Uncertainties in Generative Deep Learning and Data Amplification for High Energy Physics

Dissertation
zur Erlangung des Doktorgrades
an der Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Physik der Universität Hamburg

vorgelegt von
Sebastian Guido Bieringer

Hamburg
2024

Gutachter/innen der Dissertation:	Prof. Dr. Gregor Kasieczka
	Prof. Dr. Mathias Trabs
Zusammensetzung der Prüfungskommission:	Prof. Dr. Gregor Kasieczka
	Prof. Dr. Mathias Trabs
	Prof. Dr. Jochen Liske
	Prof. Dr. Johannes Lederer
	Dr. Frank Gaede
Vorsitzende/r der Prüfungskommission:	Prof. Dr. Jochen Liske
Datum der Disputation:	10.12.2024
Vorsitzender des Fach-Promotionsausschusses PHYSIK:	Prof. Dr. Markus Drescher
Leiter des Fachbereichs PHYSIK:	Prof. Dr. Wolfgang J. Parak
Dekan der Fakultät MIN:	Prof. Dr.-Ing. Norbert Ritter

Abstract

The upcoming high-luminosity upgrade of the LHC requires an increase in simulated data. Due to the high computational cost of detector simulation, this demand threatens to surpass the computational resources. As a consequence, it is important to develop faster, less compute intensive alternatives to classical detector simulation with Markov chain Monte Carlo (MCMC). Generative Deep Learning surrogates are one possible candidate for speeding up the simulation and are already applied in ATLAS fast simulation tools. However, the quality of the surrogate data is intrinsically limited by the training statistics.

We demonstrate that the amount of training data poses as an upper limit on the precision of global properties of observables constructed from the data. Such global properties include for example means or variances. Nevertheless, the inductive bias of the Neural Network fit allows to surpass the training statistics when analyzing smaller regions of the data space. We show that the relaxed limit, which still depends on the training data, can be estimated from uncertainties predicted by Bayesian Neural Networks. To achieve a truthful estimate, the uncertainty prediction needs to be well calibrated. We show one way to calibrate uncertainties for generative Bayesian Neural Networks and find that the common variational inference method is hard to calibrate.

We therefore develop a new method based on stochastic gradient MCMC. This method is called `AdamMCMC`. It is easy to apply and replaces the stochastic optimization commonly employed in Deep Learning. In contrast to variational inference, the variance of the uncertainty prediction can be adapted effectively through variation of a single parameter. Diverse predictions indicate out-of-distribution application. Overall, we find that the stochastic gradient MCMC produces more reliable predictions than variational inference in multiple applications.

Classifier Surrogates are one possible application of generative Machine Learning, where reliable uncertainties are crucial. This class of surrogates predicts the behavior of jet taggers working on detector data from more accessible data. Experimental analysis employing such taggers can be reinterpreted without the need for detector simulation. This cuts computational cost and enables sharing of the analysis outside the collaboration. However, the uncertainties introduced by the approximation need to be controlled and application to new data spaces needs to be prevented. We show that Continuous Normalizing Flows, in combination with `AdamMCMC`, can fulfill these requirements. Similar surrogates can be of high value for the community and could be implemented with every jet tagger employed at ATLAS or CMS.

Zusammenfassung

Das bevorstehenden "High-Luminosity" Upgrade des LHC erfordert eine entsprechende Steigerung der simulierten Datenmenge. Aufgrund der hohen Rechenkosten der Detektorsimulation könnte diese Nachfrage jedoch die Rechenressourcen zu übersteigen. Daher müssen schnellere, weniger rechenintensive Alternativen zur klassischen Detektorsimulation mit Markov Chain Monte Carlo (MCMC) entwickelt werden. So genannte "Surrogates", die Methoden der generativen künstlichen Intelligenz anwenden, sind ein möglicher Kandidat für die Beschleunigung der Simulation. Sie werden bereits im ATLAS-Experiment zur Simulation genutzt. Die Qualität der Daten, welche von einem Surrogate erstellt werden, ist jedoch von Natur aus durch die Statistik seiner Trainingsdaten begrenzt.

Wir zeigen, dass die Menge der Trainingsdaten eine Obergrenze für die Präzision globaler Eigenschaften von aus den Daten konstruierten Observablen darstellt. Mittelwerte oder Varianzen sind ein Beispiel solcher Eigenschaften. Die Beschränktheit des Funktionsraumes Neuronaler Netzwerke ermöglicht es jedoch, die Trainingsdaten bei der Analyse kleinerer Abschnitte der Datenverteilung zu übertreffen. Die durch die Trainingsdaten definierte Grenze kann auch aus den Unsicherheiten geschätzt werden, welche Bayessche Neuronale Netzwerke vorhersagen. Um eine wahrheitsgetreue Schätzung der Grenze zu gewährleisten, muss die Unsicherheitsvorhersage gut kalibriert sein. Wir zeigen eine Möglichkeit, Unsicherheiten für Bayessche generative Neuronale Netze zu kalibrieren. Wir demonstieren ebenfalls, dass die weit verbreitete "Variational Inference"-Methode schwer zu kalibrieren ist.

Wir entwickeln daher unsere eigene Methode basierend auf "stochastic gradient MCMC". Diese Methode, `AdamMCMC`, ist einfach anzuwenden und ersetzt die Methoden der stochastische Optimierung, welche üblicherweise im maschinellen Lernen eingesetzt werden. Im Gegensatz zu "Variational Inference" kann die Varianz der Unsicherheitsvorhersage anhand eines einzigen Parameters effektiv angepasst werden. Darüber hinaus zeigt die Methode die Anwendung des Netzes auf unbekannte Daten an, in dem sie hohe Unsicherheiten vorhersagt und liefert allgemein zuverlässigere Vorhersagen.

"Classifier Surrogates" sind eine mögliche Anwendung des generativen maschinellen Lernens, bei der zuverlässige Vorhersage der Unsicherheiten entscheidend ist. Diese Klasse von Modellen sagt das Verhalten von "Jet-Klassifizierungsmodellen" voraus, die direkt mit Detektordaten arbeiten. Für diese Vorhersage verwenden diese besser zugänglichen Daten anstelle der Detektordaten. Experimentelle Analysen, die solche Klassifizierungsmodelle verwenden, können mit Hilfe eines "Classifier Surrogates" neu interpretiert werden, ohne dass eine weitere Simulation des Detektors notwendig ist. Dadurch sinkt der Rechenaufwand. Die Analyse kann auch von Forscher weiter genutzt werden, die keinen Zugriff auf die Detektorspezifikationen haben. Damit solche ein Modell zuverlässig ist, müssen die Unsicherheiten abgeschätzt werden, welche durch die Modellieriung eingeführt wurden. Darüber hinaus muss vor der Anwendung auf Daten, welche für Klassifizierungsmodell und Surrogate neu sind, gewarnt werden. Wir zeigen, dass "Continuous Normalizing Flows" in Kombination mit `AdamMCMC` diesen Anforderungen genügen. Ein solcher "Classifier Surrogates" stellt einen großen Mehrwert für die Forschungsgemeinschaft dar und könnte für jeden Jet-Tagger, der bei ATLAS oder CMS eingesetzt wird, implementiert werden.

Acknowledgements

First and foremost, I would like to thank Prof. Dr. Gregor Kasieczka and Prof. Dr. Mathias Trabs for the opportunity to conduct the research presented in this thesis, as well as the supervision during the projects. Interdisciplinary projects can be hard to manage and take time. A common research language needs to be found, common research interest defined and possible ways to leverage the expertise of all collaborators found. I think we did a good job at that! During my PhD I felt well-connected to the research of both groups and supervised from both sides. Thank you for all the advice and encouragement!

I would like to thank Maximilian for the productive cooperation. I highly appreciate your commitment to finding statistical justification for algorithms I dreamed up and your endless patience explaining the intricate reasoning why certain ideas where not statistically sound!

Thanks as well to the organizers of my graduate school, Christiane and Heike, for providing an invaluable supporting structure to all the students of the school! My gratitude extends to all the students of the school, who participated in the community and formed a lively peer group. I would specifically thank my friends Thea, Carlos, Yahya, Amir, Jörn, Sebastian and Lars for the discussions and support!

I also thank the whole working group in Hamburg for all the good times we had! I fondly remember playing Mario Cart, going out, sharing home cooked food and cake or just eating lunch at the cafeteria. This also includes all the valuable discussions we had!

Once more I want to express my appreciation for Mathias, Lars, Sascha and Maximilian for making time to proofread this thesis! I know you are very busy and did this after hours. I hope I can return the favor!

Last, but in no ways least, I would like to thank my family for the unconditional support of my academic endeavors, during the past years as well as the last months of writing. This would not have been possible without you! This also includes my partner Charlotta. Thank your for being understanding when I needed to prioritize work, thank you for calming me down when I was overwhelmed and thank you for reminding me to rest! Thank you for always being there when I need you!

This work was supported by the Helmholtz Information and Data Science Schools via DASHH (Data Science in Hamburg - HELMHOLTZ Graduate School for the Structure of Matter) with the grant HIDSS-0002 and the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy – EXC 2121 Quantum Universe – 390833306 and via the KISS consortium (05D23GU4) funded by the German Federal Ministry of Education and Research BMBF in the ErUM-Data action plan. Furthermore, the computational resources for the research were provided in part through the Maxwell computational resources operated at Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany.

Contents

Preface

This cumulative thesis entails research conducted at the Institute for Experimental Physics at the University of Hamburg in the time between 2021 and 2024. It has been (pre-)published in five papers. These are distinguished from other citations by an initial letter "P".

Publications

- [P1] S. Bieringer, A. Butter, S. Diefenbacher, et al. "Calomplification — the power of generative calorimeter models". In: *JINST* 17.09 (2022), P09028. DOI: 10.1088/1748-0221/17/09/P09028.
- [P2] S. Bieringer, G. Kasieczka, M. F. Steffen, et al. "AdamMCMC: Combining Metropolis Adjusted Langevin with Momentum-based Optimization". *Submitted to the 39th AAAI Conference on Artificial Intelligence, February 25 – March 4, 2025, Philadelphia, Pennsylvania, USA; passed first phase of rejections.* Dec. 2023. arXiv: 2312.14027 [stat.ML].
- [P3] S. Bieringer, G. Kasieczka, M. F. Steffen, et al. "Statistical guarantees for stochastic Metropolis-Hastings". *Submitted to JLMR; third phase of reviews.* Oct. 2023. arXiv: 2310.09335 [stat.ML].
- [P4] S. Bieringer, G. Kasieczka, J. Kieseler, et al. "Classifier surrogates: sharing AI-based searches with the world". In: *Eur. Phys. J. C* 84.9 (2024), p. 972. DOI: 10.1140/epjc/s10052-024-13353-w.
- [P5] S. Bieringer, S. Diefenbacher, G. Kasieczka, et al. "Calibrating Bayesian generative machine learning for Bayesiamplification". In: *Mach. Learn. Sci. Tech.* 5.4 (2024), p. 045044. DOI: 10.1088/2632-2153/ad9136.

Contributions

The publications are joint efforts of multiple researchers. Individual contributions of the author are:

[P1]: This paper contains research proposed by the authors of Reference [1] as a follow-up study in 2020. The Neural Network training has been conducted by D. Hundshausen, before the author joined the institute. The consecutive evaluations, plots and interpretations where conducted by the author. The text of Section 1 and 2 thus is a joint effort of all authors, while the remainder of the paper was composed by the author.

[P2]: The second presented paper has developed from the joined work on Reference [P3]. To keep the central theme of this thesis running, they are presented in reversed chronological order. The idea of the algorithm has been developed by the author based on the need for a fast and applicable MCMC sampling, with assistance from Dr. M. F. Steffen and Prof. Dr. M. Trabs. As such, the implementation, studies and writing are work of the author, except for Theorem 1 and 2, as well as their proofs.

[P3]: The third presented study was initiated by Dr. M. F. Steffen and Prof. Dr. M. Trabs as a purely statistical study. For the numerics of the first revision, the author was consulted. The main body of text and the theoretical consideration behind can thus be allocated to both collaborators, while the text of the numerics section and the calculations within are efforts of the author.

[P4]: The need for surrogate networks was discussed at the LHC Reinterpretation Forum and the 2023 PhysTeV workshop at Les Houches [2]. It was followed by technical discussions between Dr. J. Kieseler and Prof. Dr. G. Kasieczka, who proposed the idea to the author. The technical implementation, employed methods and evaluations, as well as the main body of the publication are work of the author.

[P5]: The idea of the fifth publication was proposed by the author to Dr. S. Diefenbacher, Prof. Dr. G. Kasieczka and Prof. Dr. M. Trabs at the end of 2023. It aims connect the projects presented above. Under the supervision of both the studies where conducted autonomously by the author. The presented study, as well as the writing, were conducted by the author.

Notation

High-Energy Physics

ψ	wave function of a fermion
ϕ	wave function of a Higgs boson
ψ^*,$\psi^\dagger$, $\bar{\psi}$	complex conjugate, Hermitian conjugate, Hermitian adjoint of ψ
$SU(N)$, $U(N)$	(special) unitary group of degree N
∂_a	partial derivative with respect to dimension a
D_a	covariant derivative with respect to dimension a
$\sqrt{s}$	center-of-mass energy
L	luminosity
Φ, Θ, η	azimuthal and polar angle relative to the beam axis, pseudorapity
Z	atomic number of a material
α	fine-structure constant
$\hbar$	reduced Planck constant
E_0	incident energy of a particle hitting a calorimeter
E_c	critical energy of the calorimeter material
c	speed of light
eV, MeV, GeV, TeV	(mega-/giga-/tera-)electronvolt
cm, m, km	(centi-/kilo-)meters
s	seconds
Hz, gHz	(giga-)Hertz
Mb	megabyte

Machine Learning

ϑ, f_ϑ	vector of network parameters, neural network
ϑ^*	position of the loss minimum
P	number of parameters in ϑ
ϕ	vector of parameters for an auxiliary network or posterior approximation
$\boldsymbol{z}$	vector of parameters and auxiliary variables
L, $\hat{L}$	loss function, estimator of the total loss function based on batches
ELBO	evidence lower bound
∂_a	partial derivative with respect to variable a
∇_a	nabla operator, vector of partial derivatives in all dimensions
η	stepsize
Y, X, Z	random variable of output data space, input data space, latent space
Ω_A	sample space of random variable A
N, M	dimensionality of output and input data space
$\mathcal{D}_n$, n	training data, number of points in data
$\overline{\mathcal{D}}_m$, m	batch of training data, number of points in a batch
$\varepsilon > 0$	small parameter to prevent nuisance
$W(t)$, ϵ	Wiener process, normal distributed noise
$\hat{p}_X$, p_X, p_X^n	(estimated) probability density of random variable X
p_X^n	product probability density of n samples from X
$x \sim p_X$	x drawn from p_X
$\hat{\pi}(\vartheta)$, $\pi(\vartheta)$, $\pi_\lambda(\vartheta \mid \mathcal{D})$	(estimated) probability density of network parameter ϑ, Gibbs posterior
$\vartheta \sim \pi(\vartheta)$	ϑ drawn from π, in slight abuse of notation the argument is often specified where no subscript indicates the random variable
$\lambda = 1/k_B T$, k_B, T	inverse temperature, Boltzmann constant, temperature
o, h, $\mathcal{M}$	observables, properties and measurement of distance for amplification
a, A	amplification sample size, amplification factor
$\hat{\mathcal{I}}$, $\mathcal{I}$	(empirical) Fisher information
TV, $\Vert \cdot \Vert_{\mathrm{TV}}$	total variation distance and norm
α	acceptance probability ratio

General

tr	trace operator
$\mathrm{diag}(a_1, ..., a_n)$	diagonal matrix with entries a_1 to a_n
$\mathbb{1}$	unit matrix
$\det(A)$	determinant of matrix A
$A \geq B$	the difference of both matrices $A - B$ has non-negative eigenvalues
ln, exp	logarithm with base e, exponential function with base e
log	logarithm with base 10
sin, cos, tan	trigonometric functions
max, min	maximum, minimum
argmax, argmin	parameters that minimize the argument of the operator
var	variance
$\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{N}$	(positive) real numbers, natural numbers
$\mathrm{uniform}(\cdot; a, b)$	probability density of uniform distribution with bounds a and b
$\mathcal{N}(\cdot; a, b)$	probability density of normal distribution with mean a and covariance b
$a^\top$	transpose of vector a
$\odot$, $\oslash$	element-wise multiplication and division
D_{KL}, D_{JS}, $\overline{D}_{\mathrm{JS}}$	Kullback-Leibler, Jensen-Shannon divergence, histogram approximation of the Jensen-Shannon divergence
$\mathbb{E}_x$, $\mathbb{E}_{x \sim p_X}$	expectation value over x
$a \circ b(x) = a(b(x))$	composition of operators a and b
$\mathcal{O}$	order

Introduction

In high-energy physics (HEP) particle collisions at immense energies are produced at colliders such as the Large Hadron Collider (LHC) [3]. The aim of such collisions is to observe new physics in the spray of particles produced. Unknown physics at high energies is assumed to constitute itself in form of new particles, such as the Higgs [4, 5]. Interactions with these new particles will affect the production of stable particles in the collisions and their energy depositions in the measuring apparatus, also called a *detector*.

From theory side, these processes are described by the *Standard Model of Particle Physics* [6–8]. Although highly successful, the Standard Model cannot explain certain physics phenomena. These include the existence and nature of dark matter and the imbalance between matter and antimatter in the universe. At this point in time, clear evidence of such physics beyond the Standard Model has not yet been found at colliders. This indicates, that either the energies of the collisions or the precision of the measurement are not yet sufficient. To reach the absolute limit of the LHC in terms of precision, before constructing a stronger accelerator, it is upgraded to the *High-Luminosity LHC* (HL-LHC) [9] starting in 2025. This includes installation of detectors with higher spatial resolution. It also includes an increase of the collision rate by 5-7.5 times to further suppress the stochastic variance in the measurements of rare processes.

Conclusions on the agreement of theory and experimental observation are drawn by analyzing the agreement between predictions and measurements. To propagate the predictions of the Standard Model to energy depositions in a detector, costly simulations of the interactions between particles and detector are required. Similar amounts of simulated and experimental data need to be compared to control the stochastic errors of the analysis. With the start of operation of the HL-LHC in 2028, the amount of simulation needed threatens to surpass the computational resources [10]. This motivates the development of simulation alternatives which are less compute intensive. One prospect for cheaper simulation are Deep Learning surrogates employing generative Neural Networks (NNs) [11]. NNs can be understood as highly flexible family of functions parameterized by a huge number of parameters. Due to their structure, such models scale better to the high dimensional data produced by detectors. The inherent logic of the undertaking is: Take a limited sample of classically simulated points to adapt the NN parameters with stochastic optimization and use the model to generate arbitrary amounts of data thereafter. As the quality of the predictions of the surrogate is limited by the training statistics, the data produced this way will never be as precise as the classical simulation. Generation speed is treated for accuracy. This can be accepted, because the simulation of detector effects is applied on top of the simulation of the collision, which remains untouched and which might include new physics. The detector effects merely constitute a complicated smearing of these initial processes.

However, if the surrogate data cannot improve the simulation beyond the training statistics, this approach is flawed. We thus need to answer the question: Can an analysis be improved through the addition of artificial surrogate data? This is the question of *data amplification*. As there are various possibilities to perform the analysis, this general phrasing of the question is ill-posed. In Reference [P1, 1], it is thus rephrased on the level of distributions: Is the approximation of the surrogate closer to the true data distribution than a histogram of its training data? And how does it compare to other classical density estimation techniques?

To answer these research questions both studies compare the surrogate data to the true data distribution, which is either known or approximated by a large amount of data held back for this approximation. While this serves as a nice proof of concept, in practice it defeats the purpose of the surrogate. A much more applicable setting would be a network that itself predicts its modeling error. *Bayesian Neural Networks* (BNNs) [12] are a class of Deep Learning algorithms that predict the uncertainty based in limited data. Instead of optimizing the values of the NN parameters, such methods infer or sample a distribution of the parameters informed by the data. Through the width of these distributions, they encode uncertainty on the parameters.

Despite the requirement for well modelled errors in HEP [13], these methods are not yet widely applied within the field. This might be due to the increased memory-footprint and inference times or due to the relative complexity of the algorithms themselves. *Variational inference* (VI) [14] at this point is the only flavor of BNNs applied in HEP [15–18]. It approximates the true distribution of the weights as an uncorrelated Gaussian distribution. The resulting simplification is then used to rephrase the inference of the parameter distribution as an optimization task. This optimization can then be performed with traditional optimization techniques. However, the approach has severe drawbacks. First, the joint optimization of the NN and the distribution of its weights can be unbalanced, need additional fine-tuning and yield worse results than the deterministic framework. In an environment that competes for the best possible performance, this hinders the acceptance. Historically, the inference of such distributions is often done with Markov chain Monte Carlo (MCMC) [19]. These methods are notorious for being computationally expensive and scaling bad to high-dimensions. Despite this reputation, recently developed stochastic gradient MCMCs [20, 21] optimize at speeds comparable to stochastic optimization methods. The term "stochastic" refers the use of batches of data rather than the entire data to drastically reduce computational cost. Stochastic gradient MCMCs further do not enforce any assumption on the shape of the weight distributions. As such, they often produce better calibrated uncertainty estimates compared to approximations of the weight distribution. In [P2], we propose our own method that acts as a drop-in replacement for stochastic optimization.

The stochastic nature of the chains however poses a problem. The sampled distribution is not guaranteed to be correct. This can be solved with a *Metropolis-Hastings* (M-H) correction. A stochastic phrasing of the correction again introduces similar issues. Multiple solutions exist. However, they often increase the computational cost. In Reference [P3], we introduce a cheap correction to the optimization objective. It guarantees sampling from a distribution with the same properties as the one that would be sampled using the entire data.

Equipped with these methods, we present a first application of MCMC sampling to generative Machine Learning for HEP in Reference [P4]. We show that the uncertainty of such a BNN can indeed be connected to the amplification power of the network [P5]. While this still requires to check the calibration of the uncertainties against a ground truth, the application is more flexible than the previous efforts. For example, it includes splitting the data into areas used for calibration and others for prediction.

The thesis is structured as follows. In Section 1, we introduce the application domain and its Deep Learning use-cases to argue the need for surrogate models. The basics of Deep Learning are presented in Section 2 and generative NN are introduced in Section 3. In the later chapters of Section 3 data amplification with generative Machine Learning is discussed. We introduce the different flavors of Bayesian Machine Learning in Section 4 and debate the intricacies of stochastic M-H corrections, as well as possible solutions in 5. In Section 6, we apply these methods to HEP problems and connect them to the encompassing theme of amplification.

1 Particle Physics

Particle physics is the research field related to the search for the most fundamental, indivisible building-blocks of matter. In the subfield of HEP, collisions of particles at large energies are observed. These collisions serve as proxies for the high energy density at the start of the expansion of the universe. With this approach information on the processes that initially formed matter can be inferred from the products of the collisions.

The foundations of this field date back to the discovery of the electron by Thomson in 1897 [22] and the scattering experiments of Rutherford [23]. Through scattering of α-particles on a gold foil, Rutherford first showed that matter is made up of nuclei distributed distantly over the volume of a body. In similar collision experiments he later (1919) found the proton [24]. The particle nature of the photon, proposed by Einstein, was first experimentally proven by Compton in 1922 [25]. After the discovery of the neutron by Chadwick in 1932 [26], all chemically relevant particles were discovered and after the electron neutrino was experimentally observed in 1956 [27], the atom model seemed complete.

This assumption proved to wrong when first colliders resolved an internal structure of the proton, so-called partons, through electron-proton collisions. These observations resulted in the discovery of quarks at the Stanford Linear Accelerator Center (SLAC) [28, 29] and of gluons at DESY (Deutsches Elektronen-Synchrotron) [30]. With increasing collision energies, the heavier mediator particles of the weak interaction, the $W^{\pm}$- and Z-boson were found in proton-antiproton collisions at CERN in 1983 [31, 32]. With further increasing energies, the heavy top quark was found at the Tevatron at Fermilab in 1995 [33, 34]. Finally, in 2012 the mass mediating Higgs boson was found at the LHC at CERN [4, 5].

These discoveries and many more, are theoretically described by the Standard Model of Particle Physics [6–8]. In Section 1.1, we give an overview on how the Standard Model connects the aforementioned particles. Section 1.2 quickly explains the shortcomings of the model. We pick up on the experimental facets of particle colliders and detectors in Section 1.3 and Section 1.4. Calorimeters for measuring particle energies and the particle showers that develop in these calorimeters are discussed in Section 1.5. The issue of attributing the energy distribution to an initial particle is shortly brought up in Section 1.6. To understand why the HEP community is in need for amplified data from fast Deep Learning surrogates, we discuss the current state of simulating collider events from theory in Section 1.7. For textbooks with more in-depth discussions of particle physics, see for example Reference [35].

1.1 The Standard Model of Particle Physics

Particle Content

The Standard Model introduces two classes of particles, spin-1/2 *fermions* which make up the matter content and spin-1 *bosons* that mediate the forces between the elementary particles. Fermions can be grouped into two classes, *leptons* (Figure 1 lower left) and *quarks* (Figure 1 upper left). There are three generations of charged fermions: electrons e^-, muons μ^- and tauons τ^-, increasing in mass, each with a full negative charge. Of these three, only the electron is stable. In every generation there exists a corresponding, electronically neutral neutrino $\nu_{e/\mu/\tau}$. Within the Standard Model neutrinos are assumed to be massless.

Quarks also admit the three-generations structure. There are three types of up-type and down-type quarks respectively. The up-type quarks, up u, charm c and top t, have a positive electromagnetic charge of 2/3, while the down-type quarks, down d, strange s and bottom b, have a negative charge of 1/3. In contrast to leptons, quarks carry a color-charge. The color-charge can be red, green or blue, where the sum of all three results in a neutral particle again.

Besides their spin, all fermions share another common feature. All fermions have a corresponding antiparticle of opposite charge, color-charge and helicity, but equal properties otherwise. The antiparticle to the electron is the positron e^+. The dynamics of a fermion are described through *Quantum Field Theory* (QFT) by a Lagrangian density. For a full introduction to the underlying principles, see for example Reference [37]. Due to the spin-1/2 nature, the Lagrangian of the fermions is specified through Dirac-algebra with γ-matrices

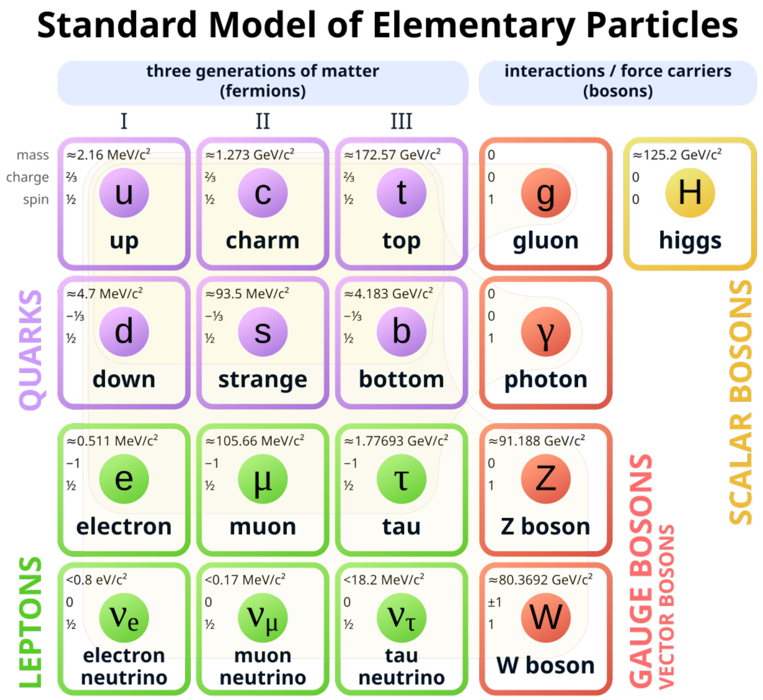


Figure 1: Diagram of the particles in the Standard Model. Image taken from Reference [36].

γ^μ and Dirac-spinor fields $\psi(x)$. With $\bar{\psi}$ the adjoint spinor, the Lagrangian of a fermion of mass m_ψ is

$$\mathcal{L}_{\text{fermion}} = \bar{\psi}(i\gamma^\mu\partial_\mu - m_\psi)\psi. \tag{1.1}$$

The Hermitian adjoint $\bar{\psi} = \psi^\dagger\gamma^0 = (\psi^*)^\top\gamma^0$ can be understood as the field of an antiparticle to a particle ψ.

Interactions

The interactions of particles can be derived by enforcing the Lagrangian to be invariant under transformations of the local gauge group $SU(3)_C \times SU(2)_L \times U(1)_Y$. A local transformation M can be written as

$$\psi(x) \to M\psi(x) = e^{ig\theta_a(x)t^a}\psi(x)\,, \tag{1.2}$$

with the generators of the group t^a, the phase of the rotation $\theta_a(x)$ and the coupling strngth g. For $SU(N)$ there are $N^2 - 1$ generators. Equation (1.1) is not invariant under such transformations as the derivative acts on the angles $\theta_a(x)$. To recover the invariance we introduce the *covariant derivative*

$$\partial_\mu \longrightarrow D_\mu = \partial_\mu + igA_{\mu a}t^a\,, \tag{1.3}$$

with gauge fields $A_{\mu a}$ for every generator of the group. By construction the gauge fields transform as

$$A_\mu \to MA_\mu M^\dagger + \frac{i}{g}(\partial_\mu M)M^\dagger\,, \tag{1.4}$$

thus recovering the invariance of the Lagrangian. The resulting $g\bar{\psi}\gamma^\mu A_{\mu a}t^a\psi$ terms describe the interaction of the fermions with the gauge fields.

The $U(1)_Y$ symmetry has only one generator. We choose $i\frac{Y_W}{2}$, with Y_W the weak hypercharge quantum number of the fermion ψ. This means, the corresponding gauge field, by design of the model, couples to fermionic particles with a non-zero weak hypercharge. The weak hypercharge is the difference of the electric charge of the particle and the third component of its weak isospin, itself a spin-like quantum number that distinguishes left- and right-handed fermions in the theoretical description.

For the $SU(2)_L$ symmetry, we choose the three Pauli matrices as generators. The weak isospin quantum number determines the coupling of the corresponding three gauge fields. This symmetry describes the *weak force*. The weak interaction is not parity conserving, as it only couples to left-handed fermions [38]. To represent this in theory, we introduce doublets under $SU(2)$ to describe left-handed particles ψ_L and singlets describing right-handed ones ψ_R. Neutrinos only appear as left-handed particles and thus only in lepton doublets.

Both gauge symmetries can be unified description in a single theory with a joint gauge symmetry $SU(2)_L \times U(1)_Y$ (*electroweak unification* [8, 39]). It is broken through the Higgs mechanism $SU(2)_L \times U(1)_Y \to U(1)_Q$ (*electroweak symmetry breaking*). This symmetry describes the *electromagnetic force*. The physical gauge boson for the electromagnetic force is the massless, neutral *photon* γ that couples to the electromagnetic charge Q. The remaining, physically observed gauge bosons can be constructed as mixtures of the four gauge bosons. The weak interaction is mediated through the neutral Z-boson, and the charged W^+- and W^--bosons. While for the other interactions the quark flavor, is conserved, the weak interaction does not respect this conservation.

The third symmetry, $SU(3)_C$, can be generated by the eight Gell-Mann matrices. These give rise to eight corresponding gauge fields, the *gluons* g, the mediators of the strong interaction. The gluons couple to the color charge of a particle. The sub-model only including the strong interaction is aptly called *Quantum Chromodynamics* (QCD). As only quarks and gluons carry such a charge, only they interact strongly. The strong interaction has two properties, that are highly relevant in collider physics. First, at high energies or small length scales this interaction is very weak and quarks and gluons move freely. This is called *asymptotic freedom* [40, 41]. For large distances however, the potential of the interaction between two quarks increases linearly with the distance. At low energies, the quarks are thus forced together and cannot exist in isolation. As a result, only color-neutral combinations of quarks, so-called hadrons, exist outside of particle colliders and neutron stars. This effect is referred to as *color confinement*. The transition from high to low-energy regimes and the connected creation of composite particles is known as *hadronization*. The most common hadrons, are either comprise quark-antiquark pairs (*mesons*) like pions or kaons, or combinations of three quarks, like protons (uud) and neutrons (udd), called *baryons*. Amongst these combinations only protons p are generally stable.

All gauge bosons are spin-1 particles with their dynamics described through a kinetic term

$$\mathcal{L}_{\text{kin}} = -\frac{1}{2} \operatorname{tr} F_{\mu\nu} F^{\mu\nu} \,, \tag{1.5}$$

with the field strength tensor $F_{\mu\nu} = -\frac{i}{g}[D_\mu, D_\nu]$.

The Higgs Mechanism

The principles introduced this far, do not explain the masses of both fermions and bosons. The naive mass term $m_\psi \bar{\psi}\psi = m_\psi(\bar{\psi}_R \psi_L + \bar{\psi}_L \psi_R)$ of (1.1) directly connects left- and right particles, doublets and singlets of $SU(2)_L$, and thus breaks the gauge invariance under this rotation group. Similarly, the naive mass terms of the gauge bosons are not invariant either.

To add a gauge invariant mass term for these particles. We introduce another auxiliary field, the spin-0 *Higgs* ϕ. Its dynamics are described by a Lagrangian with a quartic potential

$$\mathcal{L}_{\text{Higgs}} = \frac{1}{2}(D_\mu \phi)^\dagger (D^\mu \phi) - \left(\frac{1}{2}\mu^2 \phi^\dagger \phi + \frac{1}{4}\lambda (\phi^\dagger \phi)^2 \right) \tag{1.6}$$

and $\lambda > 0$ and $\mu^2 < 0$. By expanding the field around one of its non-zero minima $v = \pm\sqrt{-\mu^2/\lambda}$ (*vacuum expectation value*), that is $\phi(x) = v + h(x)$, the covariant derivative gives rise to mass and interaction terms for gauge bosons. In fact, this interaction is what allows us to rewrite the gauge bosons as physical combinations of mass eigenstates in the previous section. This approach is known as *spontaneous symmetry breaking*, as expanding Equation 1.6 around the non-zero minimum breaks the invariance of the Lagrangian.

To include interaction terms of fermions with the Higgs we write ϕ as a $SU(2)_L$ doublet. With this choice, the fermion interaction terms, so-called Yukawa couplings

$$g_f(\bar{\psi}_L \phi \psi_R)\,, \tag{1.7}$$

are invariant under the local gauge transformation. By expanding ϕ around its minimum, the interaction terms can be written as an invariant mass term and an interaction with the Higgs particle $h(x)$. As the Yukawa coupling g_f appears in the mass term, as well as the interaction term upon expansion, we find the coupling strength of fermions to the Higgs is proportional to its mass. To gauge the Higgs section in experiment, the exceptionally heavy top quark is thus especially interesting. This is one reason why techniques distinguishing the top from the remaining quarks, that is QCD background, are especially relevant for experimental analysis.

The entire derivation of boson and fermion masses via the Higgs potential is commonly referred to as the *Higgs Mechanism* and has been postulated in 1964 [42–44]. Because the most likely decay channels of the Higgs come with large backgrounds, the Higgs boson was not found until 2012 [4, 5].

1.2 Beyond the Standard Model

The Standard Model is an outstandingly successful theory and does explain the fundamental interactions of matter to high precision. It relies on experimental observations to tune the values of 19 internal parameters. Similarly, the number of lepton and quark generations is constrained by observation rather than explained from a more fundamental theory.

While ongoing efforts try to determine the internal parameters to the highest possible precision, the HEP community is also searching for physics beyond the Standard Model (BSM) to motivate a more general theory and explain phenomena that cannot be described in the Standard Model. The unexplained observations include:

- **Neutrino masses:** The Higgs mechanism only explains masses for fermions and bosons, but not for neutrinos. It has however been observed that solar neutrinos change flavor. Electron neutrinos convert to muon and tau neutrinos and vice versa, when propagating through space [45, 46]. Similar observation have been reported for man-made neutrino beams [47]. This indicates that the physical neutrinos are a mixture of the mass eigenstates allowing the mixing between the neutrinos. Hence, at least one neutrino is massive.

  The seesaw mechanism [48–51] is one theory to explain the mass of neutrinos. In it one assumes a second, very heavy, right-handed neutrino. The mass prediction of the second neutrino gets bigger, the smaller the mass of the Standard Model neutrino. As such heavy neutrinos cannot interact directly via any of the three forces of the Standard Model, the heavy neutrino also is one candidate for the explanation of dark matter.

- **Dark matter:** From astronomical observations of gravitational effects such as the radial velocities of galaxies, gravitational lensing and structure formation in the early universe, we find discrepancies between the mass predictions and the observed, luminous matter. This indicates that the universe is largely filled with matter that interacts only weakly with Standard Model particles. Multiple ideas for dark matter exist within cosmology, such as brown dwarf stars or primordial black holes. Particle physics explanations usually propose a weakly interacting massive particle, such as the heavy neutrinos or axions.

  The visible, baryonic matter of the universe makes up $\approx 5\%$ of all energy in the universe and dark matter about $\approx 25\%$. The remaining, $\approx 70\%$, so-called dark energy, are needed to explain the expansion of the universe. For a more detailed introduction, see for example Reference [52].

- **Baryon asymmetry:** The Big Bang model is the assumption that the universe expanded from a singular point of infinite temperature and density. For this model, the Standard Model predicts the creation of matter and antimatter in equal proportions. The fact that matter in the universe today is predominantly matter indicates the incompleteness of our model.

- **Unification:** The coupling strengths of the three forces of the Standard Model depend on the energy through quantum loop corrections. The coupling strength of the weak and strong interactions decrease, while that of the electromagnetic interaction increases. This brings the coupling of the forces closer together at high energies. If the coupling were to meet at the same point, a unified theory of all three forces in $SU(5)$, rather than only the electromagnetic and weak, would be possible [53]. This however is not the case for the content of the Standard Model. Additional particles are required to correct the running of the three couplings to meet at a single energy point [35].

- **Gravity:** Up to this point, we have discussed three of the four fundamental forces of nature: The electromagnetic, weak and strong force. The fourth fundamental force, gravity, is not part of the Standard Model. A unified theory of all four forces would truly be a “theory of everything”. The fundamentally different formulation of QFT (perturbation theory and path integrals) and General Relativity (differential geometry) make their unification especially tricky. Theories that manage to do so, such as string theory, often introduce additional dimensions and large parameter spaces that cannot be constrained from experiment [54].

- **Hierarchy problem:** The QFT prediction of the Higgs mass includes quantum loop corrections. These are quadratic in the highest mass scale of the theory. For a grand unified theory, this scale is on the order of $\approx 10^{16}$ GeV, while the Planck scale is even higher. As the Higgs mass is at ≈ 125 GeV, these corrections need to cancel to a high degree of precision. While such a theory can be constructed, the fine-tuning required for the corrections to cancel seems unnatural. A supersymmeteric symmetry, in which all particles are matched with a superparticle which differs in spin by $1/2$, would naturally introduce this cancellation, but has not yet been verified experimentally [35].

  Supersymmetric models also require a second Higgs doublet introducing 3 additional, heavy Higgs bosons. As the addition of these particles changes the interaction of the Standard Model Higgs with the fermions, discrepancies in the Higgs branching ratios are one interesting gauge of BSM physics. This motivates building future electron-positron colliders that produces Higgs bosons more efficiently.

1.3 Particle Colliders

In the introduction to this chapter, we have already encountered multiple different particle colliders, each famous for the discovery of new particles: The Stanford Linear Collider (SLC) at SLAC, the Positron-Electron Tandem Ring Accelerator (PETRA) at DESY, the Super Proton–Antiproton Synchrotron (S$p\bar{p}$S) at CERN and the LHC at CERN.

They differ in construction, accelerated particle and target. These choices in design determine the maximum energy released in the collision and the number of collisions observed per time. The former is referred to as the *center-of-mass energy* $\sqrt{s}$ while the latter is called *luminosity* L. The center-of-mass energy determines the processes possible within the collision, as well as their probability.

Of the introduced colliders, only the SLC is constructed as a linear collider accelerating charged particles through electromagnetic fields along a line. In a linear accelerator, only the strength of the accelerarting fields and the length of the track limit the center-of-mass energy. The other three colliders are constructed as circular tracks. In a circular collider, the particles can accelerate for multiple rounds, until the amount of energy lost to synchrotron radiation (Bremsstrahlung) equals the energy put into the system. As the energy loss to synchrotron radiation is antiproportional to the fourth power of the mass of the particle, circular colliders are effective for colliding heavier particles, such as protons or ions. As the energy loss is also antiproportional to the radius, upgrading to a higher energy usually means building a larger accelerator ring. The circular design also lends itself to the use of multiple bunches of particles to boost the luminosity and collision of bunches travelling in opposite directions to increase the center-of-mass energy without the need for additional tunnels. It also allows the construction of multiple interaction points for independent experiments. In linear accelerator designs, multiple experiments need to share one interaction point.

The design of the accelerator is usually determined by the type of collision desired. The most common choices are electron-positron, proton-proton and heavy ion collisions. A notable exception is the Hadron-Electron Ring Accelerator (HERA) at DESY [55], that used electrons and protons as beam particles. While heavier particles reach higher center-of-mass energies, the initial states of the process are unknown as the energy within a hadron is probabilistically distributed amongst its partons, that is quarks and gluons. Furthermore, lepton colliders prevent QCD background processes. Hadron colliders thus require a higher amount of data cleaning and classification to filter out the processes of interest. A more detailed report on particle colliders is given in Reference [56].

The Large Hadron Collider and its High-Luminosity Upgrade

The LHC [3] is the largest particle collider to date. Proton-proton collisions currently reach a center-of-mass energy of up to 13.6 TeV. The electromagnetic field for the acceleration is provided by 16 superconducting radio-frequency cavities along a 26.7 km circular beam line. Four large-scale experiments are located at different crossing points of the LHC beams. The general purpose detectors ATLAS [57] and CMS (Compact Muon Solenoid) [58], as well as the specialized experiments LHCb [59] and ALICE (A Large Ion Collider Experiment) [60]. LHCb is specifically designed for B-meson physics and ALICE specializes in heavy ion collisions.

The LHC was originally designed to for a luminosity of $L = 10^{34}\,\text{cm}^{-1}\text{s}^{-1}$. After the end of LHC Run-3 in 2025, the machine will be updated to increase the luminosity by a factor of 5 to 7.5 to gather more statistics on rare processes such as Higgs pair production [9]. This project is aptly called the HL-LHC. The increase in collisions observed per time will be achieved through higher collision rates and stronger focused beams. Upgrades include stronger superconducting magnets, more precise radio-frequency cavities and more efficient links and beam collimators. Simultaneously, updates and replacements to the detectors need to be implemented to guarantee higher radiation tolerance, faster data processing and higher granularity. For the CMS detector, these include stronger cooling, new endcaps for the calorimeters, a higher granular tracking, as well as updates to related readout electronics [61]. On software side, necessary updates are faster triggers, classifying interesting processes from background at measurement time, and speed-up in theory simulations, to provide comparable datasets to the increasing amounts of observations. The need for faster simulation motivates the augmentation of the simulation chain by generative Machine Learning.

Plans for Future Colliders

The main proposals for future colliders include the construction of large scale electron-positron colliders that efficiently produce Higgs particles [62]. To this end, a future collider needs to reach energies $\sqrt{s} \geq 250$ GeV and thus has to be sufficiently large.

- The *Future Circular Collider* (FCC) [63] proposes the constructing of a circular collider of ≈ 100 km circumference at CERN, that can operate at ≈ 350 GeV to study top pair production. After the operation of the FCC as a electron-positron collider, the tunnels could be reused for hadron collisions. Such a collider could reach energies up to ≈ 100 TeV. A decision on this project is scheduled for 2027 - 2028 with constructing beginning in the 2030s

- The *Circular Electron Positron Collider* (CEPC) [64] is the counterproposal to the FCC in China. Its design achieves similar center-of-mass energies at a comparable circumference. The project awaits approval by the Chinese government in 2025 and construction could start as early as 2027.

- Rather than using a circular design, the *International Linear Collider* (ILC) [65] study suggests the construction of two opposing linear accelerators in northern Japan. The footprint of the machine has a total length of about 31km. The linear construction would generate electron-positron collisions at $\sqrt{s} = 250$ GeV up to 1 TeV. Two general detectors are planned at the ILC, the International Large Detector (ILD) and the Silicon Detector (SiD) [66]. To facilitate high resolution jet reconstruction, both detectors are designed with high-granular calorimeters.

- A similar project, the *Compact Linear Collider* (CLIC)[67], was proposed for the construction at CERN. Starting out from a 11 km long collider capable of $\sqrt{s} = 380$ GeV, the study proposes to gradually upgrade the length of the accelerator ins two steps to 29km (1.5 TeV) and 50km (3 TeV). Following the proposal, the increased center-of-mass energy over the ILC could be achieved through a novel two-beam setup with increased accelerator gradients.

1.4 Detectors

In this section, we introduce the general building blocks of particle detectors and their function. General-purpose detectors, such as the ATLAS and CMS detectors, usually consist of several building blocks. These always include a tracking system for charged particles, electromagnetic and hadronic calorimeters, a magnet and muon chambers. They are arranged around the beam crossing in layers.

In a detector, positions are usually indicated in terms of the pseudorapity η, the azimuthal angle Φ and a radial distance. The pseudorapity can be calculated from the angle to the beam axis Θ (polar angle) as

$$\eta = -\ln\left[\tan\left(\frac{\Theta}{2}\right)\right] . \tag{1.8}$$

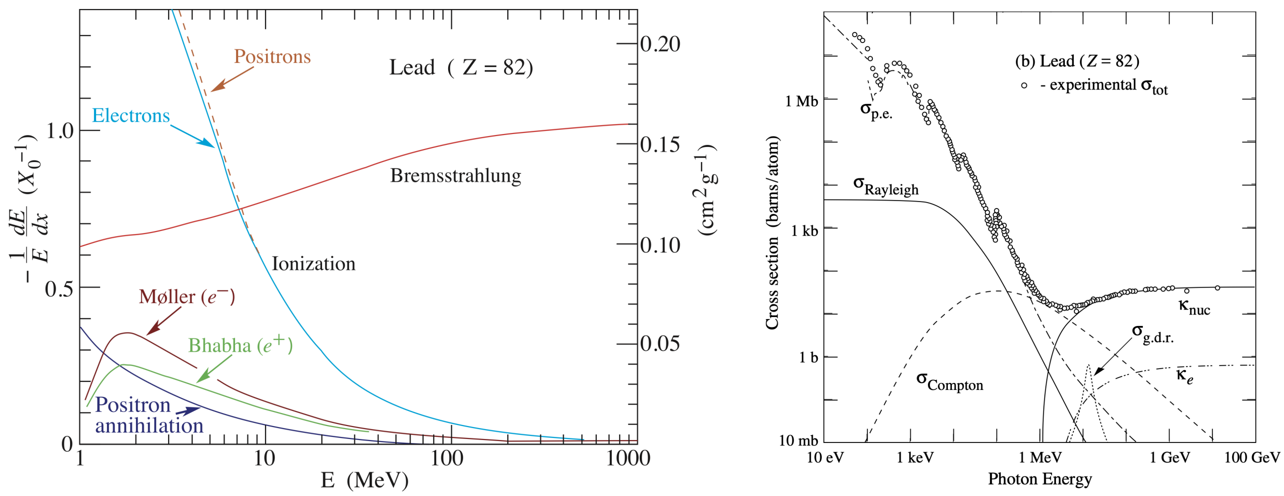
The magnet applies a magnetic field parallel to the beam axis. Charged particles thus experience a Lorentz force orthogonal to the direction of the field and their momentum. The radius of this bend is proportional to the momentum of the charged particle. From reconstructing the curve of the particle trajectories in the tracking system, one can thus infer the momentum of the particle and the sign of its charge. If the transverse momentum of the particle is too high, the trajectory will appear as a curve and the reconstruction fails. The tracking system itself is designed to record the path of charged particles while impacting the momentum of the particles as little as possible. This facilitates detailed reconstruction of the primary vertex. While CMS and ATLAS use silicon strips and pixels to create a high resolution tracking (124 million pixels for CMS), ALICE uses time projection chambers. The former use the creation of electron-hole pairs in silicon for measurement, the latter the ionization of a gas, such as xenon or argon. Arranged around the tracking system are calorimeters. Their purpose is to measure the full energy of the produced particles. A muon system usually encases the three aforementioned blocks. In contrast to electrons, the heavier muons only deposit a small part of their energy in the calorimeters. The muon system uses similar designs as the tracker, but scaled to a higher volume. Again, the bend of the muon tracks enables the reconstruction of the muon momenta.

Usually, the collision rate is dominated by uninteresting, background events. An *event* is the response of the detector to a collision. At a collision rate of about 1 gHz at the LHC, an elaborate preselection of the events is necessary, as the collected data would be too large to handle otherwise. This thinning-out is handled by multiple layers of *triggers*. Hardware-based triggers decide which events enter the data pipeline. Subsequent software-based triggers reduce the data rate by about 6 orders of magnitude and determine which events are saved. At about 3 Mb per event, this still translates to gigabytes of data per second.

1.5 Showers and Calorimeters

As introduced in the last section, the aim of a calorimeter is to destructively measure the energy of incident particles produced in the collision. For low energetic charged particles, this is already possible from the curve of its trajectory in the tracker. For neutral particles, such as photons or neutral pions, we need more specialized equipment.

In a calorimeter, the incoming particles deposit most of their energy in the calorimeter material. This happens in a cascade of interactions, each producing new particles. This cascade is called a *shower*.



(a) electrons and positrons

(b) photons

Figure 2: Processes responsible for the energy loss of electrons and photons in lead. Figure 2a shows the energy loss per radiation length of an electron as a function of the electron energy. Figure 2b shows the total cross-section of a photon as a function of its energy. "p.e." is an abbreviation for the photoelectric effect and both κ denote the cross-section of pair production in the nuclear and electron field respectively. Figures taken from Reference [68].

Electromagnetic Showers

Electrons and photons dispense energy in matter through multiple different mechanisms. To understand the creation of electromagnetic showers, we need to understand the interactions of both with the calorimeter matter.

- **Electrons:** Depending on their energy, electrons and positrons interact with matter mainly through *ionization*, at low energies, and *Bremsstrahlung*, at high energies.

  At high energies ($>$ 10 MeV), the deflection of the electrons through the electromagnetic potential of the nuclei dominates. During the interaction, the electron emits a photon, so-called Bremsstrahlung that carries part of the energy of the electron.

  Lower energetic electrons ($<$ 1 MeV) transfer their energy to electrons bound in the atoms of the calorimeter material. If the transfer is large enough, the receiving electron is freed and leaves behind an ionized atom. When not enough energy is transferred to ionize the material, the exited bound electron, will fall back to its original energy level, thereby emitting the energy in form of a photon (*scintillation*).

  Direct scattering of the electrons (*Bhabha scattering*) or positrons (*Møller scattering*), as well as electron-positron annihilation do not contribute significantly to the energy dispersion of electrons in material with high atomic numbers. The relations of these processes for the energy loss of electrons and positrons in lead is shown in Figure 2a.

- **Photons:** Photons dispense their energy in the calorimeter material through the *photoelectric effect*, *Rayleigh* and *Compton scattering* and *pair production*.

  High energetic photons create electron-positron pairs upon interaction with the electric potential of atomic electrons or nuclei. This process is only available for energies higher than the rest mass of the produced pair, that is above $\approx$ 1 MeV.

  For intermediate photon energies ($1 - 10$ MeV) the Rayleigh and Compton scattering contribute significantly to the energy deposition. Both describe scattering of the photon on the electrons of the material. Rayleigh scattering describes the elastic scattering, where no energy is passed to the electron and the photon is deflected. Compton scattering describes the inelastic process at higher energies that produces an unbound electron.

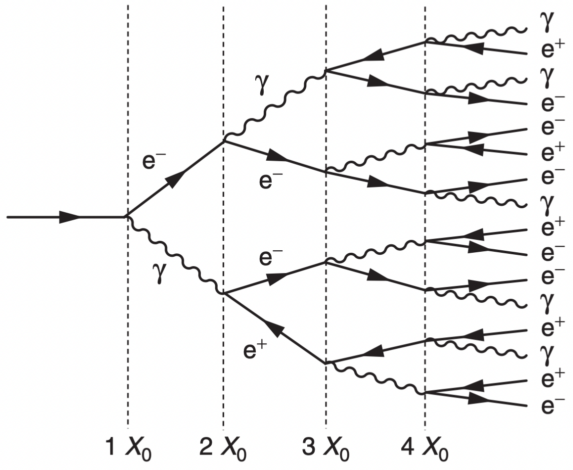


Figure 3: Schematic of an electromagnetic shower. Figure taken from Reference [35].

> Eventually, at low energies (< 1 MeV), photons predominantly are absorbed by electrons bound in an atom of the calorimeter material. This causes the electron to be emitted from the atom leaving behind a positive charge.

High energy electrons produce photons through Bremsstrahlung and high energy photons produce electron-positron pairs through pair creation. As long as the energy of produced particles is high enough, the resulting photons and electrons again transfer energy through Bremsstrahlung and pair production. The generated positrons eventually recombine with electrons producing more photons. This cascade is illustrated in Figure 3. It only stops when the energy of the particles drops below the critical energy E_c, where the ionization loss equals the electron energy. For solid materials, it can be approximated as [68]

$$E_c = \frac{610\ \text{MeV}}{Z + 1.24}\,. \tag{1.9}$$

Low energy photons ultimately produce electrons through Compton scattering and the photoelectric effect. If the shower is fully contained in the calorimeter, the number of particles created in this shower is proportional to the energy E_0 of the incident particle.

To design a calorimeter that will contain the produced showers, we need to predict the average length of a shower. The characteristic length of a shower is the *radiation length* X_0. It is defined as the mean distance a high-energy electron traverses until its energy has reduced to $1/e$ through Bremsstrahlung. For materials with atomic number $Z > 0$, the radiation length can be approximated as [35]

$$X_0 \approx \frac{1}{4\alpha n Z^2 r_e^2 \ln(287\, Z^{-1/2})}\,, \tag{1.10}$$

with n the number density of nuclei, $r_e = 2.8 \cdot 10^{-15}$ m the classical approximation of the radius of an electron and α the fine-structure constant. The mean free path length of the photons decay via pair production is $7/9 \cdot X_0$ of. As a consequence, the number of particles doubles and the average energy per particle halves after every radiation length. The maximum depth of a shower for an electron or photon is then approximately

$$\frac{\ln(E_0/E_c)}{\ln(2)} X_0\,. \tag{1.11}$$

The number of particles produced by the cascade of particles in the detector is proportional to E_0. These particles, are amplified and recorded within the calorimeter. To ensure no electrons or photons leave the calorimeter, the calorimeter material needs to ensure a low radiation length. The CMS detector uses solely lead tungstate ($PbWO_4$), which is a transparent scintillator with low $X_0 = 0.83$ cm and is hence a *homogeneous*

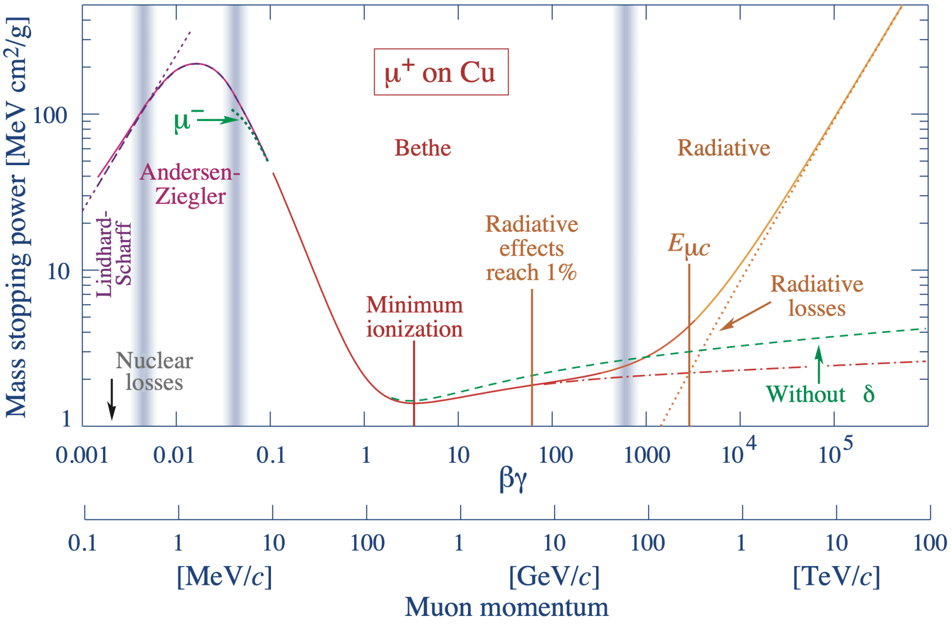


Figure 4: Mean energy loss of a muon traveling through copper per distance traveled (mass stopping power) over the momentum. The red (dash-dotted) line shows the modeling of the uncorrected Bethe-Bloch equation (1.12). Figure taken from Reference [68].

calorimeter. Alternatively, the calorimeter can also be made from alternating layers of passive and active material. The passive material features a low X_0 at low costs, while the active material is used to measure the ionization by the shower [35]. This is usually referred to as a *sampling calorimeter* and requires an additional calibration as only parts of the shower are recorded. This can be done very precisely with Deep Learning [69, 70].

Heavy Charged Particles

Charged particles generally interact with detector matter through the same processes as electrons and positrons. As the critical energy E_c depends on the mass of the particles, it grows with the square of the mass, heavier charged particles deposit energy mainly through ionization. The mean energy loss of a charged particle through ionization per distance traveled, also named *mass stopping power*, is given by the Bethe-Bloch equation [35]

$$\frac{dE}{dx} \approx -4\pi\hbar^2\alpha^2 \frac{nZ}{m_e\beta^2}\left[\ln\left(\frac{2\beta^2\gamma^2c^2m_e}{I_e}\right) - \beta^2\right] . \tag{1.12}$$

Here, $\beta = v/c$, $\hbar$ is the reduced Planck constant, m_e is the electron rest mass, $\gamma = 1/\sqrt{1-v^2/c^2}$ the Lorentz factor and I_e the effective ionization. For high energies, the mass stopping power depends logarithmically on the particle momentum squared. The Bethe-Bloch curve of a muon in copper (Figure 4) exhibits a minimum at around 300 MeV. Particles at this minimum energy are called *minimum ionizing particles* (MIP). As the energy loss per interaction is the lowest for the MIP, the number of energy deposits in the calorimeter has a peak at the MIP energy, also called the *MIP peak*. Due to its dependence only on material properties, the MIP peak is often used for calibration. If the particle momentum falls below the MIP peak, it quickly deposits the remaining energy.

Hadronic Showers

Hadronic particles, can undergo additional interactions with the calorimeter material. These include scattering interactions with the nuclei of the calorimeter material. For nuclear interactions, the *nuclear interaction length* λ_l is roughly proportional to the atomic mass of the nuclei and further dependent on the size of the hadron through the interaction cross section. It is always bigger than the radiation length. In the case of iron about a factor of 10 (λ_l(Fe) $\approx$ 17 cm, X_0(Fe) $\approx$ 1.8 cm) [35]. In contrast to electromagnetic shower ($\mathcal{O}$(10cm)), a hadronic shower can thus span multiple meters. As hadronic calorimeters in consequence need to include a much larger volume, they are commonly implemented as sampling calorimeters.

The nuclear interactions can be very diverse and produce multiple different final states. An inelastic scattering between a hadron and a nucleus (*spallation*) can produce multiple pions and eject parts of the nucleus. The neutral pions again decay almost instantaneously into photons causing electromagnetic showers. The remaining energetically exited nucleus can radiate further parts of the nucleus (*fission*) or radiation (*evaporation*). The produced hadrons again undergo interactions with further nuclei. A hadron shower is thus exhibit a more variable and complex structure than an electromagnetic shower.

As the ration of hadronic and electromagnetic deposition is unknown, hadronic calorimeters are inherently less precise. This compounds with the additional losses through the design as a sampling calorimeter.

1.6 Jets

The initial collision, described by the *hard process*, frequently produces quarks. For hadron colliders, such quarks or gluons can also be radiated of even before the actual collisions (initial state radiation). These QCD particles themselves can produce a cascade of quarks and gluons through quark-gluon and triple quark interactions. At low energies the color-confinement comes into play and the quarks hadronize by generating further quark-antiquark pairs. The produced hadrons interact with the detector in hadronic showers [71].

This chain can approximately be reversed by *jet reconstruction algorithms*, to assign sprays of particles to initial products of the hard process. The particles grouped together by such an algorithm are then called a *jet*. These algorithms commonly work by repeatedly summing particles closest in pseudorapidity η and azimuthal angle Φ to mother particles until the closest remaining distance exceeds a cutoff distance, the jet radius. Different measures of distance define different algorithms such as the k_T- [72], the Cambridge/Aachen- [73] or the anti-k_T-algorithm [74].

Classifying the particle type of the initial particle from the jet energy depositions is referred to as *jet tagging*. It is another common application for Deep Learning in HEP [75, 76].

1.7 Monte Carlo Simulation

To compare the experimental data to theory prediction, a simulation of the full chain of hard process, QCD effects and detector interaction is needed. Due to the probabilistic nature of the processes, the simulation generally entails MCMC. This is the highly complex distribution of possible collision results at the detector is sampled via consecutive random draws. A defining feature of MCMC is, that the next simulation step can only depend on the results, that is the intermediate particles, of the previous. Treating the generation as a chain of splitting and interactions occurring with a probability determined by QFT and the Lagrangian of the theory is an intuitive way to think about the intractable distribution of energy depositions.

This simulation chain is commonly divided into individual, specialized parts:

- **Event generation:** Initially, the hard process is simulated to generate a sample of particles generated from the colliding particles, as well as their kinematics. On theory side, calculating probability of certain particles to be produced can include multiple processes and their interactions, corrections and integration over momenta of intermediate particles. Common event generators are MADGRAPH [77], PYTHIA8 [78], HERWIG [79] and SHERPA [80].

- **Parton showers and hadronization:** The chain of quark and gluon splittings and pair productions that lead to hadronized particles, as well as their decay, are commonly simulated as the next step. For hadronization two competing models exist, the String and the Cluster model. The event generators PYTHIA8, HERWIG and SHERPA include parton shower and hadronization simulations [78–80].

- **Detector simulation:** Lastly, tools like Geant4 [81] are used to simulate the energy deposition in detectors. This includes calculating the evolution of the particle shower by integrating the possibility of all available splittings for all particles over discrete steps. The stepsize is limited by the requirement that a single step cannot cross a calorimeter cell without interacting with the calorimeter material.

  The integration is implemented as an integration of the inverse mean free path length over a section of the trajectory

  $$n_\lambda = \int_{x'}^{x'+s} \frac{1}{\lambda(x)} dx \tag{1.13}$$

  Here the mean free path length $\lambda(x)$ by definition gives the distance in which the probability for no interaction to occur falls to $1/e$. The radiation length X_0 and the radiation nuclear interaction length λ_l are examples of interaction lengths of the different processes. Equivalently, the probability for an interaction to occur between x and $x+s$ is $1-e^{-n_\lambda}$. A stepsize likely to occur from step x under this probability, can then be sampled by drawing $u \sim \text{uniform}[0,1]$ as

  $$s = -\log(u) \cdot \lambda(x)\,. \tag{1.14}$$

  Geant4 executes this calculation for the mean free path lengths of all possible processes and chooses the smallest s. With the exponentially increasing number of particles in electromagnetic and hadronic showers, this simulation can get very costly. Every particle needs to be calculated individually without possibility of parallelization.

With the increase in luminosity with the HL-LHC, the cost of simulation increases by the same amount (5-7.5×), because the amount of simulation should match the number of observations as not to be limited by Poisson errors. For CMS, the upgrades to the tracker and calorimeter end caps further imply increased granularity. To increase the precision of the simulation to match the improved spatial resolution, the MCMC needs to apply smaller steps, adding to the rise in computational cost of the simulation. The computational need of the LHC experiments is thus predicted to outgrow the resources with the start of HL-LHC data taking in 2028 [10]. This motivates the development for cheaper, more precise simulation tools.

Multiple approaches for faster detector simulation have been developed [82]. These generally trade accuracy for faster simulation speed. If the resolution of the detector is known, the detector effects can be approximated by a smearing of the momenta of the incoming particles. As the resolution is energy dependent, the smearing is parameterized by the particle energy and the magnetic field applied. This approximation is useful for cheaply generating insights on the effect of theory parameters, but does not predict a detector response that can be compared to experimental results. A popular tool applying this approximation is Delphes [83].

Electromagnetic showers can also be approximated in terms of their longitudinal, radial, and azimuthal energy distribution marginals. Adaptations for hadronic showers or sampling calorimeters exist. Such shower parametrizations are included in Geant4 under the name GFlash or in the ATLAS toolkit `AtlFast3` as `FastCaloSim` [84].

Another, older approach is to use large libraries of precalculated showers. After a simulation of the trajectories of high energetic particles, these shower substitute the costly simulation for the low energetic depositions that make up the largest part of the shower [85].

Currently, the application of generative Machine Learning models trained on Geant4 simulation is widely discussed in the HEP community. Pioneered by the early adaption of Generative Adversarial Networks (GANs) to electromagnetic showers in References [86–88], GANs are now part of the ATLAS fast simulation tools (`FastCaloGAN` [82, 84]). Numerous studies have been conducted for the application of different generative architectures to both electromagnetic and hadronic shower simulations. These architectures include GANs, Variational Autoencoders (VAEs), Normalizing Flows and methods based on differential equations, such as Continuous Normalizing Flows and Diffusion Models. An exhaustive, regularly updated overview over these models is provided in Reference [89].

2 Deep Learning

The term *Deep Learning* (DL) has become a seemingly mystical set of tools promising of improved accuracy and seemingly new applications. When applying DL to squeeze the last percent out of a given system, we thus need to make sure to know its mechanisms and limitations. In HEP, the primary system is the LHC, and we are searching for every possibility to access any information in the reported data that has not yet been found. As such, the application of these tools comes with high hopes. We thus need to be very clear about what these modern tools can achieve. It is important to remember:

1. **Deep Learning is applied statistics:** Deep Learning, that is the application of *Deep Neural Networks*, is a sub-category to *Machine Learning* (ML) which itself is part of the broader idea of *Artificial Intelligence* (AI). All these terms describe statistical methods to make a computer program perform given tasks without giving specific rules in data space. They do so by iteratively adapting the parameters of a model to the data.

   Over the last 30 years we have seen a change in models driven by the increase in computational power and availability of ever-larger dataset. Before Large-Language Models were used for summarizing and classifying text, Latent Dirichlet Allocation was applied. Before (graph-)convolutional NNs were deployed for classification, much lighter Decision Trees were used. And before we used generative NNs for density estimation or sampling, histograms, Kernel Density Estimators (KDE) and Pair Copula Constructions were utilized.

   Mostly these methods are just different solutions to a common objective. As such, they share the underlying statistical phrasing of the task at hand and are limited to the well-definedness of the task. NNs will thus not open any groundbreaking paths for data analysis and handling, but rather enable us to apply statistical insight to bigger datasets and higher dimensional problems by leveraging current hardware developments.

2. **Results are only as good as the data:** The term "Artificial Intelligence" is misleading. Methods from this realm do not generate new insight about a problem. They perform a high dimensional fit to adapt a model to perform a task in the best possible way. So if the quality or amount of data is limited, the results will be biased through these limitations as well. And if the data is ambiguous or non-existent in some region in data space, the model results cannot be trusted at this region. Critically, NNs often still indicated high levels of certainty on their prediction in such settings.

3. **Neural Networks are a black box:** In the past, we have employed parametrized models carefully designed to best fit the task. Using a model that was to simple was detrimental to the performance as a lacked flexibility to describe the data. Similarly, using a complex model did also decrease performance through overfitting.

   With DL we enter the space of largely over-parametrized fits. This over-parametrization has been shown to increase performance and generalizability. The functional form of the model does not have to be specifically chosen for the task. This comes at the cost of interpretability. The output of a network cannot easily be connected to the input. The interpretation is lost in the number of parameters involved in the computation.

   Thus, to lift the curtain of the black-box, often methods relating the outputs to the inputs are employed, for example Shapley Values [90]. In Section 4 we advocate for Bayesian Learning as one alternative to produce more reliable NN applications through incorporating uncertainty in the learning procedure.

We will now introduce our mathematical formulation of NNs in 2.1, optimization objectives in 2.2 and stochastic optimization in 2.3, before discussing generative architectures in Section 3 and Bayesian NNs in Section 4. In this thesis, some notations might have different meaning in physics and ML context, for example the pseudorapidity η and the learning rate η_t. The distinction will be clear from context.

2.1 Neural Networks

A NN is a highly flexible family of functions consisting of nested linear functions (*neurons*) and non-linearities (*activation functions*), so-called layers. The family is characterized by the high-dimensional vector of network parameters ϑ. The number of parameters can range between hundreds, in small, efficiency-driven applications, to trillions for the newest Large-Language Models (as of September 2024 [91]). This large number of parameters warrants thinking about NNs in terms of non-parametric statistics.

The NN performs a mapping from a, possibly high-dimensional, space $\Omega_X \subseteq \mathbb{R}^M$ to a second space $\Omega_Y \subseteq \mathbb{R}^N$

$$f_\vartheta(\cdot) : \Omega_X \to \Omega_Y, \quad x \mapsto f_\vartheta(x)\,. \tag{2.1}$$

In general, architectures for dimensionality reduction as well as upscaling exist. Specific implementations however restrict the dimensions of this mapping, for example to $N = M$.

The most simple layer architecture is the *fully-connected* or *dense* layer. It can be written as a single matrix multiplication

$$y'_j = \mathrm{fc}(x'_i) := \sum_i w_{ji} x'_i + b_j\,. \tag{2.2}$$

Both, the weights $w \in \mathbb{R}^{N' \times M'}$ and the biases $b \in R^{N'}$ are part of the network parameters ϑ.

By considering the individual dimensions of the function, it can be separated into individual neurons. Broadly following the biological archetype, this function takes multiple (here M') inputs and transforms them into a single output variable. In biological neurons an output spike is triggered, once the accumulated inputs reach a specific threshold. The deep learning analog to this non-linearity is an activation function. Arguably the most simple, and still widely used, activation function is the *Rectified Linear Unit (ReLU)*

$$\mathrm{ReLU}(x') := \max(x', 0)\,. \tag{2.3}$$

While it is fast to compute, its derivative is undefined at $x' = 0$ and 0 for $x' < 0$. This can lead to vanishing gradients and stalling optimization. One solution is assigning a small negative slope for negative inputs (leaky ReLU) or switching to a different activation altogether. The *Exponential Linear Unit (ELU)*

$$\mathrm{ELU}(x') := \begin{cases} \alpha\left(e^{x'} - 1\right) & \text{if } x' \leq 0 \\ x' & \text{if } x' > 0 \end{cases}, \text{ with } \alpha > 0, \tag{2.4}$$

retains a small gradient for $x' < 0$ nad is differentiable for the default choice of $\alpha = 1$.

With these tools, a k-layer *Multi-Layer Perceptron* (MLP) is composed of alternating layers

$$f_\vartheta(x) = \mathrm{ELU}^{(k)} \circ \mathrm{fc}^{(k)} \circ \ldots \circ \mathrm{ELU}^{(1)} \circ \mathrm{fc}^{(1)}(y)\,. \tag{2.5}$$

For $k \geq 3$, this is a simple example of a Deep NN, that is a NN with hidden layers that are not immediately connected to input or output. Early literature on MLPs goes back to the 1950s [92].

Other commonly used layers include *convolutional* layers, *pooling* layers and *dropout* layers. The idea of concatenating linear and non-linear functions to parametrize a powerful family of functions remains the same, independent of the adaptation of the architecture to the data.

2.2 Loss Functions

To adapt the NN to a task at hand we need an empirical phrasing of the learning objective. We therefore consider a n-point dataset $\mathcal{D}_n$ of matched events $\{(x_i, y_i)\}_{i \in 1,\ldots,n}$ or unmatched events $\{x_i\}_{i \in 1,\ldots,n}$ and denote the corresponding *loss* function

$$L_n(\cdot) := L(\,\cdot\,; \mathcal{D}_n) : \Omega_\vartheta \to \mathbb{R}, \quad \vartheta \mapsto L(\vartheta; \mathcal{D}_n)\,. \tag{2.6}$$

This function has to be continuously differentiable over the full domain Ω_ϑ of possible parameter values to employ optimization using gradient descent (see Section 2.3). Here $\Omega_\vartheta \subseteq \mathbb{R}^P$ is a bounded set.

The choice of loss function needs to reflect the task. For classification of $C \in \mathbb{N}$ classes, traditionally a *cross entropy* (CE) loss is employed

$$\mathrm{CE}(\vartheta; \mathcal{D}_n) = -\sum_{i=1}^{n} \sum_{c=1}^{C} y_{i,c} \log\left(f_\vartheta(x_i)_c\right). \tag{2.7}$$

Here we assume $y_{i,c}$ gives the target probability for the result to be of class c and $f_\vartheta(x_i)_c$ gives a probability as well. To allow an interpretation of the ouput as a probability, commonly a softmax-normalization $\mathrm{softmax}(f_\vartheta(x_i)_c) = \exp f_\vartheta(x_i)_c / \sum_c \exp f_\vartheta(x_i)_c$ is applied in the last layer. For regression, a *mean squared error* (MSE)

$$\mathrm{MSE}(\vartheta; \mathcal{D}_n) = \frac{1}{n} \sum_{i=1}^{n} (y_i - f_\vartheta(x_i))^2 \tag{2.8}$$

often suffices. And for density estimation often Maximum Likelihood Estimation (MLE) is applied by maximizing the joint likelihood $\hat{p}(\mathcal{D}_n; \vartheta) = \prod_{i=1}^{n} \hat{p}(x_i; \vartheta)$ of the data under the model specified by ϑ. Here, the NN f_ϑ either acts as a parametrization of the likelihood distribution directly or predicts the parameters of a parametrization. A common parametrization of the likelihood for the latter case is a normal distribution with the network output defining mean and covariance. To prevent numerical nuisance caused by the large product, the logarithm of the joint likelihood is often used. Because the logarithm is a monotonic function, the maxima of the joint likelihood coincide with the minima of the *negative* log-*likelihood* (NLL)

$$\mathrm{NLL}(\vartheta; \mathcal{D}_n) = -\sum_{i=1}^{n} \log \hat{p}(x_i; \vartheta). \tag{2.9}$$

In fact, all of the objectives above are closely related to NLL-losses. The CE-loss is the expectation value of the NLL for

$$p(x_i \in c \mid \vartheta) = f_\vartheta(x_i)_c\,,$$

with respect to the actual probability distribution of the data. And the MSE is the NLL of a normal likelihood parametrization

$$p(x_i, y_i \mid \vartheta) = \mathcal{N}(f_\vartheta(x_i); y_i, \Sigma) = (2\pi)^{-M/2} \det(\Sigma)^{-1/2} \exp\left(-\frac{1}{2}(y_i - f_\vartheta(x_i))^\top \Sigma^{-1} (y_i - f_\vartheta(x_i))\right),$$

with a diagonal covariance of $\Sigma = \mathbb{1}$, after dropping the constant additive prefactor.

Multiple other loss objectives are employed in deep learning. Not all can be easily related to a likelihood of the data. We will for example encounter the idea of adversarial losses in Section 3. For the connection to Bayesian Learning however, the interpretation of the loss as a term derived from the likelihood of the data under the model is essential.

2.3 Stochastic Optimization

Given the model f_ϑ and a continuously differentiable loss $L(\,\cdot\,; \mathcal{D}_n)$, we can employ gradient descent to try and find the global minimum

$$\vartheta^* := \underset{\vartheta \in \Omega_\vartheta}{\operatorname{argmin}}\, L(\vartheta; \mathcal{D}_n). \tag{2.10}$$

This parameter setting will result in the best possible estimator $\hat{f} := f(\vartheta^*)$ of the true data generating function f. In general, for the convergence of the presented optimization algorithms, it is assumed that $f - \hat{f}$ is a convex function. This can only be proven for NNs in very specific cases. However, the optimization can usually be adapted to find a good estimator even for multi-modal distributions.

The idea of gradient descent is to follow the gradient of the loss landscape towards the minimum by iteratively performing update steps in this direction

$$\vartheta_{t+1} = \vartheta_t - b_t := \vartheta_t - \eta_t \nabla_\vartheta L_n(\vartheta_t)\,. \tag{2.11}$$

Here, $\eta > 0$ is the learning rate of the optimization. Using a low learning rate might artificially slow down the convergence and increase the risk of getting stuck in a local minimum of $L_n(\vartheta)$. A large learning rate on the other hand might hinder convergence by being insensitive to loss modes smaller than the stepsize. A decreasing learning rate scheduling $\eta_t \geq \eta_{t+1}$ can combine the advantages of both: Fast exploration after initialization, stability against small local minima and stable convergence to the minimum of the targeted mode in later parts of the optimization.

Calculating the gradient over the full batch of data is computationally intensive. Instead, a stochastic estimator based on random m-point subsets $\overline{\mathcal{D}}_m$ of the full dataset $\mathcal{D}_n$ is commonly used to construct an unbiased estimator $\hat{L}_m(\vartheta)$ of the full loss. Usually n/m disjoint subsets called *(mini-)batches* are constructed in the beginning of each iteration over $\mathcal{D}_n$. Each iteration over the full dataset is also referred to as one *epoch*. Using the gradient of this estimator for the update

$$\vartheta_{t+1} = \vartheta_t - \hat{b}_t := \vartheta_t - \eta_t \nabla_\vartheta \hat{L}_m(\vartheta_t)\,, \tag{2.12}$$

is also known as *stochastic gradient descent* (SGD) [93]. The statistical fluctuations of the gradient approximations help stabilize convergence of the optimization, as local minima vary between the different subsets.

Still, the optimization is susceptible to getting stuck at larger local minima and saddle points of the loss landscape. The optimization behavior can be further improved by including a running average of the previous steps

$$\hat{b}_t = \gamma \hat{b}_{t-1} + \eta_t \nabla_\vartheta \hat{L}_m(\vartheta_t)\,, \tag{2.13}$$

with $\gamma < 1$. The additional term is usually called a *momentum* term. It helps the network build up a direction of optimization and carry on even if the gradient vanishes at the current parameter configuration.

RMSprob

The effective stepsize $|\vartheta_{t+1} - \vartheta_t|$ of SGD (2.12) depends on the gradient. A consistent stepsize is thus hard to set, making the learning rate parameter hard to interpret. To improve the interpretability of the learning rate η_t, one can use a running average of the squared gradient

$$v_{t+1} = \alpha v_t + (1-\alpha) \nabla \hat{L}_m(\vartheta_t)^2\,, \tag{2.14}$$

with α regulating the exponential decay of the running average. This running average can be used to normalize the update steps

$$\vartheta_{t+1} = \vartheta_t - \eta_t \nabla_\vartheta \hat{L}_m(\vartheta_t) / \left(\sqrt{v_{t+1}} + \varepsilon\right)\,, \text{ with } \vartheta_0 = 0 \tag{2.15}$$

initially and $\varepsilon >$ a small parameter preventing division by zero. On average, the resulting update steps are of the same size over the course of opimization. The small constant ε avoids division by zero. The resulting algorithm is known as RMSprob [94] and works very well with non-stationary schemes, such as learning rate scheduling. It is a first example of gradient preconditioning and will find its equivalent for the MCMC algorithms of Section 4.2.

AdaGrad

By not distinguishing between individual parameter dimensions, the RMSprob algorithm as presented above struggles with sparse data and gradients that only seldomly generate non-zero updates on a subset of the weights. The popular AdaGrad algorithm [95] fixes this through the use of an element-wise (!) continuous sum over squared gradients for v

$$\begin{aligned} v_{t+1} &= v_t + \nabla \hat{L}_m(\vartheta_t) \odot \nabla \hat{L}_m(\vartheta_t) \\ \vartheta_{t+1} &= \vartheta_t - \eta_t \nabla_\vartheta \hat{L}_m(\vartheta_t) \oslash \left(\sqrt{v_{t+1}} + \varepsilon\right) \ . \end{aligned} \tag{2.16}$$

Here $\odot$ and $\oslash$ are element-wise multiplication and division. Sparsely updated parameters now have a larger effective stepsize than ones with regular, sizable parameter updates. However, the effective stepsize of AdaGrad also decreases strongly over time. Algorithms like AdaDelta [96] try to fix this by adapting the learning rate accordingly.

Current implementations of RMSprob, for example in PyTorch, also use a running average vector rather than scalar to adapt the learning rate of every parameter dimension individually. Here, no adaptation of the learning rate is needed as the running average does not accumulate over time. The resulting updates of the dimension-wise RMSprob depend strongly on the sign of the gradient in the respective direction, rather than its magnitude.

Adam

The most commonly used tool for stochastic optimization to this point is the Adam algorithm [97]. It combines the dimension-wise RMSprob with a momentum-like running average of the gradient and handles the bias from initializing the running average (2.15) at 0. Two parameters, β_1 and β_2, now control the exponential decay of the moving averages

$$\begin{aligned} m_{t+1} &= \beta_1 m_t + (1-\beta_1)\, \nabla \hat{L}_m(\vartheta_t) \\ v_{t+1} &= \beta_2 v_t + (1-\beta_2)\, \nabla \hat{L}_m(\vartheta_t) \odot \nabla \hat{L}_m(\vartheta_t) \\ \vartheta_{t+1} &= \vartheta_t - \eta_t \frac{m_{t+1}}{1-\beta_1^{t+1}} \oslash \left(\left(\frac{v_{t+1}}{1-\beta_2^{t+1}} \right)^{1/2} + \varepsilon \right) \ . \end{aligned} \tag{2.17}$$

The division by the decay parameters counteracts the initialization bias and a small parameter $\varepsilon > 0$ prevents division by zero. For this algorithm v_t is an estimate of the uncentered variance of the gradient. It is large, either if previous updates have been large in the same direction or if the fluctuation in the gradient is high. Thus two desirable features are achieved. Sparse data is handled through increased learning at low v_t and steps in directions of high variance are limited. Looking forward to Section 4.2, v_t can also be understood as an approximation to the diagonal of the Fisher matrix [98], a popular preconditioner for stochastic gradient MCMCs.

For Adam, the effective stepsize is bounded from above by the learning rate in most cases, making it easy to set. Sparse data is handled by treating every parameter dimension separately and saddle points do not pose an issue due to the use of a gradient average.

3 Generative Modeling

In generative ML, like in density estimation, we try to generate an estimator

$$\hat{p}_X(x;\vartheta) \quad \text{of the unobservable probability density} \quad p_X(x)$$

underlying the data of a training set $\mathcal{D}_n = \{x_i\}_{i\in 1,\dots,n}$. Here x_i are samples from the data-generating random variable X. As such, generative methods do not require matched datasets and are considered part of unsupervised learning. While for some methods $\hat{p}_X$ is analytically tractable and can be used for training with a NLL-loss (2.9), for others only samples $x' \sim \hat{p}_X$ of the estimate are accessible.

All generative DL methods have in common, that they construct the estimator as a mapping from samples of a random variable Z to those of X

$$f_\vartheta(\cdot) : \Omega_Z \to \Omega_X, \quad z \mapsto x' := f_\vartheta(z)\,. \tag{3.1}$$

The random variable Z is often referred to as a *latent variable* and its sample space Ω_Z as the *latent space* of the model. In most common applications, and all cases presented here, the probability density function of the latent variable is a standard normal distribution $p_Z(z) = \mathcal{N}(z; 0, \mathbb{1})$.

In this chapter, we cover the DL architectures relevant for understanding the publications. These include *Generative Adversarial Networks* (GANs, Section 3.1), *Variational Autoencoders* (VAEs, Section 3.2), and *Normalizing Flows* (NFs, Section 3.3). For completeness, the core concepts of *Diffusion Models* and *Transformers* are explained in Section 3.4. Data amplification is defined in Section 3.5 and discussed for a toy example in Section 3.6 and on calorimeter data in Section 3.7. The concept is revisited in Section 6.2 in the context of BNNs.

Note that classical methods for density estimation, such as histograms, KDEs or Pair Copula Constructions, allow easy sampling from the estimated distribution. They can therefore be understood as generative ML.

Kullback-Leibler and Jensen-Shannon Divergence

Before going into details about the employed architecture, let us quickly discuss how the performance of a generative network can be assessed. Estimating the quality of a generative network translates to comparing the similarity between $\hat{p}_X(x;\vartheta)$ and $p_X(x)$. To fix the notation, let us restrict the introduction to an application generating calorimeter images and assume X is a continuous random variable.

The *Kullback–Leibler divergence* (KLD)

$$D_{\mathrm{KL}}(g\,|\,q) = \int g(x) \log \frac{g(x)}{q(x)}\, \mathrm{d}x \tag{3.2}$$

is one way to measure the similarity of two continuous probability densities g and q. It can be written more generally in terms of probability measures. However, the important takeaway is that in order for KLD to be well-defined, g has to vanish wherever q is zero. The KLD is non-negative and 0 only if both distributions are identical. While this quantity evaluates the similarity of the distributions, it is neither symmetric nor satisfies the triangle inequality. It thus is not a metric.

To generate a metric based on the KLD, one can consider the square root of the symmetrized term

$$D_{\mathrm{JS}}(g,q) = \frac{1}{2} D_{\mathrm{KL}}\left(g \,\middle|\, \frac{g+q}{2}\right) + \frac{1}{2} D_{\mathrm{KL}}\left(q \,\middle|\, \frac{g+q}{2}\right)\,. \tag{3.3}$$

This term is referred to as the *Jensen-Shannon divergence* (JSD) and its square root as the *Jensen-Shannon distance*. The JSD inherits its lower bound from the KLD but is upper bounded by log(2). As we usually cannot access both the data distribution p_X and its estimator $\hat{p}_X$ analytically, the integration in Equation (3.2) cannot be performed. Instead, we often construct an estimator of the KLD or JSD from histograms of the true and generated data by summing over the bins. The quality of this estimate then strongly depends on the number of samples available.

3.1 Generative Adversarial Networks

One way to have a NN learn input-like data is through adversarial training [99]. The general idea is, to introduce a second network

$$d_\phi(\cdot) : \Omega_X \to [0,1], \quad x \mapsto d_\phi(x)\,, \tag{3.4}$$

which classifies input from the data space into generated and real data. This network is referred to as the *discriminator*, while the generative network in this setup is also referred to as a *generator*. For the data subset $\overline{\mathcal{D}}_m$, the discriminator is updated to minimize the cross entropy (2.7) of the classification

$$L_{\text{disc}}(\vartheta, \phi; \overline{\mathcal{D}}_m) := -\left(\sum_{i=1}^{m} \log(d_\phi(x_i)) + \log(1 - d_\phi(f_\vartheta(z_i)))\right)\,. \tag{3.5}$$

The generator is then optimized to maximize confusion in the discriminator by producing more data-like samples

$$L_{\text{gen}}(\vartheta, \phi; \overline{\mathcal{D}}_m) := -\left(\sum_{i=1}^{m} \log(1 - d_\phi(f_\vartheta(z_i)))\right)\,. \tag{3.6}$$

Both optimization steps are alternated during training to solve the min-max-problem

$$\min_\vartheta \max_\phi V(\vartheta, \phi) := \min_\vartheta \max_\phi \mathbb{E}_{x\sim p_X} \log(d_\phi(x)) + \mathbb{E}_{z\sim p_Z} \log(1 - d_\phi(f_\vartheta(z)))\,. \tag{3.7}$$

Note that, to be in line with literature, the sign of the previous cross entropy losses (3.5) and (3.6) is flipped here, and so is the minimization and maximization. It can be shown, in the non-parametric limit (infinitely large networks), given a Neyman-Pearson optimal [100, Theorem 3.87] discriminator

$$d_{\phi^*}(x) := \frac{p_X(x)}{p_X(x) + \hat{p}_X(x;\vartheta)} \tag{3.8}$$

the minimum of

$C(\vartheta) := V(\vartheta, \phi^*) = -\log(4) + 2D_{\text{JS}}(p_X(x), \hat{p}_X(x;\vartheta))$ is reached if and only if $p_X(x) = \hat{p}_X(x;\vartheta)$. Furthermore, adversarial training is proven to converge, if the discriminator is always updated until optimal according to Equation (3.8) before alternating to the generator optimization.

While this result seems quite interpretable, we need to bear in mind that an optimal classifier is assumed. In practice this assumption does not hold for multiple reasons. Most prominently, the non-parametric limit is never implemented and optimizing the discriminator to optimal performance in every iteration is computationally infeasible. GANs thus suffer multiple drawbacks. For one, the loss of a GAN is uninformative in practice. If training is not converging, it is unclear whether the performance of the generator or discriminator is at fault. Similarly, improvements in generation performance can be found with a simultaneous increase in loss if the discriminator improves more than the generator. Further, if the discriminator is too strong or to weak, it will give only small gradients for optimizing the generator. To achieve a stable convergence, the capabilities is both nets need to be balanced very well. In addition, the generator is not incentivized to learn the full distribution of p_X as the discriminator is only applied to individual samples. GANs thus often only to learn single modes of the distribution. This is called *mode collapse*.

3.2 Variational Autoencoders

While GANs usually expand low-dimensional latent data into high-dimensional output, Autoencoders [101] pursue the converse strategy: Compress data x into a dimensionality reduced representation $z = e_\phi(x)$ and use a simultaneously optimized decompression $x' = f_\vartheta(z)$ in combination with an estimator of the distribution of the reduced representation $\hat{p}_Z$ to generate new data. *Dimensionality reduction* here refers to $\dim \Omega_X > \dim \Omega_Z$. In autoencoding, a common loss determining the distance of the reconstructed sample $x' = f_\vartheta(e_\phi(x))$ from the original sample x is the MSE (2.8)

$$L_{\mathrm{AE}}(\vartheta, \phi; \overline{\mathcal{D}}_m) := \mathrm{MSE}(\vartheta, \phi; \overline{\mathcal{D}}_m) = \frac{1}{m} \sum_{i=1}^{m} \left(y_i - f_\vartheta(e_\phi(x_i))\right)^2 . \tag{3.9}$$

If the *encoder* e_ϕ and the *decoder* f_ϑ are perfectly aligned, that is $f_\vartheta \circ e_\phi = \mathrm{id}$, the loss is zero. However, due to the loss of information in the dimensionality reduction, this value can only be achieved if the sub-manifold of the data is of the same dimension as the latent space.

VAEs combine the scalability of autoencoding to high dimensions with the idea of Bayesian inference [102]. The aim is, once more, to find the parameters ϑ^* that minimize the NLL of the data under the generative part of the model $\sum_{i=1}^{n} -\log p_X(x_i; \vartheta)$. Following Bayes theorem (see Section 4 for an introduction), we understand the data marginal distribution as a combination of the conditional probabilities of data and continuous latent variables z

$$p_X(x; \vartheta) = \frac{p(x \mid z; \vartheta)\, p_Z(z)}{p(z \mid x; \vartheta)} = \frac{p(x, z; \vartheta)}{p(z \mid x; \vartheta)} . \tag{3.10}$$

Subscripts i are omitted for brevity. The NN decoder f_ϑ defines the likelihood distribution $p(x \mid z; \vartheta)$, while the posterior density $p(z \mid x; \vartheta)$ is intractable. In the following, we thus approximate this distribution with the distribution $\hat{p}(z \mid x; \phi)$ imposed by the encoder network e_ϕ. As $p_X(x; \vartheta)$ is independent of z and $\hat{p}(z \mid x; \phi)$ by construction integrates to 1, we can expand the NLL as

$$\begin{aligned}
-\log p_X(x; \vartheta) &= -\int \hat{p}(z \mid x; \phi) \log p_X(x; \vartheta)\, dz \\
&= -\int \hat{p}(z \mid x; \phi) \log \frac{p(x, z; \vartheta)}{p(z \mid x; \vartheta)}\, dz \\
&= -\int \hat{p}(z \mid x; \phi) \log \frac{p(x, z; \vartheta)}{p(z \mid x; \vartheta)} \frac{\hat{p}(z \mid x; \phi)}{\hat{p}(z \mid x; \phi)}\, dz \\
&= -\int \hat{p}(z \mid x; \phi) \left[\log \frac{p(x, z; \vartheta)}{\hat{p}(z \mid x; \phi)} + \log \frac{\hat{p}(z \mid x; \phi)}{p(z \mid x; \vartheta)}\right] dz \\
&= \underbrace{-\int \hat{p}(z \mid x; \phi) \log \left(\frac{p(x, z; \vartheta)}{\hat{p}(z \mid x; \phi)}\right) dz}_{= -D_{\mathrm{KL}}(\hat{p}(z \mid x; \phi) \mid p(x, z; \vartheta)) =: \mathrm{ELBO}} - D_{\mathrm{KL}}(\hat{p}(z \mid x; \phi) \mid p(z \mid x; \vartheta))
\end{aligned} \tag{3.11}$$

where we used Equation (3.10) and multiplied by 1. Because the KLD is non-negative, the first term gives a lower bound to the log-probability. This term is thus often referred to as the *evidence lower bound* (ELBO) and is solely maximized, as the second KLD term is inaccessible. By factoring the joint probability in the lower bound into the conditionals, it can itself be restructured as

$$\begin{aligned}
\mathrm{ELBO} &= \int \hat{p}(z \mid x; \phi) \log \left(\frac{p(z, x; \vartheta)}{\hat{p}(z \mid x; \phi)}\right) dz \\
&= \int \hat{p}(z \mid x; \phi) \log \left(\frac{p_Z(z)}{\hat{p}(z \mid x; \phi)}\right) + \hat{p}(z \mid x; \phi) \log p(x \mid z; \vartheta) dz \\
&= \int \hat{p}(z \mid x; \phi) \log p(x \mid z; \vartheta) dz - D_{\mathrm{KL}}(\hat{p}(z \mid x; \phi) \mid p_Z(z))
\end{aligned} \tag{3.12}$$

The two terms appearing are the log-likelihood as defined by the decoder and the KLD to the prior distribution in latent space. The last term can be understood as a regularization term that has to be balanced against the reconstruction power of the network. To decrease computational complexity, the posterior estimate is often parametrized as an uncorrelated normal distribution

$$\hat{p}(z \mid x; \phi) = \mathcal{N}(z; \boldsymbol{\mu}, \Sigma)\,, \text{ with } \Sigma = \operatorname{diag}(\boldsymbol{\sigma}_1, ..., \boldsymbol{\sigma}_{\dim \Omega_Z})\,. \tag{3.13}$$

The vector of means and variances is given by the encoder network $e_\phi(x) = (\boldsymbol{\mu}(x), \boldsymbol{\sigma}(x))^\top$. With the prior choice $p_Z(z) = \mathcal{N}(z; \mathbf{0}, \mathbb{1})$, the computation of the regularization can be executed as

$$\begin{aligned} D_{\mathrm{KL}}(\hat{p}(z \mid x; \phi) \mid p_Z(z)) &= D_{\mathrm{KL}}(\mathcal{N}(z; \boldsymbol{\mu}(x), \Sigma(x)) \mid \mathcal{N}(z; \mathbf{0}, \mathbb{1}) \\ &= \sum_{i=1}^{\dim \Omega_Z} D_{\mathrm{KL}}(\mathcal{N}(z; \boldsymbol{\mu}_i(x), \boldsymbol{\sigma}_i(x)) \mid \mathcal{N}(z; 0, 1)) \\ &= \frac{1}{2} \sum_{i=1}^{\dim \Omega_Z} \left(\sigma_i(x)^2 + \mu_i(x)^2 - 1 + \log(\sigma_i(x)^2)\right)\,. \end{aligned} \tag{3.14}$$

With this parametrization of the posterior estimate, the integration in the first term can be approximated by a summation over samples drawn from the normal distribution

$$\int \hat{p}(z \mid x; \phi) \log p(x \mid z; \vartheta) dz \approx \sum_{z \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)} \log p(x \mid z_j; \vartheta)\,. \tag{3.15}$$

Often, a single drawing suffices for the optimization of both networks. To ensure differentiability of the ELBO for the parameters of the encoder, the sampling $z \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ has to be recast as a sum of the encoder output

$$z = \boldsymbol{\mu} + \boldsymbol{\sigma}\epsilon\,, \text{ with a random element } \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbb{1})\,. \tag{3.16}$$

This can be seen as one instance of a more general *reparametrization trick* [102]. The same technique is also applied for BNNs in Section 4.1.

Classically, the likelihood is also parametrized as a normal distribution centered in x with covariance $\mathbb{1}$. The log-likelihood then reduces to an MSE. With these Gaussian approximations, the per-point ELBO loss is now easy to compute. The VAE loss is then derived from the ELBO by summation over the subset $\overline{\mathcal{D}}_m$

$$\begin{aligned} L_{\mathrm{VAE}}(\vartheta, \phi; \overline{\mathcal{D}}_m) &:= -\frac{n}{m} \sum_{i=1}^{m} \mathrm{ELBO} \\ &= -\frac{n}{m} \sum_{i=1}^{m} \left(\sum_{z \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)} \log p(x_i \mid z_j; \vartheta) + \frac{1}{2} \sum_{i=1}^{\dim \Omega_Z} \left(\sigma_i(x_i)^2 + \mu_i(x_i)^2 - 1 + \log(\sigma_i(x_i)^2)\right) \right)\,. \end{aligned} \tag{3.17}$$

After minimizing this loss with one of the gradient descent methods from Section 2.3, we are left with an encoder that tries to map any input onto a standard normal distribution and a decoder that can infer data like points from samples of the standard normal prior p_Z through

$$p_X(x; \vartheta) = \int p(x \mid z; \vartheta)\, p_Z(z)\, dz\,. \tag{3.18}$$

As for the GAN, there are multiple drawbacks of this method. The first, is the use of an MSE loss. One prominent local minimum of this loss, is the mean prediction over all data. VAE training is susceptible to getting stuck in this minimum. The resulting VAE only reproduces the mean of the data. Using a GAN-like discriminator on the decoder output instead of the MSE can improve this behavior [103]. This architecture is referred to as a VAE-GAN.

Furthermore, the modeling capacity of a VAE is limited through the strong regularization in latent space. By using an adversarial discriminator on the latent space, one can enforce arbitrary prior shapes [104]. Alternatively, one can use the data samples to estimate

$$\hat{p}_Z(z) := \int \hat{p}(z \mid x; \phi) p_X(x) \, dx$$

from the encoder mapping, for example by using an KDE [105, 106]. This relaxes the need for a strong regularization. It also addresses the issue that the latent space distribution can never actually be a standard Gaussian in order to retain information for the decoder.

The VAE can be understood as an application of an information bottleneck [107, 108]. This information theoretic reasoning, connects GANs, VAEs, VAE-GANs and more in a single framework [109]. It also motivates an architecture, the BIB-AE, that combines all the introduced features: A regularization of the latent space distribution and a reconstruction MSE term, as well as discriminator networks on both distributions. This model has been very influential in the initial development of fast calorimeter simulation with generative ML [106, 110, 111].

3.3 Normalizing Flows

Both GANs and VAEs learn to model a probability density indirectly, as the probability of the data under the model itself is not accessible. The class of NFs [112, 113] is by construction easily invertible, with inverse $g_\vartheta = f_\vartheta^{-1}$, and both directions are differentiable almost everywhere, with an easy to compute Jacobian determinant $\det \frac{\partial g_\vartheta}{\partial x}$. This implies $\dim \Omega_Z = \dim \Omega_X$. For simplicity, we use $\Omega_Z = \Omega_X = \mathbb{R}^N$.

With these features, the approximated data space density can then be calculated as the push-forward of the latent distribution

$$\hat{p}_X(x; \vartheta) = p_Z\left(g_\vartheta(x)\right) \left| \det \frac{\partial g_\vartheta}{\partial x} \right| . \tag{3.19}$$

In DL literature, this is often called the *change-of-variables* formula. The forward direction is often talked about as the *generating* direction, as it maps random samples from the latent space to data-like samples The inverse can be referred to as the *normalizing* direction. It maps data to a latent distribution of known shape.

The expectation of the data log-probability under the model can directly be used to optimize the model parameters with a subset-based approximation

$$L_{NF}(\vartheta; \overline{\mathcal{D}}_m) := -\sum_{i=1}^{m} \log \hat{p}_X(x_i; \vartheta) = -\sum_{i=1}^{m} \left(\log p_Z(g_\vartheta(x)) + \log \left| \det \frac{\partial g_\vartheta}{\partial x} \right| \right) . \tag{3.20}$$

This can also be understood as the ELBO of $D_{\mathrm{KL}}(p_X(x) \mid \hat{p}_X(x; \vartheta))$. For simplicity, we have assumed a fixed latent space distribution $p_Z(z)$. However, one can also use a parametrization $p_Z(z) := p_Z(z; w)$ that is optimized simultaneously with the flow parameters. In the following, we always enforce a standard normal distribution in latent space. In this case, the *log*-probability in latent space reduces to

$$\log p_Z(g_\vartheta(x)) = (g_\vartheta(x))^2/2 + \text{const.}.$$

The direct optimization of the log-probability as a loss makes flows very stable to train. It comes at the cost of limiting oneself to architectures that are easily invertible. This choice can be restrictive and introduces obstructions on the topology of the learned distribution. The flow mapping f_ϑ is a homoeomorphism and, as such, preserves the topological structure of the input space [114, 115]. See for example Reference [116] or [117] for techniques reducing this effect.

Flows can also be used to infer a conditional distribution $\hat{p}(x \mid c; \vartheta)$ given a dataset $\mathcal{D}_n = \{(x_i, c_i)\}_{i \in 1, \ldots, n}$, without further modification to the reasoning above [118–120].

Block-Based Normalizing Flows

A popular strategy to improve the expressive power of a flow is, to concatenate multiple bijective instances

$$f = f^{(N_B)} \circ f^{(N_B-1)} \circ ... \circ f^{(1)} \iff g = g^{(1)} \circ g^{(2)} \circ ... \circ g^{(N_B)} .$$

The compound flow then again is a bijection with a compound Jacobian determinant

$$\log\det \frac{\partial g(x)}{\partial x} = \sum_{i=1}^{N_B} \log\det \frac{\partial g^{(i)}(x^{(i)})}{\partial x^{(i)}} \,, \text{ where } x^{(i)} = g^{(i+1)} \circ ... \circ g^{(N_B)}(x) \text{ and } x^{(N_B)} = x \,. \tag{3.21}$$

Different choices of these individual blocks are feasible to construct a mapping that is cheap to invert and calculate the Jacobian determinant of. *Coupling flows*, first explored in Reference [121], split the input into two parts $z = (z^A, z^B)$ and use an invertible function h to construct the forward direction and inverse as

$$\begin{array}{l} x^A = h(z^A, s_\vartheta(z^B)) \\ x^B = z^B \end{array} \iff \begin{array}{l} z^A = h^{-1}(x^A; s_\vartheta(z^B)) \\ z^B = x^B \end{array} \,. \tag{3.22}$$

The Jacobian of this transformation is block-triangular and its determinant reduces to the determinant of the Jacobian of h. The function h is called the *coupling function* and its parameters are predicted by an arbitrarily complex NN s_ϑ. The NN is never inverted and does not need to be invertible.

Autoregressive Flows [122, 123] are another way of constructing a mapping with a triangular Jacobian. The autoregressive property relates to the dependence of the modelled output on the inputs. The N coupling functions of an autoregressive architecture can only use the previous entries of the input to predict the parameters of the coupling function

$$x_t = h(z_t; s_{t,\vartheta}(z_{t-1}, ..., z_1))) \,. \tag{3.23}$$

The normalizing direction of such a flow can be evaluated in a single pass using elaborate masking [124]. However, the generating direction requires sequential evaluation of all entries and is thus slow to evaluate. By exchanging the dependence on $z_{t-1}, ..., z_1$ through $x_{t-1}, ..., x_1$ in Equation (3.23), the converse can be achieved [123]. Still, one direction of an autoregressive flow will always be slow to evaluate. Both, coupling and autoregressive flows, benefit from using random permutations between the blocks to ensure mixing of all dimensions. In addition to the Jacobian structure, the coupling functions themselves can be chosen. Popular choices are affine coupling functions $h(z, \vartheta) = \vartheta_1 z + \vartheta_2$ [121, 123–126] and rational quadratic splines [127].

In contrast to GANs and VAEs, flows lack any kind of dimensionality reduction. They are therefore parameter-intensive and slow in comparison. To reduce the computational complexity, *residual flows* [128, 129] can be constructed by applying the same block multiple times

$$x^{(i+1)} = f^{(i)}(x^{(i)}) := x^{(i)} + f_{\text{res}}(x^{(i)}) \,. \tag{3.24}$$

However, the Jacobians of proposed residual flows are inefficient to compute, introducing a different bottleneck for computation. Thinking of residual flows with an infinite number of blocks, that is $i \in \{1, ..., N_B\} \to t \in [0, 1]$, motivates the use of an ordinary differential equation (ODE) to define the flow mapping.

Continuous Normalizing Flows

First introduced in Reference [130], *Continuous Normalizing Flows* (CNFs) define a flow transformation $f\colon \mathbb{R}^N \times [0,1] \to \mathbb{R}^N$ dependent on a continuous *time* t. For simplicity, we write $x(t) := f(x,t)$ for the transformed variable and set $f_0(x) = x$. Instead of having multiple flow instances, the dependence of f on t is defined through the ODE

$$dx(t) = v(x(t), t)dt \,, \tag{3.25}$$

by the time dependent *vector-field* $v\colon \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$. This vector-field itself is parametrized as a NN

$$\tilde{v}_\vartheta(\cdot, t) = v(\cdot, t)\,.$$

This network can in principle be arbitrarily complex. By convention, the flow transforms data from a latent space distribution, usually $\mathcal{N}(0,1)$, for $t = 0$ into data-like output at $t = 1$. This choice sets the boundaries of the *probability path* $p\colon \mathbb{R}^d \times [0,1] \to \mathbb{R}_{>0}$ induced by the vector-field trough Equation (3.25). The change in probability distribution for a continuous transformation in time follows another ODE [130]

$$\frac{\partial}{\partial t} \log p(x(t), t) = -\mathrm{tr}\left(\frac{d\tilde{v}_\vartheta(x(t), t)}{dx(t)}\right)\,, \tag{3.26}$$

the change of variables for continuous t. While the calculation of the discretized change of variables (3.19) requires the calculation of the Jacobian determinant, in the continuous case we only need to calculate the trace. This allows us to freely choose the network $\tilde{v}_\vartheta(\cdot, t)$ as it does not need to be invertible.

To optimize the log-probability with stochastic gradient descent methods, we need to calculate the derivative of the log-probability from Equation (3.26) in the parameters ϑ of $\tilde{v}_\vartheta(\cdot, t)$. This requires the evaluation of [130]

$$\frac{d}{d\vartheta} \log p(x(t), t) = -\int_0^1 \left(\frac{\partial \log p(x(t), t)}{\partial x(t)}\right)^\top \frac{\partial \tilde{v}_\vartheta(x(t); t)}{\partial \vartheta}\, dt\,. \tag{3.27}$$

To minimize the NLL of input data at $t = 1$, we need to solve this ODE equation at every step of the training through discrete numerical integration schemes with a sufficient number of steps. This is especially expensive for higher dimensional models, because both, the trace operation and the ODE solving, scale linearly in N. The cost of the trace calculation can be reduced by using a Monte Carlo estimate [131].

To facilitate a more efficient training, the *Conditional Flow Matching* (CFM) objective [132–134]

$$L_{\mathrm{CFM}}(\vartheta) = \mathbb{E}_{t, x(t=1), x} \left\| u_t(x|x(t=1)) - \tilde{v}_\vartheta(x; t) \right\|^2 \tag{3.28}$$

can be applied. The samples are drawn as $t \sim \mathrm{uniform}(0,1)$, $x(t=1) \sim p_X$ and $x \sim p(x, t \mid x(t=1))$. This loss avoids solving the ODE during training altogether and enables the scaling of CNFs to very high dimensions. It reduces the calculation of the optimization criterion to the calculation of a mean-squared error between the network output $\tilde{v}_t(x; \vartheta)$ and an analytical solution u_t. A good choice of u_t and corresponding $p(x,t)$ is a Gaussian conditional probability path with mean and variance changing linear in time (optimal transport) [132]. The CFM-loss (3.28) then reduces even further to

$$L_{\mathrm{CFM-OT}}(\vartheta) = \mathbb{E}_{t, x(t=1), x_0} \bigg\| \left(x(t=1) - (1 - \sigma_{\min})\, x_0\right) - \tilde{v}_\vartheta(\sigma_t x_0 + \mu_t; t) \bigg\|^2, \tag{3.29}$$

where $\mu_t = t x(t=1)$, $\sigma_t = 1 - (1 - \sigma_{\min})t$, $x_0 \sim p(x_0) = \mathcal{N}(x_0; 0, 1)$ and $\sigma_{\min}$ a small parameter, that can be chosen to match the noise level of the training data.

As the generative direction of a CNF employs an ODE solver, CNFs, be it with or without Flow Matching, are slower in generation than discrete flows of similar size. However, CNF training has shown to scale more efficiently with the number of network parameters [130] and require less parameters for a given task overall.

3.4 Diffusion Models and Transformers

Diffusion Models

CNFs are closely related to Diffusion Models [135–137], a class of generative architectures that has been very popular in prompt-to-image generation (see amongst other References [138, 139]). The differential equation at the core of Diffusion Models differs from Equation (3.25) only through the addition of a noise term

$$dx(t) = v(x(t), t)dt + \sigma(x(t), t)dW(t)\,. \tag{3.30}$$

Here, $W(t)$ is the Wiener process (Brownian motion) in N dimensions, v the *drift* coefficient and σ the *diffusion* coefficient. The change in the time dependent probability density now is described by the Fokker-Planck equation (also Kolmogorov's forward equation)

$$\frac{\partial}{\partial t} p(x(t), t) = -\nabla_x \big(v(x(t), t) p(x(t), t)\big) + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\partial^2}{\partial x_i \partial x_j} \big(D_{ij}(x, t) p(x(t), t)\big)\,, \tag{3.31}$$

with $D(x, t) = \frac{1}{2}\sigma\sigma^\top$. A similar equation (Kolmogorov's backward equation) can be used for the backwards direction. This can be understood as adding noise until the data fully follows the noise distribution (diffusion direction) or subtracting noise to generate data like output (generative direction).

Different implementations of this idea exist. Denoising diffusion probabilistic models [136] in generative direction add Gaussian noise at fixed intervals with mean and variance predicted by a NN dependent on the previous step. In diffusion direction, fixed Gaussian noise is added. The model is trained on an ELBO of the data log-probability. This technique is computationally intensive, as it does not leverage advances in modern differential equation solvers, but relies on multiple equal length steps for the solution of the stochastic differential equation (SDE).

Score matching [140–142] poses a different way of constructing and optimizing a Diffusion Model. The term *score* refers to the gradient of the log-probability of the underlying data distribution

$$s(x) = \nabla_x \log p_X(x)\,. \tag{3.32}$$

Score matching models approximate the score with a NN $\hat{s}_\vartheta(x) \approx s(x)$ by optimizing another lower bound. The score approximation can then be used to solve the reverse SDE

$$dx(t) = \left[v(x(t), t) - \sigma(x(t), t)^2 s(x(t))\right] dt + \sigma(x(t), t)d\tilde{W}(t)\,, \tag{3.33}$$

where $\tilde{W}(t)$ is the time reversed Wiener process. This equation can then be solved with numerical SDE solvers. A study of different solvers available can be found in Reference [143].

Similar to CNFs, the drift $v(x(t), t)$ can also be used to model the diffusion process. In this approach, the CFM loss (3.29) can be employed for SDE based sampling as well.

Autoregressive Transformers

Autoregressive Transformers are fundamentally different from the ODE/SDE based architectures introduced previously. The autoregressive property (see Equation (3.23)) efficiently relates new input to previous input through the use of an attention mechanism [144]. This has been very popular in Large-Language Models, such as ChatGPT [91], and can be leveraged for the cascade-like structure of particle showers [17, 145].

Again, the autoregressive property refers to a modeling, in which the output can only depend on previously seen input. For a data point $x \in \mathbb{R}^N$ this refers to a factorization

$$\hat{p}(x; \vartheta) = \prod_{i=1}^{N} \hat{p}(x_i \mid x_1, ..., x_{i-1}; \vartheta_i)\,. \tag{3.34}$$

The approximated distribution per dimension can be parametrized for example as a Gaussian mixture model [17], with parameters predicted by a NN using the attention mechanism. For such a construction, the NLL of the data can be easily evaluated and used for optimization of the network parameters.

3.5 Data Amplification

If we want to use the previously introduced methods to accelerate detector simulations, there are two factors that determine the success of this undertaking:

- **Generation speed:** The idea of emulating MC simulation with generative DL is based on the better scaling to high dimensional data spaces of NNs over GEANT4. This creates the opportunity to generate a speed-up in simulation. The concrete speed varies between the different models, with architectures based on differential equations in general being slower than more classical architectures. This speed-up has always been part of the discussion around fast detector simulation, such as in References [110, 111, 146–148].
- **Data quality:** A lesser discussed factor is the quality of the generated data, that is the alignment between $\hat{p}_X(x;\vartheta)$ and $p_X(x)$. This alignment depends strongly on the number of points used for training the model, as well as the network type and training. If we can simulate data very fast, but the systematic bias introduced by the misalignment with the true data is larger than the benefits we gain from suppressing stochastic variance through simulating more points, we did not gain an advantage.

In the following, and throughout this work, we will focus on the latter aspect of generating artificial data. In detail, we will focus on data amplification.

Definition

Let $o : \Omega_X \to \Omega_O$ be an *observable* of a data sample, with Ω_O the discrete or continuous space of observable values. This could for example be the visible energy in a calorimeter image. The observable values follow p_O, the true distribution enforced by the mapping of the random variable X. For the purpose of discussing the generation quality in data space, this mapping can be chosen to be the identity mapping.

For D_{Ω_O} the set of all distributions over observable space, $h : D_{\Omega_O} \to \mathbb{R}^H$ extracts a vector of *properties* of the observable's distribution. Illustrative examples of such properties are the mean, variance or histogram bin counts. As the underlying distribution is not accessible, and thus sampled with complex MCMC algorithms, usually a numerical estimator $\hat{h} \circ o : \Omega_X^l \to \mathbb{R}^H$, is employed. The exponent l here denotes the arbitrary size of a set sampled from data space and will be replaced by a more specific variable name whenever possible. Similar definitions can also be found in statistics literature [149].

Using a measurement of distance $\mathcal{M} : \mathbb{R}^H \times \mathbb{R}^H \to \mathbb{R}$, where a smaller value indicates better agreement, we can formally define the data amplification A. One possible choice of $\mathcal{M}$, used in the next section, is the MSE. The amplification is the ratio of true data points n_{true}, that equals the high statistics limit of the emulator data in $\mathcal{M}$, and the number of training samples n

$$A := \frac{a}{n}\,, \text{ with } a := \min\left(n_{\text{true}} \in \mathbb{N} \,\middle|\, \underbrace{\mathop{\mathbb{E}}_{\mathcal{D}_{n_{\text{true}}} \sim p_X^{n_{\text{true}}}} \left[\mathcal{M}\left(\hat{h}\left(o(\mathcal{D}_{n_{\text{true}}})\right), h(p_O)\right)\right]}_{\mathcal{M}_{\text{true}}} \right. \\ \left. \leq \underbrace{\lim_{n_{\text{gen}} \to \infty} \mathop{\mathbb{E}}_{\mathcal{D}_{n_{\text{gen}}} \sim \hat{p}_X^{n_{\text{gen}}}} \left[\mathcal{M}\left(\hat{h}\left(o(\mathcal{D}_{n_{\text{gen}}})\right), h(p_O)\right)\right]}_{\mathcal{M}_{\text{gen}}} \right). \tag{3.35}$$

If the amplification is larger than 1, the generated samples do improve the estimation of the observable distribution p_O in this measurement of distance. For cases where p_X is not accessible, p_O is likewise not accessible. In these cases, we thus approximate $h(p_O) \approx \hat{h} \circ o(\mathcal{D}_{n_{\text{val}}})$ for $\mathcal{D}_{n_{\text{val}}} \sim p_X^{n_{\text{val}}}$ with a validation set size much bigger than the number of data points compared against $n_{\text{true}} \ll n_{\text{val}}$. The definition already gives us some intuition to the pitfalls of discussing amplification. The reported value is dependent on the observable, the distribution property and the chosen distance measure. A generalization to different choices is non-trivial.

Training Data Limits the Available Information

Fundamentally, the amount of information provided can never surpass the level present in the training data [150]. As a consequence, if the hypothesis test performed in the subsequent analysis uses the given data optimally (for example in the sense of the Neyman-Pearson lemma), generated data cannot improve the analysis. Oversampling a generative NN should in many cases be understood as a way of improving the analysis, rather than an augmentation of the dataset. Due to the inductive bias of a NN, that is the inherent smoothness conditions [151], the estimate of the data underlying distribution outperforms other methods based on more traditional density estimation [P1]. For a numerical proof we refer to Section 3.7 and Reference [P1].

This limits the tasks in which amplification can be pursued. In HEP, we compare simulated data to experimental data to decipher the nature of the hard processes in the collision. The cascade of particle emissions following the collision and the limited detector resolution smear our view of these processes, but are time consuming to simulate. It is common practice to optimize the nuisance parameters of both detector simulation and hadronization to better fit the experimental data. The enforced smearing through the detector model thus, even in a full simulation pass, is an approximation. While we cannot amplify information from the hard process, replacing the smearing in the detector with a much faster DL emulator is a much more feasible prospect and analogous to other fast simulation approaches introduced in Section 1.7.

Comparing Bias against Variance

Independent of the amount of samples drawn from the generative NN, the fit $\hat{p}_X(x;\vartheta) \approx p_X(x)$ will never be perfect. The network approximation is always biased by the statistical limitations of the training data. However, in the infinite sample limit the variance of the estimator $\hat{h} \circ o$ goes to zero. For an unbiased estimator, the limit thus reduces to

$$\begin{aligned} \mathcal{M}_{\text{gen}} &= \lim_{n_{\text{gen}} \to \infty} \mathop{\mathbb{E}}_{\mathcal{D}_{n_{\text{gen}}} \sim \hat{p}_X^{n_{\text{gen}}}} \left[\mathcal{M}\left(\hat{h}\left(o(\mathcal{D}_{n_{\text{gen}}}) \right),\, h(p_O) \right) \right] \\ &= \lim_{n_{\text{gen}} \to \infty} \mathcal{M}\left(\hat{h}\left(o(\mathcal{D}_{n_{\text{gen}}}) \right),\, h(p_O) \right) = \mathcal{M}\left(h(\hat{p}_O),\, h(p_O) \right) , \end{aligned} \tag{3.36}$$

where $\hat{p}_O(o(x), \vartheta)$ is the distribution of the generated samples mapped to the observable.

On the other hand, sampling an infinite amount n_{true} from the true data distribution $\mathcal{M}_{\text{true}}$ will approach the minimal value of $\mathcal{M}$, if $\hat{h} \circ o$ is constructed as an unbiased estimator of $h(p_O)$. A finite amount of samples however will lead to high variance in the estimator. This variance is captured by the distance measurement $\mathcal{M}_{\text{true}}$ and leads to an increased value.

With the amplification setup we are thus comparing two different effects: The bias caused by the statistical fluctuations and to some extent remedied by the inductive bias of the NN against the statistical fluctuations of an estimator constructed from a finite sample.

3.6 GANplification

The first experiments on amplification with specific focus on particle physics are conducted in Reference [1]. While we ultimately want to prove amplification for calorimeter emulation, these first experiments are conducted on a toy example. The authors investigate emulating a one-dimensional camel-back distribution

$$p_X(x) = \frac{\mathcal{N}(x;-4,1) + \mathcal{N}(x;4,1)}{2} ,$$

as well as its multidimensional generalizations defined in spherical coordinates through

$$p_X(x) = p_X(r,\varphi) = (\mathcal{N}(r;-4,1) + \mathcal{N}(r;4,1)) \times \text{uniform}(\varphi;0,\pi) .$$

These distributions are simplification of the typical resolution encountered for Breit-Wigner propagators of intermediate particles [152].

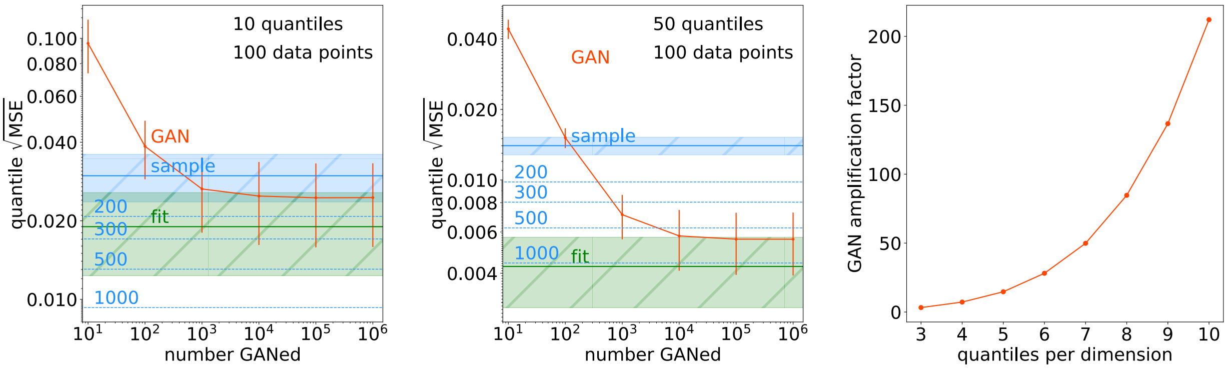


Figure 5: Left and middle: Distance to the true data distribution (one-dimensional camel-back) for a histogram estimator based on different numbers of samples drawn from a GAN distribution (red). For comparison the quality of estimators based on 100, 200, 300, 500 and 1000 samples from the data distribution (blue) and for a functional fit with a double Gaussian (green) is given. We find increasing the number of quantiles evaluated from 10 (left) to 50 (middle) seemingly increases the possible amplification. The error bands show the standard deviation of 100 independently sampled sets and consequent fits or GAN trainings respectively.
Right: Increase in amplification with the number of quantiles in the evaluation for a 5-dimensional hypersphere (500 training points). Images taken from Reference [1].

For the one- and two-dimensional experiments a GAN is trained on 100 samples. For the concluding 5-dimensional study, 500 samples are used. The employed GAN is fairly large, at a latent dimension of 1000, seven fully-connected layers of 256 nodes for the generator and a DeepSets-like discriminator [153, 154] using 3 equally sizeable convolutional layers. The GAN is optimized for 10000 epochs to generate the best possible approximation $\hat{p}_X(x, \vartheta)$. For more details, also see Reference [1] itself.

With the perspective of eventually amplifying particle showers in detectors, judging the amplification on the full distribution rather than on defined observables is of interest. As the toy example is low-dimensional, this is also computationally feasible. The authors thus choose $o(x) = x$ and histogram bin values for h. To avoid effects from arbitrarily choosing the edges of the histogram, they opt for bins with equal expected numbers of sample points from the truth distribution. We will refer to bins of equal probability mass as *quantiles* Q_i in the following and denote the set of quantiles as $\boldsymbol{Q} = \{Q_1, ..., Q_{n_\text{quant}}\}$. The distribution properties are then

$$\hat{h}_i(o(\mathcal{D})) = \hat{h}_i(\mathcal{D}) = \frac{\#\{x' \in \mathcal{D} \mid x' \in Q_i\}}{\#\mathcal{D}} \,, \tag{3.37}$$

which is the Monte Carlo estimate of

$$h_i(p'_X) = \int_{Q_i} p'_X(x)\, dx \,.$$

They evaluate the distance to the true distribution using a MSE

$$\mathcal{M}(\hat{h}(o(\mathcal{D}_{n_\text{gen}})), h(p_O)) = \mathcal{M}(\hat{h}(\mathcal{D}_{n_\text{gen}})) = \frac{1}{n_\text{quant}} \sum_{i=1}^{n_\text{quant}} \left(\hat{h}_i(\mathcal{D}_{n_\text{gen}}) - \frac{1}{n_\text{quant}} \right)^2 \,, \tag{3.38}$$

where we used that $h_i(p_X) = 1/n_\text{quant}$ by the definition of the quantiles. Interestingly, this measure differs from the χ^2-test statistic only through a factor of n^2_quant. As such, it appears on both sides of the inequality (3.35) and cancels. The amplification estimate is thus indifferent to the exchange of the MSE and χ^2 statistic. The same is true for taking arbitrary powers of $\mathcal{M}$, additive constants or factors.

Sparse Bins

In this setup the authors numerically verify amplification with a generative network. They find large samples from the GAN distribution behave similar to those of a fit using the correct functional shape. As expected,

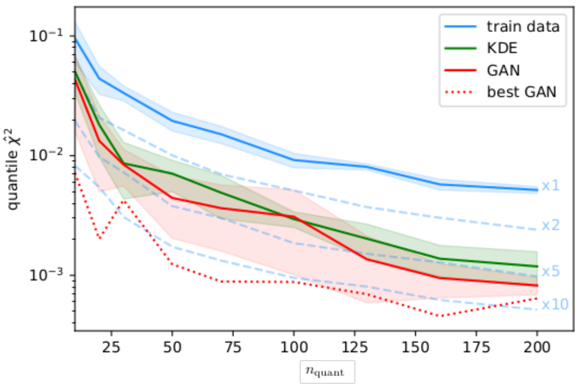


Figure 6: Distance to the true data distribution (Gaussian ring) for a histogram estimator with increasing number of bins based on samples drawn from a GAN (red) or KDE density estimator (green). This time we scale the size of the training data (solid blue) and the validation sets (dashed blue) with $n \propto n_{\rm quant}^2$. Thus, the variance of the estimator on limited data decreases to the right in the same way as for the oversampled artificial data ($n_{\rm gen} = 100 \cdot n$).

the quality of the estimator based on GAN samples surpasses the quality of one only using the training data at around 10× oversampling (compare Figure 5).

However, they also find a strong dependence on $n_{\rm quant}$. This observation only gives limited information on the quality of the GAN fit. It rather shows the effect of the increasing variance of the estimator $\hat{h}\,(o(\mathcal{D}_{n_{\rm true}}))$ due to limited true data on $\mathcal{M}_{\rm true}$. For example, for the left panel of Figure 5 (10 quantiles, 100 data points) the probability for the Poisson distributed bin value to be 0 is only $4.5 \cdot 10^{-5}$, while for the middle panel (50 quantiles, 100 data points) it is 13%. It is thus crucial to report amplification in realistic a setting, as the number for a histogram based distance measure can always be inflated artificially.

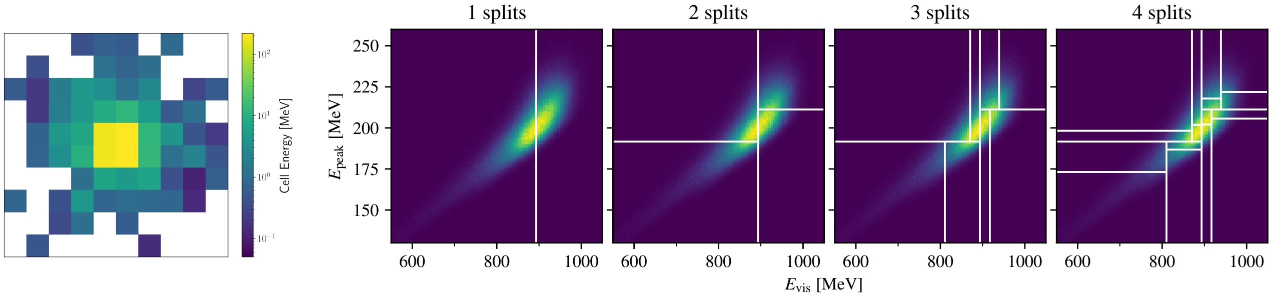
The increase in variance for increasing $n_{\rm quant}$ can be circumvented by scaling the number of training points with number of quantiles in the evaluation. When scaling the number of training points $n \propto n_{\rm quant}^2$ the variance of the estimator $\hat{h}\,(o(\mathcal{D}_n))$ converges to 0 as well. We repeat the experiments from Reference [1] for the 2-dimensional Gaussian ring with the same GAN setup, but scale the training data and retrain when evaluating higher numbers of quantiles. To make sure we are close to the low-variance limit $n_{\rm gen} \to \infty$, we always sample $n_{\rm gen} = 100 \cdot n$ samples with the GAN.

In contrast to the paper, in Figure 6 we use the χ^2 measure which differs from the previously shown MSE results through a factor of $n_{\rm quant}^2$ and leaves the amplification estimate invariant as argued above. We find that without the effect of sparse data, the GAN reports an amplification of approximately 5 over the full range of quantile numbers. However, we also find similar, albeit slightly lower, amplification from a KDE at significantly lower computational cost.

3.7 Calomplification [P1]

As discussed before, our goal is to prove the merit of replacing the costly MCMC calorimeter simulation in the high-energy physics simulation pipeline with a DL emulator. We do so in Reference [P1] by demonstrating amplification for calorimeter images.

For this numerical experiment, we simulate calorimeter depositions of 50 GeV photons using Geant4 [155]. The detector design used in the simulation is the electromagnetic calorimeter of the ILD [156], a 30 layer silicon-tungsten (Si-W) sampling calorimeter proposed for the ILC. The full dataset contains 268k showers of $30 \times 30 \times 30$ calorimeter cells. As the density $p_X(x)$ is not available in closed form, we hold $n_{\rm val} = 218$k showers back as a validation sample. Of the remaining 50k samples, we only use sets of 1k for NN training and compare the NN generated data against up to $n_{\rm true} \leq 50$k true data points. To facilitate training from such a small sample, we reduce the dimensionality of the data to 10×10 pixels by summing along the propagation axis of



(a) Calorimeter image

(b) Approximating quantiles through iterative splitting.

Figure 7: Experimental setup for the detector amplification study. Figure 7a shows a preprocessed calorimeter image after pooling. Figure 7b shows the iterative procedure for finding the approximate position of the quantiles by iteratively splitting the validation set into subsets of equal size. Figures originally published in Reference [P1]. Figures originally published in Reference [P1].

the incident photon and pooling the resulting image in patches of 3×3. One point in the training data is show in Figure 7a.

As a surrogate model, we train a VAE-GAN [103]. The generator and discriminator layer architecture is inspired by the LAGAN [157]. Both feature locally connected layers, a more flexible version of convolutional layers. The discriminator is trained using spectral norm [158] and label smoothing, meaning artificial noise is added to the true-vs-generated label. The encoder uses three convolutional layers followed by one linear layer. This setup is a simplification of the BIB-AE network developed specifically for precision simulation of particle showers [106, 110, 111]. We train the full setup for 50000 epochs and chose the best epoch amongst the last 10000 epochs by calculating the mean $\mathcal{M}_{\mathrm{gen}}$ for h as in Equation (3.37). For the selection, we use constant width bins and o the common observables of calorimeter images specified below. To gauge the stability of the training and the validity of the results, we train 3 instances of the VAE-GAN and sample in equal proportions from them.

For the evaluation, we use five common observables for calorimeter images to circumvent the computational complexity of evaluating the 100-dimensional data space. With x'_{ab} a pixel (energy value) of calorimeter image $x' \in \mathbb{R}^{10} \times \mathbb{R}^{10}$, these observables are

$$\begin{aligned}
\text{visible energy} \quad & o_1(x') = E_{\mathrm{vis}}(x') = \textstyle\sum_{a,b} x'_{ab}\,, \\
\text{peak energy} \quad & o_2(x') = E_{\mathrm{peak}}(x') = \max_{a,b} x'_{ab}\,, \\
\text{per-pixel energy} \quad & o_3(x') = E_{\mathrm{pixel}}(x') = (x'_{1\,1}, \ldots, x'_{10\,10})^\top\,, \\
\text{center of gravity in } x\text{-dimension} \quad & o_4(x') = \mathrm{cg}_x(x') = \textstyle\sum_a a\, \frac{\sum_b x'_{ab}}{E_{\mathrm{vis}}(x')}\,, \\
\text{and center of gravity in } y\text{-dimension} \quad & o_5(x') = \mathrm{cg}_y(x') = \textstyle\sum_b b\, \frac{\sum_a x'_{ab}}{E_{\mathrm{vis}}(x')}\,.
\end{aligned}$$

From the validation data we construct a set of quantiles $\boldsymbol{Q} = \{Q_1, \ldots, Q_{n_{\mathrm{quant}}}\}$ similar to the previous section. Because no closed form of either of p_O is available, we construct the quantiles by dividing the validation set into equal size halve, which we further, iteratively, divide into halves themselves. This procedure constructs bins of approximately equal probability mass over one or multiple dimensions. When examining the joint distributions of the observables and their NN surrogate, we alternate the dimensions we split the sets in. Figure 7b shows the iterative construction of the quantiles for the joint distribution $p_{O_{1,2}}$. As in Equation (3.37), we again use the relative population of the quantiles as the distribution properties h of interest.

Under the assumption of an optimal discriminator, the adversarial training optimizes the JSD between $p_X(x)$ and $\hat{p}_X(x;\vartheta)$, see Section 3.1. We thus change the measurement of distance from the previously used MSE/χ^2

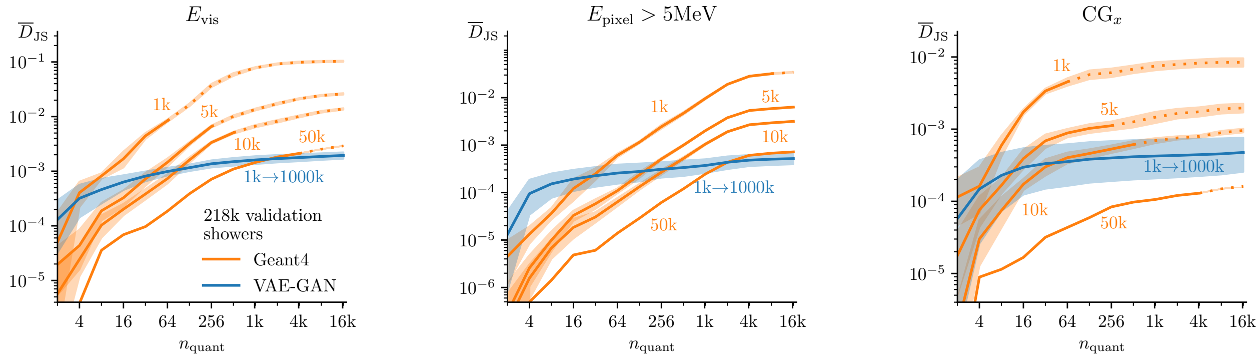


Figure 8: Distance to the observable distribution of the true data (MCMC simulated photon showers) for a histogram estimator with increasing number of bins based on samples drawn from a VAE-GAN (blue). For comparison, we also show the results for histogram estimators calculated from sets of true data in 4 different sizes (orange). Error envelopes are calculated from 5 distinct sets of data and subsequent VAE-GAN training. The observables shown are the visible energy, all cell energies and the center of gravity in x-dimension. These Figures were originally published in Reference [P1].

to a JSD based measurement

$$\begin{aligned}\overline{D}_\mathrm{JS}(\hat{h}(o(\mathcal{D}_{n_\mathrm{gen}})), h(p_O)) &:= \mathcal{M}(\hat{h}(o(\mathcal{D}_{n_\mathrm{gen}})), h(p_O)) \\ &= \frac{1}{2}\sum_{i=1}^{n_\mathrm{quant}}\left(\hat{h}_i(o(\mathcal{D}_{n_\mathrm{gen}}))\log\frac{\hat{h}_i(o(\mathcal{D}_{n_\mathrm{gen}}))}{\frac{1}{2}(\hat{h}_i(o(\mathcal{D}_{n_\mathrm{gen}}))+h_i(p_O))} \quad + h_i(p_O)\log\frac{h_i(p_O)}{\frac{1}{2}(\hat{h}_i(o(\mathcal{D}_{n_\mathrm{gen}}))+h_i(p_O))}\right),\end{aligned} \tag{3.39}$$

with $h_i(p_O) \approx \frac{1}{n_\mathrm{quant}}$. If we interpret the set of h_i and $\hat{h}_i$ as histogram approximations of the densities p_O and $\hat{p}_O$, this measurement is exactly the JSD (3.3) between both histograms. To prevent the sparse bins, we only report results where $\#\mathcal{D} < 10 \cdot n_\mathrm{quant}$. As in the last section, the probability for an empty bin in this configuration is then $< 4.5 \cdot 10^{-5}$.

In this setup, we find a similar dependence on n_quant as in Section 3.6. For low numbers ($n_\mathrm{quant} < 8$), no significant amplification can be observed. This is in line with the observation that an estimator of a mean of a Gaussian distribution cannot be improved with surrogate data [150]. At $n_\mathrm{quant} > 64$, the amplification surpasses 5 and can reach up to 50 for very small bins. This can be observed for all observables individually (Figure 8), as well as for the joint distributions of the observables (Figure 9).

To find out whether this effect is universal for different density estimation techniques, we fit a histogram and a KDE to the training data. For these more classical density estimation techniques, we optimize on the observable values rather than the 100-dimensional images. Still we find in Figure 10, for high numbers of quantiles, the distance to the truth distribution for the VAE-GAN approximation is close to that of KDEs and histograms using five times the data. For higher-dimensional observables, the performance difference between the methods gets more substantial. This is testament to the superior interpolation of NNs in high-dimensional, more data sparse, applications. Unsurprisingly, for low numbers of quantiles we again find that no improvement can be generated by using a generative NN.

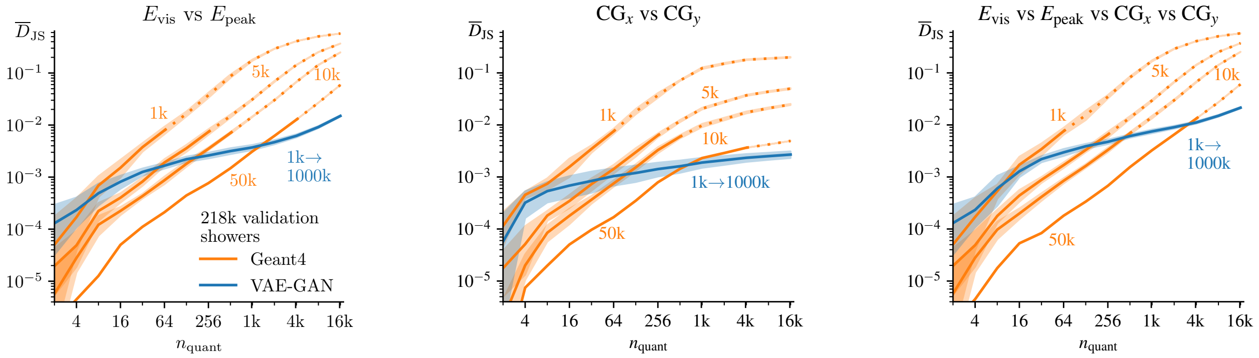


Figure 9: Distance to the observable distribution of the true data (MCMC simulated photon showers) for a histogram estimator with increasing number of bins based on samples drawn from a VAE-GAN (blue). For comparison, we also show the results for histogram estimators calculated from sets of true data in 4 different sizes (orange). Error envelopes are calculated from 5 distinct sets of data and subsequent VAE-GAN training. The observables shown are the vector of visible energy and peak energy, both centers of gravity and joint vector of all four observables. These Figures were originally published in Reference [P1].

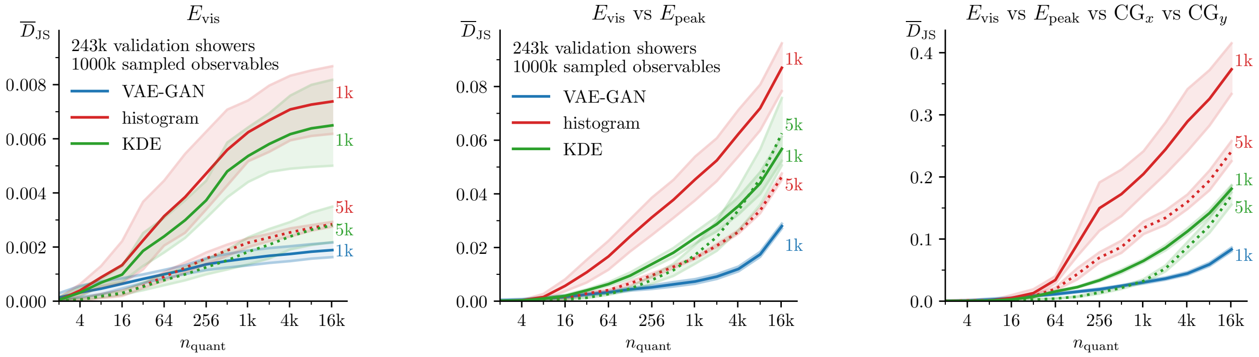


Figure 10: Distance to the observable distribution of the true data (MCMC simulated photon showers) for a histogram estimator with increasing number of bins based on samples drawn from a VAE-GAN (blue), a histogram (red) and a KDE (green). For the histogram and KDE we report results for trainings on 1k and 5k showers, while for the VAE-GAN we only show results on the smaller sample size. Error envelopes are calculated from 5 distinct sets of data and subsequent VAE-GAN training. The observables shown are the visible energy, the vector of visible energy and peak energy, and the joint vector of visible energy, peak energy and both centers of gravity. These Figures were originally published in Reference [P1].

4 Bayesian Neural Networks

This far, we have discussed the effects of limited training statistics on generative NNs trained with stochastic gradient descent. We found that the resulting mismodeling decreases with increasing the amount of data. In the greater context of learning uncertainties, this mismodeling is captured by the *epistemic* uncertainty [159]. In physics terms, the epistemic uncertainty of a model is often referred to as its systematic uncertainty. A mismodeling which in this case is based in the statistical uncertainty of the available data.

It has to be distinguished from the *aleatoric* uncertainty, the systematic uncertainty of the data itself. This could be the stochasticity of the detector simulation or the measurement error of the detector in the experiment. It is irreducible. While we cannot reduce this uncertainty through more data, we can model the probability of the continuous (or discrete) outcome. We often assume a Gaussian distribution and adapt the mean and variance to achieve the highest possible likelihood of the data. More complex distributions can be approximated using a NFs, see Section 3.3. For some cases it is sufficient to vary only the mean, but not the variance or shape of the distribution, dependent on the input. One example for such a case is the error caused by rounding to a specified decimal place. This is called *homoscedastic* uncertainty. In physics, errors often scale with the input, for example the precision of a detector with the deposited energy. The variance or shape of the distribution thus needs to be a function of the input. This is understood as *heteroscedastic* uncertainty.

Epistemic uncertainty can have multiple sources. One is that the optimization of a loss on limited dataset will not result in the best possible solution over the true, inaccessible data distribution. This is even more drastic if a subset-based, empiric estimate of the loss on the full set is applied. This is called *approximation uncertainty*. Furthermore, the functional form encoded by the chosen architecture of the NN does not necessarily include the optimal solution to the task (*model uncertainty*). For a more detailed introduction into uncertainty in ML, see References [160–162] and Reference [13] for a review dedicated to applications in HEP.

The model uncertainty is hard to quantify. One possibility is to use an ensemble of NNs to construct frequentist confidence intervals of the network output [163–166], but vary the network architecture within the ensemble [167]. It is also theoretically conceivable, although computationally inefficient, to sample the architecture as part of a Markov chain, as analyzed in Reference [P3]. As NNs are largely overparametrized, they have the flexibility to approximate an optimal solution. We can thus safely assume the model to be valid [162] and focus on gauging the approximation uncertainty.

Similarly, we assume the data distribution to be smooth and stationary during training and testing. To gauge it we employ Bayesian inference, that is, we infer a distribution on the function space of the NNs specified through the network parameters ϑ. This distribution is called *posterior* distribution

$$\pi(\vartheta \mid \mathcal{D}_n) = \frac{\pi(\mathcal{D}_n \mid \vartheta)\,\pi(\vartheta)}{\pi(\mathcal{D}_n)} \propto \pi(\mathcal{D}_n \mid \vartheta)\,\pi(\vartheta) \tag{4.1}$$

and is formed from our *prior* beliefs $\pi(\vartheta)$ and the *likelihood* $\pi(\mathcal{D}_n \mid \vartheta)$ of the data $\mathcal{D}_n$ given ϑ. The marginal probability $p(\mathcal{D}_n) = \int \pi(\mathcal{D}_n \mid \vartheta)\,\pi(\vartheta)\,d\vartheta$ is often intractable if the likelihood is not available in closed form. The likelihood distribution is in fact not new to us. We have encountered it in the previous chapters on probabilistic learning

$$\pi(\mathcal{D}_n \mid \vartheta) = \prod_{i=1}^{n} \hat{p}_X(x_i; \vartheta)\,. \tag{4.2}$$

To distinguish conditional densities of the data that are part of the learning task, from those of the model parameters, we use p for the probability densities of the former and π for those of the latter. In the previous sections, we also only reported results for a single parameter estimate. With access to the posterior distribution however, we are interested in the posterior mean distribution

$$\hat{p}_X(x) = \int \hat{p}_X(x; \vartheta)\pi(\vartheta \mid \mathcal{D}_n)\,d\vartheta\,. \tag{4.3}$$

Similarly, we can analyze the spread of the approximated distributions over different drawings of the parameters from the posterior. In Section 6.2, that is Reference [P5], we further discuss the interpretation of this uncertainty.

Bayesian learning also gives us insights into stochastic optimization. For example, the parameter estimate maximizing the posterior distribution

$$\vartheta_{\mathrm{MAP}} = \mathrm{argmax}_{\vartheta \in \Omega} \pi(\vartheta \mid \mathcal{D}_n) \tag{4.4}$$

is commonly referred to as the *maximum a-posteriori* (MAP) estimate. Using the log-posterior

$$L_{\mathrm{MAP}}(\vartheta; \mathcal{D}_n) := \log \pi(\vartheta \mid \mathcal{D}_n) = \log \pi(\mathcal{D}_n \mid \vartheta) + \log \pi(\vartheta) - \mathrm{const.} \tag{4.5}$$

as the optimization criterion for a gradient descent optimization then naturally gives rise to common weight regularization terms. For an exponential or Gaussian prior, the second term of L_{MAP} reduces to L_1- or L_2-weight regularization respectively. We see that in the context of Bayesian learning, the prior does not necessarily need to be the distribution used for initializing the network parameters. It should rather be understood as a regularization term that implements the effects of our prior assumptions in the posterior in addition to the data likelihood.

For the following methods, we will often encounter the assumption that the posterior is a multi-variate Gaussian. To better understand this assumption, we will take a step back and discuss some statistical tools and statements for MLE.

Fisher Information

An important term in this discussion will be the *Fisher information*, or short the *Fisher* [168]. Let us assume $p_X(x; \vartheta)$ is a family of probability densities parametrized by parameters ϑ and the data is generated by drawing from the true distribution $\mathcal{D}_n \sim p_X^n = p_X(x; \vartheta_0)^n$. If $p_X(x; \vartheta)$ is concentrated tightly around ϑ_0, finding a good estimate from $\mathcal{D}_n$ is easy. The Fisher information

$$\mathcal{I}(\vartheta') = \mathop{\mathbb{E}}_{x \sim p_X}\left[\left(\frac{\partial}{\partial \vartheta} \log p_X(x; \vartheta) \right)^2 \middle| \vartheta' \right] = - \mathop{\mathbb{E}}_{x \sim p_X}\left[\frac{\partial^2}{\partial \vartheta^2} \log p_X(x; \vartheta) \middle| \vartheta' \right] \tag{4.6}$$

measures the amount of information that samples $x \sim p_X$ carry on ϑ. The Fisher only exists if $p_X(x; \vartheta)$ is differentiable with respect to ϑ almost everywhere. The second equality is only well-defined if it is twice differentiable with respect to ϑ and integration over p_X and the derivative commute. If the Fisher is approximated from a set of data $\mathcal{D}_n$

$$\hat{\mathcal{I}}(\vartheta') = \mathop{\mathbb{E}}_{x \in \mathcal{D}_n}\left[\left(\frac{\partial}{\partial \vartheta} \log p_X(x; \vartheta) \right)^2 \middle| \vartheta' \right] , \tag{4.7}$$

it is referred to as the *empirical* Fisher information. In the case where ϑ is multidimensional, the Fisher is a matrix with entries

$$\mathcal{I}(\vartheta')_{ij} = \mathop{\mathbb{E}}_{x \sim p_X}\left[\left(\frac{\partial}{\partial \vartheta_i} \log p_X(x; \vartheta) \right) \left(\frac{\partial}{\partial \vartheta_j} \log p_X(x; \vartheta) \right) \middle| \vartheta' \right] = - \mathop{\mathbb{E}}_{x \sim p_X}\left[\frac{\partial^2}{\partial \vartheta_i \partial \vartheta_j} \log p_X(x; \vartheta) \middle| \vartheta' \right] . \tag{4.8}$$

Cramér–Rao Bound and Efficiency

Under certain regularity conditions, the inverse Fisher can be used as a lower bound of the variance of an unbiased estimator $\hat{\vartheta}(\mathcal{D}_n)$ of ϑ_0

$$\mathrm{var}_{\vartheta}(\hat{\vartheta}(\mathcal{D}_n)) \geq \mathcal{I}(\vartheta_0)^{-1} . \tag{4.9}$$

Here $A \geq B$ means $A - B$ has non-negative eigenvalues. Of course, this requires the Fisher matrix to be well-defined and invertible. Again, the proof uses that the order of integration in the expectation value and differentiation can be reversed. Similar statements can also be derived for the more general case of a biased estimator. For the proof and generalizations of the Cramér–Rao bound [169, 170], we refer to the statistics

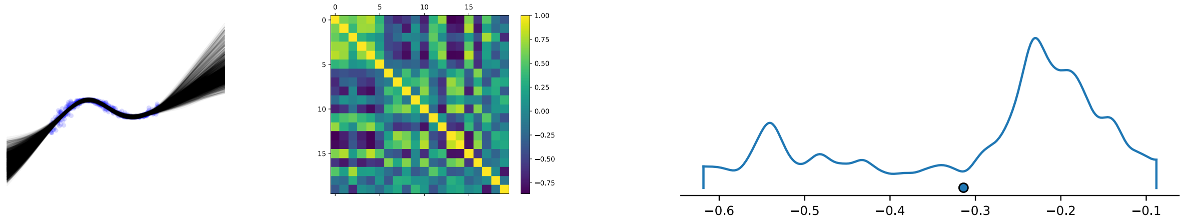


Figure 11: Correlations of the parameters in the first layer of a NNs for a simple regression example (left). The parameters are highly correlated (center) and the marginal posterior of a single weight exhibits a multi-modal shape (right). The posterior samples are generated with a piece-wise deterministic Markov process [172]. Figures taken from [172].

literature [100].

An estimator is said to be fully efficient, if the Cramér–Rao bound (4.9) is reached. For MLE, as it is often applied in generative ML

$$\hat{\vartheta}_n = \underset{\vartheta \in \Omega}{\operatorname{argmin}} \operatorname{NLL}(\vartheta; \mathcal{D}_n)\,, \tag{4.10}$$

this bound is achieved in the limit of large data $n \to \infty$ under certain regularity conditions on the family $p_X(x;\vartheta)$ [171, Chapter 8.9]. More specifically

$$\sqrt{n}\left(\hat{\vartheta}_n - \vartheta_0\right) \xrightarrow{n\to\infty} \mathcal{N}\left(0, \mathcal{I}(\vartheta_0)^{-1}\right)\,. \tag{4.11}$$

Bernstein-Von Mises Theorem

Equation (4.11) can also be connected to Bayesian inference. The Bernstein-von Mises Theorem states that the posterior distribution of the parameter ϑ converges to a normal distribution in total variation distance for the limit of large samples [171, Chapter 10.2]

$$\operatorname{TV}\left(\pi(\vartheta \mid \mathcal{D}_n), \mathcal{N}\left(\hat{\vartheta}_n, n^{-1}\mathcal{I}(\vartheta_0)^{-1}\right)\right) = \left\|\pi(\vartheta \mid \mathcal{D}_n) - \mathcal{N}\left(\hat{\vartheta}_n, n^{-1}\mathcal{I}(\vartheta_0)^{-1}\right)\right\|_{\mathrm{TV}} \xrightarrow{n\to\infty} 0\,. \tag{4.12}$$

Here, $\|f(\vartheta)\|_{\mathrm{TV}}$ is the total variation norm, $\mathcal{I}$ is the Fisher information matrix and $\hat{\vartheta}_n$ the estimator from MLE (4.10). Equation (4.12) is only true under a set of assumptions. First, the Fisher matrix needs to exist and be invertible. The log-probability thus needs to be differentiable at ϑ_0. Furthermore, $p_X(x;\vartheta)$ needs to be differentiable in quadratic mean [171, Equation 5.38] and allow separating ϑ_0 from compliments of balls around ϑ_0 for arbitrary radii through a series of hypothesis tests [171, Chapter 10.2].

The symmetric, over-parameterized shape of $p_X(x;\vartheta)$ for NNs breaches the last assumption, as many parameter configurations can lead to the same probability predictions. The Bernstein-von Mises Theorem (4.12) can thus be a bad approximation. BNN posteriors are expected to be highly correlated [173] and multi-modal. Reference [172] shows the correlations of the parameters in the first layer of a NN and the marginal posterior for one weight for a simple regression task (Figure 11). Even for this small example, the posterior shows strong correlations between parameters, multiple modes and long tails and is thus not in agreement with the assumptions of Equation (4.12).

Common Methods

Full Hamiltonian Monte Carlo [174] is considered the gold standard of inferring NN posterior distributions [175]. This Markov chain requires calculation of the loss over the entire dataset at every update step. In an example on the ResNet20 architecture [175], 240 samples of the posterior are drawn at the cost of $6 \cdot 10^7$ SGD steps. The stochastic MCMC methods introduced in 4.2 achieve more efficient sampling at the cost of mixing and accuracy.

They also compare the results of this substantial computational effort to other, computationally more efficient methods. The most common methods include deep ensembles [166], dropout MCMC [176], Gaussian processes [177], Laplace approximations [178] and mean-field variational inference [14]. Less common but very computationally efficient methods are SWAG [179] and evidential deep learning [180]. See References [12, 181–183] for an overview. For a review of non-Bayesian methods for quantifying epistemic uncertainty, consider Reference [162]. These methods include set-valued and conformal prediction, as well as outlier detection with anomaly detection methods and classification with rejections.

Besides Reference [175], multiple studies have compared subsets of these methods [184–186]. In general, these studies find better uncertainty estimation and generalization properties with deep ensembles and MCMC based methods over methods inferring a Gaussian approximation of the posterior. Furthermore, analogously to pruning of a network, studies have found that inferring a marginal posterior for only a subset of the parameters yields similar results at significantly reduced cost [187–190]. A popular technique of these subset-BNNs is treating only the last-layer of a NN as Bayesian [189, 190].

Training an ensemble of networks is not feasible for the large generative tasks in HEP. We will thus focus on approximations of the posterior as a Gaussian (as it is the most widespread in HEP) in Section 4.1, as well as stochastic gradient MCMC methods in Section 4.2. In Section 4.3, we then introduce our own stochastic gradient MCMC method.

4.1 Gaussian Approximations to the Posterior

As already hinted at in the introduction, a common way to achieve scalable implementations of BNNs is through an approximation of the posterior distribution with a Gaussian distribution

$$\pi(\vartheta \mid \mathcal{D}) \approx \tilde{\pi}(\vartheta;\phi) = \mathcal{N}\left(\vartheta; \boldsymbol{\mu}(\phi), \Sigma(\phi)\right) \,. \tag{4.13}$$

Usually, the entries of $\boldsymbol{\mu}$ and Σ themselves are the parameters ϕ. For evidential learning [180], $p_X(x;\vartheta)$ is a Gaussian parametrization and the functions $\boldsymbol{\mu}(\phi)$ and $\Sigma(\phi)$ are implemented as NNs.

In the previous section, we have seen that this can be a bad approximation of the true posterior. This is especially true if the covariance matrix of the Gaussian is assumed to be diagonal, that is the parameters are not correlated. However, due to their easy application, this class of has found the most widespread use in HEP [15–18]. We will thus discuss the main techniques to infer such an approximation in the following.

Variational Inference

The most popular technique is to translate the inference task into a variational optimization problem, thus dubbed *variational inference* (VI) [191] or *Bayes-by-Backprop* [14]. It has developed from earlier ideas on ensemble learning [173, 192]. This inference task can be achieved trough minimizing the KLD between true posterior and approximation

$$L_{\mathrm{VI}}(\phi;\mathcal{D}) = D_{\mathrm{KL}}\left[\tilde{\pi}(\vartheta;\phi), \pi\left(\vartheta|\mathcal{D}\right)\right] = \underbrace{-\int \log \pi\left(\mathcal{D}|\vartheta\right)\,\tilde{\pi}(\vartheta;\phi)d\vartheta + D_{\mathrm{KL}}\left[\tilde{\pi}(\vartheta;\phi), \pi(\vartheta)\right]}_{\mathrm{ELBO}} + \text{const.}\,. \tag{4.14}$$

Here, as in Section 3.2, we encounter the evidence lower bound of the KLD. Analogously to the calculation in Equation (3.14), for an uncorrelated posterior approximation and a Gaussian prior with the same mean and covariance on all independent parameters

$$\tilde{\pi}(\vartheta;\phi) = \mathcal{N}\left(\vartheta; \begin{pmatrix}\boldsymbol{\mu}_1\\ \ldots \\ \boldsymbol{\mu}_P\end{pmatrix}, \begin{pmatrix}\sigma_1^2 & & \\ & \ldots & \\ & & \sigma_P^2\end{pmatrix}\right) \quad \text{and} \quad \pi(\vartheta) = \mathcal{N}\left(\vartheta; \begin{pmatrix}\mu_p\\ \ldots \\ \mu_p\end{pmatrix}, \begin{pmatrix}\sigma_p^2 & & \\ & \ldots & \\ & & \sigma_p^2\end{pmatrix}\right) \,,$$

the calculation of the KLD reduces to

$$\begin{aligned} D_{\mathrm{KL}}\left[\tilde{\pi}(\vartheta;\phi),\pi(\vartheta)\right] &= \frac{1}{2}\sum_{i=1}^{P}\left((\log\sigma_p^2-\log\sigma_i^2)+\frac{\sigma_i^2-\sigma_p^2}{\sigma_p^2}+\frac{(\mu_i-\mu_p)^2}{\sigma_p^2}\right) \\ &\approx \frac{1}{2\sigma_p^2}\sum_{i=1}^{P}\left(\frac{\left(\sigma_p^2-\sigma_i^2\right)^2}{\sigma_i^2}+(\mu_i-\mu_p)^2\right)\,. \end{aligned} \tag{4.15}$$

Here, we also introduce P as the number of parameters in the BNN. The log-likelihood of the first term of the ELBO can be calculated differently for different applications. In 2.2, we discuss different optimizations objectives with interpretations as log-probabilities and in Section 3.3 we discuss how to directly access the log-probability of the data for generative ML with flows. For these implementations, we need to calculate the expectation value over the approximation of the posterior. This is done through Monte Carlo approximation, that is sampling from the posterior distribution

$$\int \log\pi\left(\mathcal{D}|\vartheta\right)\,\tilde{\pi}(\vartheta;\phi)\,d\vartheta \approx \mathbb{E}_{\vartheta\sim\tilde{\pi}(\vartheta;\phi)}\log\pi\left(\mathcal{D}|\vartheta\right)\,. \tag{4.16}$$

Again as in Section 3.2, we need to ensure differentiability of the loss for the parameters to use gradient descent on the parameters of the posterior approximation. Because we assumed the posterior to be an uncorrelated Gaussian, we can use the same reparametrization trick (3.16) as for VAEs [14]

$$\vartheta_i = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i\epsilon\,, \text{ with } \epsilon\sim\mathcal{N}(0,1) \text{ for } i\in\{1,...,P\}\,. \tag{4.17}$$

Due to the cost of loss calculation and parameter storing, only a limited amount of samples is drawn from the posterior for the integration (4.16). Often only a single sample is used. This leads to a large variance in the MC estimate. Several techniques try to reduce this variance through different sampling [193] or by sampling pseudo-independent perturbations [194]. We have found great improvements in the stability of VI with the latter method.

Similar to dropout regulation [195], which can be understood as a limiting case of sampling a neural network posterior itself [176], the *flipout* method uses the activations of the nodes as perturbations to decrease correlations within one batch of data. Due to the symmetry of $\epsilon\sim\mathcal{N}(\mathbf{0},\mathbb{1})$ around $\mathbf{0}$, the reparametrization (3.16) is independent of multiplying with a random sign ±1. One can thus generate pseudo-independent samples

$$\vartheta = \boldsymbol{\mu} + e\boldsymbol{\sigma}\epsilon \sim \tilde{\pi}(\vartheta;\phi) \tag{4.18}$$

by multiplying with $e\sim\text{uniform}(\{-1,+1\})^P$ for a single fixed perturbation $\epsilon\sim\mathcal{N}(\mathbf{0},\mathbb{1})$ and every point in the training data $\mathcal{D}_n$. This can be efficiently implemented by binary masks and parallelized. It approximately doubles the cost, but guarantees the best possible variance reduction $1/n$ [194].

The two terms in L_{VI} (4.14) often have largely different scales. While the log-likelihood scales linearly in the number of points n, the KLD is often close to 0. The convergence of the prior regularization is often significantly slowed down by the dominance of the first term. It can thus be beneficial to initiate the posterior approximation at the result of a classical maximum likelihood training as motivated by the Bernstein-von Mises theorem (4.12). This can be interpreted as specifying an informative prior through empirical Bayes [196].

For a derivative of VI with a less restricted posterior approximation, see Reference [197].

Laplace Approximation

Using a Gaussian distribution based on the second derivative of the log-posterior to approximate the posterior distribution has been a popular, scalable method for BNNs [198–200]

$$\pi(\vartheta \mid \mathcal{D}_n) \approx \mathcal{N}\left(\vartheta; \vartheta_{\mathrm{MAP}}, \frac{\partial^2}{\partial\vartheta^2} \underbrace{\log \pi(\vartheta \mid \mathcal{D}_n)}_{=L_{\mathrm{MAP}}(\vartheta;\mathcal{D}_n)} \Bigg|_{\vartheta_{\mathrm{MAP}}}\right) . \tag{4.19}$$

It is often referred to as the *Laplace approximation*, as it can also be understood as a second-order Taylor expansion of the log-posterior at the MAP estimate

$$\log \pi(\vartheta \mid \mathcal{D}_n) \approx L_{\mathrm{MAP}}(\vartheta_{\mathrm{MAP}}; \mathcal{D}_n) + \frac{1}{2}(\vartheta - \vartheta_{\mathrm{MAP}})^\top \underbrace{\frac{\partial^2}{\partial\vartheta^2} L_{\mathrm{MAP}}(\vartheta; \mathcal{D}_n)\Bigg|_{\vartheta_{\mathrm{MAP}}}}_{\approx\mathcal{I}(\vartheta_{\mathrm{MAP}})+\partial^{\in}/\partial\vartheta^{\in}\log\pi(\vartheta)} (\vartheta - \vartheta_{\mathrm{MAP}})\,. \tag{4.20}$$

The first order vanishes, as ϑ_{MAP} is a maximum of the posterior. Exponentiating both sides recovers the Gaussian shape of the approximation. The Laplace approximation can thus be inferred after training of the model, that is estimation of the MAP estimate.

Recent developments in optimization libraries accomplish the fast calculation of second derivatives. Based on these libraries, different matrix representations of the empirical Fisher can be used for the approximation (4.20) [178]. This includes the calculation of more expressive factorizations than the diagonal factorization, which establishes the same functional shape of the approximation as is commonly used in VI. One popular choice of block-diagonal factorization is the *Kronecker-factored approximate curvature* (KFAC) [201, 202].

4.2 Stochastic Markov Chain Monte Carlo for Posterior Sampling

As mentioned in the introduction to Section 4, the Bernstein-von Mises approximation, as any approximation assuming a Gaussian shape of the posterior, can be far off for the over-parametrized regime of BNNs. Such an approximation might thus not yield accurate credible sets. This motivates the development of more flexible methods for Bayesian learning in NNs.

Traditionally, Monte Carlo sampling has often been used for sampling from intractable posteriors. The literature on sampling NN weights is as old as the concept of stochastic, gradient-based optimization, with early literature dating back to the early 1990s and before [174, 203]. With the ever steady increase of model- and data-complexity, multiple problems with the basic theory of Monte Carlo sampling occur. The most critical issues are:

- **Efficient convergence and mixing in high dimensional sampling spaces:** NNs consist of up to hundreds of trillions of parameters. Applications in HEP often have millions of parameters, for example 2.14M parameters for ParT [76] or 560k for EPIC flow matching [148]. Even smaller toy examples still require tens of thousands of parameters [P5]. Due to the random-walk like nature of Markov chains, the parameter space exploration of such algorithms is inefficient, especially for high numbers of parameters. This is often solved by using gradient-based chains (with momentum) to profit from the efficiency of network optimization during Monte Carlo sampling.

- **Computational complexity:** Based on the immense size of the datasets (200k showers of up to 150 constituents for EPIC flow matching or 100M of up to 100 particles for `JetClass` [204]) it is not feasible to calculate the parameter gradient for full datasets in every step of the chain. However, using a stochastic approximation, that is batches of data, introduces a bias on the chains invariant distribution. This bias needs to be accounted for with corrections or applying correct asymptotics.

  For applying corrections to the chain, the M-H correction is the most common one. It corrects the sampled invariant distribution, by accepting or rejecting a step of the chain based in the log-likelihood (loss) of the corresponding parameters. Again, employing a stochastic estimate of the likelihood (loss) reduces the accuracy of the correction and might lead to further biases. The M-H correction thus needs its own correction terms.

MCMC algorithms that feature both, gradients and batch-wise computation, are referred to as *stochastic gradient MCMC* (sgMCMC). We will concentrate on this class of algorithms as the only viable solution for parameter sampling with limited resources.

Let us assume a labeled or unlabeled n-point dataset $\mathcal{D}_n$, the vector of network weights ϑ, a generic loss (also risk or cost) function $L(\vartheta)$ and its empirical counterpart $L_n(\vartheta) = L(\vartheta; \mathcal{D}_n)$. The goal of every sgMCMC algorithm then is to sample from the Gibbs posterior distribution

$$\pi_\lambda(\vartheta \mid \mathcal{D}_n) \propto \exp(-\lambda L_n(\vartheta))\ \pi(\vartheta)\,, \tag{4.21}$$

with an inverse temperature parameter $\lambda > 0$ and a prior density on the network weights $\pi(\vartheta)$. The Gibbs posterior matches the classical Bayesian posterior through Bayes theorem up to a multiplicative constant if the empirical loss is exactly the log-likelihood of the data under the model distribution. That is if $\lambda = 1$ and

$$L_n(\vartheta; \mathcal{D}_n) = -\log \pi(\mathcal{D}_n \mid \vartheta) = -\sum_{\mathbf{x}' \in \mathcal{D}_n} \log \pi(x' \mid \vartheta)\,. \tag{4.22}$$

Reference [205] introduces a unified framework in which all the major established stochastic gradient MCMC methods can be developed from a general SDE that ensures the invariant distribution of the chain is the Gibbs posterior. The general SDE is constructed from a positive semi-definite diffusion matrix $\mathbf{D}(\mathbf{z})$ and a skew-symmetric curl matrix $\mathbf{Q}(\mathbf{z})$. The chain is calculated for $\mathbf{z} = (\vartheta, r)$, a vector of the network parameters ϑ and auxiliaries of the chain r. For this vector of parameters, a joint distribution

$$\pi(\mathbf{z} \mid \mathcal{D}_n) \propto \exp(-H(\mathbf{z}))\,,\ \text{with } H(\mathbf{z}) = \lambda L_n(\vartheta) - \log \pi(\vartheta) + g(\vartheta, r)\,, \tag{4.23}$$

is sampled. In imitation of physics terminology, the combined energy function $H(\mathbf{z})$ is called a *Hamiltonian* and can include various auxiliary terms satisfying $\int \exp(-g(\vartheta, r))dr = \text{const.}$. The marginal of $\pi(\mathbf{z} \mid \mathcal{D}_n)$ in the network parameters is then again the Gibbs posterior (4.21).

Assuming ergodicity of the process, any SDE that can be written as

$$d\mathbf{z} = \mathbf{f}(\mathbf{z})\ dt + \sqrt{2\mathbf{D}(\mathbf{z})}\ dW(t)\,, \tag{4.24}$$

with $W(t)$ the Wiener process (Brownian motion) in d dimensions samples from $\pi(\mathbf{z} \mid \mathcal{D}_n)$ as its stationary distribution [205]. This is only true if

$$\mathbf{f}(\mathbf{z}) = -\left(\mathbf{D}(\mathbf{z}) + \mathbf{Q}(\mathbf{z})\right) \nabla H(\mathbf{z}) + \Gamma(\mathbf{z})\,,\ \text{where } \Gamma_i(\mathbf{z}) = \sum_{j=1}^{d} \frac{\partial}{\partial \mathbf{z}_j} \left(\mathbf{D}_{ij}(\mathbf{z}) + \mathbf{Q}_{ij}(\mathbf{z})\right)\,. \tag{4.25}$$

Conversely, they also show that any Markov chain with desired stationary distribution $\pi(\mathbf{z} \mid \mathcal{D}_n)$ can be written in terms of Equation (4.25), if

$$\mathbf{f}_i(\mathbf{z})\ \pi(\mathbf{z} \mid \mathcal{D}_n) - \sum_{j=1}^{d} \frac{\partial}{\partial \vartheta_j} \left(\mathbf{D}_{ij}(\mathbf{z})\ \pi(\mathbf{z} \mid \mathcal{D}_n)\right)$$

is integrable with respect to the Lebesgue measure. Through discretization into steps of size η_t, we get the update rule of the chain

$$\mathbf{z}_{t+1} = \mathbf{z}_t - \eta_t \left[\left(\mathbf{D}(\mathbf{z}_t) + \mathbf{Q}(\mathbf{z}_T)\right) \nabla H(\mathbf{z}_t) + \Gamma(\mathbf{z}_t)\right] + \sqrt{2\eta_t \mathbf{D}(\mathbf{z}_t)}\epsilon_t\,,\ \text{with } \epsilon_t \sim \mathcal{N}(0, 1)\,. \tag{4.26}$$

For a random m-point subset $\overline{\mathcal{D}}_m$ of the full dataset $\mathcal{D}_n$ drawn uniformly without replacement,

$$\hat{L}_m(\vartheta) = -\log \pi(\overline{\mathcal{D}}_m \mid \vartheta) = -\frac{n}{m} \sum_{\mathbf{x}' \in \overline{\mathcal{D}}_m} \log \pi(x' \mid \vartheta)$$

is an unbiased estimator of the full loss L_n and its gradient $\nabla \hat{L}_m(\vartheta)$ is an unbiased estimator of the full gradient $\nabla L_n(\vartheta)$. The variance of the loss estimator propagates to the estimator of the Hamiltonian gradient $\nabla \hat{H}(\mathbf{z})$. The introduced noise thus changes the dynamics of the system and has to be countered by adapting the scale of the Brownian motion

$$\sqrt{2\eta_t \mathbf{D}(\mathbf{z}_t)}\epsilon_t \longrightarrow \left(\sqrt{2\eta_t \mathbf{D}(\mathbf{z}_t)} - \eta_t \hat{\mathbf{B}}_t\right) \epsilon_t \, . \tag{4.27}$$

Here, $\hat{\mathbf{B}}_t$ is an estimate of the variance of the stochastic gradient $\mathbf{V}(\vartheta) = \mathrm{var}(\nabla \hat{L}_m(\vartheta))$. It is assumed that the batches are large enough for the central limit theorem (CLT) to apply (couple hundreds [206]) such that the noise on the stochastic gradient is approximately Gaussian

$$\nabla \hat{L}_m(\vartheta) \approx \nabla L_n(\vartheta) + \mathcal{N}(0, \mathbf{V}(\vartheta)) \, .$$

As $\eta_t \hat{\mathbf{B}}_t$ approaches 0 more quickly than $\sqrt{2\eta_t \mathbf{D}(\mathbf{z}_t)}$ for $\eta_t \to 0$, in the limit to small stepsizes the invariant distribution is unchanged by the variance introduced by the batch-wise computation.

The general SDE specified by Equation (4.24), Equation (4.25) and Equation (4.27) and its discretization (4.26) allows us to derive the different families of sgMCMC from a general theory through the choice of the auxiliary variables, curl- and diffusion matrices.

4.2.1 Stochastic Gradient Langevin Dynamics

The simplest family includes *stochastic gradient Langevin Dynamics* (sgLD) [20, 203] and derivatives thereof. The sgLD algorithm is closely related to stochastic gradient descent and only differs from it through the addition of isotropic noise

$$\vartheta_{t+1} = \vartheta_t - \eta_t \nabla \hat{L}_m(\vartheta_t) + \sqrt{\frac{2\eta_t}{\lambda}}\epsilon_t \, , \ \text{with } \epsilon_t \sim \mathcal{N}(0,1) \, . \tag{4.28}$$

Here of course, the subset $\overline{\mathcal{D}}_m$ is drawn anew for each step. This update rule describes a discretization of the Itô-SDE of Brownian dynamics [20, 207]

$$d\vartheta(t) = -\nabla_\theta \hat{L}_m(\vartheta(t))dt + \sqrt{\frac{2}{\lambda}}dW(t) \, . \tag{4.29}$$

For a suitable loss, the unique invariant distribution of this SDE is indeed given by the Gibbs posterior (4.21) [208]. In terms of the general theory (4.25), sgLD can be constructed from the general SDE by removing auxiliaries, $\mathbf{z} = \vartheta$ and $H(\mathbf{z}) = \lambda L_n(\vartheta) - \log \pi(\vartheta)$, and using a uniform prior. The matrices are chosen to be $\mathbf{D} = \mathbb{1}/\lambda$, $\mathbf{Q} = 0$ and $\hat{\mathbf{B}}_t = 0$.

In addition to the error of the stochastic approximation of the gradient, the discretization of the SDE introduces a *discretization error*. Rather than correcting for this error with an M-H correction, commonly the limit $\eta_t \to 0$, such that $\sum_{t=0}^{\infty} \eta_t = \infty$ and $\sum_{t=0}^{\infty} \eta_t^2 < \infty$, is considered with a polynomial decay of the stepsize. In this limit, the discretization error of the chain vanishes and a small error can be achieved by decreasing the stepsize to 0 after an initial burn-in phase [20]. Doing so, the stochasticity in the gradient will also be dominated by the added noise and the acceptance rates of a possible M-H correction will approach 1. Such a correction can thus be omitted [20]. To circumvent vanishing mixing rates, in practice the stepsize is reduced to a small fixed value, where the acceptance step can still be neglected. Finding a good final learning rate, as well as scheduling, is notoriously tricky. A detailed analysis of the bias introduced through finite stepsizes is performed in Reference [209]. The authors find that the final learning rate for a controlled bias depends on the batchsize in such a way that the computational cost remains roughly constant over all sub-sample sizes. For larger batches, larger steps can be used to reach the same bias. Thus, the chain convergences faster and requires the computation of fewer steps. This motivates the introduction of controls on the sample variance that scale sub-linearly in the batchsize, as we discuss in Section 5.

4.2.2 Stochastic Gradient Hamiltonian Monte Carlo

Due to the random-walk like nature of sgLD, these algorithms often exhibit slow convergence and mode exploration [210]. Hamiltonian Monte Carlo (HMC), originally proposed as Hybrid Monte Carlo, achieves a more efficient algorithm for high-dimensional parameter spaces by simulating the motion of an object of mass (matrix) $\mathbf{M}$ at position ϑ and momentum r. The full system is described by the Hamiltonian

$$H(\mathbf{z}) = H(\vartheta, r) = \lambda L_n(\vartheta) - \log \pi(\vartheta) + \frac{1}{2} r^T \mathbf{M}^{-1} r \,. \tag{4.30}$$

Similar to optimization with momentum, this additional momentum term results in a smoother exploration of the loss landscape.

Unlike sdLD, the noise of the *stochastic gradient HMC* (sgHMC) process is solely introduced by the variance $\mathbf{V}(\vartheta)$ of the stochastic estimator of the gradient. By assuming Gaussianity and absorbing it into the diffusion matrix $\mathbf{D}$, we find an additional friction term from

$$\mathbf{D}(\mathbf{z})\nabla H(\mathbf{z}) = \begin{pmatrix} 0 & 0 \\ 0 & \eta_t \mathbf{V}(\vartheta) \end{pmatrix} \begin{pmatrix} \nabla_\vartheta \\ \nabla_r \end{pmatrix} H(\mathbf{z}) = \eta_t \mathbf{V}(\vartheta) \nabla_r H(\mathbf{z}) = \eta_t \mathbf{V}(\vartheta) \mathbf{M}^{-1} r \,.$$

By using an estimate of the random friction caused by the variance of the gradient estimator $\hat{\mathbf{B}} \approx \mathbf{V}(\vartheta)$ and introducing a user specified friction variable $\mathbf{C}$ that dominates the random friction $\mathbf{C} \geq \hat{\mathbf{B}}$, the update rules for parameters and momentum read [21, 205]

$$\begin{aligned} \vartheta_{t+1} &= \vartheta_t + \eta_t \mathbf{M}^{-1} r_t \\ r_{t+1} &= r_t - \eta_t \left(\lambda \nabla \hat{L}_m(\vartheta_t) - \nabla \log \pi(\vartheta_t) \right) - \eta_t \mathbf{C} \mathbf{M}^{-1} r_t + \left(\sqrt{2\eta_t \mathbf{C}} - \eta_t \sqrt{\hat{\mathbf{B}}_t} \right) \epsilon_t \,. \end{aligned} \tag{4.31}$$

As in Equation (4.27), the estimate $\hat{\mathbf{B}}_t$ of the variance of the stochastic gradient estimator is subtracted to reduce the noise during sampling to the necessary amount. This fits the general SDE with

$$\mathbf{Q} = \begin{pmatrix} 0 & -\mathbb{1} \\ \mathbb{1} & 0 \end{pmatrix} .$$

In the limit $\eta_t \to 0$, the controlled noise $\mathbf{C}$ dominates and the resulting updates admit the correct invariant distribution. Similarly to sgLD [20], the limit is not applied in practice and a small residual bias through the finite learning rate is accepted. As pointed out in References [211, 212], this algorithm can lead to arbitrarily poor performance when the Gaussianity assumption on the noise is violated. For a study including multiple applications of sgHMC to BNN see Reference [213].

4.2.3 Further Developments and Improvements

Many improvements on the dynamics of sgLD and sgHMC have been proposed. They can all be understood as employing the freedom of the general SDE introduced in Section 4 in one way or another:

- **Preconditioning:** The process diffusion matrix $\mathbf{D}$ can be chosen as any positive semi-definite matrix. To improve the convergence of both sgLD or sgHMC, it can be used to parameterize the posteriors geometry and introduce larger noise levels in the dominant directions of the posterior. This is often referred to as *preconditioning* and the use of such a preconditioning matrix to “transform all directions to the same scale" [20] has already been proposed with the original sgLD [20]. Note that, through Equation (4.25), all choices of $\mathbf{D}$ require a correction term.

From the Bernstein-von Mises approximation (4.12), we know for well-parameterized models and for large enough datasets the posterior is shaped like a Gaussian. The covariance of the posterior is given by the inverse Fisher Information matrix (4.6). The empirical Fisher is thus a popular choice as a preconditioner

$$\mathbf{D} = \mathbf{D}(\vartheta) \overset{!}{=} \mathcal{I}^{-1}(\vartheta) \approx \hat{\mathcal{I}}^{-1}(\vartheta)\,.$$

The earliest use of the inverse Fisher metric as a preconditioning of sgLD can be found in Reference [206]. The authors propose to transition from sgLD to sampling from the Bernstein-von Mises approximation directly at large stepsizes using an online average of the empirical Fisher. Of course for BNNs, where the Bernstein-von Mises approximation is violated, this approach does yield limited improvements and the small stepsize asymptotics still need to be invoked. A similar approach using a running average of the Hessian matrix is presented in Reference [214].

Later schemes propose to determine the inverse Fisher metric for every sampled parameter point $\mathcal{I}^{-1}(\vartheta_t)$ to use as preconditioning for the next update step [205, 215]. The experiments are however only performed for latent Dirichlet allocations rather than BNNs. Once again, the over-parametrized nature BNNs has us questioning whether convergence improvements are on the same high level for NNs. Furthermore, the calculation of the (empirical) Fisher metric significantly increases computational cost.

Similar preconditioning of sgLD has been explored in Reference [216] for BNNs. The authors use the preconditioning from dimension-wise RMSprob (2.15)

$$\mathbf{D}(\vartheta_{t+1}) = \operatorname{diag}\left(\mathbb{1} \oslash (\varepsilon\mathbb{1} + \sqrt{V(\vartheta_{t+1})})\right) \tag{4.32}$$

$$V(\vartheta_{t+1}) = \alpha V(\vartheta_t) + (1-\alpha)\nabla\hat{L}_m(\vartheta_t) \odot \nabla\hat{L}_m(\vartheta_t)/m^2\,, \tag{4.33}$$

with element-wise multiplication and division and a small, positive constant ε preventing numerical nuisance. Here, the steps and noise are adapted to the gradients of the posterior landscape. This algorithm admits the same asymptotic guarantees as sgLD, but converges much quicker for the examined NN examples, both fully-connected and convolutional.

- **Relativistic Monte Carlo:** A similar effect to preconditioning can be derived from a relativistic description of the momentum in sgHMC. In Section 4.2.2, we have used a kinetic energy of $K(r) = \frac{1}{2}r^\top \mathbf{M}^{-1} r$ for setting the Hamiltonian (4.30). In theory, this Hamiltonian can produce infinitely large parameter updates. An alternative solution to preconditioning averting the resulting instability is to use a relativistic energy function [217]

$$K_{\text{rel}}(r) = \sum_{j=1}^{P} m_j c_j^2 \left(\frac{r_j^2}{m_j^2 c_j^2} + 1\right)^{\frac{1}{2}}\,. \tag{4.34}$$

This limits the size of the parameter updates to c_j, a speed of light individually set for a parameter dimension. The resulting mass matrix is diagonal $\mathbf{M} = \mathbf{M}(r) = \operatorname{diag}(m_1(r), ..., m_W(r))$, with

$$m_j(r) = m_j \left(\frac{r_j^2}{m_j^2 c_j^2} + 1\right)^{\frac{1}{2}} = \sqrt{\frac{r_j^2}{c_j^2} + m_j^2}\,.$$

The parameter update $\eta_t \mathbf{M}^{-1} r_t$ thus yields

$$\eta_t \mathbf{M}^{-1(r_t)} r_t = \eta_t \left(\frac{r_{t,1}}{\sqrt{\frac{r_{t,1}^2}{c_1^2} + m_1^2}}, ..., \frac{r_{t,W}}{\sqrt{\frac{r_{t,W}^2}{c_W^2} + m_W^2}}\right)^\top\,, \tag{4.35}$$

where $r_{t,j}$ is the entry of the momentum vector in dimension j at time t. A rescaling similar to those in Section 2.3 thus naturally arises without the need for additional preconditioning. It limits the update in dimension j to $\eta_t c_j$. In the limit of low temperature, a similar adaptive stochastic optimizer can indeed be obtained. Analogously to stochastic optimization, the authors show that the use of a relativistic momentum improves the stability with regard to the variance of the stochastic gradients compared to Newtonian momenta.

A similar relativistic kinematic term is also employed in the Adaptive Thermostat Monte Carlo sampler [218].

- **Monomial gamma HMC:** To improve transitioning between different modes, the kinetic function of HMC (4.30) can be generalized

$$r^\top \mathbf{M} r \longrightarrow \left(|r|^{\frac{1}{2a}}\right)^\top \mathbf{M}\, |r|^{\frac{1}{2a}}, \tag{4.36}$$

  with the exponent applied component-wise. It can be shown that for increasing $a \to \infty$ the probability to get trapped in a posterior mode goes to zero. Analogous improvements are found in application [219] at the cost of poor convergence and numerical difficulties for $a \geq 1$. In subsequent work, the authors show that more efficient generalizations of the kinetic energy in combination with an additional thermostat parameter can be applied to sgHMC [220].

- **Thermostats and tempering:** Further insights from physics are used to improve sampling. In Reference [221] the number of weights is promoted to a canonical ensemble that has to maintain thermal equilibrium. The ensemble has to fulfill

$$\pi(\theta, r) = \exp(-H(\theta, r)/(k_B T))\,, \text{ with } k_B T = \frac{1}{P}\mathbb{E}(r^\top r)\,. \tag{4.37}$$

  Here, k_B is the Boltzmann constant and T the temperature $\lambda = 1/(k_B T)$. This condition is not guaranteed when using stochastic gradients in sgLD or sgHMC and thus a further auxiliary variable ξ needs to be added into the chain to correct the dynamics

$$\xi_{t+1} = \xi_t + (\frac{1}{P} r_t^\top r_t - k_B T)\,, \text{ with } \xi_{(0)} = \mathbf{C}\,. \tag{4.38}$$

  This variable replaces the friction variable $\mathbf{C}$ in Equation (4.31) and adapts the friction in concordance with the equilibrium condition (4.37). The authors show, that this stabilizes the sampled chain with respect to sgHMC for large stepsizes η.

  Similar techniques based on the idea of a thermal equilibrium, as well as the general SDE from Equations (4.24) and (4.25) are applied in References [217] and [218]. Increased friction will however also slow down convergence and negative friction terms can lead to exploding momenta [218]. They thus clip the friction term at the variance of the momentum update.

  One further idea to leverage the temperature of the simulated distribution is *simulated annealing* as presented in Reference [222]. When starting sgLD out with a high temperature posterior, the algorithm can explore the posterior landscape efficiently as the Gibbs posterior is effectively flattened. However, if the temperature is too large, the chain will escape the desired posterior modes. To improve the convergence to high probability modes, the temperature is decreased later in the chain (as well as the stepsize, to achieve sampling from the true posterior asymptotically). The authors show analytically, that such an algorithm can find the minimum of a convex function, even in the presence of multiplicative, non-bounded, noise. The stochastic optimization algorithm obtained through temperature annealing of sgHMC with a thermal equilibrium variable is studied numerically in Reference [223]. The MAP estimate is shown to outperform the stochastic optimization algorithms from Section 2.3 in MNIST classification and other examples.

Interestingly, Reference [224] schedules the temperature the other way around. During mode exploration, they use the stochastic optimization limit $T \to 0$, that is $\lambda \to \infty$, to force the algorithm into the modes of the posterior and switch to an untempered posterior only shortly before drawing samples from the chain.

- **Contour sgLD:** A comparable flattening effect to tempering the posterior is achieved in Reference [225] with a gradient multiplier in Equation (4.28). It is calculated from a histogram of the Gibbs posterior and updated similarly to a Wang-Landau algorithm (importance sampling). This improves the sampling of multi-modal distributions as the barriers between modes are continually reduced. Note that for contour sgLD the samples are drawn from the flattened posterior rather than the untempered posterior. In Reference [226], this idea is combined with a multi-chain approach. A more stable convergence is achieved, through estimating the flattening histogram as a mean over the chains.

- **Multiple chains:** Using multiple, interacting chains computed in parallel to improve the performance is a popular idea. These chains can work on batches of different sizes and interact at given times to sample the posterior distribution [227]. Theoretic evidence on the superior efficiency of interacting chains over individual computations is provided by Reference [228]. More complex interaction schemes [229–231], as well as interactions of chains at different temperatures [232] have been proposed specifically for the sampling of multi-modal posteriors.

  Large applications in HEP already require distributed computation over multiple nodes. A further factorization of the computation cost to accommodate parallel chains thus seems unfeasible. In the following, we thus concentrate on single machine, non-distributed, MCMC approaches.

- **Cyclical scheduling schemes:** When applying sgLD and sgHMC, one has to balance fast convergence against mode exploration capabilities. Once a minimum mode is reached, the probability of transferring to a different mode is low. This regime is mostly determined by sampling with $\lambda = 1$ and diminishing stepsizes, as to circumvent the asymptotical bias of the invariant distribution. The chain can be divided into different phases, an *exploration phase* and a *sampling phase*.

  To improve the characteristics of both phases, mode exploration can be stimulated by high temperatures [222] or high learning rates [224]. After sufficient time, the sampling phase can be initiated by decreasing temperature or learning rate. Reference [224] proposes to alternate exploration and sampling by reestablishing the large stepsize after a predefined number of samples is drawn. In this scheme a cosine schedule is used for to decay the learning rate after reinitialization. The authors prove convergence of sgLD and sgHMC in this scheme and demonstrate the improvements in sampling multi-modal posteriors of BNNs.

  With multiple chains, the problem in exploration is circumvented by starting from different parameter configurations and likely sampling different modes with different chains. The cyclical scheduling can be understood as a concatenation of chains from parallel MCMC.

Furthermore, the integration scheme can be adapted. To this point, we only discussed discretizations of the SDE (4.24) that alternate the updates of auxiliary variables and parameters. This is commonly referred to as the *Euler integrator*. As already mentioned, this discretization introduces an error, which can be corrected by an M-H correction. Alternatively this error can also be reduced by using higher-order integration schemes.

- **Higher order integrators:** The leapfrog integrator [233] for HMC is one popular example of a second-order integrator. It alternates the update of the momentum $r_t \to r_{t+1/2}$, $\vartheta_t \to \vartheta_{t+1}$ and a second momentum update $r_{t+1/2} \to r_{t+1}$. This approach can be generalized to arbitrary orders by introducing the generator $\mathcal{G}$ of the diffusion (4.24) [234]

$$\mathcal{G} f(\vartheta_t) := \lim_{\eta \to 0_+} \frac{\mathbb{E}_{\vartheta_{t+\eta}}(f(\vartheta_{t+\eta})) - f(\vartheta_t)}{\eta} \,. \tag{4.39}$$

The function f here is an arbitrary, twice differentiable function. The generator defines the true evolution following the SDE as $\mathbb{E}_{\vartheta_{t+T}}(f(\vartheta_{t+T})) = e^{T\mathcal{G}} f(\vartheta_t)$. The effect of the generator on a small step h is usually approximated by a numerical *integrator* $P_\eta \approx e^{\eta \mathcal{G}}$, such that

$$\mathbb{E}_{\vartheta_{t+T}}(f(\vartheta_{t+T})) = \underbrace{e^{\eta\mathcal{G}} \circ ... \circ e^{\eta\mathcal{G}}}_{T/\eta\,\times} f(\vartheta_t) \approx P_\eta \circ ... \circ P_\eta f(\vartheta_t)\,. \tag{4.40}$$

It is said to be of order K if

$$P_h f(\vartheta_t) = e^{h\mathcal{G}} f(\vartheta_t) + \mathcal{O}(h^{K+1})\,. \tag{4.41}$$

This means, the error per step size η, is of the order η^{K+1}. In sgMCMC, we further approximate $\mathcal{G}$ through a subset-based approximation $\hat{\mathcal{G}}$, introducing the stochastic approximation error.

For sgLD (and sgHMC), the Euler integration specified through Equation (4.26) can be shown to be of order 1. Its error scales with η^2. For higher-order integrators, the interplay of the approximation and discretization errors for sgMCMC has been analyzed in detail in References [234] and [235]. The authors find, that varying the stepsize of the Euler integrator (4.26) leaves the convergence rate unchanged. They also propose a concrete second-order integrator and show improved convergence. Furthermore, they show that convergence rates of higher-order integration schemes can approach those of full MCMC for high orders. References [218] and [217] also consider higher order integrators in their work.

- **Split Symmetric HMC:** The factorization (4.40) can also be used to distribute one full HMC step into M/N individual subset steps by splitting the Hamiltonian itself into a sum over the subsets. This retains the same dynamics and ensuring reversibility of the chain [236].

  Albeit the dynamics of this sampling seem advantageous in that they do not require handling of the variance of the subset-based gradient estimator, the scheme still requires an M-H correction to correct for the error of the time discretization. A combination with controlled stochastic M-H methods (see Section 5) might thus yield a truly efficient algorithm.

Markov chains also do not need to satisfy the symmetry conditions under which sgLD and sgHMC are developed. Chains with favorable properties can be constructed using weaker conditions.

- **Piecewise deterministic Markov processes [237]:** The reversibility of the introduced methods hinders fast convergence through either the limitation on symmetric proposal distributions or low M-H acceptance rates. Thus, multiple non-reversible, continuous time Markov chains [238–241] have been proposed, with specific focus on datasets with large numbers of points.

  Rather than sampling updates at constant time intervals η, as suggested through the segmentation in Reference 4.40, the times between updates are simulated from an inhomogeneous Poisson process. The intensity function of the Poisson process, that is the phase space dependent rate of the steps to occur, is a function of the log-probability of the model. For the Boomerang Sampler [239], as well as the Bouncy Particle Sampler [240, 241], it is the inner product of the trajectory velocity and the gradient of the log-probability. The trajectory velocity itself is generated from arbitrarily complex update rules that might themselves contain random elements. Strategies like adaptive preconditioning can be applied for these chains similarly as for sgHMC [216, 239]. Through treating the update time intervals as another sampled variable, these algorithms usually do not have discretization errors. They can be constructed such that subset-based versions retain the exact posterior as invariant distribution.

  The main complication of this class of Markov chains, is integrating the intensity function, and thus the log-probability, along the trajectories of the sampling update intervals. This is analytically impossible for BNNs, even if the stochasticity from subset-based gradient estimation does not change the invariant distribution in this scheme [237]. Similar to sgLD the Monte Carlo error of this integration at a fixed number of integration steps increases for smaller sample sizes. Control algorithms similar to the ones discussed in Section 5 could thus be beneficial.

Recently, an application of piece-wise deterministic Markov processes to BNNs that efficiently implements this sampling using adaptive bounds to the gradient of the log-probability was presented [172]. While they find performance on the same level as sgLD and sgHMC, the sampling efficiency of such chains is strongly increased. They also find similar dependencies on the introduced noise (here on the velocity), where too much noise leads to bad convergence and to little will result in bad mode exploration. The authors provide a package for running this sampling with BNNs, which might be a good starting point for future studies.

4.2.4 Metropolis-Hastings Correction and MALA

As pointed out in Reference [242], the naive use of subset-based gradient estimators can lead to large deviations between the chain on full gradients and its stochastic approximation. The presented implementations of Section 4.2.1 Section 4.2.2 and Section 4.2 thus rely on controlled, artificially introduced noise to control the bias of the algorithm. In the limit of diminishing stepsizes, the artificial noise dominates the noise of the stochastic gradient approximation and leads to asymptotically correct sampling as both, the discretization and the influence of the gradient estimator variance, go to zero. In practice, this limit affects the capabilities of the sampling to explore the mode landscape and is thus never realized. Cyclic learning rate schemes try to solve this predicament by distinguishing between exploration and sampling phases [224]. However, through the dependence of the chain during the sampling phase on the exploration steps, as well as the time-discretization, some bias remains.

One way to restore the exact sampling is an M-H correction [243, 244].

1. First, we draw a possible update step $\tilde{\mathbf{z}}_{t+1} \sim q(\mathbf{z}' \mid \mathbf{z})$ from a *proposal distribution*. It gives the probability of proposing $\mathbf{z}'$ starting at $\mathbf{z}$.

2. For this step, we calculate the acceptance probability (ratio)

$$\alpha(\tilde{\mathbf{z}}_{t+1} \mid \mathbf{z}_t) = 1 \wedge \frac{\pi_\lambda(\tilde{\mathbf{z}}_{t+1} \mid \mathcal{D}_n)}{\pi_\lambda(\mathbf{z}_t \mid \mathcal{D}_n)} \frac{q(\tilde{\mathbf{z}}_{t+1} \mid \mathbf{z}_t)}{q(\mathbf{z}_t \mid \tilde{\mathbf{z}}_{t+1})} = 1 \wedge \frac{\exp(-\lambda L_n(\tilde{\mathbf{z}}_{t+1}))}{\exp(-\lambda L_n(\mathbf{z}_t))} \frac{\pi(\tilde{\mathbf{z}}_{t+1})}{\pi(\mathbf{z}_t)} \frac{q(\tilde{\mathbf{z}}_{t+1} \mid \mathbf{z}_t)}{q(\mathbf{z}_t \mid \tilde{\mathbf{z}}_{t+1})}, \tag{4.42}$$

   where $a \wedge b = \min\{a, b\}$.

3. We draw $u \sim \text{Uniform}(0, 1)$. If $u \leq \alpha$, we update to the proposed state $\mathbf{z}_{t+1} \leftarrow \tilde{\mathbf{z}}_{t+1}$. Else, we reject the proposal and revert to the previous state $\mathbf{z}_{t+1} \leftarrow \mathbf{z}_t$.

This construction ensures that the stationary distribution of the chain is the tempered Gibbs posterior $\pi_\lambda(\cdot \mid \mathcal{D}_n)$. For a more in depth discussion of the M-H algorithm, also see Reference [245, Chapter 7.3].

The combination of the stochastic gradient centered proposal distribution from Langevin dynamics (4.28) with an M-H acceptance step is known as the *Metropolis-adjusted Langevin algorithm* (MALA) [245, Section 7.8.5]. As hinted at before, the M-H step is applied to correct for the inherent discretization error of Langevin dynamics. It is thus not limited to small stepsizes to guarantee sampling from the posterior distribution [246]. However, there are three major drawbacks to employing an M-H correction. For one, whenever the stationary distribution of the chain dictated by drawing of $\tilde{\mathbf{z}}_{t+1}$ without correction does not agree well with the desired posterior, acceptance probabilities will be low. The algorithm is thus slowed down significantly by the large amount of rejected samples calculated in vain. This can be solved by carefully designing the proposal distribution.

Furthermore, the correction introduces a sensitive dependence on the width and shape of the proposal distribution. On one hand, a very wide proposal distribution will, with high probability, sample a $\tilde{\mathbf{z}}_{t+1}$ far from the loss minima. This results in an increased corresponding loss $L_n(\tilde{\mathbf{z}}_{t+1})$. On the other hand, low noise will lead to a diminishing probability of the backwards direction under the proposal distribution $q(\mathbf{z}_t \mid \tilde{\mathbf{z}}_{t+1})$, due to the narrow shape of the distribution. This has to be met by carefully choosing the batchsize of the stochastic updates and the shape and width of the proposal distribution.

The biggest issue however, is the cost of the ideal correction. The calculation of the acceptance probability requires the computation of the log-likelihood over the full dataset in every step. For the large datasets encountered in particle physics, this is unfeasible. To solve this, approximations of the ideal correction (4.42) have been proposed. Early proposals for correction schemes based on variable size subsets to reduce the cost of an M-H correction while controlling the introduced bias have been introduced in Reference [211] and [247]. Later publications introduce sub-sampling schemes with fixed size batches, see for example Reference [248]. In Section 5, we give a detailed overview over all proposed methods and elaborate on our own design of a correction to the log-likelihood loss that restores the original scaling with the data size.

4.3 AdamMCMC [P2]

The algorithms presented in the previous section start out by discretizing a differential equation that leaves the Gibbs posterior invariant and usually only sample the true posterior in the limit for small stepsizes. This comes at the cost of slowing down convergence and mixing. The algorithms thus compromise on the speed-up gained by designing the steps on a subset of the data. Thus, if we want to guarantee fast convergence and sampling from a controlled distribution, an M-H correction is indispensable. In Section 5, we discuss different possibilities for performing the correction based on batches of data. We also present our own view on stochastic M-H corrections.

For now, we accept that a discrete-time algorithm needs to employ a correction. As the correction step ensures the stationarity of the chain, the update proposal can be chosen more freely. Specifically, this allows us to start out from a parameter update we know is very efficient for Deep NNs: the Adam algorithm [97]. The update rules (2.17) of Adam are

$$\begin{aligned} m_{t+1} &= \beta_1 m_t + (1-\beta_1)\, \nabla \hat{L}_m(\vartheta_t) \\ v_{t+1} &= \beta_2 v_t + (1-\beta_2)\, \nabla \hat{L}_m(\vartheta_t) \odot \nabla \hat{L}_m(\vartheta_t) \\ \vartheta_{t+1} &= \vartheta_t - \underbrace{\eta_t \frac{m_{t+1}}{1-\beta_1^{t+1}} \oslash \left(\left(\frac{v_{t+1}}{1-\beta_2^{t+1}} \right)^{1/2} + \varepsilon \right)}_{u_t(m_{t+1}, v_{t+1})} . \end{aligned}$$

We construct a parameter proposal $\tilde{\vartheta}_{t+1}$, by sampling from a normal distribution centered in the Adam step

$$\tilde{\vartheta}_{t+1} \sim q_1(\vartheta \mid \vartheta_t, m_{t+1}, v_{t+1}) = \mathcal{N}\left(\vartheta;\, \vartheta_t - u_t(m_{t+1}, v_{t+1}), \Sigma_t(m_{t+1}, v_{t+1})\right) . \tag{4.43}$$

When choosing equal noise levels in all dimensions $\Sigma_t = \sigma^2 \mathbb{1}_P$, the subsequent correction step will result in diminishing acceptance rates, due to the smoothing properties of the momentum auxiliary variables. We thus need to employ another preconditioning on the noise. In contrast to the preconditions introduced in Section 4.2.3, our goal is not to balance the stepsize in all dimensions. The Adam update itself is preconditioned already. We rather want to align the direction of the proposal with that of the momentum. This can be achieved by using an elliptical proposal distribution by choosing

$$\Sigma_t = \sigma^2 \mathbb{1}_P + \sigma_\nabla^2 u_t(m_{t+1})\, u_t(m_{t+1})^\top . \tag{4.44}$$

While we do not add any artificial noise on the auxiliary variables, the variance of the subset based gradient estimator does effectively impose a distribution

$$\begin{aligned} m_{t+1}, v_{t+1} \sim q_2(m, v|\vartheta_t, m_t, v_t) =& \mathcal{N}(m;\, \beta_1 m_t + (1-\beta_1)\, \nabla L_n(\vartheta_t),\, \rho_1^2 \mathbb{1}_P) \\ & \cdot \mathcal{N}(v;\, \beta_2 v_t + (1-\beta_2)\, \nabla L_n(\vartheta_t) \odot \nabla \hat{L}_m(\vartheta_t),\, \rho_2^2 \mathbb{1}_P) , \end{aligned} \tag{4.45}$$

with low noise values $\rho_1, \rho_2 > 0$. We have seen similar reasoning for example in Section 4.2.2. Using the same auxiliary values for the sampling probability $q_1(\tilde{\vartheta}_{t+1} \mid \vartheta_t, m_{t+1}, v_{t+1})$ and the reverse direction

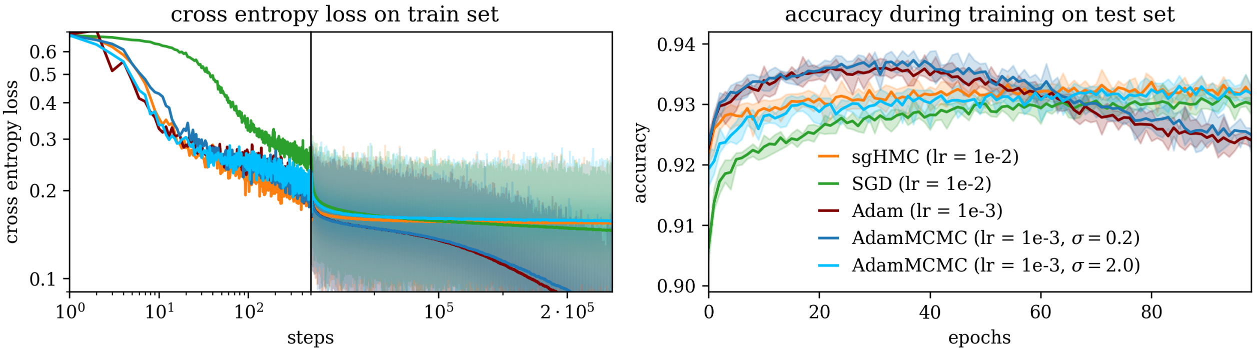


Figure 12: Optimization performance of `AdamMCMC` in comparison to Adam, SGD and sgHMC for Top-tagging with ParticleNet. Left: Cross entropy loss on the training set over steps of the chain/training. The first 500 steps are plotted in log-scaling. For the remaining ≈ 230k steps, the moving-average over 2400 steps (solid), as well as the unsmoothed cross entropy, are shown in linear scaling. Right: Test set accuracy on the (400k jets) over training epochs, of 2400 steps each. `AdamMCMC` ($\sigma = 0.2$) closely resamples the behavior of Adam including overfitting. For larger noise values, `AdamMCMC` ($\sigma = 2.0$) shows no signs of overfitting and a similar optimization performance as sgHMC. Originally published in Reference [P2].

$q_1(\vartheta_t \mid \tilde{\vartheta}_{t+1}, m_{t+1}, v_{t+1})$, we can calculate the acceptance probability ratio as

$$\alpha_t(\tilde{\vartheta}_{t+1}|\vartheta_t, m_{t+1}, v_{t+1}) = 1 \wedge \left(\frac{\pi_\lambda(\tilde{\vartheta}_{t+1}|\mathcal{D}_n)}{\pi_\lambda(\vartheta_t|\mathcal{D}_n)} \frac{q_1(\vartheta_t|\tilde{\vartheta}_{t+1}, m_{t+1}, v_{t+1})}{q_1(\tilde{\vartheta}_{t+1}|\vartheta_t, m_{t+1}, v_{t+1})} C(\vartheta_t, \tilde{\vartheta}_{t+1}) \right) . \tag{4.46}$$

We set $0/0 = 0$ for the case where $p_\lambda(\vartheta_t|\mathcal{D}_n) = 0$. The correction terms C accounts for the drawing of auxiliary variables

$$\begin{aligned} C(\vartheta_t, \tilde{\vartheta}_{t+1}) = \frac{q_2(m_{t+1}, v_{t+1}|\tilde{\vartheta}_{t+1}, m_t, v_t)}{q_2(m_{t+1}, v_{t+1}|\vartheta_t, m_t, v_t)} = \exp \Bigg(&-\frac{|m_{t+1} - \nabla L_n(\tilde{\vartheta}_{t+1})|^2}{2\rho_l^2/(1-\beta_1^2)} + \frac{|m_{t+1} - \nabla L_n(\vartheta_t)|^2}{2\rho_l^2/(1-\beta_1^2)} \\ &-\frac{|v_{t+1} - \nabla L_n(\tilde{\vartheta}_{t+1})^2|^2}{2\rho_l^2/(1-\beta_2^2)} + \frac{|v_{t+1} - \nabla L_n(\vartheta_t)^2|^2}{2\rho_l^2/(1-\beta_2^2)} \Bigg) . \end{aligned} \tag{4.47}$$

After an initial period, the running averages m_{t+1} and v_{t+1} will be close to gradients which decrease over time. Setting $C(\vartheta_t, \tilde{\vartheta}_{t+1}) = 1$ thus is a well justified simplification. In Reference [P2], we prove this chain admits an invariant distribution with the correct marginal distribution $\pi_\lambda(\cdot|\mathcal{D}_n)$ in ϑ. We also show the distribution of ϑ_t converges to the Gibbs posterior distribution in total variation distance.

We test the `AdamMCMC` algorithm by sampling the parameters of the state-of-the-art jet tagging architecture ParticleNet [249]. This architecture constructs a graph from the point cloud of incoming particles and processes this graph with edge convolution blocks [250]. The architecture circumvents evaluation of sparse input and is invariant under shuffling of the particles. In comparison to the Particle Transformer architecture [76] this architecture is very parameter efficient. As this is a classification task, the categorical log-posterior takes on the shape of a binary CE-loss (2.7).

As data, we use the TopLandscape dataset [75]. For training, it contains 600k simulated top and QCD jet events each. We run Adam for 100 epochs (2400 batches of size 512 each) with $\eta_t = 1 \times 10^{-3}$ and $\beta_1 = \beta_2 = 0.99$ as ground truth. Figure 12 shows the resulting test accuracy during training. It exhibits strong overfitting. We can closely reproduce this behavior when running our sampling at low overall noise of $\sigma = 0.2$. We chose $\lambda = 1$ and $\sigma_\Delta = P/100 = 3661.6$ and learning rate and $\beta_{1/2}$ as before. Increasing the noise to $\sigma > 2.0$ prevents this overfitting. The convergence of the adjusted `AdamMCMC` is similar to that of sgHMC.

We find that using a prolate proposal distribution, as specified in Equation (4.44), results in an algorithm that maintains an efficient mean acceptance ratio in the limit $\sigma \to 0$. A scan over multiple orders of magnitude for values of σ is show in Figure 13. In combination with a large span of σ_Δ values that yield solid performance, this robustness to low σ makes `AdamMCMC` easy to use. While other algorithms can be notoriously hard to tune,

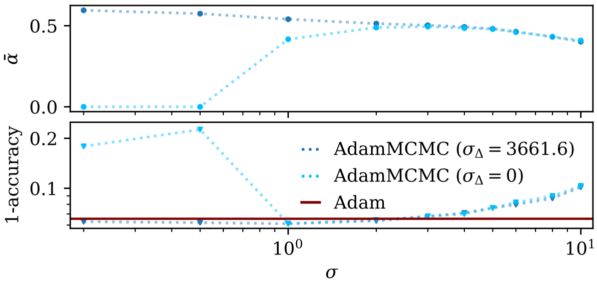


Figure 13: Mean acceptance rate (upper) and accuracy of the posterior mean prediction on test data (lower) over the width of the proposal distribution σ. An algorithm without aligned proposal and momentum (light blue) is strongly dependent on a correct choice of σ. Aligning both (dark blue) enables efficient sampling at low proposal width. The algorithm reproduces the accuracy of the deterministic optimization in this limit. Starting at the low σ limit, noise can be added to prevent overfitting or achieve well-calibrated uncertainties. Originally published in Reference [P2].

for `AdamMCMC` one can simply start out at a working Adam optimization, reuse learning rate and betas, set a low σ and increase σ_Δ until the mean acceptance rate is sufficient. This setting reproduces the Adam results closely. Starting from there, one can increase σ until overfitting is suppressed and the uncertainty prediction is well-calibrated (Figure 14). Other parameters do not affect the error prediction significantly, as long as efficient sampling is guaranteed.

When running this algorithm, we use a stochastic estimator $\hat{L}_m$ of the loss L_n for the M-H acceptance step. This allows efficient computation in batches, but changes the sampled distribution to a mixture of subset posteriors. In theory, this is quite unattractive, even more so as our algorithm heavily relies on the correction for sampling the true Gibbs posterior. However, we do not find dramatic differences between the true posterior and the mixture in practice, In the next section, we introduce several stochastic M-H correction methods with better control of the sampled distribution. This includes our own proposal for stochastic M-H algorithms in Section 5.4.

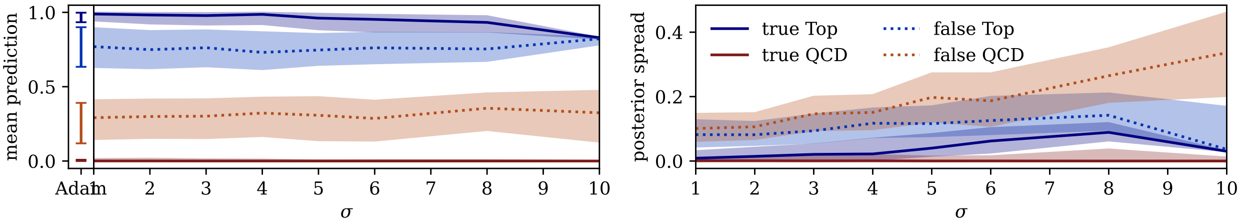


Figure 14: Posterior mean prediction (left) and posterior spread (right), that is difference between the 75%- and 25%-quantile of the prediction for 10 posterior sample, over the noise level σ. The median (center line) and the 75%- and 25%-quantile (envelopes) are reported for both classes and true and false assignments for a test set of 400k jets. A slight decrease in classification performance shows for increasing σ for a large section of the scanned space. The performance of an Adam optimization is recovered well for low noise values. The uncertainty prediction strongly depends on σ. A significant posterior spread can already be found for $\sigma = 1$. Originally published in Reference [P2].

5 Weight Sampling with Stochastic Metropolis-Hastings

Given the high computational cost of the log-likelihood loss computation in Equation (4.42), the search for stochastic equivalents of the M-H correction has experienced a new rise in popularity with the current push to big, that is tall, data. The corresponding literature is scattered within the diverse field of MCMC sampling. A first review of the early approaches is given in Reference [212]. The authors categorize the efforts into two groups:

- **Divide-and-conquer approaches:** This class of algorithms runs multiple chains on subsets of the data and infers an approximation of the posterior from the combination of the chains. It has been shown that this procedure is very sensitive to the quality of the posteriors sampled for every subset [251].
- **Sub-sampling:** Sub-sampling algorithms use a (newly drawn) subset of data at every iteration of the chain to perform an approximate acceptance step.

The former class merely distributes the computation cost over a longer time or multiple machines and often relies on a Gaussian shaped posterior or other approximations for the combination of the posteriors deduced from every sub-sample [212, Section 3]. In this work, we will thus focus on the domain of sub-sampling algorithms as the best current solution to the excessive cost of M-H corrections for NNs in HEP. For the same reason, we will also not further discuss the option of distributing the calculation of a full likelihood over multiple machines.

5.1 Naive Sub-Sampling

Naively exchanging the negative log-likelihood in the calculation of the acceptance probability ratio for the M-H correction with a stochastic estimator

$$\hat{\alpha}(\tilde{\mathbf{z}}_{t+1} \mid \mathbf{z}_t) = 1 \wedge \frac{\exp(-\lambda \hat{L}_m(\tilde{\mathbf{z}}_{t+1}))\ \pi(\tilde{\mathbf{z}}_{t+1})}{\exp(-\lambda \hat{L}_m(\mathbf{z}_t))\ \pi(\mathbf{z}_t)} \frac{q(\tilde{\mathbf{z}}_{t+1} \mid \mathbf{z}_t)}{q(\mathbf{z}_t \mid \tilde{\mathbf{z}}_{t+1})}, \tag{5.1}$$

will lead to a complicated invariant distribution that is hard to interpret [212]. While $\hat{L}_m$ is an unbiased estimator of L_n, the same is not true after exponentiation. $\exp(-\hat{L}_m)$ will clearly introduce a bias over $\exp(-L_n)$. The admitted invariant distribution is thus a mixture of all sub-sampled posteriors. While such a mixture is hard to read, we show in Section 5.4 that it sustains a lot of characteristics of the posterior distribution.

An algorithm employing an estimator with uncontrolled variance can also get stuck at a parameter proposal accepted due to an outlier estimate [212]. This is a phenomenon we often encountered in the early stages of testing `AdamMCMC`.

For the naive sub-sampling scheme, Reference [252] shows the connection between using the log-likelihood estimator in a M-H step and sampling from a tempered posterior at higher, known temperature. For sub-sampling with $m = n^\rho$ and $\lambda = 1$, they show that rescaling the estimator as $n^{\gamma-1}\hat{L}_m$, where γ is a new temperature parameter such that $0 < \gamma < \rho < 1$, will correct the posterior distribution and result in

$$\pi_{n^{\gamma-1}}(\mathbf{z} \mid \mathcal{D}_n) \propto \exp\left(-n^{\gamma-1} L_n(\mathbf{z})\right)\ \pi(\mathbf{z}) \tag{5.2}$$

as an invariant distribution in the limit of large datasets and $1/\lambda \ll n$. Their argument is based on an augmented chain including the difference between estimator and full log-likelihood as an auxiliary variable. If we choose γ close to ρ, the resulting tempering gives $n^{\gamma-1} \approx m/n$ as an upper limit. For this choice, the change in temperature reflects only the change in batchsize analogous to our observations in Section 5.4.

Reference [252] also shows, that sampling at higher temperature, that is lower λ, due to the stochasticity of the acceptance improves the probability of transitions between modes of the posterior. A similar argument is also given in Reference [212, Section 6.3]. In Parallel Tempering [229], this effect of high temperature is leveraged to construct multiple chains at different temperatures and thus different mode exploration properties. Similarly, a temperature scheduling can be applied to achieve better mixing in an exploration phase of the chain.

5.2 Inexact Algorithms with Controlled Bias

In the naive approach, the bias on the sampled distribution is not gauged. A method to control this bias, is to assume a Gaussian distribution for drawings of the log-likelihood estimator $\hat{L}_m(\tilde{\mathbf{z}}_{t+1}) - \hat{L}_m(\mathbf{z})$ justified by the CLT. From this assumption, a statistical test can be employed to test whether the true log-likelihood fulfills

$$\sum_{i=1}^{n} -\lambda \left(L_n(\tilde{\mathbf{z}}_{t+1}) - L_n(\mathbf{z}_t) \right) \geq \log u - \log \left(\frac{\pi(\tilde{\mathbf{z}}_{t+1}) q(\tilde{\mathbf{z}}_{t+1} \mid \mathbf{z}_t)}{\pi(\mathbf{z}_t) q(\mathbf{z}_t \mid \tilde{\mathbf{z}}_{t+1})} \right) \tag{5.3}$$

with high probability. If it does not, the batchsize is increased until the correct acceptance decision is taken with a previously fixed p-value of the test. Note that Equation (5.3) is just a reshuffling of $u \leq \alpha$ for an acceptance rate as in Equation (4.42). This procedure, applying a T-test, has been proposed in Reference [247] as *AusterityMH*. One can achieve a good reduction in computational cost at reliable sampling whenever the Gaussian assumption from the CLT applies, but runs the risk of large and uncontrolled biases for example for posteriors with strong tails [212].

A similar adaption of the batchsize in combination with concentration inequalities to construct confidence bounds $c_m(\delta)$ satisfying

$$\mathbb{P}\left(\frac{\lambda}{n} \left| \left(\hat{L}_m(\mathbf{z}_t) - \hat{L}_m(\tilde{\mathbf{z}}_{t+1}) \right) - \left(L_n(\mathbf{z}_t) - L_n(\tilde{\mathbf{z}}_{t+1}) \right) \right| > c_m(\delta) \right) \geq 1 - \delta \tag{5.4}$$

is applied in Reference [211]. The algorithm predicts a subset size m for a user-specified confidence $1 - \delta$ to perform the M-H step. The authors however find that in non-favorable cases, this yields almost no reduction of the computation as the predicted m is close to the size of the full dataset. The follow-up improved confidence sampler [212], uses a proxy of the full log-likelihood in form of a second order Taylor expansion to replace many likelihood estimations and achieve a notable speed-up compared to Reference [211].

In a similar manner, a Barker test [253] (logarithmic M-H test) is employed in Reference [248] to achieve controllable biases from adaptive batches. They leverage the first three moments of the batch of log-likelihood ratios to estimate the confidence of the accept-reject decision.

All three papers test the algorithms on Gaussian distributions and logistic regression, but not on large scale NNs. For BNN applications, the iterative nature of the tests will slow down the computation considerably in comparison to stochastic optimization, all while the multi-modal structure of an over-parameterized NN posterior further reduces the efficiency of the sampling.

5.3 Exact Algorithms

For the introduced inexact algorithms, one can show that for arbitrary constants $\delta \in (0, 1)$ and $\rho > 0$ one can find a target posterior π and proposal distribution q such that the actual stationary distribution of the inexact algorithm $\tilde{\pi}$ satisfies [254, Theorem 1]

$$\mathrm{TV}(\pi, \tilde{\pi}) \geq \delta \quad \text{and} \quad D_{\mathrm{KL}}(\pi, \tilde{\pi}) \geq \rho \,.$$

In theory there will thus always be a target distribution for which the sampling included a controlled stochastic M-H correction is arbitrarily bad. In practice, this can often be circumvented by a smart design of the proposal (see for example Section 4.2.2 and Section 4.3). However, it still motivates the design of stochastic M-H corrections that are exact, that is they sample from the full posterior $\pi_\lambda(\vartheta \mid \mathcal{D}_n)$.

The easiest way to reduce computation cost while retaining the exact invariant distribution of choice is a *two-stage M-H correction* [255]. In a first step, a point drawn from the proposal distribution is accepted or rejected based on a sub-sample estimate of the log-likelihood. If it is accepted, a second accept-reject step based on the full sample is calculated such that the combined transition kernel reduces to the full kernel exactly. While this might yield large savings for the case of low acceptance rates, we want to employ gradient based proposal distributions to navigate the large dimension of the BNN posterior efficiently. We thus aim at high acceptance rates, where the gains from such a two-step approach are marginal.

Amortized Metropolis Adjustment (AMA), as introduced with in AMAGOLD [256], cuts the cost by using an infrequent M-H corrections only every $T > 0$ steps. The idea is charmingly simple:

1. Sample T consecutive steps $\tilde{\vartheta}_{t+1}, ..., \vartheta_{t+T}$ from the corresponding proposal distributions
$$q(\tilde{\vartheta}_{t+1} \mid \vartheta_t), ..., q(\tilde{\vartheta}_{t+T} \mid \tilde{\vartheta}_{t+T-1})\,.$$
2. Accept all T steps with
$$\alpha_{\text{AMA}}(\tilde{\vartheta}_{t+T} \mid \vartheta_t) = 1 \wedge \frac{\exp(-\lambda \hat{L}_m(\tilde{\vartheta}_{t+T}))\ \pi(\tilde{\vartheta}_{t+T})}{\exp(-\lambda \hat{L}_m(\vartheta_t))\ \pi(\vartheta_t)} \prod_{i=1}^{T-1} \frac{q(\tilde{\vartheta}_{t+i+1} \mid \tilde{\vartheta}_{t+i})}{q(\tilde{\vartheta}_{t+i} \mid \tilde{\vartheta}_{t+i+1})} \frac{q(\tilde{\vartheta}_{t+1} \mid \vartheta_t)}{q(\vartheta_t \mid \tilde{\vartheta}_{t+1})} \tag{5.5}$$
to restore detailed balance, that is the reversibility, of the chain.

Contrary to the previously introduced two-stage M-H, this algorithm shows the best improvements for high acceptance rates and proposal distributions that are already close to reversible. It should thus work especially well with the directional proposal distribution of Section 4.3. For the combination of AMA with sgHMC, that is the AMAGOLD algorithm, the authors note that sgHMCs proposals are not close to reversible without negating the momentum term. They thus introduce a notion of "skew-reversibility" including the negation, that can be converted back to a regular reversible chain through momentum resampling, to reach high acceptance rates. The resulting algorithm is shown to converge to the exact posterior at finite stepsizes at a rate at most a constant factor slower than sgHMC. The algorithm is demonstrated to run efficiently and robustly on various examples, including a two-layer MLP classification task with 60000 data points. Code is shared with the paper and might be interesting for future applications in HEP.

For high acceptance rates, better reductions can be achieved with strong assumptions on the target, that is the posterior [257, 258]. Both require a tight and easy to compute (global) lower bound of the log-likelihood and sum the difference between bound and estimator over a subset of data. As such a bound is not easily available for NN posteriors, the application of these methods to BNNs seems impractical.

References [259, 260] also introduce exact algorithms for posterior sampling from batches. They however come at increased computational cost, limiting their merit over the full correction. The algorithm proposed in Reference [259] calculates the acceptance ratio for each step from a product of parallel batch-wise estimates of the log-likelihood, effectively performing the calculation over the all data points. Similar to the algorithms proposed in Reference [211, 212], the *Scalable M-H* of Reference [260] relies on a Gaussian posterior to construct a factorized M-H Kernel from a Taylor expansion of the log-posterior. The Taylor expansion has to be calculated at every step of the chain, limiting the possible speed-up of the algorithm. Scalable M-H [260] also assumes bounds on the energy difference, that is difference in log-prior and log-likelihood. This is easily implemented through a Lipschitz constant and limited stepsizes.

The same bound is assumed for the *tunable M-H algorithm* (TunaMH) [254]. It however does not impose any further assumptions on the posterior shape. The authors propose drawing batchsizes according to $B \sim \text{Poisson}(\chi C^2 M^2(\vartheta_t, \tilde{\vartheta}_{t+1}) + C M(\vartheta_t, \tilde{\vartheta}_{t+1}))$ for $C = \sum_{i=1}^n c_i$, $c_1, ..., c_n \in \mathbb{R}_+$, $\chi \in \mathbb{R}_{>0}$ and a symmetric function M such that $|U_i(\vartheta_t) - U_i(\tilde{\vartheta}_{t+1})| \leq c_i M(\vartheta_t, \tilde{\vartheta}_{t+1})$. Here for the dataset $\mathcal{D}_n = x_1, ..., x_n$ the energy per point
$$U_i(\vartheta) = -\log \pi(x_i|\vartheta) - \frac{1}{n} \log \pi(\vartheta)$$
is used. Subsequently, they choose B points uniformly distributed over the dataset and add them to the sub-sample I based on the relation of the individual energy difference $U_i(\vartheta_t) - U_i(\tilde{\vartheta}_{t+1})$ and their bound $c_i M(\vartheta_t, \tilde{\vartheta}_{t+1})$. They make the correct M-H acceptance ratio out to be
$$\alpha_{\text{TunaMH}}(\tilde{\vartheta}_{t+1} \mid \vartheta_t) = 1 \wedge \exp\left(2 \sum_{i \in I} \text{artanh}\left(\frac{U_i(\vartheta_t) - U_i(\tilde{\vartheta}_{t+1})}{c_i M(\vartheta_t, \tilde{\vartheta}_{t+1})(1 + 2\chi M(\vartheta_t, \tilde{\vartheta}_{t+1}))}\right)\right) \frac{q(\tilde{\vartheta}_{t+1} \mid \vartheta_t)}{q(\vartheta_t \mid \tilde{\vartheta}_{t+1})}\,.$$

The resulting sampling is shown to be exact with a convergence speed that is tunable through the choice of χ. This is the first proof of a connection between batchsize and convergence rate, that is between scalability and efficiency. However, the assumption of a-priori knowledge of the c_i and M make this algorithm hard to apply for the arbitrarily complex energy landscape of BNNs, as for loose bounds the sampled batchsize B will be close to the full set size n. As such, the numerical experiments in Reference [254] only employ the sampling to a Gaussian mixture model and logistic regression.

Penalty BNNs [261] solve the same problem with a different approach. Once again, they assume the loss (here an estimator of the log-posterior including the bias) is normal distributed around the true values with variance $\sigma^2(\tilde{\vartheta}_{t+1}, \vartheta_t)$ due to the stochasticity of the sub-sampling. They then apply the concept of noise penalty [262] to the M-H correction, that is penalize the acceptance rate (5.1)

$$\alpha_{\mathrm{PBNN}}(\tilde{\vartheta}_{t+1} \mid \vartheta_t) = 1 \wedge \hat{\alpha}(\tilde{\vartheta}_{t+1} \mid \vartheta_t) \exp(-\sigma^2(\tilde{\vartheta}_{t+1}, \vartheta_t)/2). \tag{5.6}$$

Here, $\hat{\alpha}$ is as in Equation (5.1), but without the clipping operation $1 \wedge \cdot$. This correction is sufficient to reestablish detailed balance on average and ensures sampling from the full posterior. It however comes at the cost of exponentially suppressing the acceptance of update steps and thus significantly slowing down the computation. To remedy this fact, the authors employ the mean over multiple batches as an unbiased estimator of the log-posterior difference in $\hat{\alpha}(\tilde{\vartheta}_{t+1} \mid \vartheta_t)$, as well as a chi-squared estimator of its variance. This improves the acceptance rates, but significantly increases computation. The authors employ their algorithm on a 1-dimensional NN regression with only 2998 data points and find significant improvements over the naive M-H correction or sgLD. The size of the example, as well as the increased computation from using multiple batches per step give reasons to question the applicability of noise penalty methods to large scale BNNs.

Similar control variates to Equation (5.6) are proposed and analyzed in detail in Reference [263] to control the variance of the likelihood estimator itself (rather than the acceptance probability). Instead of the naive rescaled sum, a difference estimator can be used for a better grasp on the estimator variance [263]. The resulting method samples from a perturbed posterior dependent on the variance of the likelihood estimator. However, the derivative of these considerations using a block-Poisson estimator of the likelihood [264] reinstates exact sampling of the posterior expectation values. The application to BNNs however is discussed in neither and the strong dependence on correctly estimating the variance of the employed estimators could make both algorithms difficult to adjust for noisy data.

5.4 Statistical Guarantees for Stochastic Metropolis-Hastings [P3]

In Reference [P3], we study the MALA algorithm with a subset-based M-H correction for regression settings and matched data $\mathcal{D}_n = \{(x_i, y_i)\}_{i \in 1,\dots,n}$. For these settings, the NLL corresponds to the MSE (2.8). To prevent the calculation of the MSE over a full dataset, we draw auxiliary variables from a Bernoulli distribution $b_i \sim \mathrm{Ber}(\rho)$ for some $\rho \in (0, 1]$. The loss on the subset defined through $B = \{b_1, \dots, b_n\}$ is then

$$L_n(\vartheta, B) = L(\vartheta, B; \mathcal{D}_n) := \frac{1}{n\rho} \sum_{i=1}^{n} b_i \underbrace{(y_i - f_\vartheta(x_i))^2}_{=: l_i(\vartheta)} . \tag{5.7}$$

We thus have to consider the joint target distribution

$$\begin{aligned} \overline{\pi}(\vartheta, B \mid \mathcal{D}_n) &\propto \prod_{i=1}^{n} \rho^{b_i} (1-\rho)^{1-b_i} \exp(-\lambda L_n(\vartheta, B))\, \pi(\vartheta) \\ &\propto \exp\left(-\lambda L_n(\vartheta, B) + \log\left(\frac{\rho}{1-\rho}\right) \sum_{i=0}^{n} b_i \right) \pi(\vartheta) \, . \end{aligned} \tag{5.8}$$

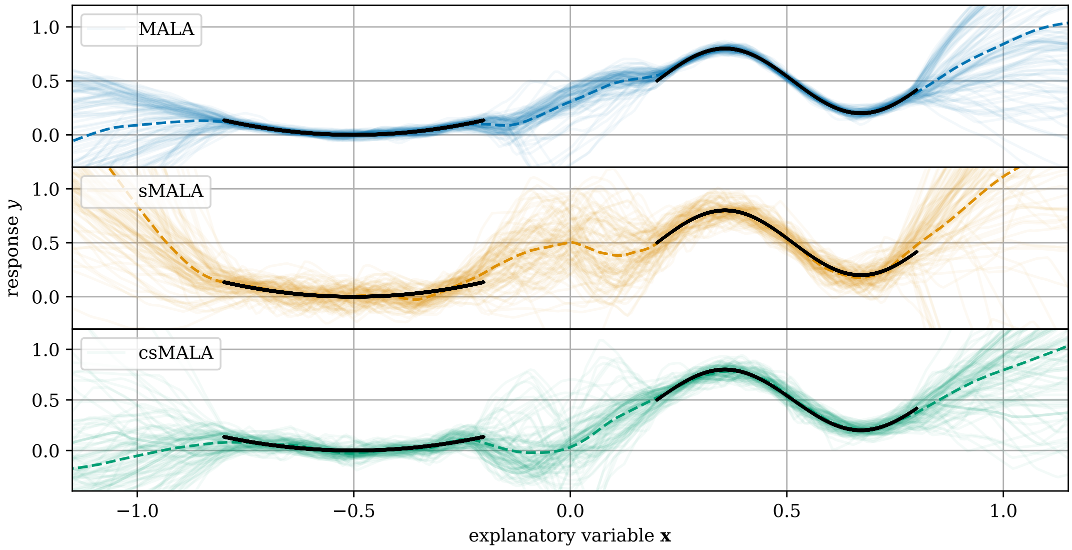


Figure 15: Training data for the regression task (black), as well as functions for 50 parameter samples drawn with MALA, uncorrected stochastic MALA and stochastic MALA including the correction from Equation (5.11).

Summing over all drawn b leaves us with the marginal in ϑ

$$\begin{aligned}\overline{\pi}(\vartheta \mid \mathcal{D}_n) &\propto \prod_{i=1}^{n}\left[\rho\exp\left(-\frac{\lambda}{n\rho}l_i(\vartheta)\right)+1-\rho\right]\pi(\vartheta)\\ &= \exp\left(-\lambda\underbrace{\left[-\frac{1}{\lambda}\sum_{i=1}^{n}\log\left(\rho e^{-\frac{\lambda}{n\rho}l_i(\vartheta)}+1-\rho\right)\right]}_{=:\overline{L}_n(\vartheta)}\pi(\vartheta)\right).\end{aligned} \tag{5.9}$$

It resembles the Gibbs posterior (4.21), but for an adapted loss $\overline{L}_n(\vartheta)$. We refer to this posterior as the *surrogate posterior* and show that for sufficiently small $\frac{\lambda}{n\rho}$ the surrogate posterior is a good approximation of the true posterior in terms of the KLD [P3, Lemma 1]. For very small ρ and thus small batches, it behaves like the mixed posterior

$$\prod_{i=1}^{n}\exp\left(\rho e^{-\frac{\lambda}{n\rho}l_i(\vartheta)}\right)\pi(\vartheta)\,. \tag{5.10}$$

This mixture largely resembles the prior for all non-optimal ϑ, that is whenever $l_i(\vartheta) > 0$.

When adapting the loss to

$$\tilde{L}_n(\vartheta, B) := \frac{1}{n}\sum_{i=1}^{n}b_i l_i(\vartheta) + \zeta\frac{\log\rho}{\lambda}\sum_{i=1}^{n}b_i\,, \tag{5.11}$$

we find the marginal of the corresponding invariant distribution

$$\tilde{\pi}(\vartheta \mid \mathcal{D}_n) \propto \prod_{i=1}^{n}\left[\exp\left(-\frac{\lambda}{n}l_i(\vartheta)\right)+1-\rho\right]\pi(\vartheta) \tag{5.12}$$

resembles the true Gibbs posterior (4.21) with a reduced inverse temperature of $\frac{\lambda}{1-\rho}$ for small $\frac{\lambda}{n}$. The dependence on the sampling probability is dropped and the scaling of MALA restored. Again this guarantee is given in terms of the KLD between both posteriors.

For a simple one dimensional regression setting (see Figure 15), we sample a NN with 10401 parameters. We examine the scaling of the validation MSE-loss of the posterior mean prediction on a validation set of 10000

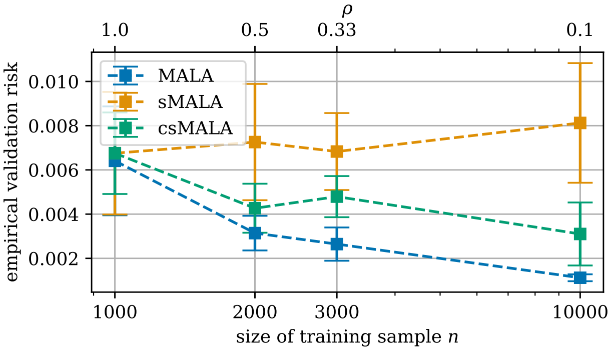


Figure 16: Scaling of the validation loss on the size of the training sample. We find the scaling of MALA is restored with the correction. Originally published in Reference [P3].

points in Figure 16. The corrected stochastic MALA exhibits the same scaling with the number of training samples n as MALA. The naive approach however, is dependent on the batch size $n\rho = 1000$, which we keep constant. It does not show any dependence on n. We find similar results for the radii of credible sets calculated from the posterior samples.

6 Uncertainty in Generative Machine Learning

In Section 3, we have discussed multiple ways to construct generative NNs. For many popular applications of generative ML, such as image generation or language modeling, uncertainties as discussed in Section 4 are of no meaning. In such applications, only single points are generated in data space. Thus, reproducing realistic images or chat responses is valued the most and the uncertainty on the distribution of all generated points is not evaluated. In science applications, for example in emulating detector simulation, we generated millions of points in data space. This could be done either to infer the distribution of a variable or to compare against experimental data. In 6.1, we give an example for a sensible application of generative uncertainties for inference and close the loop back to the idea of amplification in Section 6.2.

The previous sections already define all the technical tools needed to construct Bayesian generative NNs. While adversarial learning only defines a loss objective which can be understood in probabilistic terms for optimal discriminator power, VAEs are inherently Bayesian. They infer the posterior distribution of latent space variables $p(z \mid x)$ from input variables as a parameterized fit with a NN. This setting can be expanded to a full Bayesian phrasing of all network weights [265]. For an efficient implementation using the LA, see Reference [266]. As VAEs can be used to reconstruct images by applying the encoder and the decoder sequentially, a Bayesian VAE can even be applied on an image-by-image basis to indicate high variance regions of the input. It is interesting to note, that the variance on a reconstructed image can be used as a method for edge detection.

In HEP, VAEs however have been replaced in favor of (continuous) flows, diffusion models and transformers to increase the expressive power of the modeling. For these architectures, the log-likelihood of the data is accessible via the Equations (3.19), (3.26), (3.31) and (3.34). Combinations with the Bayesian methods of Section 4 therefore seem self-evident. To apply VI, the log-likelihood in Equation (4.14) can be calculated as one of the above [17]. For the LA (4.19) the MAP-loss (4.5) also consists of the log-likelihood and a prior term. And sgMCMC methods sample the Gibbs posterior (4.21), which resembles the true posterior for NLL-losses (4.22). This gives us a large variety of possible methods to choose from.

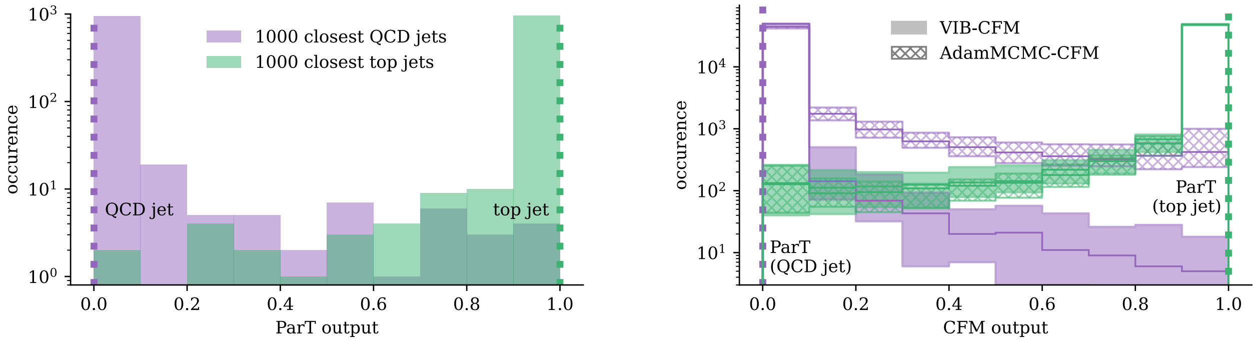
For the studies presented in References [P4] and [P5], we select CNFs due to their parameter efficient nature, as well as methodical proximity to diffusion models and block-based flows. We combine this architecture with both, the widely popular VI scheme and our proposed `AdamMCMC`. Following Reference [17], we use the CFM-objective (3.29) in combination with VI for efficient optimization. The CFM-loss however has no clear interpretation as a log-likelihood. We thus need to introduce an additional, tunable parameter k to account for the difference

$$L_{\mathrm{VIB-CFM}}(\phi;\mathcal{D}_n) = \mathbb{E}_{\vartheta\sim\tilde{\pi}(\vartheta;\phi)}\left[L_{\mathrm{CFM-OT}}(\vartheta;\mathcal{D}_n)\right] + kD_{\mathrm{KL}}\left[\tilde{\pi}(\vartheta;\phi),\pi(\vartheta)\right] . \tag{6.1}$$

6.1 Classifier Surrogates [P4]

As hinted at in Section 3.3, one popular application of NFs is to infer conditional likelihood distributions. In the domain of inverse problems, where $p(y \mid x)$ is defined through an irreversible forward process, using a flow to estimate the inverse direction $p(x \mid y) \approx \hat{p}(x \mid y; \vartheta)$ is a popular approach. It is often referred to as *neural posterior estimation* [118–120]. Such a setup can be interesting for tuning nuisance parameters or measuring theory parameters of the irreversible MCMC simulations used in HEP [267]. In Reference [P4], we look at the converse problem, that is approximating the forward direction itself. This is sometimes called *neural likelihood estimation*.

Modern analysis pipelines at CMS and ATLAS often employ NN-based tagging algorithms. The tagging output is a complex function of the detector response and hard to understand in terms of physical representations. This is problematic when sharing the analysis within and outside the collaboration. Reinterpretation from outside the collaboration is impossible, as the detailed detector schematics are only available within and the tagger input cannot be simulated. Furthermore, reproducing the result, if possible, still requires the costly simulation of the detector. For these reasons, it would be desirable to have a prediction of the tagger output from high-level jet observables or simulation data before or after hadronization. Such a *surrogate* would need to



(a) Detector smearing of the tagger output

(b) Learned approximation of the tagger output

Figure 17: Prediction of the distribution of ParT output imposed through the detector smearing of a QCD jet event (purple) and top jet event (green). Figure 17a shows an approximation by a histogram of the 1000 points closest in transverse momentum, energy and particle number. Figure 17b shows the posterior mean approximation of a CNF samples from a VI-posterior (solid) as well as the approximation inferred by stochastic optimization of the CFM-objective (checkered). The envelopes depict the min-max envelope over 11 samples drawn starting a `AdamMCMC` chain at the MAP-estimate or from the VI result. Figures originally published in Reference [P4].

reproduce the tagger data well. Through the stochasticity of the detector simulation, multiple detector events can correspond to the same high-level observables or simulation results preceding the detector. Each of the events can differ in tagger output. The tagger output for fixed observables thus follows a distribution that needs to be approximated with a conditional NF, the *detector smearing distribution.* In addition, in cases where input is given, that is not included in the training of the tagger or its surrogate, neither the tagger nor the surrogate output is not to be trusted. The surrogate needs to indicate this through balanced predictions or large uncertainties.

As an example for our proof-of-principle study, we choose top-tagging on the JetClass dataset [204] with the state of the art ParT architecture [76]. ParT uses the kinematics, particle identification, and trajectory displacement information measured by the detector. We compile a new dataset including the ParT predictions, as well as observables of the of the jet events. The observables are the transverse momentum p_T, pseudorapidity η, scattering angle Φ, jet energy E_{jet}, number of particles n_{const}, soft drop mass m_{SD} [268] and N-subjettiness τ_N [269] for $N = 1, ..., 4$. Figure 17a shows the distribution of the ParT output for the 1000 events in the 10^8 point dataset that are closest to one randomly chosen QCD and top jet event in p_T, E_{jet}, and n_{const}. This is a

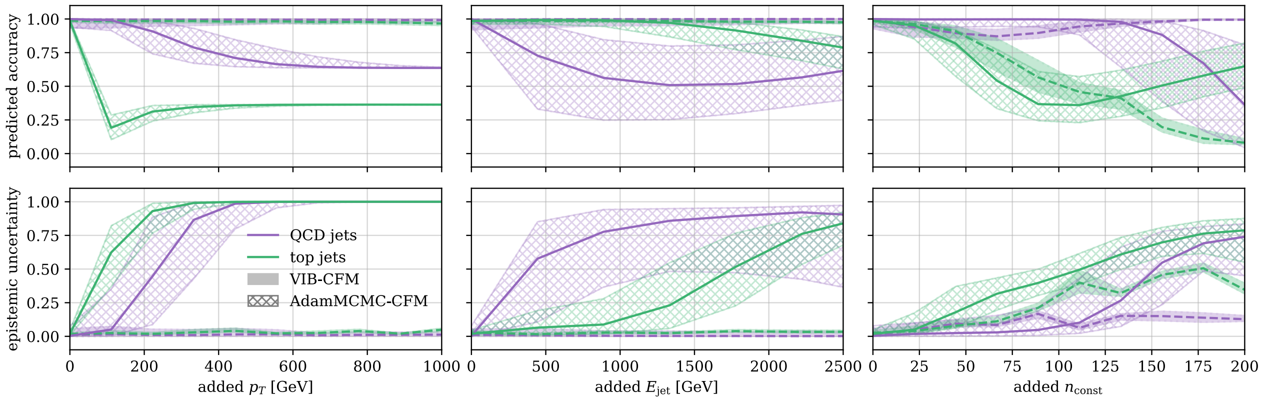


Figure 18: Dependence of the accuracy (top row) and mean epistemic uncertainty (bottom row) when distorting the input by artificially adding fixed amounts to the depicted high-level observables. Solid lines show the median over 1000 QCD events (purple) and top events (green) drawn at random. Envelopes indicate the 10%-90%-quantile envelope of the VIB and `AdamMCMC` ensemble respectively. Figures originally published in Reference [P4].

first approximation of the detector smearing distribution.

We approximate the true distribution with a CNF with CFM and with VI for the CFM-objective (6.1) using Adam. Starting at the result of the stochastic optimization, we sample 10 points from the network posterior with `AdamMCMC`. Qualitatively, we find similar shapes in the conditioned output of the model (Figure 17b) as for the proximity based approximation (Figure 17a). However, we find significant discrepancies between both models. When checking the calibration of the uncertainty prediction on 10^5 validation samples however, we find both setups give well-calibrated predictions [P4].

To investigate the indication of unknown data, we examine the prediction when adding increasing amounts of unphysical distortions to the jet observables. We report the mean accuracy of the prediction for a class distinction at 0.5 (predicted accuracy). We also give the mean distance between the minimal and maximal prediction in the set of posterior samples as a measure of the epistemic uncertainty. In Figure 18, we see that the `AdamMCMC` ensemble is much more sensitive to distortions than the VI one. Large distortions are indicted, once all events are outside the expected intervals. This can be seen in both measures. The epistemic uncertainty measure goes to 1 and the predicted accuracy approaches a balanced prediction of 0.5.

6.2 Bayesiamplification [P5]

When generating large numbers of samples with a generative NN, the bias left on the generated data is purely based on the limited statistics of the training set of the model. This bias can be estimated in terms of the epistemic uncertainty with the Bayesian setups introduced previously. The question thus arises: Can we construct an estimator of the amplification (3.35) from the epistemic uncertainty prediction? We try to answer this question in Reference [P5].

To this end, we use the Bayesian CNFs introduced in the previous sections. These are VI with a CFM-loss and `AdamMCMC` sampling of a CNF based on its log-likelihood. The toy data we use is closely related to that of the original GANplification study [1] introduced in Section 3.6. Both studies use data from a ring distribution. To increase to difficulty of the task, we sample the radial coordinates from a Gamma distribution

$$p_X(x) = p_X(r, \varphi) = \Gamma(r; 2, 2) \times \text{uniform}(\varphi; 0, 2\pi)\,,$$

which is not differentiable at $r = 4$. The histogram of the resulting two-dimensional data distribution in Cartesian coordinates, as well as a plot of the marginal distribution along the radius, is shown in Figure 19.

The samples $\boldsymbol{\vartheta} = \{\vartheta^1, ..., \vartheta^{n_\text{stat}}\}$, drawn either with `AdamMCMC` or from the VI posterior approximation $\tilde{\pi}(\vartheta; \phi)$, give us two sets of approximations of the toy distribution

$$\{\hat{p}_X(x, \vartheta')\}_{\vartheta' \in \boldsymbol{\vartheta}_\texttt{AdamMCMC}} \quad \text{and} \quad \{\hat{p}_X(x, \vartheta')\}_{\vartheta' \in \boldsymbol{\vartheta}_\text{VI}}\,.$$

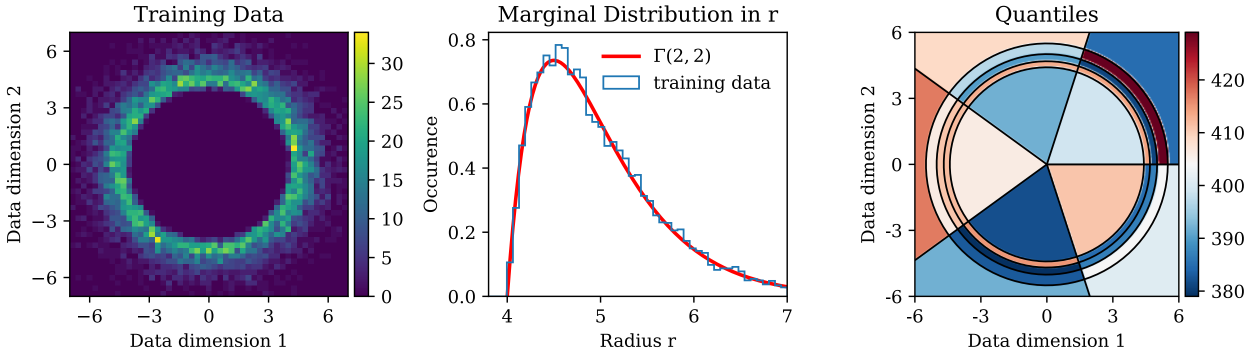


Figure 19: Left and middle: Histogram of the 10000 point training data in Cartesian coordinates (left), as well as marginal histogram and marginal data distribution in radial direction (middle). Right: Count of the training data in 5 quantiles constructed form a 10^7 point validation set. Figures originally published in Reference [P5].

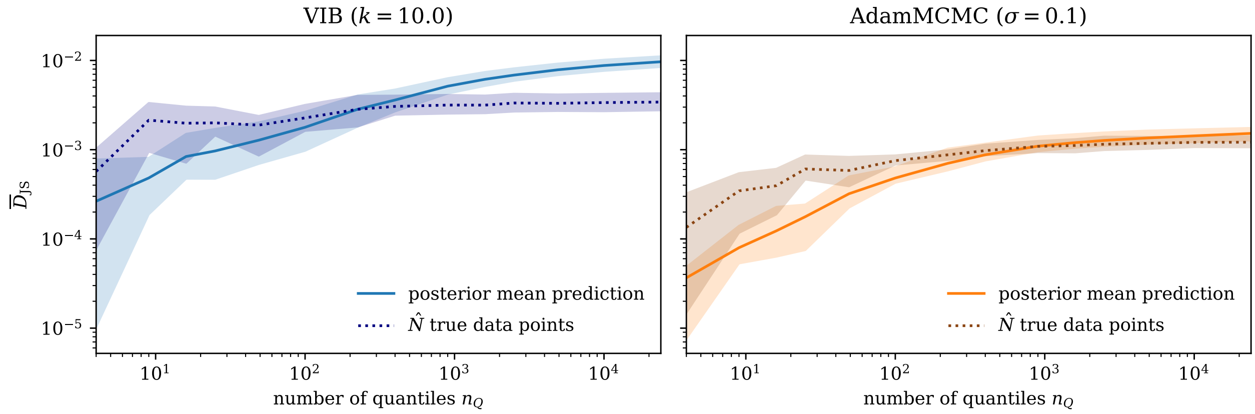


Figure 20: JSD between the posterior mean prediction of the data approximation and the data distribution, as well as JSD between an independent set of the size $\hat{N}$ estimated from the relative error predicted on the bin counts by a BNN. Both JSD measurements exhibit a dependence on the number of quantiles n_Q through the increase in amplification power shown in Figure 21. Error bars are calculated as the minimum and maximum of 5 independent repetitions. Figures originally published in Reference [P5].

To evaluate the approximations we sample sets $\mathcal{D}^i_{n_{\mathrm{gen}}} \sim \hat{p}_X(x, \vartheta^i)^{n_{\mathrm{gen}}}$ and count the number of draws in bins of equal expected probability weight $\boldsymbol{Q} = \{Q_1, ..., Q_{n_Q}\}$. As an example, the right panel of Figure 19 shows the quantiles for 5 bin in radial and angular direction each. For the distribution properties h_j, we again use the count in quantile Q_j defined in Equation (3.37), but without normalizing by the number points. To compare to the results of Section 3.6 and Section 3.7, we calculate the JSD based measurement of distance

$$\overline{D}_{\mathrm{JS}}(\hat{h}(o(\mathcal{D}^i_{n_{\mathrm{gen}}})), h(p_O))$$

(3.39) for all parameter samples. The mean results over $\boldsymbol{\vartheta}_{\texttt{AdamMCMC}}$ and $\boldsymbol{\vartheta}_{\mathrm{VI}}$ are reported in Figure 20. We compare them against

$$\overline{D}_{\mathrm{JS}}(\hat{h}(o(\mathcal{D}_{\hat{N}})), h(p_O))\,.$$

Here,

$$\hat{N} := \sum_{j=1}^{n_Q} \frac{\mu^2_{\hat{h}_j}}{\sigma^2_{\hat{h}_j}} := \sum_{j=1}^{n_Q} \frac{\mathbb{E}_{i\in\{1,\dots,n_{\mathrm{stat}}\}}\left[\hat{h}_j(o(\mathcal{D}^i_{n_{\mathrm{gen}}}))\right]^2}{\mathbb{E}_{i\in\{1,\dots,n_{\mathrm{stat}}\}}\left(\hat{h}_j(o(\mathcal{D}^i_{n_{\mathrm{gen}}})) - \mu_{\hat{h}_j}\right)^2} \tag{6.2}$$

is an estimator of the amplification estimate a (3.35). It is designed as the number of independently drawn true data points in $\mathcal{D}_{\hat{N}}$, whose Poisson error on $\hat{h}_j$ equals the relative error predicted by the Bayesian CNFs. The dependence of the amplification power $\hat{N}/n$ on the number of quantiles n_Q is given in Figure 21. Again we see

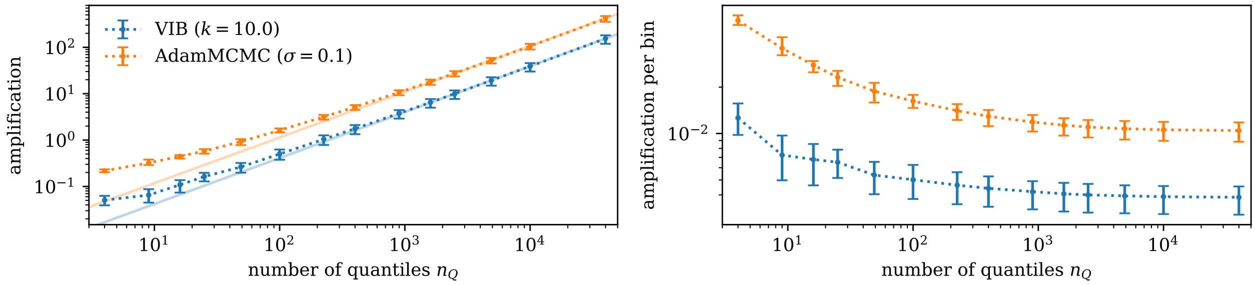


Figure 21: Dependence of the amplification estimate $\hat{N}/n$ (left), as well as $\hat{N}/(n \cdot n_Q)$ (right), on the number of quantiles n_Q. Error bars are calculated as the minimum and maximum of 5 independent repetitions. Figures originally published in Reference [P5].

a strong, exponential dependence. This is in line with the results of Section 3.6 in Figure 5.

Furthermore, $\overline{D}_{\mathrm{JS}}(\hat{h}(o(\mathcal{D}^i_{n_{\mathrm{gen}}})), h(p_O))$ and $\overline{D}_{\mathrm{JS}}(\hat{h}(o(\mathcal{D}_{\hat{N}})), h(p_O))$ align well for high numbers of quantiles and well-calibrated uncertainties, as shown in Figure 20. We test the calibration of the $\hat{h}_j$ with 5 independent sets of samples and find that choosing a k for VI or σ for `AdamMCMC`, that shows good calibration over multiple orders of magnitude in n_Q, is hard. In line with the figure, we find the predictions are underconfident where $\overline{D}_{\mathrm{JS}}(\hat{h}(o(\mathcal{D}_{\hat{N}})), h(p_O))$ is below $\overline{D}_{\mathrm{JS}}(\hat{h}(o(\mathcal{D}^i_{n_{\mathrm{gen}}})), h(p_O))$ and overconfident where they are above. A setting that produces well-calibrated uncertainties for a large spread of n_Q and underconfident ones for low numbers was only found for `AdamMCMC`. For further details on how to judge calibration, as well as training the models we refer to Reference [P5].

7 Conclusion and Outlook

Motivated by the surge in required event simulation due to the upcoming HL-LHC runs, replacing the detector simulation in the HEP simulation chain by NN surrogates has recently been proposed. Previous studies have shown great agreement between data and surrogate output and large possible speed-ups. However, a rigorous study of data amplification has only been conducted in Reference [1] for a low-dimensional toy example.

In Reference [P1], we provide numerical proof for amplification in a realistic setting. For 10×10 images of calorimeter data, we examine the agreement between histograms of surrogate data and large amounts of true data in terms of the JSD. We estimate the amplification by comparing the results to the JSD of histograms of training data and the same validation data. This amplification estimate strongly depends on the number of histogram bins. We conclude, that estimates of global properties of the data distribution, such as the mean and standard deviation, cannot be improved with surrogate data. Detector simulation however can be understood as a convolution with a complex smearing distribution. As such die functional shape itself and thus small bins have to be considered. In this regime we find the surrogate output can resemble up to 50 times the size of the training data. We further find that for small bins our NN model performs on the same level as a histogram or KDE surrogate trained on 5 times the amount of data. The NN surrogate however scales much better to higher dimensions.

To break the dependence on large amounts of data for the estimation of the mismodeling of the surrogate distribution, we turn to Bayesian ML. We proof that well-calibrated epistemic uncertainties can indeed be used to construct on estimate of the amplification [P5]. This amplification estimate agrees with the one from comparing JSD values studied in Reference [P1]. It also depends on the number of bins in the same way. However, we also find that well-calibrated uncertainty predictions are crucial for a valid estimate.

In two studies on different applications, we observe that the widespread VI method for BNNs when applied to CNFs is hard to calibrate [P4, P5]. Out-of-distribution data is not indicated, due to the imposed Gaussian shape of the weight posterior distribution. And the quality of the approximation lacks behind the deterministic implementation or MCMC approaches. In comparison, our own implementation of stochastic gradient MCMC, `AdamMCMC` [P2], provides tunable uncertainty predictions and out-of-distribution indication through large uncertainties and balanced predictions. It constitutes a drop-in replacement for common stochastic optimization methods and can thus be used for arbitrary architectures employing a log-likelihood objective without being reliant on fixed layer implementations. The reliable indication of out-of-distribution use makes `AdamMCMC`, and sgMCMC in general, suitable for applications where error estimation is critical. Classifier Surrogates [P4], that predict the output of detector events from more accessible simulation results, are one example for such an application.

The method relies on an M-H correction, that for sake of limited computational resources can only be executed with stochastic estimates of the log-likelihood. This introduces additional bias into the MCMC and multiple algorithms have been proposed to control or mitigate this bias. In Reference [P3], we examine this issue and find that a simple, additive correction to the stochastic loss objective can restore the scaling of the sampled posterior. Including the correction, the sampled distribution approaches the data distribution in the same way as for log-likelihoods calculated from the full data. The only difference is a small tempering of the sampled distribution.

In HEP a discovery is claimed, when the experimental data differs from the null-hypothesis by more than 5σ. As such, the field is largely dependent on well-calibrated error estimates. If the estimated uncertainty is too low, the variance of the evaluation might be interpreted as new physics and if it is too high, possible discoveries are missed. In consideration of the significant value uncertainties hold in HEP and the widespread use of Deep Learning within the field, the low frequency at which BNNs are applied is staggering. Modern stochastic gradient MCMC methods, such as `AdamMCMC` or AMAGOLD [256], promise optimization performance on par with commonly used stochastic optimization algorithms such as Adam at similar computational cost. They also include the estimation of epistemic uncertainties at low additional cost. Moreover, they are independent of the initialization and can be applied after NN optimization to further decrease their cost or analyze uncertainties in hindsight. While they are currently widely unknown within the community, they have shown huge potential for applications with generative methods and beyond.

References

[P1] S. Bieringer, A. Butter, S. Diefenbacher, et al. "Calomplification — the power of generative calorimeter models". In: *JINST* 17.09 (2022), P09028. DOI: 10.1088/1748-0221/17/09/P09028.

[P2] S. Bieringer, G. Kasieczka, M. F. Steffen, et al. "AdamMCMC: Combining Metropolis Adjusted Langevin with Momentum-based Optimization". *Submitted to the 39th AAAI Conference on Artificial Intelligence, February 25 – March 4, 2025, Philadelphia, Pennsylvania, USA; passed first phase of rejections.* Dec. 2023. arXiv: 2312.14027 [stat.ML].

[P3] S. Bieringer, G. Kasieczka, M. F. Steffen, et al. "Statistical guarantees for stochastic Metropolis-Hastings". *Submitted to JLMR; third phase of reviews.* Oct. 2023. arXiv: 2310.09335 [stat.ML].

[P4] S. Bieringer, G. Kasieczka, J. Kieseler, et al. "Classifier surrogates: sharing AI-based searches with the world". In: *Eur. Phys. J. C* 84.9 (2024), p. 972. DOI: 10.1140/epjc/s10052-024-13353-w.

[P5] S. Bieringer, S. Diefenbacher, G. Kasieczka, et al. "Calibrating Bayesian generative machine learning for Bayesiamplification". In: *Mach. Learn. Sci. Tech.* 5.4 (2024), p. 045044. DOI: 10.1088/2632-2153/ad9136.

[1] A. Butter, S. Diefenbacher, G. Kasieczka, et al. "GANplifying event samples". In: *SciPost Phys.* 10.6 (2021), p. 139. DOI: 10.21468/SciPostPhys.10.6.139.

[2] J. Y. Araz et al. "Les Houches guide to reusable ML models in LHC analyses". Dec. 2023. arXiv: 2312.14575 [hep-ph].

[3] L. Evans and P. Bryant. "LHC Machine". In: *Journal of Instrumentation* 3.08 (Aug. 2008), S08001. DOI: 10.1088/1748-0221/3/08/S08001.

[4] S. Chatrchyan et al. "Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC". In: *Phys. Lett. B* 716 (2012), pp. 30–61. DOI: 10.1016/j.physletb.2012.08.021.

[5] G. Aad et al. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC". In: *Phys. Lett. B* 716 (2012), pp. 1–29. DOI: 10.1016/j.physletb.2012.08.020.

[6] A. Salam and J. C. Ward. "Weak and electromagnetic interactions". In: *Il Nuovo Cimento Series 10* (1959). ISSN: 00296341. DOI: 10.1007/BF02726525.

[7] S. L. Glashow. "Partial-symmetries of weak interactions". In: *Nuclear Physics* (1961). ISSN: 00295582. DOI: 10.1016/0029-5582(61)90469-2.

[8] S. Weinberg. "A Model of Leptons". In: *Phys. Rev. Lett.* 19 (21 Nov. 1967), pp. 1264–1266. DOI: 10.1103/PhysRevLett.19.1264.

[9] I. Zurbano Fernandez et al. *High-Luminosity Large Hadron Collider (HL-LHC): Technical design report.* Tech. rep. Dec. 2020. DOI: 10.23731/CYRM-2020-0010.

[10] *ATLAS Software and Computing HL-LHC Roadmap.* Tech. rep. Geneva: CERN, 2022. URL: https://cds.cern.ch/record/2802918.

[11] A. Butter and T. Plehn. "Generative Networks for LHC events". Aug. 2020. arXiv: 2008.08558 [hep-ph].

[12] L. V. Jospin, H. Laga, F. Boussaid, et al. "Hands-On Bayesian Neural Networks - A Tutorial for Deep Learning Users". In: *IEEE Comput. Intell. Mag.* 17.2 (2022), pp. 29–48. DOI: 10.1109/MCI.2022.3155327.

[13] T. Y. Chen, B. Dey, A. Ghosh, et al. "Interpretable Uncertainty Quantification in AI for HEP". In: *Snowmass 2021.* Aug. 2022. DOI: 10.2172/1886020.

[14] C. Blundell, J. Cornebise, K. Kavukcuoglu, et al. “Weight Uncertainty in Neural Network”. In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. Ed. by F. R. Bach and D. M. Blei. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 1613–1622. URL: http://proceedings.mlr.press/v37/blundell15.html.

[15] S. Bollweg, M. Haußmann, G. Kasieczka, et al. “Deep-Learning Jets with Uncertainties and More”. In: *SciPost Phys.* 8.1 (2020), p. 006. DOI: 10.21468/SciPostPhys.8.1.006.

[16] A. Butter, T. Heimel, S. Hummerich, et al. “Generative networks for precision enthusiasts”. In: *SciPost Phys.* 14.4 (2023), p. 078. DOI: 10.21468/SciPostPhys.14.4.078.

[17] A. Butter, N. Huetsch, S. Palacios Schweitzer, et al. “Jet Diffusion versus JetGPT – Modern Networks for the LHC”. May 2023. arXiv: 2305.10475 [hep-ph].

[18] V. Carvalho, M. Ferreira, T. Malik, et al. “Decoding neutron star observations: Revealing composition through Bayesian neural networks”. In: *Phys. Rev. D* 108.4 (2023), p. 043031. DOI: 10.1103/PhysRevD.108.043031.

[19] J. S. Denker and Y. LeCun. “Transforming Neural-Net Output Levels to Probability Distributions”. In: *Advances in Neural Information Processing Systems 3, [NIPS Conference, Denver, Colorado, USA, November 26-29, 1990]*. Ed. by R. Lippmann, J. E. Moody, and D. S. Touretzky. Morgan Kaufmann, 1990, pp. 853–859. URL: http://papers.nips.cc/paper/419-transforming-neural-net-output-levels-to-probability-distributions.

[20] M. Welling and Y. W. Teh. “Bayesian Learning via Stochastic Gradient Langevin Dynamics”. In: *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*. Ed. by L. Getoor and T. Scheffer. Omnipress, 2011, pp. 681–688. URL: https://icml.cc/2011/papers/398%5C_icmlpaper.pdf.

[21] T. Chen, E. B. Fox, and C. Guestrin. “Stochastic Gradient Hamiltonian Monte Carlo”. In: *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*. Vol. 32. JMLR Workshop and Conference Proceedings. JMLR.org, 2014, pp. 1683–1691. URL: http://proceedings.mlr.press/v32/cheni14.html.

[22] J. J. Thomson. “Cathode Rays”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 44.269 (1897), pp. 293–316. DOI: 10.1080/14786449708621070.

[23] E. Rutherford. “The scattering of α and β particles by matter and the structure of the atom”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 21.125 (1911), pp. 669–688. DOI: 10.1080/14786440508637080.

[24] E. Rutherford. “Collision of α particles with light atoms. IV. An anomalous effect in nitrogen”. In: *Philosophical Magazine* 90.sup1 (2010), pp. 31–37. DOI: 10.1080/14786431003659230.

[25] A. H. Compton. “A Quantum Theory of the Scattering of X-rays by Light Elements”. In: *Phys. Rev.* 21 (5 May 1923), pp. 483–502. DOI: 10.1103/PhysRev.21.483.

[26] J. Chadwick. “Possible Existence of a Neutron”. In: *Nature* 129.3252 (Feb. 1932), pp. 312–312. ISSN: 1476-4687. DOI: 10.1038/129312a0.

[27] F. Reines and C. L. Cowan. “The Neutrino”. In: *Nature* 178.4531 (Sept. 1956), pp. 446–449. ISSN: 1476-4687. DOI: 10.1038/178446a0.

[28] E. D. Bloom, D. H. Coward, H. DeStaebler, et al. “High-Energy Inelastic e^-p Scattering at 6° and 10°”. In: *Phys. Rev. Lett.* 23 (16 Oct. 1969), pp. 930–934. DOI: 10.1103/PhysRevLett.23.930.

[29] M. Breidenbach, J. I. Friedman, H. W. Kendall, et al. “Observed Behavior of Highly Inelastic Electron-Proton Scattering”. In: *Phys. Rev. Lett.* 23 (16 Oct. 1969), pp. 935–939. DOI: 10.1103/PhysRevLett.23.935.

[30] D. P. Barber, U. Becker, H. Benda, et al. "Discovery of Three-Jet Events and a Test of Quantum Chromodynamics at PETRA". In: *Phys. Rev. Lett.* 43 (12 Sept. 1979), pp. 830–833. DOI: 10.1103/PhysRevLett.43.830.

[31] J. Aubert, G. Bassompierre, K. Becks, et al. "The ratio of the nucleon structure functions F2N for iron and deuterium". In: *Physics Letters B* 123.3 (1983), pp. 275–278. ISSN: 0370-2693. DOI: 10.1016/0370-2693(83)90437-9.

[32] G. Arnison, A. Astbury, B. Aubert, et al. "Experimental observation of lepton pairs of invariant mass around 95 GeV/c2 at the CERN SPS collider". In: *Physics Letters B* 126.5 (1983), pp. 398–410. ISSN: 0370-2693. DOI: 10.1016/0370-2693(83)90188-0.

[33] F. Abe, H. Akimoto, A. Akopian, et al. "Observation of Top Quark Production in $\overline{p}p$ Collisions with the Collider Detector at Fermilab". In: *Phys. Rev. Lett.* 74 (14 Apr. 1995), pp. 2626–2631. DOI: 10.1103/PhysRevLett.74.2626.

[34] S. Abachi, B. Abbott, M. Abolins, et al. "Observation of the Top Quark". In: *Phys. Rev. Lett.* 74 (14 Apr. 1995), pp. 2632–2637. DOI: 10.1103/PhysRevLett.74.2632.

[35] M. Thomson. *Modern Particle Physics.* Cambridge University Press, 2013. DOI: 10.1017/CBO9781139525367.

[36] Wikimedia Commons. *File:Standard Model of Elementary Particles.svg — Wikimedia Commons, the free media repository.* [Online; accessed 15-October-2024]. 2024. URL: https://commons.wikimedia.org/w/index.php?title=File:Standard_Model_of_Elementary_Particles.svg&oldid=917147923.

[37] M. E. Peskin and D. V. Schroeder. *An introduction to quantum field theory.* Boulder, CO: Westview, 1995.

[38] C. S. Wu, E. Ambler, R. W. Hayward, et al. "Experimental Test of Parity Conservation in Beta Decay". In: *Phys. Rev.* 105 (4 Feb. 1957), pp. 1413–1415. DOI: 10.1103/PhysRev.105.1413.

[39] S. L. Glashow. "The renormalizability of vector meson interactions". In: *Nuclear Physics* 10 (1959), pp. 107–117. ISSN: 0029-5582. DOI: 10.1016/0029-5582(59)90196-8.

[40] D. J. Gross and F. Wilczek. "Ultraviolet Behavior of Non-Abelian Gauge Theories". In: *Phys. Rev. Lett.* 30 (26 June 1973), pp. 1343–1346. DOI: 10.1103/PhysRevLett.30.1343.

[41] H. D. Politzer. "Reliable Perturbative Results for Strong Interactions?" In: *Phys. Rev. Lett.* 30 (26 June 1973), pp. 1346–1349. DOI: 10.1103/PhysRevLett.30.1346.

[42] P. Higgs. "Broken symmetries, massless particles and gauge fields". In: *Physics Letters* 12.2 (1964), pp. 132–133. ISSN: 0031-9163. DOI: 10.1016/0031-9163(64)91136-9.

[43] P. W. Higgs. "Broken Symmetries and the Masses of Gauge Bosons". In: *Phys. Rev. Lett.* 13 (16 Oct. 1964), pp. 508–509. DOI: 10.1103/PhysRevLett.13.508.

[44] F. Englert and R. Brout. "Broken Symmetry and the Mass of Gauge Vector Mesons". In: *Phys. Rev. Lett.* 13 (9 July 1964), pp. 321–323. DOI: 10.1103/PhysRevLett.13.321.

[45] Y. Fukuda, T. Hayakawa, E. Ichihara, et al. "Evidence for Oscillation of Atmospheric Neutrinos". In: *Phys. Rev. Lett.* 81 (8 July 1998), pp. 1562–1567. DOI: 10.1103/PhysRevLett.81.1562.

[46] Q. R. Ahmad, R. C. Allen, T. C. Andersen, et al. "Direct Evidence for Neutrino Flavor Transformation from Neutral-Current Interactions in the Sudbury Neutrino Observatory". In: *Phys. Rev. Lett.* 89 (1 June 2002), p. 011301. DOI: 10.1103/PhysRevLett.89.011301.

[47] K. Abe, N. Abgrall, H. Aihara, et al. "Evidence of electron neutrino appearance in a muon neutrino beam". In: *Phys. Rev. D* 88 (3 Aug. 2013), p. 032002. DOI: 10.1103/PhysRevD.88.032002.

[48] P. Ramond. "The Family Group in Grand Unified Theories". In: *International Symposium on Fundamentals of Quantum Theory and Quantum Field Theory.* Feb. 1979. arXiv: hep-ph/9809459.

[49] O. Sawada and A. Sugamoto, eds. *Proceedings: Workshop on the Unified Theories and the Baryon Number in the Universe: Tsukuba, Japan, February 13-14, 1979.* Tsukuba, Japan: Natl.Lab.High Energy Phys., 1979, p. 95.

[50] M. Gell-Mann, P. Ramond, and R. Slansky. "Complex Spinors and Unified Theories". In: *Conf. Proc. C* 790927 (1979), pp. 315–321. arXiv: 1306.4669 [hep-th].

[51] S. L. Glashow. "The Future of Elementary Particle Physics". In: *Quarks and Leptons.* Ed. by M. Levy, J.-L. Basdevant, D. Speiser, et al. Boston, MA: Springer US, 1980, pp. 687–713. ISBN: 978-1-4684-7197-7.

[52] M. Bauer and T. Plehn. *Yet Another Introduction to Dark Matter: The Particle Physics Approach.* Vol. 959. Lecture Notes in Physics. Springer, 2019. DOI: 10.1007/978-3-030-16234-4.

[53] H. Georgi and S. L. Glashow. "Unity of All Elementary-Particle Forces". In: *Phys. Rev. Lett.* 32 (8 Feb. 1974), pp. 438–441. DOI: 10.1103/PhysRevLett.32.438.

[54] K. Krasnov and R. Percacci. "Gravity and Unification: A review". In: *Class. Quant. Grav.* 35.14 (2018), p. 143001. DOI: 10.1088/1361-6382/aac58d.

[55] H. Abramowicz and A. Caldwell. "HERA collider physics". In: *Rev. Mod. Phys.* 71 (1999), pp. 1275–1410. DOI: 10.1103/RevModPhys.71.1275.

[56] H. Wiedemann. *Particle Accelerator Physics.* Cham: Springer International Publishing, 2015. ISBN: 978-3-319-18317-6. DOI: 10.1007/978-3-319-18317-6_1.

[57] T. A. Collaboration, G. Aad, E. Abat, et al. "The ATLAS Experiment at the CERN Large Hadron Collider". In: *Journal of Instrumentation* 3.08 (Aug. 2008), S08003. DOI: 10.1088/1748-0221/3/08/S08003.

[58] T. C. Collaboration, S. Chatrchyan, G. Hmayakyan, et al. "The CMS experiment at the CERN LHC". In: *Journal of Instrumentation* 3.08 (Aug. 2008), S08004. DOI: 10.1088/1748-0221/3/08/S08004.

[59] T. L. Collaboration, A. A. A. Jr, L. M. A. Filho, et al. "The LHCb Detector at the LHC". In: *Journal of Instrumentation* 3.08 (Aug. 2008), S08005. DOI: 10.1088/1748-0221/3/08/S08005.

[60] T. A. Collaboration, K. Aamodt, A. A. Quintana, et al. "The ALICE experiment at the CERN LHC". In: *Journal of Instrumentation* 3.08 (Aug. 2008), S08002. DOI: 10.1088/1748-0221/3/08/S08002.

[61] A. Tumasyan et al. *The Phase-2 Upgrade of the CMS Tracker.* Tech. rep. June 2017. DOI: 10.17181/CERN.QZ28.FLHW.

[62] *2020 Update of the European Strategy for Particle Physics.* Geneva: CERN Council, 2020. ISBN: 978-92-9083-575-2. DOI: 10.17181/ESU2020.

[63] A. Abada et al. "FCC Physics Opportunities: Future Circular Collider Conceptual Design Report Volume 1". In: *Eur. Phys. J. C* 79.6 (2019), p. 474. DOI: 10.1140/epjc/s10052-019-6904-3.

[64] W. Abdallah et al. "CEPC Technical Design Report: Accelerator". In: *Radiat. Detect. Technol. Methods* 8.1 (2024), pp. 1–1105. DOI: 10.1007/s41605-024-00463-y.

[65] *The International Linear Collider Technical Design Report - Volume 1: Executive Summary.* Tech. rep. June 2013. arXiv: 1306.6327 [physics.acc-ph].

[66] H. Abramowicz et al. *The International Linear Collider Technical Design Report - Volume 4: Detectors.* Tech. rep. June 2013. arXiv: 1306.6329 [physics.ins-det].

[67] T. K. Charles et al. *The Compact Linear Collider (CLIC) - 2018 Summary Report.* Tech. rep. Dec. 2018. DOI: 10.23731/CYRM-2018-002.

[68] S. Navas et al. "Review of particle physics". In: *Phys. Rev. D* 110.3 (2024), p. 030001. DOI: 10.1103/PhysRevD.110.030001.

[69] D. Belayneh et al. "Calorimetry with deep learning: particle simulation and reconstruction for collider physics". In: *Eur. Phys. J. C* 80.7 (2020), p. 688. DOI: 10.1140/epjc/s10052-020-8251-9.

[70] R. Rusack, B. Joshi, A. Alpana, et al. "Electron Energy Regression in the CMS High-Granularity Calorimeter Prototype". Sept. 2023. arXiv: 2309.06582 [hep-ex].

[71] J. Campbell, J. Huston, and F. Krauss. *The Black Book of Quantum Chromodynamics*. Vol. 1. 2018, pp. 1–11. ISBN: 9780199652747. DOI: 10.1093/oso/9780199652747.001.0001.

[72] S. D. Ellis and D. E. Soper. "Successive combination jet algorithm for hadron collisions". In: *Phys. Rev. D* 48 (1993), pp. 3160–3166. DOI: 10.1103/PhysRevD.48.3160.

[73] M. Wobisch and T. Wengler. "Hadronization corrections to jet cross-sections in deep inelastic scattering". In: *Workshop on Monte Carlo Generators for HERA Physics (Plenary Starting Meeting)*. Apr. 1998, pp. 270–279. arXiv: hep-ph/9907280.

[74] M. Cacciari, G. P. Salam, and G. Soyez. "The anti-ktjet clustering algorithm". In: *Journal of High Energy Physics* 2008.04 (Apr. 2008), pp. 063–063. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2008/04/063.

[75] A. Butter et al. "The Machine Learning landscape of top taggers". In: *SciPost Phys.* 7 (2019). Ed. by G. Kasieczka and T. Plehn, p. 014. DOI: 10.21468/SciPostPhys.7.1.014.

[76] H. Qu, C. Li, and S. Qian. "Particle Transformer for Jet Tagging". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by K. Chaudhuri, S. Jegelka, L. Song, et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, July 2022, pp. 18281–18292. URL: https://proceedings.mlr.press/v162/qu22b.html.

[77] J. Alwall, R. Frederix, S. Frixione, et al. "The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations". In: *JHEP* 07 (2014), p. 079. DOI: 10.1007/JHEP07(2014)079.

[78] T. Sjostrand, S. Mrenna, and P. Z. Skands. "A Brief Introduction to PYTHIA 8.1". In: *Comput. Phys. Commun.* 178 (2008), pp. 852–867. DOI: 10.1016/j.cpc.2008.01.036.

[79] J. Bellm et al. "Herwig 7.0/Herwig++ 3.0 release note". In: *Eur. Phys. J. C* 76.4 (2016), p. 196. DOI: 10.1140/epjc/s10052-016-4018-8.

[80] E. Bothmann et al. "Event Generation with Sherpa 2.2". In: *SciPost Phys.* 7.3 (2019), p. 034. DOI: 10.21468/SciPostPhys.7.3.034.

[81] S. Agostinelli et al. "GEANT4–a simulation toolkit". In: *Nucl. Instrum. Meth. A* 506 (2003), pp. 250–303. DOI: 10.1016/S0168-9002(03)01368-8.

[82] G. Aad et al. *Software and computing for Run 3 of the ATLAS experiment at the LHC*. Tech. rep. Apr. 2024. arXiv: 2404.06335 [hep-ex].

[83] J. de Favereau, C. Delaere, P. Demin, et al. "DELPHES 3, A modular framework for fast simulation of a generic collider experiment". In: *JHEP* 02 (2014), p. 057. DOI: 10.1007/JHEP02(2014)057.

[84] G. Aad et al. "AtlFast3: The Next Generation of Fast Simulation in ATLAS". In: *Comput. Softw. Big Sci.* 6.1 (2022), p. 7. DOI: 10.1007/s41781-021-00079-7.

[85] E. Barberio et al. "Fast simulation of electromagnetic showers in the ATLAS calorimeter: Frozen showers". In: *J. Phys. Conf. Ser.* 160 (2009). Ed. by M. Fraternali, G. Gaudio, and M. Livan, p. 012082. DOI: 10.1088/1742-6596/160/1/012082.

[86] M. Paganini, L. de Oliveira, and B. Nachman. "Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters". In: *Phys. Rev. Lett.* 120.4 (2018), p. 042003. DOI: 10.1103/PhysRevLett.120.042003.

[87] M. Paganini, L. de Oliveira, and B. Nachman. "CaloGAN : Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks". In: *Phys. Rev. D* 97.1 (2018), p. 014021. DOI: 10.1103/PhysRevD.97.014021.

[88] L. de Oliveira, M. Paganini, and B. Nachman. "Controlling Physical Attributes in GAN-Accelerated Simulation of Electromagnetic Calorimeters". In: *J. Phys. Conf. Ser.* 1085.4 (2018), p. 042017. DOI: 10.1088/1742-6596/1085/4/042017.

[89] HEP ML Community. *A Living Review of Machine Learning for Particle Physics*. URL: https://iml-wg.github.io/HEPML-LivingReview/.

[90] B. Rozemberczki, L. Watson, P. Bayer, et al. "The Shapley Value in Machine Learning". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*. Ed. by L. D. Raedt. ijcai.org, 2022, pp. 5572–5579. DOI: 10.24963/IJCAI.2022/778.

[91] OpenAI. "GPT-4 Technical Report". 2023. arXiv: 2303.08774.

[92] F. Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65 6 (1958), pp. 386–408. URL: https://api.semanticscholar.org/CorpusID:12781225.

[93] L. Bottou. "Stochastic Gradient Descent Tricks". In: *Neural Networks: Tricks of the Trade - Second Edition*. Ed. by G. Montavon, G. B. Orr, and K. Müller. Vol. 7700. Lecture Notes in Computer Science. Springer, 2012, pp. 421–436. DOI: 10.1007/978-3-642-35289-8_25.

[94] T. Tieleman and G. Hinton. "RMSProp: Divide the gradient by a running average of its recent magnitude". In: *Neural Networks for Machine Learning*. COURSERA, 2012.

[95] J. C. Duchi, E. Hazan, and Y. Singer. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". In: *J. Mach. Learn. Res.* 12 (2011), pp. 2121–2159. DOI: 10.5555/1953048.2021068.

[96] M. D. Zeiler. "ADADELTA: An Adaptive Learning Rate Method". Dec. 2012. arXiv: 1212.5701.

[97] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2015. arXiv: 1412.6980.

[98] R. Pascanu and Y. Bengio. "Revisiting Natural Gradient for Deep Networks". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2014. arXiv: 1301.3584.

[99] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, et al. "Generative adversarial networks". In: *Commun. ACM* 63.11 (2020), pp. 139–144. DOI: 10.1145/3422622.

[100] M. Schervish. "Theory of Statistics". In: Springer Series in Statistics. Springer New York, 2012, pp. 301–306. ISBN: 9781461242505. DOI: 10.1007/978-1-4612-4250-5.

[101] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[102] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2014. arXiv: 1312.6114.

[103] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, et al. "Autoencoding beyond pixels using a learned similarity metric". In: *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*. Ed. by M. Balcan and K. Q. Weinberger. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 1558–1566. URL: http://proceedings.mlr.press/v48/larsen16.html.

[104] A. Makhzani, J. Shlens, N. Jaitly, et al. "Adversarial Autoencoders". May 2015. arXiv: 1511.05644.

[105] S. Otten, S. Caron, W. de Swart, et al. "Event Generation and Statistical Sampling for Physics with Deep Generative Models and a Density Information Buffer". In: *Nature Commun.* 12.1 (2021), p. 2985. DOI: 10.1038/s41467-021-22616-z.

[106] E. Buhmann, S. Diefenbacher, E. Eren, et al. "Decoding Photons: Physics in the Latent Space of a BIB-AE Generative Network". In: *EPJ Web Conf.* 251 (2021), p. 03003. DOI: 10.1051/epjconf/202125103003.

[107] N. Tishby, F. C. N. Pereira, and W. Bialek. "The information bottleneck method". Apr. 2000. eprint: physics/0004057.

[108] N. Tishby and N. Zaslavsky. "Deep learning and the information bottleneck principle". In: *2015 IEEE Information Theory Workshop, ITW 2015, Jerusalem, Israel, April 26 - May 1, 2015.* IEEE, 2015, pp. 1–5. DOI: 10.1109/ITW.2015.7133169.

[109] S. Voloshynovskiy, M. Kondah, S. Rezaeifar, et al. "Information bottleneck through variational glasses". Dec. 2019. arXiv: 1912.00830.

[110] E. Buhmann, S. Diefenbacher, E. Eren, et al. "Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed". In: *Comput. Softw. Big Sci.* 5.1 (2021), p. 13. DOI: 10.1007/s41781-021-00056-0.

[111] E. Buhmann, S. Diefenbacher, D. Hundhausen, et al. "Hadrons, better, faster, stronger". In: *Mach. Learn. Sci. Tech.* 3.2 (2022), p. 025014. DOI: 10.1088/2632-2153/ac7848.

[112] I. Kobyzev, S. J. D. Prince, and M. A. Brubaker. "Normalizing Flows: An Introduction and Review of Current Methods". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 43.11 (2021), pp. 3964–3979. DOI: 10.1109/TPAMI.2020.2992934.

[113] G. Papamakarios, E. T. Nalisnick, D. J. Rezende, et al. "Normalizing Flows for Probabilistic Modeling and Inference". In: *J. Mach. Learn. Res.* 22 (2021), 57:1–57:64. URL: https://jmlr.org/papers/v22/19-1028.html.

[114] L. Younes. *Shapes and Diffeomorphisms.* Applied Mathematical Sciences. Springer Berlin Heidelberg, 2010. ISBN: 9783642120558. DOI: 10.1007/978-3-662-58496-5.

[115] E. Dupont, A. Doucet, and Y. W. Teh. "Augmented Neural ODEs". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada.* Ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, et al. 2019, pp. 3134–3144. URL: https://proceedings.neurips.cc/paper/2019/hash/21be9a4bd4f81549a9d1d241981cec3c-Abstract.html.

[116] R. Winterhalder, M. Bellagente, and B. Nachman. "Latent Space Refinement for Deep Generative Models". June 2021. arXiv: 2106.00792 [stat.ML].

[117] L. Dinh, J. Sohl-Dickstein, R. Pascanu, et al. "A RAD approach to deep mixture models". In: *Deep Generative Models for Highly Structured Data, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019.* OpenReview.net, 2019. URL: https://openreview.net/forum?id=HJeZNLIt%5C_4.

[118] C. Winkler, D. E. Worrall, E. Hoogeboom, et al. "Learning Likelihoods with Conditional Normalizing Flows". 2019. arXiv: 1912.00042.

[119] S. T. Radev, U. K. Mertens, A. Voss, et al. "BayesFlow: Learning complex stochastic models with invertible neural networks". Mar. 2020. arXiv: 2003.06281.

[120] S. T. Radev, M. Schmitt, L. Schumacher, et al. "BayesFlow: Amortized Bayesian Workflows With Neural Networks". In: *J. Open Source Softw.* 8.90 (2023), p. 5702. DOI: 10.21105/JOSS.05702.

[121] L. Dinh, D. Krueger, and Y. Bengio. "NICE: Non-linear Independent Components Estimation". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings.* Ed. by Y. Bengio and Y. LeCun. 2015. arXiv: 1410.8516.

[122] M. Germain, K. Gregor, I. Murray, et al. "MADE: Masked Autoencoder for Distribution Estimation". In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015.* Ed. by F. R. Bach and D. M. Blei. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 881–889. URL: http://proceedings.mlr.press/v37/germain15.html.

[123] D. P. Kingma, T. Salimans, and M. Welling. "Improving Variational Inference with Inverse Autoregressive Flow". June 2016. arXiv: 1606.04934.

[124] G. Papamakarios, I. Murray, and T. Pavlakou. "Masked Autoregressive Flow for Density Estimation". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by I. Guyon, U. von Luxburg, S. Bengio, et al. 2017, pp. 2338–2347. URL: https://proceedings.neurips.cc/paper/2017/hash/6c1da886822c67822bcf3679d04369fa-Abstract.html.

[125] L. Dinh, J. Sohl-Dickstein, and S. Bengio. "Density estimation using Real NVP". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=HkpbnH9lx.

[126] D. P. Kingma and P. Dhariwal. "Glow: Generative Flow with Invertible 1x1 Convolutions". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montreal, Canada*. Ed. by S. Bengio, H. M. Wallach, H. Larochelle, et al. 2018, pp. 10236–10245. URL: https://proceedings.neurips.cc/paper/2018/hash/d139db6a236200b21cc7f752979132d0-Abstract.html.

[127] C. Durkan, A. Bekasov, I. Murray, et al. "Neural Spline Flows". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, et al. 2019, pp. 7509–7520. URL: https://proceedings.neurips.cc/paper/2019/hash/7ac71d433f282034e088473244df8c02-Abstract.html.

[128] A. N. Gomez, M. Ren, R. Urtasun, et al. "The Reversible Residual Network: Backpropagation Without Storing Activations". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by I. Guyon, U. von Luxburg, S. Bengio, et al. 2017, pp. 2214–2224. URL: https://proceedings.neurips.cc/paper/2017/hash/f9be311e65d81a9ad8150a60844bb94c-Abstract.html.

[129] J. Jacobsen, A. W. M. Smeulders, and E. Oyallon. "i-RevNet: Deep Invertible Networks". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=HJsjkMb0Z.

[130] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, et al. "Neural Ordinary Differential Equations". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montreal, Canada*. Ed. by S. Bengio, H. M. Wallach, H. Larochelle, et al. 2018, pp. 6572–6583. URL: https://proceedings.neurips.cc/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html.

[131] W. Grathwohl, R. T. Q. Chen, J. Bettencourt, et al. "FFJORD: Free-Form Continuous Dynamics for Scalable Reversible Generative Models". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=rJxgknCcK7.

[132] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, et al. "Flow Matching for Generative Modeling". In: *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL: https://openreview.net/forum?id=PqvMRDCJT9t.

[133] X. Liu, C. Gong, and Q. Liu. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow". In: *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL: https://openreview.net/forum?id=XVjTT1nw5z.

[134] M. S. Albergo and E. Vanden-Eijnden. “Building Normalizing Flows with Stochastic Interpolants”. In: *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL: https://openreview.net/forum?id=li7qeBbCR1t.

[135] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, et al. “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”. In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. Ed. by F. R. Bach and D. M. Blei. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 2256–2265. URL: http://proceedings.mlr.press/v37/sohl-dickstein15.html.

[136] J. Ho, A. Jain, and P. Abbeel. “Denoising Diffusion Probabilistic Models”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, et al. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

[137] Y. Song, J. Sohl-Dickstein, D. P. Kingma, et al. “Score-Based Generative Modeling through Stochastic Differential Equations”. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=PxTIG12RRHS.

[138] J. Betker, G. Goh, L. Jing, et al. “Improving Image Generation with Better Captions”. In: URL: https://api.semanticscholar.org/CorpusID:264403242.

[139] R. Rombach, A. Blattmann, D. Lorenz, et al. “High-Resolution Image Synthesis with Latent Diffusion Models”. 2021. arXiv: 2112.10752 [cs.CV].

[140] A. Hyvärinen. “Estimation of Non-Normalized Statistical Models by Score Matching”. In: *J. Mach. Learn. Res.* 6 (2005), pp. 695–709. URL: https://jmlr.org/papers/v6/hyvarinen05a.html.

[141] P. Vincent. “A Connection Between Score Matching and Denoising Autoencoders”. In: *Neural Comput.* 23.7 (2011), pp. 1661–1674. DOI: 10.1162/NECO_a_00142.

[142] Y. Song and S. Ermon. “Generative Modeling by Estimating Gradients of the Data Distribution”. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, et al. 2019, pp. 11895–11907. URL: https://proceedings.neurips.cc/paper/2019/hash/3001ef257407d5a371a96dcd947c7d93-Abstract.html.

[143] T. Karras, M. Aittala, T. Aila, et al. “Elucidating the Design Space of Diffusion-Based Generative Models”. In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, et al. 2022. URL: http://papers.nips.cc/paper%5C_files/paper/2022/hash/a98846e9d9cc01cfb87eb694d946ce6b-Abstract-Conference.html.

[144] A. Vaswani, N. Shazeer, N. Parmar, et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by I. Guyon, U. von Luxburg, S. Bengio, et al. 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[145] T. Finke, M. Krämer, A. Mück, et al. “Learning the language of QCD jets with transformers”. In: *JHEP* 06 (2023), p. 184. DOI: 10.1007/JHEP06(2023)184.

[146] E. Buhmann, S. Diefenbacher, E. Eren, et al. “CaloClouds: fast geometry-independent highly-granular calorimeter simulation”. In: *JINST* 18.11 (2023), P11025. DOI: 10.1088/1748-0221/18/11/P11025.

[147] E. Buhmann, F. Gaede, G. Kasieczka, et al. “CaloClouds II: ultra-fast geometry-independent highly-granular calorimeter simulation”. In: *JINST* 19.04 (2024), P04020. DOI: 10.1088/1748-0221/19/04/P04020.

[148] E. Buhmann, C. Ewen, D. A. Faroughy, et al. “EPiC-ly Fast Particle Cloud Generation with Flow-Matching and Diffusion”. Sept. 2023. arXiv: 2310.00049 [hep-ph].

[149] Y. Hao, A. Orlitsky, A. T. Suresh, et al. “Data Amplification: A Unified and Competitive Approach to Property Estimation”. In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montreal, Canada.* Ed. by S. Bengio, H. M. Wallach, H. Larochelle, et al. 2018, pp. 8848–8857. URL: https://proceedings.neurips.cc/paper/2018/hash/a753a43564c29148df3150afb4475440-Abstract.html.

[150] K. T. Matchev, A. Roman, and P. Shyamsundar. “Uncertainties associated with GAN-generated datasets in high energy physics”. In: *SciPost Phys.* 12.3 (2022), p. 104. DOI: 10.21468/SciPostPhys.12.3.104.

[151] T. M. Mitchell. *The need for biases in learning generalizations.* Tech. rep. Rutgers University, 1980.

[152] A. Butter, T. Plehn, and R. Winterhalder. “How to GAN LHC Events”. In: *SciPost Phys.* 7.6 (2019), p. 075. DOI: 10.21468/SciPostPhys.7.6.075.

[153] M. Zaheer, S. Kottur, S. Ravanbakhsh, et al. “Deep Sets”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA.* Ed. by I. Guyon, U. von Luxburg, S. Bengio, et al. 2017, pp. 3391–3401. URL: https://proceedings.neurips.cc/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html.

[154] P. T. Komiske, E. M. Metodiev, and J. Thaler. “Energy Flow Networks: Deep Sets for Particle Jets”. In: *JHEP* 01 (2019), p. 121. DOI: 10.1007/JHEP01(2019)121.

[155] J. Allison, K. Amako, J. Apostolakis, et al. “Recent developments in Geant4”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 835 (2016), pp. 186–225. ISSN: 0168-9002. DOI: 10.1016/j.nima.2016.06.125.

[156] H. Abramowicz et al. *International Large Detector: Interim Design Report.* Tech. rep. Mar. 2020. arXiv: 2003.01116 [physics.ins-det].

[157] L. de Oliveira, M. Paganini, and B. Nachman. “Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis”. In: *Comput. Softw. Big Sci.* 1.1 (2017), p. 4. DOI: 10.1007/s41781-017-0004-6.

[158] T. Miyato, T. Kataoka, M. Koyama, et al. “Spectral Normalization for Generative Adversarial Networks”. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.* OpenReview.net, 2018. URL: https://openreview.net/forum?id=B1QRgziT-.

[159] S. C. Hora. “Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management”. In: *Reliability Engineering & System Safety* 54.2 (1996). Treatment of Aleatory and Epistemic Uncertainty, pp. 217–223. ISSN: 0951-8320. DOI: 10.1016/S0951-8320(96)00077-4.

[160] Y. Gal. “Uncertainty in Deep Learning”. PhD thesis. University of Cambridge, 2016.

[161] A. Kendall and Y. Gal. “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?” In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA.* Ed. by I. Guyon, U. von Luxburg, S. Bengio, et al. 2017, pp. 5574–5584. URL: https://proceedings.neurips.cc/paper/2017/hash/2650d6089a6d640c5e85b2b88265dc2b-Abstract.html.

[162] E. Hüllermeier and W. Waegeman. “Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods”. In: *Mach. Learn.* 110.3 (2021), pp. 457–506. DOI: 10.1007/S10994-021-05946-3.

[163] L. K. Hansen and P. Salamon. “Neural Network Ensembles”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.10 (1990), pp. 993–1001. DOI: 10.1109/34.58871.

[164] A. Krogh and J. Vedelsby. “Neural Network Ensembles, Cross Validation and Active Learning”. In: *Proceedings of the 7th International Conference on Neural Information Processing Systems*. MIT Press, 1994, pp. 231–238. DOI: 10.5555/2998687.2998716.

[165] T. G. Dietterich. “Ensemble Methods in Machine Learning”. In: *Multiple Classifier Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 1–15. ISBN: 978-3-540-45014-6.

[166] B. Lakshminarayanan, A. Pritzel, and C. Blundell. “Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by I. Guyon, U. von Luxburg, S. Bengio, et al. 2017, pp. 6402–6413. URL: https://proceedings.neurips.cc/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html.

[167] S. Zaidi, A. Zela, T. Elsken, et al. “Neural Ensemble Search for Uncertainty Estimation and Dataset Shift”. In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. Ed. by M. Ranzato, A. Beygelzimer, Y. N. Dauphin, et al. 2021, pp. 7898–7911. URL: https://proceedings.neurips.cc/paper/2021/hash/41a6fd31aa2e75c3c6d427db3d17ea80-Abstract.html.

[168] S. Amari. “Natural Gradient Works Efficiently in Learning”. In: *Neural Comput.* 10.2 (1998), pp. 251–276. DOI: 10.1162/089976698300017746.

[169] H. Cramér. *Mathematical Methods of Statistics*. Princeton: Princeton University Press, 1946. ISBN: 9781400883868. DOI: 10.1515/9781400883868.

[170] C. R. Rao. “Information and the Accuracy Attainable in the Estimation of Statistical Parameters”. In: *Breakthroughs in Statistics: Foundations and Basic Theory*. Ed. by S. Kotz and N. L. Johnson. New York, NY: Springer New York, 1992, pp. 235–247. ISBN: 978-1-4612-0919-5. DOI: 10.1007/978-1-4612-0919-5_16.

[171] A. W. v. d. Vaart. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998. ISBN: 9780511802256. DOI: 10.1017/CBO9780511802256.

[172] E. Goan, D. Perrin, K. L. Mengersen, et al. “Piecewise Deterministic Markov Processes for Bayesian Neural Networks”. In: *Uncertainty in Artificial Intelligence, UAI 2023, July 31 - 4 August 2023, Pittsburgh, PA, USA*. Ed. by R. J. Evans and I. Shpitser. Vol. 216. Proceedings of Machine Learning Research. PMLR, 2023, pp. 712–722. URL: https://proceedings.mlr.press/v216/goan23a.html.

[173] D. Barber and C. M. Bishop. “Ensemble learning in Bayesian neural networks”. In: *Neural Networks and Machine Learning* (1998), pp. 215–237.

[174] R. M. Neal. “Bayesian learning for neural networks”. PhD thesis. University of Toronto, Canada, 1995. URL: https://librarysearch.library.utoronto.ca/permalink/01UTORONTO%5C_INST/14bjeso/alma991106438365706196.

[175] P. Izmailov, S. Vikram, M. D. Hoffman, et al. “What Are Bayesian Neural Network Posteriors Really Like?” In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 4629–4640. URL: http://proceedings.mlr.press/v139/izmailov21a.html.

[176] Y. Gal and Z. Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, June 2016, pp. 1050–1059. URL: https://proceedings.mlr.press/v48/gal16.html.

[177] M. W. Seeger. “Gaussian Processes For Machine Learning”. In: *Int. J. Neural Syst.* 14.2 (2004), pp. 69–106. DOI: 10.1142/S0129065704001899.

[178] E. A. Daxberger, A. Kristiadi, A. Immer, et al. "Laplace Redux - Effortless Bayesian Deep Learning". In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. Ed. by M. Ranzato, A. Beygelzimer, Y. N. Dauphin, et al. 2021, pp. 20089–20103. URL: https://proceedings.neurips.cc/paper/2021/hash/a7c9585703d275249f30a088cebba0ad-Abstract.html.

[179] W. J. Maddox, P. Izmailov, T. Garipov, et al. "A Simple Baseline for Bayesian Uncertainty in Deep Learning". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, et al. 2019, pp. 13132–13143. URL: https://proceedings.neurips.cc/paper/2019/hash/118921efba23fc329e6560b27861f0c2-Abstract.html.

[180] A. Amini, W. Schwarting, A. Soleimany, et al. "Deep Evidential Regression". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, et al. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/aab085461de182608ee9f607f3f7d18f-Abstract.html.

[181] J. Mena, O. Pujol, and J. Vitrià. "A Survey on Uncertainty Estimation in Deep Learning Classification Systems from a Bayesian Perspective". In: *ACM Comput. Surv.* 54.9 (2022), 193:1–193:35. DOI: 10.1145/3477140.

[182] E. Goan and C. Fookes. "Bayesian Neural Networks: An Introduction and Survey". June 2020. arXiv: 2006.12024.

[183] J. Gawlikowski, C. R. N. Tassi, M. Ali, et al. "A survey of uncertainty in deep neural networks". In: *Artif. Intell. Rev.* 56.S1 (2023), pp. 1513–1589. DOI: 10.1007/S10462-023-10562-9.

[184] J. Snoek, Y. Ovadia, E. Fertig, et al. "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, et al. 2019, pp. 13969–13980. URL: https://proceedings.neurips.cc/paper/2019/hash/8558cb408c1d76621371888657d2eb1d-Abstract.html.

[185] F. K. Gustafsson, M. Danelljan, and T. B. Schön. "Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*. Computer Vision Foundation / IEEE, 2020, pp. 1289–1298. DOI: 10.1109/CVPRW50498.2020.00167.

[186] N. Band, T. G. J. Rudner, Q. Feng, et al. "Benchmarking Bayesian Deep Learning on Diabetic Retinopathy Detection Tasks". In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1*. 2021. URL: https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/ac1dd209cbcc5e5d1c6e28598e8cbbe8-Abstract-round2.html.

[187] P. Izmailov, W. J. Maddox, P. Kirichenko, et al. "Subspace Inference for Bayesian Deep Learning". In: *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*. Ed. by A. Globerson and R. Silva. Vol. 115. Proceedings of Machine Learning Research. AUAI Press, 2019, pp. 1169–1179. URL: http://proceedings.mlr.press/v115/izmailov20a.html.

[188] E. A. Daxberger, E. T. Nalisnick, J. U. Allingham, et al. "Bayesian Deep Learning via Subnetwork Inference". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 2510–2521. URL: http://proceedings.mlr.press/v139/daxberger21a.html.

[189] J. Snoek, O. Rippel, K. Swersky, et al. “Scalable Bayesian Optimization Using Deep Neural Networks”. In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. Ed. by F. R. Bach and D. M. Blei. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 2171–2180. URL: http://proceedings.mlr.press/v37/snoek15.html.

[190] A. Kristiadi, M. Hein, and P. Hennig. “Being Bayesian, Even Just a Bit, Fixes Overconfidence in ReLU Networks”. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 5436–5446. URL: http://proceedings.mlr.press/v119/kristiadi20a.html.

[191] A. Graves. “Practical Variational Inference for Neural Networks”. In: *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*. Ed. by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, et al. 2011, pp. 2348–2356. URL: https://proceedings.neurips.cc/paper/2011/hash/7eb3c8be3d411e8ebfab08eba5f49632-Abstract.html.

[192] G. E. Hinton and D. van Camp. “Keeping the Neural Networks Simple by Minimizing the Description Length of the Weights”. In: *Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory, COLT 1993, Santa Cruz, CA, USA, July 26-28, 1993*. Ed. by L. Pitt. ACM, 1993, pp. 5–13. DOI: 10.1145/168304.168306.

[193] S. Farquhar, M. A. Osborne, and Y. Gal. “Radial Bayesian Neural Networks: Beyond Discrete Support In Large-Scale Bayesian Deep Learning”. In: *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*. Ed. by S. Chiappa and R. Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1352–1362. URL: http://proceedings.mlr.press/v108/farquhar20a.html.

[194] Y. Wen, P. Vicol, J. Ba, et al. “Flipout: Efficient Pseudo-Independent Weight Perturbations on Mini-Batches”. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=rJNpifWAb.

[195] D. P. Kingma, T. Salimans, and M. Welling. “Variational Dropout and the Local Reparameterization Trick”. In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, et al. 2015, pp. 2575–2583. URL: https://proceedings.neurips.cc/paper/2015/hash/bc7316929fe1545bf0b98d114ee3ecb8-Abstract.html.

[196] R. Krishnan, M. Subedar, and O. Tickoo. “Specifying weight priors in bayesian deep neural networks with empirical bayes”. In: *Proceedings of the AAAI conference on artificial intelligence*. AAAI Press, 2020, pp. 4477–4484.

[197] T. G. Rudner, Z. Chen, Y. W. Teh, et al. “Tractable Function-Space Variational Inference in Bayesian Neural Networks”. Dec. 2023. arXiv: 2312.17199.

[198] C. Bishop. “Exact Calculation of the Hessian Matrix for the Multilayer Perceptron”. In: *Neural Comput.* 4.4 (1992), pp. 494–501. DOI: 10.1162/NECO.1992.4.4.494.

[199] D. J. C. MacKay. “A Practical Bayesian Framework for Backpropagation Networks”. In: *Neural Computation* 4.3 (1992), pp. 448–472. DOI: 10.1162/neco.1992.4.3.448.

[200] B. A. Pearlmutter. “Fast Exact Multiplication by the Hessian”. In: *Neural Comput.* 6.1 (1994), pp. 147–160. DOI: 10.1162/NECO.1994.6.1.147.

[201] H. Ritter, A. Botev, and D. Barber. “A Scalable Laplace Approximation for Neural Networks”. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=Skdvd2xAZ.

[202] H. Ritter, A. Botev, and D. Barber. "Online Structured Laplace Approximations for Overcoming Catastrophic Forgetting". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montreal, Canada*. Ed. by S. Bengio, H. M. Wallach, H. Larochelle, et al. 2018, pp. 3742–3752. URL: https://proceedings.neurips.cc/paper/2018/hash/f31b20466ae89669f9741e047487eb37-Abstract.html.

[203] S. B. Gelfand and S. K. Mitter. "Recursive Stochastic Algorithms for Global Optimization in $\mathbb{R}^d$". In: *SIAM Journal on Control and Optimization* 29.5 (1991), pp. 999–1018. DOI: 10.1137/0329055.

[204] H. Qu, C. Li, and S. Qian. *JetClass: A Large-Scale Dataset for Deep Learning in Jet Physics*. Version 1.0.0. Zenodo, June 2022. DOI: 10.5281/zenodo.6619768.

[205] Y. Ma, T. Chen, and E. B. Fox. "A Complete Recipe for Stochastic Gradient MCMC". In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, et al. 2015, pp. 2917–2925. URL: https://proceedings.neurips.cc/paper/2015/hash/9a4400501febb2a95e79248486a5f6d3-Abstract.html.

[206] S. Ahn, A. K. Balan, and M. Welling. "Bayesian Posterior Sampling via Stochastic Gradient Fisher Scoring". In: *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012. URL: http://icml.cc/2012/papers/782.pdf.

[207] V. S. Borkar and S. K. Mitter. "A strong approximation theorem for stochastic recursive algorithms". English. In: *Journal of optimization theory and applications* (Mar. 1999). ISSN: 0022-3239. DOI: 10.1023/A:1022630321574.

[208] T.-S. Chiang, C.-R. Hwang, and S. J. Sheu. "Diffusion for Global Optimization in $\mathbb{R}^n$". In: *SIAM Journal on Control and Optimization* 25.3 (1987), pp. 737–753. DOI: 10.1137/0325042.

[209] T. Nagapetyan, A. B. Duncan, L. Hasenclever, et al. "The true cost of stochastic gradient Langevin dynamics". June 2017. arXiv: 1706.02692.

[210] M. Raginsky, A. Rakhlin, and M. Telgarsky. "Non-convex learning via Stochastic Gradient Langevin Dynamics: a nonasymptotic analysis". In: *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*. Ed. by S. Kale and O. Shamir. Vol. 65. Proceedings of Machine Learning Research. PMLR, 2017, pp. 1674–1703. URL: http://proceedings.mlr.press/v65/raginsky17a.html.

[211] R. Bardenet, A. Doucet, and C. C. Holmes. "Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach". In: *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*. Vol. 32. JMLR Workshop and Conference Proceedings. JMLR.org, 2014, pp. 405–413. URL: http://proceedings.mlr.press/v32/bardenet14.html.

[212] R. Bardenet, A. Doucet, and C. C. Holmes. "On Markov chain Monte Carlo methods for tall data". In: *J. Mach. Learn. Res.* 18 (2017), 47:1–47:43. URL: http://jmlr.org/papers/v18/15-205.html.

[213] J. T. Springenberg, A. Klein, S. Falkner, et al. "Bayesian Optimization with Robust Bayesian Neural Networks". In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. Ed. by D. D. Lee, M. Sugiyama, U. von Luxburg, et al. 2016, pp. 4134–4142. URL: https://proceedings.neurips.cc/paper/2016/hash/a96d3afec184766bfeca7a9f989fc7e7-Abstract.html.

[214] U. Simsekli, R. Badeau, A. T. Cemgil, et al. "Stochastic Quasi-Newton Langevin Monte Carlo". In: *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*. Ed. by M. Balcan and K. Q. Weinberger. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 642–651. URL: http://proceedings.mlr.press/v48/simsekli16.html.

[215] S. Patterson and Y. W. Teh. “Stochastic Gradient Riemannian Langevin Dynamics on the Probability Simplex”. In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. Ed. by C. J. C. Burges, L. Bottou, Z. Ghahramani, et al. 2013, pp. 3102–3110. URL: https://proceedings.neurips.cc/paper/2013/hash/309928d4b100a5d75adff48a9bfc1ddb-Abstract.html.

[216] C. Li, C. Chen, D. E. Carlson, et al. “Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks”. In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*. Ed. by D. Schuurmans and M. P. Wellman. AAAI Press, 2016, pp. 1788–1794. DOI: 10.1609/AAAI.V30I1.10200.

[217] X. Lu, V. Perrone, L. Hasenclever, et al. “Relativistic Monte Carlo”. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*. Ed. by A. Singh and X. (Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, 2017, pp. 1236–1245. URL: http://proceedings.mlr.press/v54/lu17b.html.

[218] J. Heek and N. Kalchbrenner. “Bayesian Inference for Large Scale Image Classification”. 2019. arXiv: 1908.03491.

[219] Y. Zhang, X. Wang, C. Chen, et al. “Towards Unifying Hamiltonian Monte Carlo and Slice Sampling”. In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. Ed. by D. D. Lee, M. Sugiyama, U. von Luxburg, et al. 2016, pp. 1741–1749. URL: https://proceedings.neurips.cc/paper/2016/hash/3cef96dcc9b8035d23f69e30bb19218a-Abstract.html.

[220] Y. Zhang, C. Chen, Z. Gan, et al. “Stochastic Gradient Monomial Gamma Sampler”. In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by D. Precup and Y. W. Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 3996–4005. URL: http://proceedings.mlr.press/v70/zhang17a.html.

[221] N. Ding, Y. Fang, R. Babbush, et al. “Bayesian Sampling Using Stochastic Gradient Thermostats”. In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, et al. 2014, pp. 3203–3211. URL: https://proceedings.neurips.cc/paper/2014/hash/21fe5b8ba755eeaece7a450849876228-Abstract.html.

[222] O. Mangoubi and N. K. Vishnoi. “Convex Optimization with Unbounded Nonconvex Oracles using Simulated Annealing”. In: *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018*. Ed. by S. Bubeck, V. Perchet, and P. Rigollet. Vol. 75. Proceedings of Machine Learning Research. PMLR, 2018, pp. 1086–1124. URL: http://proceedings.mlr.press/v75/mangoubi18a.html.

[223] C. Chen, D. E. Carlson, Z. Gan, et al. “Bridging the Gap between Stochastic Gradient MCMC and Stochastic Optimization”. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*. Ed. by A. Gretton and C. C. Robert. Vol. 51. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 1051–1060. URL: http://proceedings.mlr.press/v51/chen16c.html.

[224] R. Zhang, C. Li, J. Zhang, et al. “Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning”. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=rkeS1RVtPS.

[225] W. Deng, G. Lin, and F. Liang. “A Contour Stochastic Gradient Langevin Dynamics Algorithm for Simulations of Multi-modal Distributions”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, et al. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/b5b8c484824d8a06f4f3d570bc420313-Abstract.html.

[226] W. Deng, S. Liang, B. Hao, et al. “Interacting Contour Stochastic Gradient Langevin Dynamics”. In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL: https://openreview.net/forum?id=IK9ap6nxXr2.

[227] S. Ahn, B. Shahbaba, and M. Welling. “Distributed Stochastic Gradient MCMC”. In: *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*. Vol. 32. JMLR Workshop and Conference Proceedings. JMLR.org, 2014, pp. 1044–1052. URL: http://proceedings.mlr.press/v32/ahn14.html.

[228] F. Futami, I. Sato, and M. Sugiyama. “Accelerating the diffusion-based ensemble sampling by non-reversible dynamics”. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 3337–3347. URL: http://proceedings.mlr.press/v119/futami20a.html.

[229] S. Syed, A. Bouchard-Côté, G. Deligiannidis, et al. “Non-Reversible Parallel Tempering: A Scalable Highly Parallel MCMC Scheme”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 84.2 (Dec. 2021), pp. 321–350. ISSN: 1369-7412. DOI: 10.1111/rssb.12464.

[230] W. Deng, Q. Feng, L. Gao, et al. “Non-convex Learning via Replica Exchange Stochastic Gradient MCMC”. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 2474–2483. URL: http://proceedings.mlr.press/v119/deng20b.html.

[231] W. Deng, Q. Feng, G. Karagiannis, et al. “Accelerating Convergence of Replica Exchange Stochastic Gradient MCMC via Variance Reduction”. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=iOnhIy-a-0n.

[232] W. Deng, Q. Zhang, Q. Feng, et al. “Non-reversible Parallel Tempering for Deep Posterior Approximation”. In: *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*. Ed. by B. Williams, Y. Chen, and J. Neville. AAAI Press, 2023, pp. 7332–7339. DOI: 10.1609/AAAI.V37I6.25893.

[233] R. M. Neal. “Handbook of Markov Chain Monte Carlo”. In: CRC Press, 2011. Chap. MCMC Using Hamiltonian Dynamics.

[234] C. Chen, N. Ding, and L. Carin. “On the Convergence of Stochastic Gradient MCMC Algorithms with High-Order Integrators”. In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, et al. 2015, pp. 2278–2286. URL: https://proceedings.neurips.cc/paper/2015/hash/af4732711661056eadbf798ba191272a-Abstract.html.

[235] X. Li, Y. Wu, and L. Mackey. “Stochastic Runge-Kutta Accelerates Langevin Monte Carlo and Beyond”. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, et al. 2019, pp. 7746–7758. URL: https://proceedings.neurips.cc/paper/2019/hash/7d265aa7147bd3913fb84c7963a209d1-Abstract.html.

[236] A. D. Cobb and B. Jalaian. “Scaling Hamiltonian Monte Carlo inference for Bayesian neural networks with symmetric splitting”. In: *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021*. Ed. by C. P. de Campos, M. H. Maathuis, and E. Quaeghebeur. Vol. 161. Proceedings of Machine Learning Research. AUAI Press, 2021, pp. 675–685. URL: https://proceedings.mlr.press/v161/cobb21a.html.

[237] P. Fearnhead, J. Bierkens, M. Pollock, et al. “Piecewise deterministic Markov processes for continuous-time Monte Carlo”. In: *Statistical Science* 33.3 (2018), pp. 386–412.

[238] J. Bierkens, P. Fearnhead, and G. Roberts. “The Zig-Zag process and super-efficient sampling for Bayesian analysis of big data”. In: *The Annals of Statistics* 47.3 (2019), pp. 1288–1320. DOI: 10.1214/18-AOS1715.

[239] J. Bierkens, S. Grazzi, K. Kamatani, et al. “The Boomerang Sampler”. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 908–918. URL: http://proceedings.mlr.press/v119/bierkens20a.html.

[240] A. Bouchard-Côte, S. J. Vollmer, and A. Doucet. “The bouncy particle sampler: A nonreversible rejection-free Markov chain Monte Carlo method”. In: *Journal of the American Statistical Association* 113.522 (2018), pp. 855–867.

[241] C. Wu and C. P. Robert. “Generalized bouncy particle sampler”. June 2017. arXiv: 1706.04781.

[242] M. Betancourt. “The Fundamental Incompatibility of Scalable Hamiltonian Monte Carlo and Naive Data Subsampling”. In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. Ed. by F. R. Bach and D. M. Blei. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 533–540. URL: http://proceedings.mlr.press/v37/betancourt15.html.

[243] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, et al. “Equation of State Calculations by Fast Computing Machines”. In: *The Journal of Chemical Physics* 21.6 (June 1953), pp. 1087–1092. ISSN: 0021-9606. DOI: 10.1063/1.1699114.

[244] W. K. Hastings. “Monte Carlo Sampling Methods Using Markov Chains and Their Applications”. In: *Biometrika* 57.1 (1970), pp. 97–109. ISSN: 00063444, 14643510. URL: http://www.jstor.org/stable/2334940 (visited on 08/19/2024).

[245] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer, 2004. ISBN: 978-1-4419-1939-7. DOI: 10.1007/978-1-4757-4145-2.

[246] N. de Freitas, P. Højen-Sørensen, M. I. Jordan, et al. “Variational MCMC”. In: *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*. UAI’01. Seattle, Washington: Morgan Kaufmann Publishers Inc., 2001, pp. 120–127. ISBN: 1558608001. URL: http://arxiv.org/abs/1301.2266.

[247] A. K. Balan, Y. Chen, and M. Welling. “Austerity in MCMC Land: Cutting the Metropolis-Hastings Budget”. In: *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*. Vol. 32. JMLR Workshop and Conference Proceedings. JMLR.org, 2014, pp. 181–189. URL: http://proceedings.mlr.press/v32/korattikara14.html.

[248] D. Seita, X. Pan, H. Chen, et al. “An Efficient Minibatch Acceptance Test for Metropolis-Hastings”. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. 2018, pp. 5359–5363. DOI: 10.24963/ijcai.2018/753.

[249] H. Qu and L. Gouskos. “ParticleNet: Jet Tagging via Particle Clouds”. In: *Phys. Rev. D* 101.5 (2020), p. 056019. DOI: 10.1103/PhysRevD.101.056019.

[250] Y. Wang, Y. Sun, Z. Liu, et al. “Dynamic Graph CNN for Learning on Point Clouds”. In: *ACM Trans. Graph.* 38.5 (2019), 146:1–146:12. DOI: 10.1145/3326362.

[251] D. A. de Souza, D. Mesquita, S. Kaski, et al. “Parallel MCMC Without Embarrassing Failures”. In: *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*. Ed. by G. Camps-Valls, F. J. R. Ruiz, and I. Valera. Vol. 151. Proceedings of Machine Learning Research. PMLR, 2022, pp. 1786–1804. URL: https://proceedings.mlr.press/v151/de-souza22a.html.

[252] D. Li and W. H. Wong. “Mini-batch Tempered MCMC”. July 2017. DOI: 10.48550/arXiv.1707.09705.

[253] A. A. Barker. “Monte Carlo calculations of the radial distribution functions for a proton-electron plasma”. In: *Australian Journal of Physics* 18 (Apr. 1965), p. 119. DOI: 10.1071/PH650119.

[254] R. Zhang, A. F. Cooper, and C. D. Sa. “Asymptotically Optimal Exact Minibatch Metropolis-Hastings”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, et al. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/e2a7555f7cabd6e31aef45cb8cda4999-Abstract.html.

[255] R. D. Payne and B. K. Mallick. “Two-stage Metropolis-Hastings for tall data”. In: *Journal of classification* 35 (2018), pp. 29–51. DOI: 10.1007/s00357-018-9248-z.

[256] R. Zhang, A. F. Cooper, and C. D. Sa. “AMAGOLD: Amortized Metropolis Adjustment for Efficient Stochastic Gradient MCMC”. In: *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*. Ed. by S. Chiappa and R. Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, 2020, pp. 2142–2152. URL: http://proceedings.mlr.press/v108/zhang20e.html.

[257] P. E. Jacob and A. H. Thiery. “On nonnegative unbiased estimators”. In: *The Annals of Statistics* 43.2 (2015), pp. 769–784. DOI: 10.1214/15-AOS1311.

[258] D. Maclaurin and R. P. Adams. “Firefly Monte Carlo: Exact MCMC with Subsets of Data”. In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. Ed. by Q. Yang and M. J. Wooldridge. AAAI Press, 2015, pp. 4289–4295. URL: http://ijcai.org/Abstract/15/610.

[259] M. Banterle, C. Grazian, A. Lee, et al. *Accelerating Metropolis-Hastings algorithms by Delayed Acceptance*. 2019. DOI: 10.3934/fods.2019005.

[260] R. Cornish, P. Vanetti, A. Bouchard-Côte, et al. “Scalable Metropolis-Hastings for Exact Bayesian Inference with Large Datasets”. In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 1351–1360. URL: http://proceedings.mlr.press/v97/cornish19a.html.

[261] E. Kawasaki and M. Holzmann. “Data Subsampling for Bayesian Neural Networks”. Oct. 2022. arXiv: 2210.09141.

[262] D. Ceperley and M. Dewing. “The Penalty Method for Random Walks with Uncertain Energies”. In: *The Journal of Chemical Physics* 110 (Dec. 1998). DOI: 10.1063/1.478034.

[263] M. V. Matias Quiroz Robert Kohn and M.-N. Tran. “Speeding Up MCMC by Efficient Data Subsampling”. In: *Journal of the American Statistical Association* 114.526 (2019), pp. 831–843. DOI: 10.1080/01621459.2018.1448827.

[264] M. Quiroz, M. Tran, M. Villani, et al. “The Block-Poisson Estimator for Optimally Tuned Exact Subsampling MCMC”. In: *J. Comput. Graph. Stat.* 30.4 (2021), pp. 877–888. DOI: 10.1080/10618600.2021.1917420.

[265] E. A. Daxberger and J. M. Hernandez-Lobato. “Bayesian Variational Autoencoders for Unsupervised Out-of-Distribution Detection”. Dec. 2019. arXiv: 1912.05651.

[266] M. Miani, F. Warburg, P. Moreno-Muñoz, et al. “Laplacian Autoencoders for Learning Stochastic Representations”. In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, et al. 2022. URL: http://papers.nips.cc/paper%5C_files/paper/2022/hash/84880289c9fcba0d4bdb198cdb8f5080-Abstract-Conference.html.

[267] S. Bieringer, A. Butter, T. Heimel, et al. “Measuring QCD Splittings with Invertible Networks”. In: *SciPost Phys.* 10.6 (2021), p. 126. DOI: 10.21468/SciPostPhys.10.6.126.

[268] A. J. Larkoski, S. Marzani, G. Soyez, et al. “Soft Drop”. In: *JHEP* 05 (2014), p. 146. DOI: 10.1007/JHEP05(2014)146.

[269] J. Thaler and K. Van Tilburg. “Identifying Boosted Objects with N-subjettiness”. In: *JHEP* 03 (2011), p. 015. DOI: 10.1007/JHEP03(2011)015.

Published by IOP Publishing for Sissa Medialab

Received: *July 4, 2022*
Revised: *July 27, 2022*
Accepted: *August 22, 2022*
Published: *September 22, 2022*

Calomplification — the power of generative calorimeter models

S. Bieringer,[a,*] A. Butter,[b] S. Diefenbacher,[a] E. Eren,[c] F. Gaede,[c] D. Hundhausen,[a] G. Kasieczka,[a] B. Nachman,[d,e] T. Plehn[b] and M. Trabs[f]

[a] *Institut für Experimentalphysik, Universität Hamburg,*
Luruper Chaussee 149, 22761 Hamburg, Germany

[b] *Institut für Theoretische Physik, Universität Heidelberg,*
Philosophenweg 16, 69120 Heidelberg, Germany

[c] *Deutsches Elektronen-Synchrotron DESY,*
Notkestr. 85, 22607 Hamburg, Germany

[d] *Physics Division, Lawrence Berkeley National Laboratory,*
Berkeley, CA 94720, U.S.A.

[e] *Berkeley Institute for Data Science, University of California,*
Berkeley, CA 94720, U.S.A.

[f] *Department of Mathematics, Karlsruhe Institute of Technology,*
Englerstr. 2, 76131 Karlsruhe, Germany

E-mail: sebastian.guido.bieringer@uni-hamburg.de

Abstract: Motivated by the high computational costs of classical simulations, machine-learned generative models can be extremely useful in particle physics and elsewhere. They become especially attractive when surrogate models can efficiently learn the underlying distribution, such that a generated sample outperforms a training sample of limited size. This kind of GANplification has been observed for simple Gaussian models. We show the same effect for a physics simulation, specifically photon showers in an electromagnetic calorimeter.

Keywords: Detector modelling and simulations I (interaction of radiation with matter, interaction of photons with matter, interaction of hadrons with matter, etc); Simulation methods and programs; Analysis and statistical methods; Calorimeter methods

ArXiv ePrint: 2202.07352

[*] Corresponding author.

https://doi.org/10.1088/1748-0221/17/09/P09028

Contents

1 Introduction

Particle physics research at colliders is defined by extremely large datasets combined with precision simulations, from first principles all the way to a detailed detector simulation. A reliable generation and simulation chain is crucial to link measurements to fundamental properties of elementary particles. This chain is factorized into two main parts, event generation based on a fundamental Lagrangian and perturbative or non-perturbative quantum field theory, and detector simulations describing the interactions of relativistic particles with the detector. For the upcoming runs of the Large Hadron Collider (LHC), both parts need to gain significantly in speed, to keep up with the size of experimental datasets. One way to achieve this speed gain is to apply modern machine learning (ML) to all levels of the simulation chain. A key tool in this speed-improvement program is deep generative neural networks (NNs) that learn to emulate slower physics-based simulations, replacing the underlying physics by fast and accurate *surrogate models*.

A foundational question with NN surrogate models is, what are the advantages of using the fast simulation compared with the original dataset used for training? Or specifically, how many more events can we sensibly generate from these models before we are limited, for instance, by the training statistics? Without any additional information, we would expect that the statistical power of a generated dataset is at most the same as the dataset used for training. A larger generated sample than the training dataset will then include successively less information per event than the training data, and eventually the information in the generated events will saturate and be dominated by limitations from the network architecture and training. With this pattern in mind [1], we can define an amplification or GANplification factor [2, 3] in terms of an effective sample size for a given surrogate model.

GANplification arises, intuitively, from the fact that neural networks work like classical parametric fits [4, 5], and they are particularly effective when we want to interpolate in many dimensions. This feature is behind the success of the NNPDF parton densities [6] as the first mainstream ML-application in particle theory.

Formally, this fit-like effect is one source of inductive bias, where the underlying assumption is that physics probability densities are smooth. Especially in particle physics, it should be possible to employ other inductive biases, such as symmetries or fundamental invariances in datasets [7–12]. Fast detector simulations benefit from the fact that we can factorize the problem into pieces. Surrogate models are trained to produce a detector response for each outgoing particle. For example, if there is an event with M outgoing particles, each one will be attached to a sampling from the surrogate model. If the training set has N detector interactions, additional combinatorial factors appear for choosing N out of M different events that could be created. These factors can lead to another statistical amplification. Finally, surrogate models with valid inductive biases require far fewer parameters to specify than the original dataset, so there will also be a benefit in the required disk space.

The goal of this paper is to study the statistical amplification of deep generative models, focusing on interpolation from the smoothness inductive bias, for detector simulations as a realistic and highly relevant application. Fast surrogate models for detector simulations have been developed [13–25] and improved [26–40] to the level that they are ready to be used in the upcoming LHC runs. In fact, the ATLAS Collaboration has already integrated a Generative Adversarial Network (GAN) into its fast calorimeter simulation and will use it to generate over a billion events [41, 42]. Initial studies exist on quantifying uncertainties of generative models in event generation [43], but there has not yet been a study of the fundamental benefits of deep generative surrogates applied to detector simulations.

In this paper, we study statistical amplification in the context of photon showers in an electromagnetic calorimeter for a GAN-like generative model (Calomplification). However, the method can be applied to gauge the merit of generative surrogates whenever the underlying distribution can be accessed either through a large number of samples or analytically. We expect similar results in all cases where the smoothness assumption on the underlying density distribution is valid.

The paper is organized as follows. In section 2, we start by introducing our data set and the established generative Variational Autoencoder-GAN (VAE-GAN) architecture adapted to this simulation [30]. Next, we describe our treatment of the comparison between generated and truth samples and the relevant observables in section 3. We then present the amplification effects of the generative networks in section 4. This comparison includes an estimate of the effective sample size to the information encoded and a comparison to standard density estimators. In section 5, we briefly summarize our promising findings.

2 Dataset and model

The International Large Detector (ILD) [44] is one of two detector concepts proposed for the International Linear Collider (ILC). It is optimized towards the Particle Flow analysis concept for optimal global event reconstruction [45, 46]. It combines high-precision tracking and vertexing capabilities with very good hermiticity and highly-granular electromagnetic and hadronic calorimeters (ECal/HCal). We choose one of its two proposed electromagnetic calorimeters, the Si-W ECal, for our dataset. It consists of 30 active silicon layers in a tungsten absorber stack with 20 layers of 2.1 mm and 10 layers of 4.2 mm thickness. The silicon sensors have a cell size of 5×5 mm^2.

ILD uses iLCSoft [47] for detector simulation, reconstruction, and analysis. The Geant4 [48] simulation uses a realistic detector model implemented in DD4hep [49]. Photons are shot into the ECal barrel at a perpendicular incident angle. We project the cells with energy depositions (hits) onto a rectangular grid of $30 \times 30 \times 30$ cells. We choose photon showers, because their structure is more regular and faster to learn than the structure of pion showers [32].

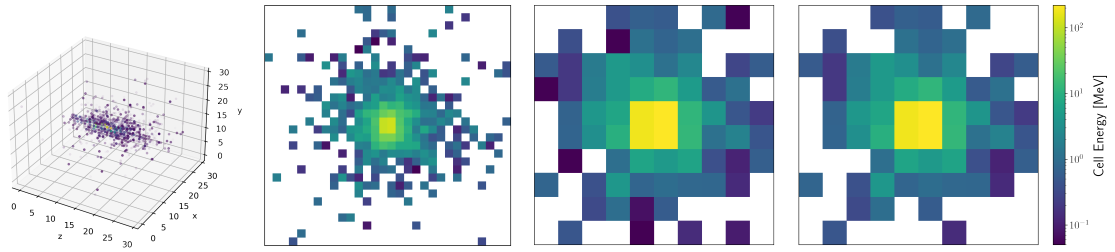


Figure 1. Illustrated transformation of the original calorimeter images from left to right. All histograms feature a logarithmic color coding, with an equal scaling for the 10 × 10 images. The final step of cutting below half the MIP energy is applied for evaluation only.

To develop a high-precision generative model, we fully simulate 268k photon showers with a fixed energy of 50 GeV. From the full set, 50k showers are randomly selected for training (1k) and for the evaluation of the network performance (all 50k). Whenever we need to estimate the generative model uncertainty, we train the network on five sets of 1k training samples. To include an error estimate for the evaluation samples, we use five sets of 5k or 10k evaluation showers, chosen as subsets of these 50k showers. The remaining 218k showers are used as a high-statistics estimate of the truth distribution.

To simplify the training of our precision-generative model, we reduce the dimensionality of the images to 10 × 10 pixels, summing along the beam axis and pooling 3 × 3 patches of the resulting 2D-images. The process is illustrated in figure 1. We will always refer to the combined calorimeter cells as pixels of the calorimeter image. The reduction allows us to obtain a powerful model from a small training set, such that the majority of the data can be used to estimate the truth distribution. Finally, we apply a cut at 0.1 MeV, which corresponds to the most probable energy deposition of a minimal ionizing particle (MIP). Cell energies below have a low signal to noise ratio. To aid the network training, this cut is not present in the training data, but applied on the full set of generated and reference data.

In comparison to studies done in context of proposed, high-granularity tracking calorimeters, these simplifications seem extensive. However, for simulation of the current ATLAS detector the AtlFast3 simulation tool [42] uses 300 individual GANs, each generating only one *eta*-slice of the calorimeter. Each network generates a 2D-image in the radius-*phi*-plane of the detector and only 1000 events are generated to learn the highest energy samples. Albeit, the models are trained including ten-thousands of lower energy samples. We see that in order to facilitate a comparison to a large validation set, the task has not been simplified further than in current applications.

Our generative architecture is a VAE-GAN [50], closely related to the network developed for precision simulations of photon showers [30] and illustrated in figure 2. It closely resembles a standard VAE setup, but deviates in its use of a GAN-like discriminator as a substitute for the usual element-wise reconstruction loss. The loss function is

$$\mathcal{L}_{\text{VAE-GAN}} = \mathcal{L}_{\text{GAN}} + \underbrace{D_{\text{KL}}\left(q_{\text{encoder}}(z|x)|p(z)\right)}_{\mathcal{L}_{\text{prior}}}$$

$$\text{with} \qquad \mathcal{L}_{\text{GAN}} = \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log D(x)\right] + \mathbb{E}_{z\sim q_{\text{encoder}}(z|x)}\left[\log(1 - D(G(z)))\right], \tag{2.1}$$

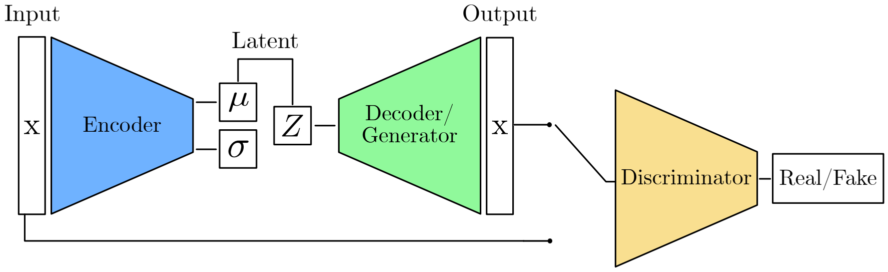


Figure 2. Illustration of the VAE-GAN architecture. The encoder and decoder form a VAE setup, while the decoder can also be understood as a GAN generator. The discriminator acts as a binary classifier, as in a classical GAN.

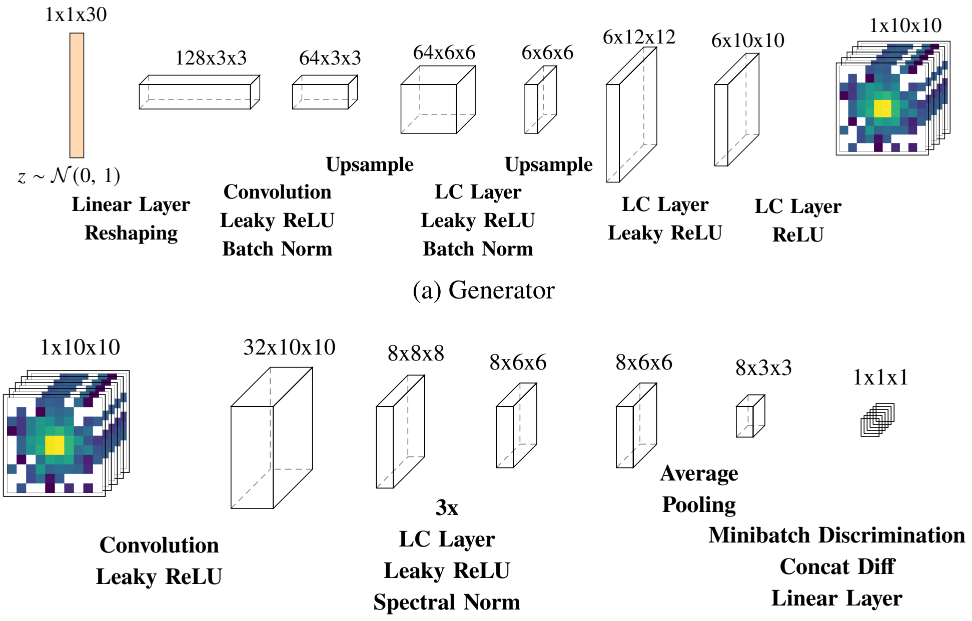
where $p(z) = \mathcal{N}(0, I)$. We maximize $\mathcal{L}_{\text{GAN}}$ during discriminator optimization. Every two discriminator steps, we update generator and encoder by minimizing the full loss function, $\mathcal{L}_{\text{VAE-GAN}}$, i.e. the generator dependent part of $\mathcal{L}_{\text{GAN}}$ and the second term $\mathcal{L}_{\text{prior}}$. The prior loss regularizes the latent space and allows us to sample from $p(z) = \mathcal{N}(0, I)$ during generation. For generator updates, we recast the GAN loss to $-\log D(G(z))$ to ensure efficient training for early epochs [51]. In every update step we sample z only once per input x. Using a GAN-like discriminator is essential, as the range of pixel values covers multiple orders of magnitude. For such images, the element-wise reconstruction loss is dominated by the central, high-energy pixels.

The GAN-like part of our network is modeled after the LAGAN [13] illustrated in figure 3. Unlike many standard applications of convolutional networks, the LAGAN features locally connected layers. Other than convolutional layers, these have the flexibility to account for the missing translation symmetry in calorimeter images. A few changes are made to the original LAGAN setup, including modifying the dimensionality of our network layers to conform to our image and latent space sizes. We also replace batch norm by spectral norm in the discriminator [52] to further stabilize the training. The discriminator uses the difference between reconstructed images and the corresponding training images as an additional input to the final, fully connected layer. For the training images themselves, this difference vector is zero. We apply label smoothing to prevent vanishing gradients from an overconfident classifier. Supplementing the information gained from the images themselves with locally connected layers and mini-batch discrimination [53] ensures better consistency between training and generated images.

The encoder network uses a convolutional input and two convolutional hidden layers, applying Leaky Rectified Linear Unit activation (LeakyReLU) [54] after the first two and ReLU after the third layer. The output of the encoder's convolutional part is fed to two separate linear layers, defining the mean and log var values of the Gaussian VAE latent space.

Our network is implemented in PyTorch 1.8.0 [55] and trained on Nvidia P100s using the `Adam` optimizer [56] with a learning rate of 8×10^{-6} for all networks. Each training on 1k showers is run for 24h, amounting to around 50000 epochs. For epochs after 40000, the distributions of the (i) pixel energy sum (visible energy), (ii) highest pixel energy (peak energy), (iii) per-pixel energy, and, (iv,v) the pixel position weighted by the pixel energy (center of gravity) in a given direction,

$$\left\{ E_{\text{vis}}, E_{\text{peak}}, E_{\text{pixel}}, \text{CG}_{x,y} \right\}, \tag{2.2}$$



(a) Generator

(b) Discriminator

Figure 3. Generator and discriminator setup including parameter space sizes in between operations. The feed-forward for both networks proceeds left to right.

are estimated using histograms of 96000 generated images. The histograms feature 100 bins and constant ranges. Finally, we select the epoch with the best agreement between the generated and training distributions averaged over all five observables, in terms of the measure discussed in the next section. This procedure is repeated for three independent trainings per set of training samples, and we draw a VAE-GAN sample in equal proportions from the resulting three models. We are aware that three independently trained models are not statistically sufficient to define a reliable standard deviation, but we have found them to be very helpful and sufficient in estimating the stability of the network training. The results in section 4 feature the standard deviation on the five different training sets. Whenever we show E_{pixel} we apply an additional minimum cut of 5 MeV, as will be discussed in detail in the next section.

3 Sample comparison

To determine the performance of the trained model, we again use distributions of the same five high-level observables as for the training. We compare showers generated by Geant4 and our VAE-GAN, but now using the high-statistics validation set. Figure 4 shows a set of distributions for 1k shower images used for a single VAE-GAN training and 1000k showers from the corresponding generative network. They are compared to the validation set of 218k Geant4 showers. In addition to the continuous distributions we also show the number of active pixels per image. First, we see that statistical fluctuations of the training set propagate into under- and over-densities of the learned distributions. One prominent difference is the number of active pixels, which can be attributed to the under-estimation of the number of low energy hits below 5 MeV. The remaining learned distributions

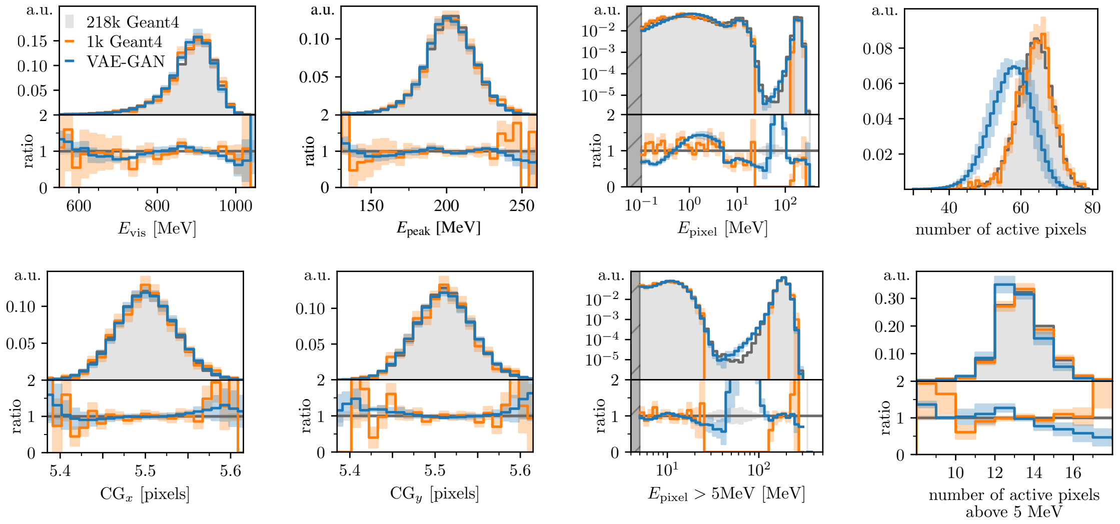


Figure 4. Differential distributions for the observables given in eq. (2.2) from Geant4 and from the VAE-GAN-generated images. Errors of the validation set (grey) and the training set (orange) correspond to the Poisson-error per bin, while the uncertainty on the VAE-GAN line (blue) is illustrated by the standard deviation of three independent trainings on the 1k training data. All histograms are normalized, such that all bins add up to one. The insets show the ratio to the high-statistics estimate of the truth distribution.

are smoother and show fewer fluctuations than the training data. For the visible per-pixel energy, the VAE-GAN interpolates into the sparsely populated interval between around 2 and 120 MeV even though the training set does not include a single pixel in this range. Previous work has shown [30] how to correct the low-energy behavior through an additional, consecutively trained post-processing network, using an maximum mean discrepancy loss [18, 57] on the pixel energy spectrum. Here we skip this post-processing and instead focus on the statistical properties of the generated data for visible pixel energies above 5 MeV.

Quantiles. We now turn to quantifying the efficacy of the VAE-GAN, given the strong performance shown in figure 4. Like in section 2, we could use standard histograms with bins of equal size. However, in this case the occupation numbers of the bins strongly depend on the assumed support of the distributions and on the binning. To avoid zero bins and sparse distributions we have to define the ranges and binnings by hand, making this strategy inconsistent in evaluation. Instead, we now split the support of the distributions into bins of equal probability weight, so-called quantiles, forming the set **Q**. We generate the quantiles for a given distribution by iteratively dividing the set of validation showers into equal-sized subsets and keeping the median as the edge of the quantile. For multi-dimensional distributions, the splitting dimensions alternate. Figure 5 illustrates this algorithm. When comparing generated with reference samples, we want to increase the number of quantiles as far as possible, to cover the entire respective distribution at sufficient resolution.

In this iterative quantile scheme, zero bins will still occur once the number of quantiles exceeds the number of generated showers. To ensure the statistical fluctuations per bin are small and do not cause empty quantiles, we discard results for more than $n/10$ bins, where n is the number of showers in the evaluation set. This leads to roughly 10 events per bin, because the evaluated data is either

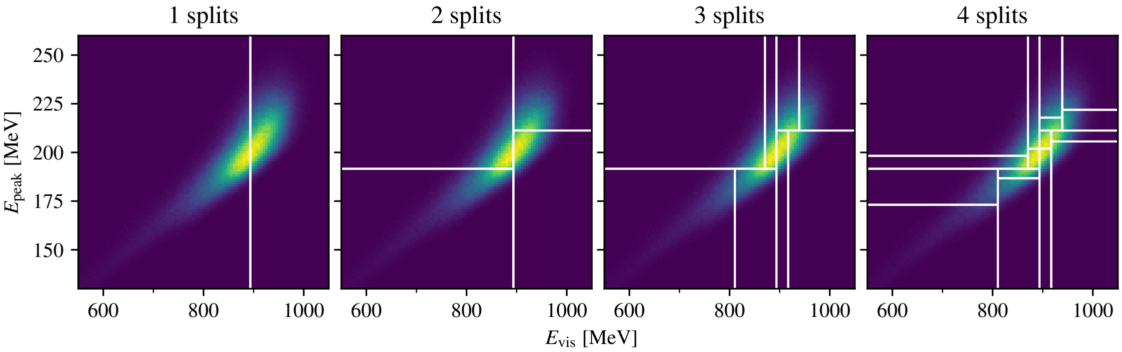


Figure 5. Quantiles developing by splitting of the validation set into subsets of equal size regarding their energy sum & peak energy.

generated from the same distribution as the validation data or is trained to resemble it well. As the event counts follow a Poisson distribution, the probability for a zero bin to occur can be calculated for the average occupation and gives around $4.5 \cdot 10^{-5}$.

Jensen-Shannon divergence. The evaluation chain for the quality of the generated samples starts by constructing quantiles from the validation set. This defines our approximate truth density p_i per quantile i. Next, we extract the density of showers g_i per quantile, either for smaller sets of Geant4 showers or 1000k VAE-GAN generated showers. Due to values appearing in our validation set multiple times, quantiles are not uniquely defined, so the p_i values may differ slightly from their constructed value $1/\#\mathbf{Q}$.

To measure the similarity of the two distributions, we use the Jensen-Shannon divergence

$$D_{\mathrm{JS}}(g,p) = \frac{1}{2} D_{\mathrm{KL}}\left(g \,\middle|\, \frac{g+p}{2}\right) + \frac{1}{2} D_{\mathrm{KL}}\left(p \,\middle|\, \frac{g+p}{2}\right) . \tag{3.1}$$

The D_{JS} can be understood as a symmetrized version of the Kullback-Leibler (KL)-divergence

$$D_{\mathrm{KL}}(g \,|\, p) = \int g(x) \log \frac{g(x)}{p(x)} \, \mathrm{d}x. \tag{3.2}$$

For the VAE-GAN results, where $g = g(x)$ is the generated distribution, the D_{JS} is the exact entity optimized by the min-max training on the GAN loss defined in eq. (2.1) [30, 51]. For GAN and Monte Carlo methods, we usually do not have an explicit form of the generated distributions, but only sets $\mathbf{G}$ and $\mathbf{P}$ generated from the estimated distribution g or the true distribution p. This is why we estimate the D_{JS} for the continuous distributions from the quantile values

$$\overline{D}_{\mathrm{JS}}(g,p) = \frac{1}{2} \sum_{Q_i \in \mathbf{Q}} \left(g_i \log \frac{g_i}{\frac{1}{2}(g_i + p_i)} + p_i \log \frac{p_i}{\frac{1}{2}(g_i + p_i)} \right) . \tag{3.3}$$

Just like the D_{JS}, this estimate lies between zero and log 2. It turns into the continuous D_{JS} between the histogram estimators

$$
\begin{aligned}
&\overline{g}(x) = \sum_{Q_i \in \mathbf{Q}} \frac{g_i}{\mathrm{vol}(Q_i)}\, 1_{Q_i}(x) = \sum_{Q_i \in \mathbf{Q}} \frac{\#\{x' \in Q_i \,|\, x' \in \mathbf{G}\}}{\#\mathbf{G} \cdot \mathrm{vol}(Q_i)}\, 1_{Q_i}(x) \\
\text{and} \qquad &\overline{p}(x) = \sum_{Q_i \in \mathbf{Q}} \frac{p_i}{\mathrm{vol}(Q_i)}\, 1_{Q_i}(x)\,,
\end{aligned}
\tag{3.4}
$$

with vol the n-dimensional volume, 1_{Q_i} the indicator function of the i-th quantile and $\mathbf{G}$ all showers in either an evaluation set of Geant4 samples or in the generated set. As for all histogram estimators, independent of the choice of bin edges, the overall number of bins, the cardinality of the fitted set, as well as the number of showers per bin have to go to infinity for the estimator to converge to the underlying distribution. As $\overline{D}_{\text{JS}}$ goes to zero, the two distributions g and p are identical.

To determine the quality of our generative model relative to truth or validation distributions, we look at the dependence of the Jensen-Shannon divergence $\overline{D}_{\text{JS}}$ on the number of quantiles n_{quant} we can reliably construct. This will allow us to gauge where the density estimation underlying the VAE-GAN beats the statistically limited training data. As discussed earlier, we estimate the uncertainty on $\overline{D}_{\text{JS}}$ for the 5k and 10k evaluation sets of Geant4 data from five independent sets each.

4 GANplification performance

Using our illustrated methodology we are now in a position to extend the toy study of ref. [1] to a relevant physics application, with the corresponding increased complexity and physics content.

Overcoming training statistics. In figure 6 we show how $\overline{D}_{\text{JS}}$ depends on the number of quantiles for the different observables given in eq. (2.2). For simple, uni-modal distributions like the energy sum, the peak energy and the centers of gravity, 1000k showers generated from the VAE-GAN achieve similar values as the 1k training data for very low numbers of bins. This means the generated data closely resembles the mean, standard deviation and low-level moments of the training data. For the more complex distribution of the visible per-pixel energy, the $\overline{D}_{\text{JS}}$ only resolves part of the high-density regions for a small number of quantiles. Increasing the numbers of quantiles, the interpolation of the generative model in the sparsely populated areas of the support starts to becomes apparent, and the $\overline{D}_{\text{JS}}$-values for the Geant4 data increases over the VAE-GAN level. As there are on average about 13 active pixels above 5 MeV, as seen in figure 4, the statistics for the per-pixel energy distribution benefits from these 13 pixel measurements per shower. For large numbers of quantiles, the $\overline{D}_{\text{JS}}$ values of the VAE-GAN are consistently below the corresponding values for the training sample and for all observables. This amplification is a result of the interpolation via the generative model's smoothing properties.

To quantify the amplification, we can compare the VAE-GAN distributions to larger Geant4 samples. Again, for small numbers of quantiles the VAE-GAN does not reach the truth $\overline{D}_{\text{JS}}$-values of larger data samples. This confirms that the neural network does not add global information to the training data and will not improve, for instance, the estimated mean of a Gaussian distribution. On the other hand, what we are really interested in are the features over the full distributions. In figure 6 we show how the network trained on 1k showers and used to generate 1000k showers plateaus in $\overline{D}_{\text{JS}}$, as a function of the resolution, and how this plateau value compares to different Geant4 sample sizes. For a large number of quantiles and probing detailed features of the distributions, our VAE-GAN surrogate description corresponds to at least 50k Geant4 showers when we look at E_{vis}, E_{peak}, or

E_{pixel}. This gives us GANplification factors as large as 50 for the relevant high-resolution features. For the reconstructed center of gravity this factor becomes a little smaller, but remains above ten.

Similar observations can be made for joint distributions, or correlations, of the different observables. Figure 7 shows how the VAE-GAN encodes the correlations between observables with a consistently smaller error than the training data. The per-pixel energy distribution cannot be included in the correlations, as it features a varying number of pixel energy values per shower, whereas all other observables give a single value per shower. An unexpected upwards slope appears when examining joint distributions containing the energy sum and the peak energy of the generated images. This can be traced back to slight, small-scale fluctuations in the correlation between them in the generated data. Still, in all of the correlations we find a GANplification factor larger than 50 for the relevant detailed features, larger than for the one-dimensional distributions, as expected from the higher dimensionality and therefore reduced per-quantile statistics.

Density estimation. After we have seen that it is beneficial to generate datasets based on a learned density estimation, the question is whether other ways to estimate densities can give similar results. While there exists literature on convergence rates of generative methods [58, 59], our physics application is defined by very specific limitations, different from those formal arguments. We therefore compare the performance of our VAE-GAN to two classical density estimation techniques. For both of them we analyze the same one-dimensional and multi-dimensional kinematic distributions as before.

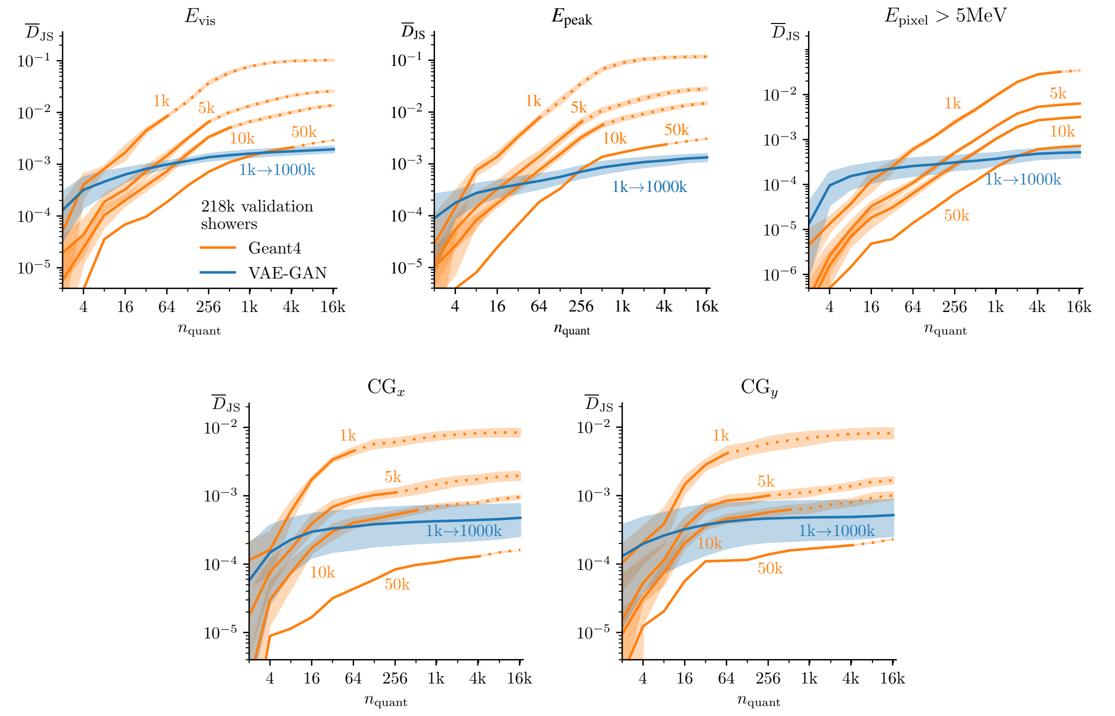


Figure 6. Dependence of $\overline{D}_{\text{JS}}$ on the number of quantiles n_{quant} for different amounts of Geant4 data (orange) and VAE-GAN data (blue) for the observables given in eq. (2.2). Solid lines indicate meaningful, non-sparse quantile sets. The 1k Geant4 samples were also used to train the VAE-GAN. Errors are calculated as the standard deviation from five datasets. For 50k we omit the negligible errors.

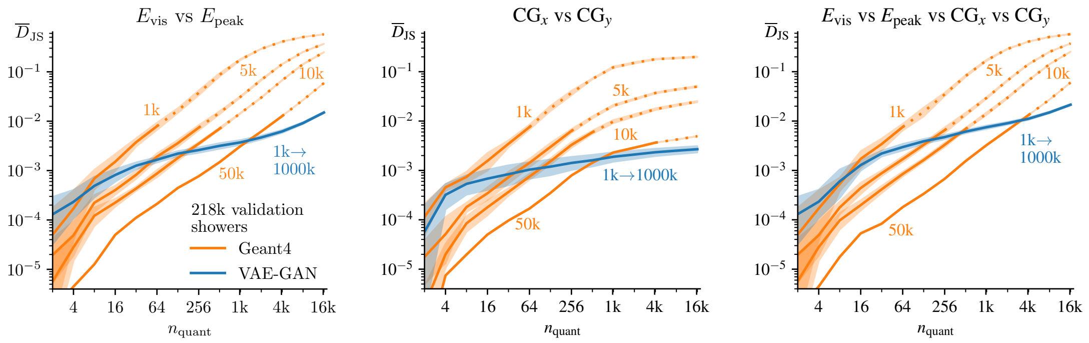


Figure 7. Dependence of $\overline{D}_{\mathrm{JS}}$ on the number of quantiles n_{quant} for different amounts of Geant4 data (orange) and VAE-GAN data (blue), now for correlations between the observables of eq. (2.2), corresponding to the 1D results in figure 6.

To each of our five training sets, we fit a kernel density estimator (KDE) and a histogram estimator, by minimizing the mean negative log-likelihood of cross-validation subsets of the training set on a grid of the estimator parameters. The results for the energy sum are shown in figure 8. For the KDE we use the `scikit-learn` [60] KernelDensity class together with the built-in GridSearchCV tool using 5-fold cross-validation to optimize the bandwidth of the Gaussian kernel. The values of the bandwidth for the individual optimizations are given in table 1. The parameters of the histogram estimator, i.e. the number of bins along the individual dimensions, are optimized using our own implementation of the same techniques. To ensure stable convergence, we form 500 cross-validation sets from the training data. The results of this optimization can again be found in table 1.

In figure 8 we see that the KDE tends to over-fit and that the histogram estimator is limited by its discrete functional form. We can analyze their performance more quantitatively using the $\overline{D}_{\mathrm{JS}}$ shown in figure 9. Due to the logarithmic nature and complex functional form of the per-pixel energy distribution, the histogram estimator and the KDE do not converge for the low number of training showers we use, so we omit this observable. First, trained on 1k showers, the histogram estimator can only use very few bins to balance over-fitting against the approximation error caused by its coarse structure and is thus outperformed by the KDE. For a larger training set and the

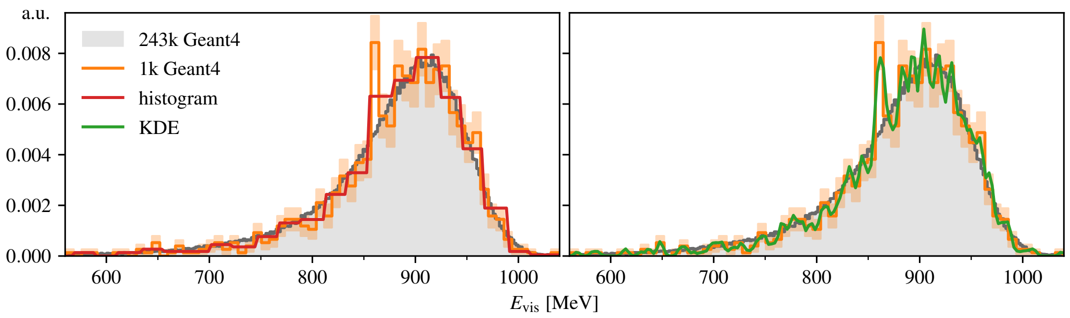


Figure 8. Example of an histogram estimator (red) and a kernel density estimator (green). The orange histogram shows the training data, including Poisson errors, that both estimators where fitted to using cross-validation.

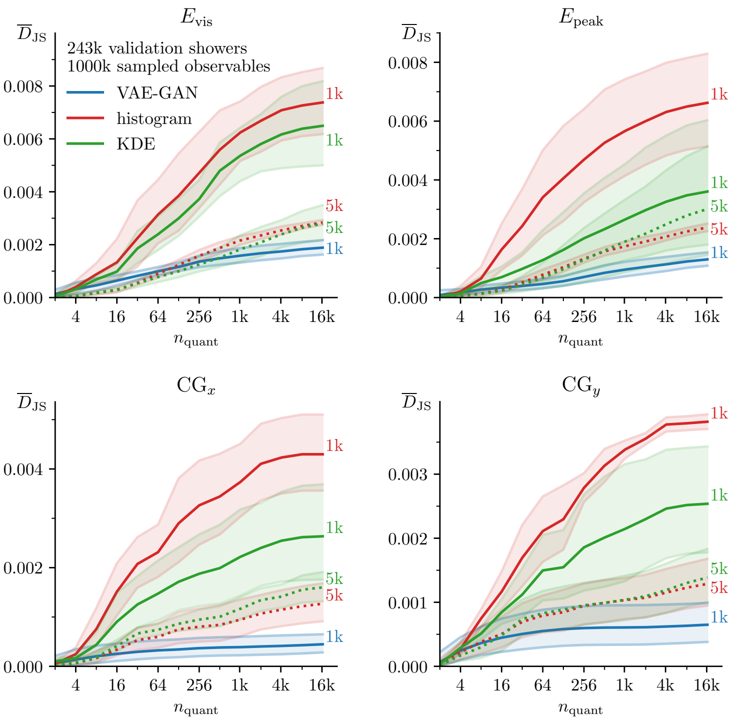


Figure 9. Dependence of $\overline{D}_{\text{JS}}$ on the number of quantiles n_{quant} for 1000k observable values sampled from histogram estimators (red) and kernel density estimators (green) and for 1000k showers sampled from VAE-GANs (blue). Errors are calculated as the standard deviation of five fits to different datasets. The size of the training sets is given to the right of the corresponding lines.

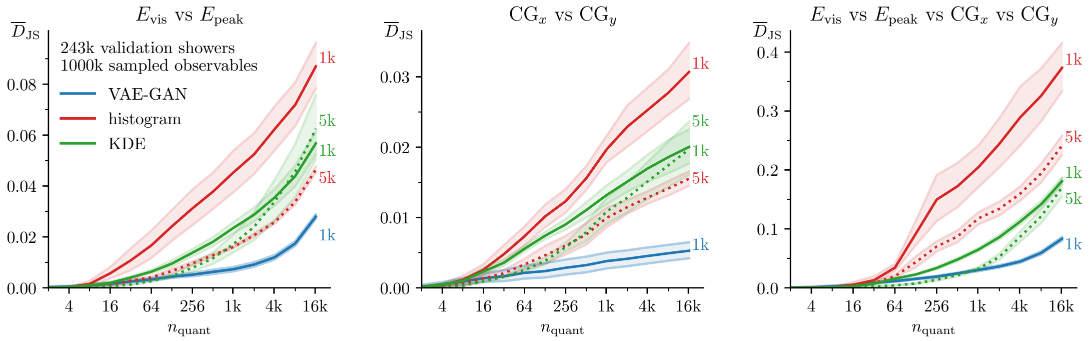


Figure 10. Dependence of $\overline{D}_{\text{JS}}$ on the number of quantiles n_{quant} for 1000k observable values sampled from different density estimators for multi-dimensional combinations of the observables given in eq. (2.2), in analogy to the 1D results in figure 9.

correspondingly larger number of bins, the approximation errors drop and both estimation methods perform similarly. However, compared to the VAE-GAN, both techniques lack descriptive power for small scales. Only for two to four bins they perform similarly to the VAE-GAN. Next, comparing the generative network to density estimators fitted to 5k showers, we can again observe the benefits of higher statistics for estimating low moments of the distributions.

For the 2-dimensional correlations shown in figure 10 we find similar limitations of the classical

Table 1. KDE bandwidths and numbers of bins in the according dimensions for the histogram estimators presented in figures 9 and 10. Estimators are fitted for five independent training sets to extract the mean and standard deviation.

	KDE bandwidth 1k	KDE bandwidth 5k	# histogram bins 1k	# histogram bins 5k
E_{vis}	0.05 ± 0.01	0.03 ± 0.01	39 ± 10	58 ± 7
E_{peak}	0.10 ± 0.03	0.03 ± 0.02	27 ± 4	50 ± 5
CG_x	0.10 ± 0.02	0.02 ± 0.01	32 ± 12	49 ± 13
CG_y	0.10 ± 0.02	0.03 ± 0.01	25 ± 4	43 ± 7
E_{vis} vs E_{peak}	0.09 ± 0.01	0.03 ± 0.01	$30 \pm 3 \times 26 \pm 7$	$40 \pm 2 \times 46 \pm 5$
CG_x vs CG_y	0.18 ± 0.02	0.07 ± 0.01	$21 \pm 1 \times 20 \pm 1$	$21 \pm 1 \times 22 \pm 2$
complete 4D	0.24 ± 0.01	0.12 ± 0.01	$20 \times 19 \times 5 \times 5 \pm 1$	$21 \times 21 \times 7 \times 8 \pm 1$

methods. Only the 4-dimensional density estimation behaves differently in that the histogram estimator is generally outperformed by the KDE. We can understand these patterns from the histogram parameters in table 1. As the histogram estimator introduces bins in every direction, the number of showers per bin drops inversely proportional to the volume of the space. To avoid over-fitting, only few bins per dimension can then be used, leading to a large approximation error. The KDE and the VAE-GAN scale better with the number of dimensions, and as before the KDE only matches the VAE-GAN performance for a very small number of quantiles.

In addition to the neural network outperforming both density estimators, we remind ourselves that the VAE-GAN actually performs the more general task of estimating the distribution of calorimeter images or low-level observables, whereas the classical methods estimate the distributions of the high-level observables.

5 Conclusions

In this paper we have shown that a realistic generative ML-model can indeed be used to generate a large number of showers, beyond a limited training statistics. Specifically, we used a VAE-GAN to generate photon showers for the electromagnetic calorimeter of the planned ILD detector design at a future linear collider. Our model is a simplification of the established precision-simulation network developed for this task [30]. This model is trained on a small number of showers from a Geant4 simulation, where a high-statistics sample of Geant4 showers serves as a truth estimate. Relative to this truth sample, we estimate the information content of finite-size samples using quantiles for standard kinematic observables and their correlations. A variable number of quantiles allows us to balance resolution with statistics.

Our study confirms earlier results based on a simple Gaussian example [1], in that for a properly trained network a set of generated showers comparable in size to the training data provides a physics-wise nearly equivalent but statistically independent copy of the training data. More generated showers will, individually, contain less information than an actual shower, but add information as a sample. This amount of information can be linked to an effective sample size of actual data. For very large numbers of generated showers, the information in the generated sample reaches a plateau, reflecting limitations of the network architecture and training.

For our problem and network at hand, we find that the effective sample sizes give an enhancement or GANplification factor of 10 to 50, for a large number of quantiles and corresponding to high-resolution kinematic features. For a training sample of 1k showers we generate up to 1000k showers from the network and find a comparable performance of up to 50k Geant4 showers for the kinematic distributions and their correlations. We also interpret the VAE-GAN as a density estimator and find that it learns the truth density from the showers better than standard density estimators on the high-level kinematic variables. This proves that the generative network can even learn and sample from implicitly defined distributions and benefit from superior interpolation or fit properties. These properties motivate deep generative detector simulations for statistical amplification in addition to computational acceleration.

Acknowledgments

We would like to thank Suada Mulgeci for the valuable discussions in the earlier stages of this project. The research of AB and TP is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant 396021762 — TRR 257 Particle Physics Phenomenology after the Higgs Discovery. This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2181/1 — 390900948 (the Heidelberg STRUCTURES Excellence Cluster). BN is supported by the U.S. Department of Energy, Office of Science under contract DE-AC02- 05CH11231. SB is supported by the Helmholtz Information and Data Science Schools via DASHH (Data Science in Hamburg — HELMHOLTZ Graduate School for the Structure of Matter) with the grant HIDSS-0002. DH and SD acknowledge support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy — EXC 2121 "Quantum Universe" — 390833306. EE is funded through the Helmholtz Innovation Pool project ACCLAIM that provided a stimulating scientific environment for parts of the research done here. This research was supported in part through the Maxwell computational resources operated at Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany.

References

[1] A. Butter, S. Diefenbacher, G. Kasieczka, B. Nachman and T. Plehn, *GANplifying event samples*, *SciPost Phys.* **10** (2021) 139 [arXiv:2008.06545].

[2] Y. Hao, A. Orlitsky, A.T. Suresh and Y. Wu, *Data amplification: A unified and competitive approach to property estimation*, arXiv:1904.00070.

[3] B. Axelrod, S. Garg, Y. Han, V. Sharan and G. Valiant, *On the Statistical Complexity of Sample Amplification*, arXiv:2201.04315.

[4] M. Bellagente, M. Haußmann, M. Luchmann and T. Plehn, *Understanding Event-Generation Networks via Uncertainties*, *SciPost Phys.* **13** (2022) 003 [arXiv:2104.04543].

[5] I. Chahrour and J.D. Wells, *Comparing machine learning and interpolation methods for loop-level calculations*, *SciPost Phys.* **12** (2022) 187 [arXiv:2111.14788].

[6] NNPDF collaboration, *Unbiased determination of the proton structure function F(2)**p with faithful uncertainty estimation*, *JHEP* **03** (2005) 080 [hep-ph/0501067].

[7] S. Krippendorf and M. Syvaeri, *Detecting Symmetries with Neural Networks*, arXiv:2003.13679.

[8] G. Barenboim, J. Hirn and V. Sanz, *Symmetry meets AI*, *SciPost Phys.* **11** (2021) 014 [arXiv:2103.06115].

[9] B.M. Dillon, G. Kasieczka, H. Olischlager, T. Plehn, P. Sorrenson and L. Vogel, *Symmetries, safety, and self-supervision*, *SciPost Phys.* **12** (2022) 188 [arXiv:2108.04253].

[10] C.G. Lester, *Chiral Measurements*, arXiv:2111.00623.

[11] R. Tombs and C.G. Lester, *A method to challenge symmetries in data with self-supervised learning*, 2022 *JINST* **17** P08024 [arXiv:2111.05442].

[12] K. Desai, B. Nachman and J. Thaler, *Symmetry discovery with deep learning*, *Phys. Rev. D* **105** (2022) 096031 [arXiv:2112.05722].

[13] L. de Oliveira, M. Paganini and B. Nachman, *Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis*, *Comput. Softw. Big Sci.* **1** (2017) 4 [arXiv:1701.05927].

[14] M. Paganini, L. de Oliveira and B. Nachman, *Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters*, *Phys. Rev. Lett.* **120** (2018) 042003 [arXiv:1705.02355].

[15] M. Paganini, L. de Oliveira and B. Nachman, *CaloGAN: Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks*, *Phys. Rev. D* **97** (2018) 014021 [arXiv:1712.10321].

[16] S. Vallecorsa, F. Carminati and G. Khattak, *3D convolutional GAN for fast simulation*, *EPJ Web Conf.* **214** (2019) 02010.

[17] S. Carrazza and F.A. Dreyer, *Lund jet images from generative and cycle-consistent adversarial networks*, *Eur. Phys. J. C* **79** (2019) 979 [arXiv:1909.01359].

[18] A. Butter, T. Plehn and R. Winterhalder, *How to GAN LHC Events*, *SciPost Phys.* **7** (2019) 075 [arXiv:1907.03764].

[19] R. Di Sipio, M. Faucci Giannelli, S. Ketabchi Haghighat and S. Palazzo, *DijetGAN: A Generative-Adversarial Network Approach for the Simulation of QCD Dijet Events at the LHC*, *JHEP* **08** (2019) 110 [arXiv:1903.02433].

[20] V. Chekalina, E. Orlova, F. Ratnikov, D. Ulyanov, A. Ustyuzhanin and E. Zakharov, *Generative Models for Fast Calorimeter Simulation: the LHCb case*, *EPJ Web Conf.* **214** (2019) 02034 [arXiv:1812.01319].

[21] P. Musella and F. Pandolfi, *Fast and Accurate Simulation of Particle Detectors Using Generative Adversarial Networks*, *Comput. Softw. Big Sci.* **2** (2018) 8 [arXiv:1805.00850].

[22] K. Deja, T. Trzcinski and Ł. Graczykowski, *Generative models for fast cluster simulations in the TPC for the ALICE experiment*, *EPJ Web Conf.* **214** (2019) 06003.

[23] L. de Oliveira, M. Paganini and B. Nachman, *Controlling Physical Attributes in GAN-Accelerated Simulation of Electromagnetic Calorimeters*, *J. Phys. Conf. Ser.* **1085** (2018) 042017 [arXiv:1711.08813].

[24] J.W. Monk, *Deep Learning as a Parton Shower*, *JHEP* **12** (2018) 021 [arXiv:1807.03685].

[25] J.N. Howard, S. Mandt, D. Whiteson and Y. Yang, *Learning to simulate high energy particle collisions from unlabeled data*, *Sci. Rep.* **12** (2022) 7567 [arXiv:2101.08944].

[26] M. Erdmann, L. Geiger, J. Glombitza and D. Schmidt, *Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks*, *Comput. Softw. Big Sci.* **2** (2018) 4 [arXiv:1802.03325].

[27] M. Erdmann, J. Glombitza and T. Quast, *Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network*, *Comput. Softw. Big Sci.* **3** (2019) 4 [arXiv:1807.01954].

[28] M. Backes, A. Butter, T. Plehn and R. Winterhalder, *How to GAN Event Unweighting*, *SciPost Phys.* **10** (2021) 089 [arXiv:2012.07873].

[29] D. Belayneh et al., *Calorimetry with deep learning: particle simulation and reconstruction for collider physics*, *Eur. Phys. J. C* **80** (2020) 688 [arXiv:1912.06794].

[30] E. Buhmann et al., *Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed*, *Comput. Softw. Big Sci.* **5** (2021) 13 [arXiv:2005.05334].

[31] E. Buhmann et al., *Decoding Photons: Physics in the Latent Space of a BIB-AE Generative Network*, *EPJ Web Conf.* **251** (2021) 03003 [arXiv:2102.12491].

[32] E. Buhmann et al., *Hadrons, better, faster, stronger*, *Mach. Learn. Sci. Tech.* **3** (2022) 025014 [arXiv:2112.09709].

[33] C. Krause and D. Shih, *CaloFlow: Fast and Accurate Generation of Calorimeter Showers with Normalizing Flows*, arXiv:2106.05285.

[34] C. Krause and D. Shih, *CaloFlow II: Even Faster and Still Accurate Generation of Calorimeter Showers with Normalizing Flows*, arXiv:2110.11377.

[35] G.R. Khattak, S. Vallecorsa, F. Carminati and G.M. Khan, *Fast simulation of a high granularity calorimeter by generative adversarial networks*, *Eur. Phys. J. C* **82** (2022) 386 [arXiv:2109.07388].

[36] R. Kansal et al., *Particle Cloud Generation with Message Passing Generative Adversarial Networks*, arXiv:2106.11535.

[37] A. Hariri, D. Dyachkova and S. Gleyzer, *Graph Generative Models for Fast Detector Simulations in High Energy Physics*, arXiv:2104.01725.

[38] F. Rehm et al., *Reduced Precision Strategies for Deep Learning: A High Energy Physics Generative Adversarial Network Use Case*, arXiv:2103.10142.

[39] F. Rehm, S. Vallecorsa, K. Borras and D. Krücker, *Validation of Deep Convolutional Generative Adversarial Networks for High Energy Physics Calorimeter Simulations*, arXiv:2103.13698.

[40] F. Rehm, S. Vallecorsa, K. Borras and D. Krücker, *Physics Validation of Novel Convolutional* 2D *Architectures for Speeding Up High Energy Physics Simulations*, *EPJ Web Conf.* **251** (2021) 03042 [arXiv:2105.08960].

[41] ATLAS collaboration, *Deep generative models for fast shower simulation in ATLAS*, ATL-SOFT-PUB-2018-001 (2018).

[42] ATLAS collaboration, *AtlFast3: the next generation of fast simulation in ATLAS*, *Comput. Softw. Big Sci.* **6** (2022) 7 [arXiv:2109.02551].

[43] A. Butter et al., *Generative Networks for Precision Enthusiasts*, arXiv:2110.13632.

[44] ILD Concept Group collaboration, *International Large Detector: Interim Design Report*, arXiv:2003.01116.

[45] M.A. Thomson, *Particle Flow Calorimetry and the PandoraPFA Algorithm*, *Nucl. Instrum. Meth. A* **611** (2009) 25 [arXiv:0907.3577].

[46] J.S. Marshall and M.A. Thomson, *The Pandora Software Development Kit for Pattern Recognition*, *Eur. Phys. J. C* **75** (2015) 439 [arXiv:1506.05348].

[47] *iLCSoft Project Page*, https://github.com/iLCSoft (2016).

[48] J. Allison et al., *Recent developments in Geant4*, *Nucl. Instrum. Meth. A* **835** (2016) 186.

[49] M. Frank, F. Gaede, C. Grefe and P. Mato, *DD4hep: A Detector Description Toolkit for High Energy Physics Experiments*, *J. Phys. Conf. Ser.* **513** (2014) 022010.

[50] A.B.L. Larsen, S.K. Sønderby, H. Larochelle and O. Winther, *Autoencoding beyond pixels using a learned similarity metric*, in *Proceedings of the 33rd International Conference on International Conference on Machine Learning — Volume 48*, pp. 1558-1566, PMLR (2016) [PDF].

[51] I.J. Goodfellow et al., *Generative Adversarial Networks*, arXiv:1406.2661.

[52] T. Miyato, T. Kataoka, M. Koyama and Y. Yoshida, *Spectral Normalization for Generative Adversarial Networks*, arXiv:1802.05957.

[53] T. Salimans et al., *Improved techniques for training gans*, in *Adv. Neural Inf. Process. Syst.*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon and R. Garnett eds., vol. 29, Curran Associates, Inc. (2016) [PDF].

[54] A.L. Maas, A.Y. Hannun and A.Y. Ng, *Rectifier nonlinearities improve neural network acoustic models*, in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, Atlanta, Georgia, U.S.A. (2013).

[55] A. Paszke et al., *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, *Adv. Neural Inf. Process. Syst.* **32** (2019) 8024.

[56] D.P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980.

[57] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf and A. Smola, *A kernel method for the two-sample-problem*, in *Adv. Neural Inf. Process. Syst.*, B. Schölkopf, J. Platt and T. Hoffman eds., vol. 19, MIT Press (2007) [PDF].

[58] G. Biau, B. Cadre, M. Sangnier and U. Tanielian, *Some theoretical properties of gans*, *Annals Statist.* **48** (2018) 1539 [arXiv:1803.07819].

[59] D. Belomestny, E. Moulines, A. Naumov, N. Puchkin and S. Samsonov, *Rates of convergence for density estimation with GANs*, arXiv:2102.00199.

[60] F. Pedregosa et al., *Scikit-learn: Machine learning in Python*, *J. Mach. Learn. Res.* **12** (2011) 2825.

AdamMCMC: Combining Metropolis Adjusted Langevin with Momentum-based Optimization

A Preprint

Sebastian Bieringer[*1], **Gregor Kasieczka**[1], **Maximilian F. Steffen**[2], **and Mathias Trabs**[2]
[1]Universität Hamburg, Institut für Experimentalphysik, Luruper Chaussee 149, 22761 Hamburg, Germany
[2]Karlsruhe Institute of Technology, Institut für Stochastik, Englerstr. 2, 76131 Karlsruhe, Germany

December 6, 2024

Abstract

Uncertainty estimation is a key issue when considering the application of deep neural network methods in science and engineering. In this work, we introduce a novel algorithm that quantifies epistemic uncertainty via Monte Carlo sampling from a tempered posterior distribution. It combines the well established Metropolis Adjusted Langevin Algorithm (MALA) with momentum-based optimization using `Adam` and leverages a prolate proposal distribution, to efficiently draw from the posterior. We prove that the constructed chain admits the Gibbs posterior as invariant distribution and approximates this posterior in total variation distance. Furthermore, we demonstrate the efficiency of the resulting algorithm and the merit of the proposed changes on a state-of-the-art classifier from high-energy particle physics.

1 Introduction

Deep learning methods are widely applied in industry, engineering, science and medicine. Especially in the latter fields, widespread application of such methods is held back by non-existent, overconfident or time-intensive error estimates. This is especially problematic when neural network outputs are used in clinical decision making, control systems of autonomous vehicles, or scientific discovery, for example in particle physics (Karagiorgi et al., 2022).

In these cases, the uncertainty in the data generation or taking, the *aleatoric* uncertainty, can by accessed by learning a data likelihood $p(\mathcal{D}|\vartheta)$ in the same framework. This is often done by parameterizing the likelihood as a Gaussian (Gal, 2016) or with a normalizing flow architecture (Radev et al., 2020).

Bayesian neural networks (BNNs) can be used to estimate an uncertainty on the neural network fit stemming from the limited training statistics, that is the *epistemic* uncertainty. Such methods understand the network parameters ϑ as random variables from a posterior distribution $p(\vartheta|\mathcal{D})$ conditioned on the training data $\mathcal{D}$. By drawing from this distribution we can produce an uncertainty prediction from an ensemble of different parameter samples.

BNN algorithms based on parametric estimates of the posterior weights, be it Gaussian-mean field (Blundell et al., 2015) or Laplace approximation based (Daxberger et al., 2021; Ritter et al., 2018), have to balance the quality of the fit with the complexity of the algorithm. To accommodate efficient evaluation and scaling with the network size, they rely on (block-) diagonal approximations of the covariance matrix, leading to bad fit performance and underconfident uncertainties.

Markov Chain Monte Carlo (MCMC) algorithms on the other hand can access the full weight posterior, but struggle with slow convergence and high computational costs. To reduce computation, Welling & Teh (2011) proposed a first chain based on stochastic gradients. The authors achieve unbiased sampling in the limit of diminishing step sizes, thus reducing the phase space exploration of the algorithm to a random walk. Chen et al. (2014) therefore adapt Hamiltonian Monte Carlo for stochastic gradients (sgHMC) to achieve more efficient exploration through use of an

[*] `sebastian.guido.bieringer@uni-hamburg.de`

auxiliary momentum variable. Thermostats Ding et al. (2014); Heek & Kalchbrenner (2019) use yet another auxiliary variable, inspired by thermodynamics, that is quadratic in the parameters to further improve the sampling.

The random walk like diffusion process underlying these stochastic gradient based chains can be tuned to the geometry of the posterior with a diffusion matrix (Patterson & Teh, 2013; Xifara et al., 2014). Ma et al. (2015) note that arbitrary, positive definite diffusion matrices can be used whenever the commonly used Fisher metric is hard to compute. This includes the preconditioning of RMSprop (Li et al., 2016; Chen et al., 2016).

Rather than using the general framework of Ma et al. (2015) to construct a chain that admits the posterior as invariant distribution, we bridge the gap to stochastic optimization by

- proposing a novel algorithm that employs the MALA algorithm (Roberts & Tweedie, 1996a) around `Adam` updates (Kingma & Ba, 2015), including first and second order momentum. In combination with a prolate deformation of the diffusion we ensure high acceptance rates. We refer to the resulting algorithm as `AdamMCMC`.
- For a state-of-the-art particle physics application, we show the changes lead to a more well-behaved algorithm, that approaches the performance of pure `Adam` for narrow proposal distributions and allows adjusting the calibration as a tradeoff between fit uncertainty and performance.

`AdamMCMC` uses a Metropolis-Hastings (M-H) step Robert & Casella (2004) that ensures convergence for fixed size learning rates. It has desirable theoretical properties in terms of contraction rates as well as credible sets (Franssen & Szabó, 2022; Bieringer et al., 2023). Our study can thus also be understood as a study of the difference between stochastic optimization and Bayesian inference in terms of M-H acceptance rates.

In Section 2, we introduce the established MALA algorithm. The changes to the proposal distribution are presented in Section 3, with the accompanying proofs in Appendix A. We employ the proposed algorithm on an application from particle physics in Section 4, where we use a stochastic approximation of the M-H step. We discuss the implications of this choice, as well as proposals from related literature, in Section 4.4 before concluding in Section 5.

2 Metropolis Adjusted Langevin Algorithm

For a labeled or unlabeled n-point dataset $\mathcal{D}_n$, the P-dimensional vector of network weights ϑ, a generic loss function $L(\vartheta)$ and its empirical counterpart $L_n(\vartheta) = L(\vartheta; \mathcal{D}_n)$, the Gibbs posterior is given by the density

$$p_\lambda(\vartheta|\mathcal{D}_n) \propto \exp(-\lambda L_n(\vartheta))\, p(\vartheta), \tag{1}$$

where $\lambda > 0$ is the inverse temperature parameter and $p(\vartheta)$ is a prior density on the network weights. The Gibbs posterior is a central object in the PAC-Bayes theory (Alquier, 2021) and it matches the classical Bayesian posterior distribution by Bayes theorem if $\lambda L_n(\vartheta)$ is the negative log-likelihood of the data-generating distribution, that is $p(\mathcal{D}_n|\vartheta) = \exp(-\lambda L_n(\vartheta))$. Throughout, we choose a uniform prior on some bounded set $\Omega \subset \mathbb{R}^P$.

Starting at some initial choice $\vartheta^{(0)}$, for every update step $k+1$ a new parameter proposal $\tau^{(k)}$ is sampled from a Gaussian *proposal density* centered in a gradient step

$$\tau^{(k)} \sim q(\tau|\vartheta^{(k)}) = \frac{1}{(2\pi\sigma^2)^{P/2}} \exp\left(-\frac{1}{2\sigma^2}\left|\tau - \vartheta^{(k)} + \gamma\nabla_\vartheta L_n(\vartheta^{(k)})\right|^2\right). \tag{2}$$

The new proposal is accepted with the *acceptance probability* (ratio)

$$\alpha(\tau^{(k)}|\vartheta^{(k)}) = \left(\exp\left(-\lambda L_n(\tau^{(k)}) + \lambda L_n(\vartheta^{(k)})\right) \cdot \mathbf{1}_\Omega(\tau^{(k)}) \frac{q(\vartheta^{(k)}|\tau^{(k)})}{q(\tau^{(k)}|\vartheta^{(k)})}\right) \wedge 1, \tag{3}$$

where $a \wedge b = \min\{a, b\}$. Here, $q(\vartheta^{(k)}|\tau^{(k)})$ is the probability of the current parameters given the parameter proposal, that is the *backwards direction*. When the gradient steps are larger than σ, this probability can be very small and lead to vanishing acceptance probabilities.

If the proposal is not accepted, the previous parameter values are kept:

$$\vartheta^{(k+1)} = \begin{cases} \tau^{(k)} \text{ with probability } \alpha(\tau^{(k)}|\vartheta^{(k)}) \\ \vartheta^{(k)} \text{ with probability } 1 - \alpha(\tau^{(k)}|\vartheta^{(k)}). \end{cases}$$

The calculation of the acceptance probability requires the evaluation of the loss function L_n and its gradient $\nabla_\vartheta L_n$ for both the current weights $\vartheta^{(k)}$ and the proposed update $\tau^{(k)}$.

The choice of proposal density ensures that for a proposal $\tau^{(k)}$ close to the previous parameters $\vartheta^{(k)}$, the transition probability $q(\tau^{(k)}|\vartheta^{(k)})$ is bounded from below. Under these assumptions, Roberts & Tweedie (1996b, Theorem 2.2) prove that $(\vartheta^{(k)})_{k\in\mathbb{N}_0}$ is a Markov chain with invariant distribution $p_\lambda(\cdot|\mathcal{D}_n)$. The resulting neural network estimators are denoted by $\hat{f}_{\vartheta^{(k)}}$.

After a *burn-in* time $b \in \mathbb{N}$, the Markov chain stabilizes at its invariant distribution and generates samples from the Gibbs posterior (1) in every iteration. To ensure approximate independence of the samples, we estimate the posterior mean prediction over samples with a gap of *gap length* $c \in \mathbb{N}$

$$\bar{f}_\lambda = \frac{1}{N}\sum_{i=1}^{N} \hat{f}_{\vartheta^{(b+ic)}}.$$

3 Improved Sampling Efficiency

Proposals unlikely under the Gibbs posterior will come with low acceptance probabilities. We thus need to make sure the proposals track the loss landscape closely as not to slow down the algorithm.For fast convergence, it is thus desirable to run MALA at low σ. This, on the contrary, reduces the probability of the backwards direction. To solve this dilemma, we adapt the diffusion process and introduce a prolate proposal distribution. In combination with momentum-smoothed trajectories (see Section 3.2), we recover high probabilities of the backwards direction even at large σ.

3.1 Directional Noise

Similarly to Ludkin & Sherlock (2023), we replace the isotropic noise in (2) with a modified Langevin proposal with a *directional noise* in gradient direction $\nabla L_n^{(k)} := \nabla_\vartheta L_n(\vartheta^{(k)})$. The new proposal distribution

$$\tau^{(k)} \sim q(\tau|\vartheta^{(k)}) = \frac{1}{\sqrt{(2\pi)^P \det(\mathbf{\Sigma}_k)}} \exp\left(-\frac{1}{2}\left(\tau - \tilde{\vartheta}^{(k+1)}\right)^\top \mathbf{\Sigma}_k^{-1}\left(\tau - \tilde{\vartheta}^{(k+1)}\right)\right), \tag{4}$$

is centered in

$$\tilde{\vartheta}^{(k+1)} = \vartheta^{(k)} + \gamma \nabla_\vartheta L_n(\vartheta^{(k)},$$

with covariance

$$\mathbf{\Sigma}_k := \mathbf{\Sigma}\left(\nabla L_n^{(k)}\right) := \sigma^2 I_P + \sigma_\nabla^2 \nabla L_n^{(k)} \big(\nabla L_n^{(k)}\big)^\top.$$

Here, $\sigma, \sigma_\nabla > 0$ are noise levels and I_P the P-dimensional unit matrix. We omit the argument and write $\mathbf{\Sigma}_k$ whenever the direction of the update is clear from context. In particular, the variance of $\tau^{(k)}$ in gradient direction is

$$\mathrm{Var}\big(\langle \tau^{(k)}, \nabla L_n^{(k)}\rangle\big) = \sigma^2 |\nabla L_n^{(k)}|^2 + \sigma_\nabla^2 |\nabla L_n^{(k)}|^4,$$

while the variance in any direction $v \in \mathbb{R}^P$ orthogonal to $\nabla L_n^{(k)}$ is $\mathrm{Var}(\langle \tau^{(k)}, v\rangle) = \sigma^2|v|^2$. As a consequence, the underlying random walk converges more reliably towards a minimum of the loss while still being sufficiently randomized to explore the whole parameter space.

To evaluate the acceptance probability under the new proposal distribution effectively, we need to circumvent matrix multiplications in the calculation of both the determinant and inverse of the covariance $\mathbf{\Sigma}_k$. Thanks to our particular choice of the anisotropic covariance structure, the Sylvester determinant identity yields

$$\det(\mathbf{\Sigma}_k) = \sigma^{2P}\Big(1 + \frac{\sigma_\nabla^2}{\sigma^2}\left|\nabla L_n^{(k)}\right|^2\Big) \tag{5}$$

and due to the rank one perturbation of the unit matrix the inverse covariance matrix can be calculated as

$$\mathbf{\Sigma}_k^{-1} = \sigma^{-2} I_P - \frac{\sigma^{-4}\sigma_\nabla^2}{1 + \sigma^{-2}\sigma_\nabla^2 |\nabla L_n^{(k)}|^2} \nabla L_n^{(k)} \big(\nabla L_n^{(k)}\big)^\top. \tag{6}$$

and the probability of the proposal (4) can thus be evaluated as two vector products.

Note that the directional noise affects the invariant distribution chain. Without a M-H step, a correction term to the gradient update, needs to be applied. Following the general framework of Ma et al. (2015), this correction would include second derivatives of the loss.

Algorithm 1: `AdamMCMC`

```
Input: empirical loss function $L_n(\vartheta)$, starting position $\vartheta^{(0)}$, inverse temperature $\lambda > 0$, learning rate $\gamma > 0$,
       momenta parameters $\beta_1, \beta_2 \in [0, 1)$, standard deviations $\sigma, \sigma_\nabla > 0$.
for $k = 0, 1, 2...$ do
    Sample parameters around Adam$(L_n(\vartheta^{(k)}))$:
    $m^{(k+1)} = (m_1^{(k+1)}, m_2^{(k+1)}) \leftarrow \Big(\beta_1 m_1^{(k)} + (1-\beta_1)\nabla L_n(\vartheta^{(k)}), \beta_2 m_2^{(k)} + (1-\beta_2)\nabla L_n(\vartheta^{(k)})^2\Big)$
    $\tilde{\vartheta}^{(k+1)} \leftarrow \vartheta^{(k)} - u_k(m^{(k+1)})$
    $\tau^{(k)} \sim \mathcal{N}\left(\tilde{\vartheta}^{(k+1)}, \boldsymbol{\Sigma}\left(u_k(m^{(k+1)})\right)\right)$

    Metropolis-Hastings correction:
    $\tilde{\tau}^{(k+1)} \leftarrow \tau^{(k)} - u_k(m^{(k+1)})$
    $\alpha_k(\tau^{(k)} | \vartheta^{(k)}, m^{(k+1)}) = \mathbf{1}_\Omega(\tau^{(k)}) \cdot \frac{\exp\left(-\lambda L_n(\tau^{(k)})\right)}{\exp\left(-\lambda L_n(\vartheta^{(k)})\right)} \frac{\varphi_{\tilde{\tau}^{(k+1)}, \boldsymbol{\Sigma}\left(u_k(m^{(k+1)})\right)}\left(\vartheta^{(k)}\right)}{\varphi_{\tilde{\vartheta}^{(k+1)}, \boldsymbol{\Sigma}\left(u_k(m^{(k+1)})\right)}\left(\tau^{(k)}\right)}$
    $a \sim \mathrm{uniform}(0, 1)$
    if $a \le \alpha_k(\tau^{(k)} | \vartheta^{(k)}, m^{(k+1)})$ then
        $\vartheta^{(k+1)} \leftarrow \tau^{(k)}$
    else
        $\vartheta^{(k+1)} \leftarrow \vartheta^{(k)}$
    end
end
```

3.2 Metropolis-Hastings with `Adam`

The change in the proposal distribution is especially effective in combination with momentum variables. We therefore replace the gradient $\nabla L_n^{(k)}$ in the above construction with an `Adam` update step (Kingma & Ba, 2015, Algorithm 1). The k-th step of `Adam` updates the momenta $m^{(k+1)} := \big(m_1^{(k+1)}, m_2^{(k+1)}\big)$ as

$$\begin{aligned} m_1^{(k+1)} &:= \beta_1 m_1^{(k)} + (1-\beta_1)\nabla L_n^{(k)}, \\ m_2^{(k+1)} &:= \beta_2 m_2^{(k)} + (1-\beta_2)(\nabla L_n^{(k)})^2, \end{aligned} \tag{7}$$

where $\beta_1, \beta_2 \in [0, 1)$ tune the importance of the momenta and the exponent is understood component wise. The network parameters are then updated as

$$\vartheta^{(k+1)} := \vartheta^{(k)} - u_k(m^{(k+1)}),$$

with

$$u_k(m^{(k+1)}) := \gamma \frac{m_1^{(k+1)}}{1-\beta_1^{k+1}} \Big/ \bigg(\Big(\frac{|m_2^{(k+1)}|_\bullet}{1-\beta_2^{k+1}}\Big)^{1/2} + \delta\bigg)$$

Again, exponents and division are understood component wise, $|\cdot|_\bullet$ denotes the entry-wise absolute value and a small constant $\delta > 0$ prevents numerical nuisance.

Due to the dependence on the momenta, the augmented chain $(\vartheta^{(k)}, m^{(k)})_{k\geq 1}$ has to be considered. The augmentation recovers the Markovian nature of the chain and allows us to calculate the acceptance probabilities, as in (3), based on the proposal distributions $q(\vartheta^{(k)}|\tau^{(k)}, m^{(k+1)})$ and $q(\tau^{(k)}|\vartheta^{(k)}, m^{(k+1)})$ using the same momenta $m^{(k+1)}$, see Algorithm 1. Therein, we denote the probability density function of the P-dimensional normal distribution $\mathcal{N}(\mu, \boldsymbol{\Sigma})$ by $\varphi_{\mu,\boldsymbol{\Sigma}}$. If the covariance matrix is diagonal, that is $\boldsymbol{\Sigma} = \sigma^2 I_P$ for some $\sigma > 0$, we abbreviate $\varphi_{\mu,\sigma^2} = \varphi_{\mu,\boldsymbol{\Sigma}}$.

The proposal for the network parameters is distributed according to

$$q_{1,k}(\tau^{(k)}|\vartheta^{(k)}, m^{(k+1)}) = \varphi_{\vartheta^{(k)} - u_k(m^{(k+1)}), \boldsymbol{\Sigma}_k}(\tau^{(k)}) \tag{8}$$

with covariance matrix

$$\boldsymbol{\Sigma}_k = \sigma^2 I_P + \sigma_\nabla^2 u_k(m^{(k+1)}) u_k(m^{(k+1)})^\top.$$

Although we do not randomize the moments in practice, for theoretical considerations we impose a momentum update distribution given by

$$q_2(m^{(k+1)}|\vartheta^{(k)}, m^{(k)}) = \prod_{l=1}^{2} \varphi_{\beta_l m_l^{(k)} + (1-\beta_l)\nabla L_n(\vartheta^{(k)})^l, \rho_l^2}(m_l^{(k+1)}) \tag{9}$$

with small noise levels $\rho_1, \rho_2 > 0$. This approximation is in line with an additional stochastic error that occurs if the gradient is replaced by a stochastic gradient for Section 4 and similar approximations are applied for example by Chen et al. (2014). The acceptance probabilities are then given by

$$\alpha_k(\tau^{(k)}|\vartheta^{(k)}, m^{(k+1)}) = 1 \wedge \Big(\frac{p_\lambda(\tau^{(k)}|\mathcal{D}_n)}{p_\lambda(\vartheta^{(k)}|\mathcal{D}_n)} \frac{q_{1,k}(\vartheta^{(k)}|\tau^{(k)}, m^{(k+1)})}{q_{1,k}(\tau^{(k)}|\vartheta^{(k)}, m^{(k+1)})} C(\vartheta^{(k)}, \tau^{(k)})\Big) \tag{10}$$

(setting $0/0 = 0$ if $p_\lambda(\vartheta^{(k)}|\mathcal{D}_n) = 0$) with a correction term

$$C(\vartheta^{(k)}, \tau^{(k)}) = \exp\Big(- \sum_{l=1}^{2} \frac{|m_l^{(k+1)} - \nabla L_n(\tau^{(k)})^l|^2}{2\rho_l^2/(1-\beta_l^2)} \quad + \sum_{l=1}^{2} \frac{|m_l^{(k+1)} - \nabla L_n(\vartheta^{(k)})^l|^2}{2\rho_l^2/(1-\beta_l^2)}\Big).$$

Note that β_1 and β_2 are chosen close to 1 and after a burn-in time the gradients $\nabla L_n(\vartheta^{(k)})^l$ and $\nabla L_n(\tau^{(k)})^l$ will be small and close to their long-term average $m_l^{(k)}$. While our theoretical results rely on the explicit form of $C(\vartheta^{(k)}, \tau^{(k)})$, the correction term can be well approximated by 1 in the sampling stage and in our algorithm `AdamMCMC` we simply set $C(\vartheta^{(k)}, \tau^{(k)}) = 1$.

The following theorem verifies that the `Adam` based Metropolis-Hastings algorithm indeed admits the desired invariant distribution.

Theorem 1. *For arbitrary proposal distributions $q_{1,k}(\tau^{(k)}|\vartheta^{(k)}, m^{(k+1)})$ and q_2 from* (9) *the Markov chain $(\vartheta^{(k)}, m^{(k)})_{k\geq 1}$ admits the invariant distribution*

$$f(\vartheta, m) = p_\lambda(\vartheta|\mathcal{D}_n)\varphi_{\nabla L_n(\vartheta), \rho_1^2/(1-\beta_1^2)}(m_1) \cdot \varphi_{\nabla L_n(\vartheta)^2, \rho_2^2/(1-\beta_2^2)}(m_2).$$

In particular, the marginal distribution of $f(\vartheta, m)$ in ϑ is the Gibbs posterior distribution $p_\lambda(\cdot|\mathcal{D}_n)$.

Moreover, the following result shows a good approximation of the Gibbs distribution in the presence of a sufficiently small momentum. The special case $\beta = 0$ corresponds to a Metropolis-Hastings-within-Gibbs algorithm where we obtain convergence to the invariant distribution with the typical geometric rate, cf. Jones et al. (2014):

Theorem 2. *Suppose that $L_n(\vartheta)$ and $\nabla L_n(\vartheta)$ are uniformly bounded for $\vartheta \in \Omega$. Choose $q_{1,k}$ and q_2 from* (8) *and* (9)*, respectively. Further, let $m^{(0)} \sim \mathcal{N}\big(\nabla L_n(\vartheta^{(0)}), \rho_1^2/(1-\beta_1^2)I_P\big) \otimes \mathcal{N}\big(\nabla L_n(\vartheta^{(0)})^2, \rho_2^2/(1-\beta_2^2)I_P\big)$, where $\vartheta^{(0)}$ is drawn from an arbitrary distribution with bounded density $f^{(0)}(\vartheta)$ and support Ω. Then, the total variation distance of the distribution of $\vartheta^{(k)}$ to the Gibbs posterior $p_\lambda(\cdot|\mathcal{D}_n)$ satisfies:*

$$\mathrm{TV}(\mathbb{P}^{\vartheta^{(k)}}, \Pi_\lambda(\cdot|\mathcal{D}_n)) \lesssim (1-a)^k + b\beta.$$

for some $a \in (0,1), b > 0$ and $\beta = \beta_1 \vee \beta_2 := \max\{\beta_1, \beta_2\}$.

Note that the geometric decay suffers from an exponential decrease of a with P as typically observed in the convergence analysis of Metropolis-Hastings algorithms.

4 Numerical Experiments

We determine the effects of the algorithm parameters λ, σ, σ_Δ and β_1, β_2 on the generated ensemble of network weights for a high-dimensional classification task from particle physics. In a parallel application on neural posterior estimation using continuous normalizing flows (Bieringer et al., 2024b), we have already found great improvements in indicating out-of-distribution input over the commonly used variational inference-based network weight posterior approximation (Blundell et al., 2015).

In neural network training code, the gradient calculation and optimizer step can be easily exchanged by an `AdamMCMC` step. As the algorithm extends a single `Adam` update step, the M-H step constitutes the main computational bottleneck. Besides the calculation of the loss-values, it only applies highly parallelisable subtractions and vector products.

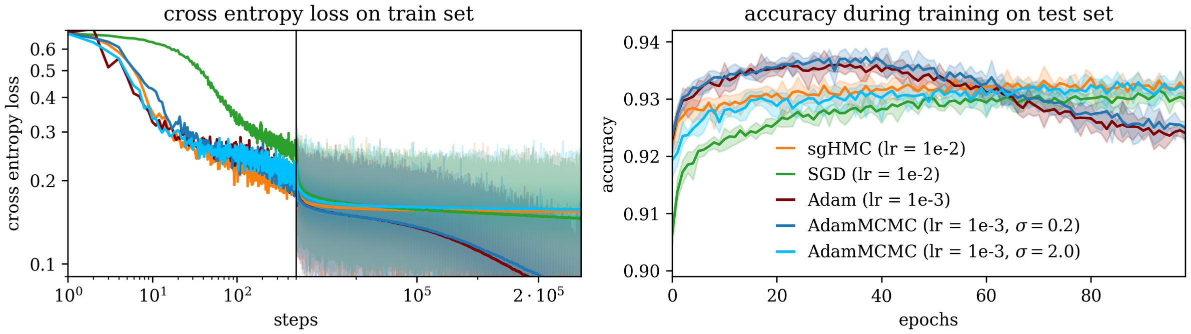


Figure 1: Optimization speed of AdamMCMC in comparison to Adam, SGD and sgHMC for the Top-tagging (binary classification) task. We report the mean curves over 5 independent runs for the highest learning rate that allows stable results from a grid search. Left: Development of the cross entropy loss on the training set during running of the chain/training. To increase the readability, we show log-scaling for the first 500 steps and linear scaling as well as the a moving-average over 2400 steps for the remaining $\approx$ 230k steps. Right: Accuracy on the test set (400k jets) over training epochs (2400 steps per epoch). While AdamMCMC ($\sigma = 0.2$) closely resamples the behavior of Adam including overfitting, the samples generated with AdamMCMC ($\sigma = 2.0$) show no signs of overfitting and a similar optimization performance as sgHMC. The error bands show the $\min$-$\max$ evelope of the 5 runs.

4.1 Top-Tagging and ParticleNet

At the Large Hadron Collider (LHC), products of particle collisions are measured using calorimeters. In these, the generated particles produce a spray of daughter particles each depositing energy in the calorimeter cells. The cascade of measured energy depositions is usually referred to as a particle shower. During evaluation, these measured showers are grouped into jets originating in one initial particle. To reconstruct the correct scattering process in the particle collisions, the reconstructed jets need to be assigned to the correct initial partons.

One particularly useful tool in the investigation of the Higgs particle is a classification of jets originating from Top-quarks from their background originating from lighter quarks (QCD). To ensure a fair comparison between the multiple efforts within the high-energy physics community, Butter et al. (2019) introduce the TopLandscape dataset. It contains 600k top and background jets each for training.

While the Particle Transformer architecture (Qu et al., 2022b) currently reports the best accuracy and rejection rates, we choose the commonly used and more parameter-efficient ParticleNet architecture (Qu & Gouskos, 2020) for our evaluation of AdamMCMC. ParticleNet constructs a graph from the per-jet point cloud of constituents by connecting each constituent to its $k = 16$ nearest neighbours in physical space. It does so for the 128 particles with highest transverse momentum. We follow the original architecture and apply three layers of edge convolutions (Wang et al., 2019) to the graph, dynamically recalculating the neighbours at the beginning of every edge convolution block and transforming the features of the graph with a three-layered perceptron based on its neighbours. The graph layers are followed by a global average pooling layer on the channels of the edge convolutions. After this, two fully connected layers, the first featuring additional dropout of 0.1 and ReLU activation, and a softmax function are applied. In total the employed ParticleNet uses $P = 366160$ parameters on an input dimension of 128 points of 2 coordinates and 7 input features.

The classification is trained using a cross entropy loss, that is the sum of the categorical-$\log$-liklihoods per event. Due to the size of network and dataset, we have to employ stochastic approximations of the loss and its gradient for update and correction steps. While the stochasticity in the update steps is corrected by the M-H correction, the stochasticity of the M-H step allows a remaining bias to the invariant distribution. In Section 4.4, we gauge the effects of this approximation numerically.

Originally, ParticleNet is trained using AdamW and weight decay. To focus on the transition from Adam to AdamMCMC, we omit these technicalities and train with a constant learning rate of 1×10^{-3} and $\beta_1 = \beta_2 = 0.99$ for 100 epochs (2400 batches of size 512 each). This barely slows down convergence and reports comparable accuracy values to the original training schedule reported by Qu & Gouskos (2020).

For an initial setup with $\lambda = 1$, $\sigma = 0.2$ and $\sigma_\Delta = P/100 = 3661.6$ shown in Figure 1, we find the sampling algorithm follows its deterministic counterpart closely for the full optimization. The performance on the 400k test training set indicates overfitting of the model for both algorithms. This is a clear indication for running AdamMCMC

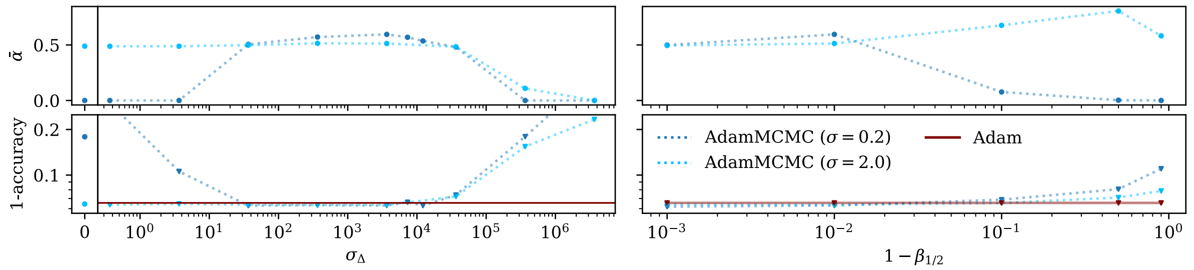


Figure 2: Dependence of the mean acceptance rate (upper) and the accuracy of the posterior mean prediction on test data (lower) on the momentum parameters of the proposal distribution σ_Δ (left) and `Adam` optimizer $\beta_1 = \beta_2$ (right). For low noise runs, we find a strong dependency of the algorithm efficiency on sufficiently large momentum terms. At higher noise, this dependence is reduced due to increased width of the proposal distribution and corresponding higher probability of the backwards direction.

with too little noise, thereby prohibiting any parameter space exploration. We thus increase σ to 2.0, which is the lowest noise setting that allows sufficient parameter space exploration to prevent overfitted samples.

With the higher noise level, `AdamMCMC` converges at similar speed as sgHMC. Both outperform simple stochastic gradient descent (SGD) optimization, which acts as a benchmark for stochastic gradient Langevin Dynamics (Welling & Teh, 2011) based chains, such as MALA.

4.2 Noise, Directional Noise and Momentum

The key novelty of the proposed algorithm is the combination of an prolate proposal distribution with momentum-based optimization. To gauge the effect of these changes, we observe the dependence of the mean acceptance rate during sampling, as well as the accuracy of the ensemble drawn from the approximate posterior, on both the noise in update direction σ_Δ and the momentum parameters β_1 and β_2 in Figure 2. For easier evaluation, we choose $\beta_1 = \beta_2$. To ensure sampling from a converged chain and approximate independence of the samples, we use a burn-in time of $b = 48 \cdot 2400$ steps and a gap length of $c = 5 \cdot 2400$ steps, that is five epochs, in the following. From the 100 epoch runs, we thus generate $N = 10$ weight samples.

We find a strong dependence on both, the directional noise and the momentum parameters, when running the algorithm with low noise of $\sigma = 0.2$. As expected for low directional noise ($\sigma_\Delta < 100$) and low momentum ($\beta_{1/2} < 0.99$), the sampling breaks down due to low acceptance rates. The low acceptance probabilities originate from the low probability of the backwards direction in (10). The accuracy drops accordingly. Using higher directional noise and momentum increases the acceptance probabilities by aligning the proposal distribution with the optimization step.

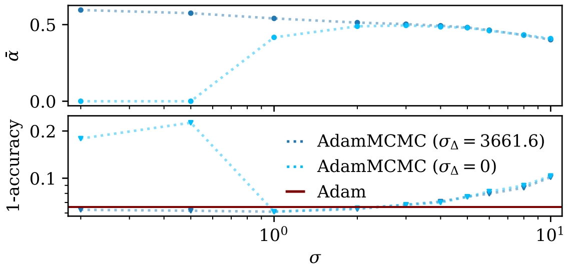


Figure 3: Dependence of the mean acceptance rate (upper) and the accuracy of the posterior mean prediction on test data (lower) on the width of the proposal distribution σ. Without directional noise (light blue), the algorithms efficiency is strongly dependent on a correct choice of σ. Applying an prolate proposal distribution however allows the algorithm to approach the deterministic optimization in the limit of low σ. Noise can then be added to guarantee sufficient parameter space exploration and achieve well-calibrated uncertainties.

For higher noise, $\sigma = 2.0$, the same dependence cannot be observed. Even without directional noise or momentum, stable sampling at good performance can be achieved due to the sufficient spread of the proposal distribution.

High and low noise runs show a strong decrease in acceptance and performance when going to very high directional noise values ($\sigma_\Delta > 10^5$). This decrease is caused by the low likelihood values, that is high NLL-loss values, of network parameters sampled with high variance and thus far away from the currently explored loss minimum. Similar scaling can be observed at very high overall noise ($\sigma > 10$).

Up to this point, the improvements of combining momentum-based optimization with an prolate proposal distribution over sgMALA appear only at low σ. In this region, `AdamMCMC` is prone to overfitting due to limited parameter space exploration. Higher values of σ, in which the effect of the prolate proposal is limited, are to be used regardless. However, in application the extended algorithm is more well-behaved. To show this, we run the sampling chain with and without the adapted proposal for an array of noise values. Figure 3 shows the scaling of the mean acceptance and test accuracy for both.

Running a chain at low noise without directional noise leads to diminishing acceptance probabilities, as does running at high noise. From applying `AdamMCMC` to multiple tasks (Bieringer et al., 2024a;b), we find the range of functional σ-values can vary strongly between different applications, data set sizes and parameter dimensionalities. In a hyperparameter search, it is thus unclear in which direction the noise parameter has to be altered to achieve efficient sampling. When including the directional noise however, low noise parameters result in increasing acceptance and close resemblance between optimizer and MCMC. Starting from this parameter space interval of semi-deterministic optimization, a practitioner can simply increase the noise value until the mode exploration capabilities are sufficient, overfitting is avoided and the desired uncertainty calibration is achieved. This renders fine-tuned learning rate schedules as required for sgHMC unnecessary.

4.3 Error Estimation

The noise parameter σ not only gives a handle on the fitting, but also determines the uncertainty in the predictions. A higher noise value will lead to larger differences between the different weight samples. It thus allows us to adjust the uncertainty quantification to calibrate the predictions.

Figure 4 shows the posterior mean prediction and the posterior spread, as an estimate of the epistemic uncertainty, for different values of σ. We refrain from evaluating at $\sigma < 1$, due to the previously reported issues with limited mode exploration at low noise. The class probability predictions are calculated for the 400k point test set and split into true and false class assignments for both classes, Top- and QCD-jets.

For low noise, we find the posterior mean predictions in the left panel align very well with class predictions from stochastic optimization. Increasing the sampling noise to $\sigma > 7$ leads to the decrease in classification power, that has already been observed in Figure 3. Distinguishing between the two classes allows us attribute this trend to a decrease in predicted probability of Top-jets.

While the performance slightly decreases between $\sigma = 1$ and 7, we find the posterior spread, determined as the distance between the 75%- and 25%-quantile of the weight samples, steadily rises. This is most prominent in the falsely assigned classes. For these, a significant uncertainty is reported at low noise already.

When running the classification on out-of-distribution data from the more comprehensive JetClass dataset (Qu et al., 2022a), we find a largely increased posterior spread. The increase of the uncertainty with increasing noise is reproduced analogously to the in-distribution sample.

From the tempered Gibbs posterior (1), we would expect a similar effect from the inverse temperature λ. We have varied the inverse temperature within $\lambda \in [10^{-4}, 10^3]$ and did not find a strong dependence of the uncertainty prediction on this parameter. Very low values will however lead to strongly suppressed acceptance rates and a corresponding loss of classification performance.

4.4 Stochastic Metropolis-Hastings

To reduce computational cost for our experiments, we have used a stochastic approximation of the M-H correction. That is a correction calculated from an unbiased, batch-based estimator of the full loss. This stochastic correction introduces a bias to the posterior estimation.

To gauge the difference between an algorithm with a full correction to one using only a batch of data, here 512 points, we run both algorithms at the same hyperparameter settings. Due to the immense computational cost of the full correction, we are limited to short chains only. We thus only evaluate the most interesting regions in Figure 5.

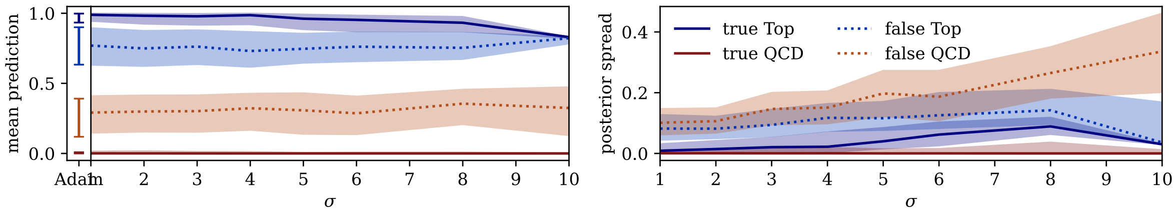


Figure 4: Scaling of the posterior mean prediction (left) and posterior spread (right) with the proposal distribution width σ. The posterior spread is calculated as the difference between the 75%- and 25%-quantile of the prediction for 10 posterior samples. We report individual results for both classes and true and false assignments for a test set of 400k jets. The center line shows the median of the jets in the according category and the envelopes depict the 75%- and 25%-quantile on the data. We find a slight decrease classification performance for rising σ for a large section of the scanned space, while for low noise values the performance of an `Adam` optimization is closely reproduced. The uncertainty prediction increases with increasing σ starting out at an already significant level for $\sigma = 1$.

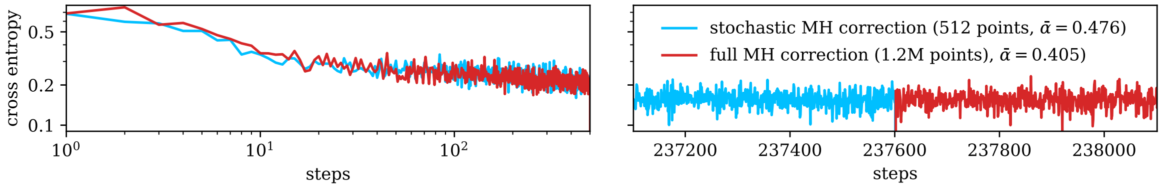


Figure 5: Comparing the loss development for `AdamMCMC` algorithms employing a full Metropolis-Hastings correction, as well as a stochastic approximation thereof from random initialization (left) and the end of the stochastically corrected chain (right). Both chains are calculated at the same hyperparameters, that is $\sigma = 2.0, \sigma_\Delta = 3661.6.\lambda = 1, \beta_{1/2} = 0.99$ and a learning rate of 10^{-3}. We find no significant differences in the dynamics of the chain, although the full correction is slightly more selective.

Starting from the same random initialization, the differences between both algorithms does not seem to exceed the random fluctuations of the proposal and stochasticity of the batches. Both algorithms explore the phase space in the same way.

To gauge the sampling after burn-in, we initialize a second chain with full corrections from the end of the chain employing the stochastic approximation. We find that both, the mean and variance of the loss-values during the chain are virtually the same. A difference in the mean acceptance probability can however be found. As expected, the full correction is more restrictive as its stochastic counterpart.

The additional noise introduced by a stochastic correction, can in part be countered by a reduction of σ and σ_Δ. Stochastic M-H corrections that control the introduced bias can be employed for `AdamMCMC`, whenever the application requires a strict control of the uncertainties. While the corrections of Balan et al. (2014) and Bardenet et al. (2014) rely on subsets of various size to perform sequential hypothesis testing, the minibatch acceptance test of Seita et al. (2018) ensures detailed balance from fixed size batches with an additive correction variable to a Barker test. Zhang et al. (2020) introduce an exact routine for M-H algorithms on subsamples of data from bounds of the difference in the loss through the update. Recent proposals (Bieringer et al., 2023; Kawasaki & Holzmann, 2022) also propose a correction term to the loss to counteract the batch-wise approximation error of the acceptance probabilities.

5 Conclusion

In this report, we proposed a generalization of the Metropolis Adjusted Langevin Algorithm with update steps calculated from `Adam`. We suggested a prolate deformation of the proposal distribution to increase the algorithms acceptance rate. Our construction allows for an efficient calculation of the proposal density which is strictly necessary in order to obtain a computationally feasible algorithm. We have proven that the resulting algorithm admits the desired Gibbs posterior distribution as invariant distribution. While a general convergence result is left open for further research, we have verified that the Gibbs posterior can be well approximated by the algorithm.

For a classification task for particle physics, we show the algorithm works well with stochastic approximation of loss values and gradients. AdamMCMC recovers the behavior of the underlying stochastic optimization, and thereby improves the robustness of the algorithm, at low injected noise. Through changing the width of the proposal distribution, it enables the user to adjust the uncertainty quantification starting out from Adam-like behavior.

A Proofs

In this section, we verify our theoretical contributions. The proof strategies are in line with the literature, see e.g. Chauveau & Vandekerkhove (2002) who have used a step-dependent proposal distribution. However, some technical modifications are necessary to handle noisy momenta.

A.1 Proof of Theorem 1

For brevity, we write $\vartheta = \vartheta^{(k)}, \tilde{\vartheta} = \vartheta^{(k+1)}$ and similarly for m. The overall transition kernel is

$$q_k(\tilde{\vartheta}, \tilde{m}|\vartheta, m) = q_{1,k}(\tilde{\vartheta}|\vartheta, \tilde{m})\alpha(\tilde{\vartheta}|\vartheta, \tilde{m})q_2(\tilde{m}|\vartheta, m) + \Big(\int q_{1,k}(\tau|\vartheta, \tilde{m})(1-\alpha(\tau|\vartheta, \tilde{m}))\,\mathrm{d}\tau\Big)q_2(\tilde{m}|\vartheta, m)\,\delta_\vartheta(\mathrm{d}\tilde{\vartheta}),$$

where the first term corresponds to accepting the proposal for $\tilde{\vartheta}$, the second one rejects the proposal and δ_ϑ denotes the Dirac measure in ϑ. By construction we can rewrite

$$\alpha_k(\tilde{\vartheta}|\vartheta, \tilde{m}) = 1 \wedge \frac{f(\tilde{\vartheta}, \tilde{m})q_{1,k}(\vartheta|\tilde{\vartheta}, \tilde{m})}{f(\vartheta, \tilde{m})q_{1,k}(\tilde{\vartheta}|\vartheta, \tilde{m})}$$

and thus the detailed balance equation

$$\begin{aligned}\alpha_k(\tilde{\vartheta}|\vartheta, \tilde{m})q_{1,k}(\tilde{\vartheta}|\vartheta, \tilde{m})f(\vartheta, \tilde{m}) &= \big(f(\tilde{\vartheta}, \tilde{m})q_{1,k}(\vartheta|\tilde{\vartheta}, \tilde{m})\big) \wedge \big(f(\vartheta, \tilde{m})q_{1,k}(\tilde{\vartheta}|\vartheta, \tilde{m})\big) \\ &= \alpha_k(\vartheta|\tilde{\vartheta}, \tilde{m})q_{1,k}(\vartheta|\tilde{\vartheta}, \tilde{m})f(\tilde{\vartheta}, \tilde{m})\end{aligned} \tag{11}$$

is satisfied. Setting $s_l^2 := \rho_l^2/(1-\beta_l^2)$ we have $s_l^2 = \beta_l^2 s_l^2 + \rho_l^2$ and thus we deduce for two independent standard-normal random vectors $Z_1, Z_2 \in \mathbb{R}^P$ and any $A = A_1 \times A_2$ with measurable $A_1, A_2 \subseteq \mathbb{R}^P$ that

$$\begin{aligned}\int_A \int q_2(\tilde{m}|\vartheta, m)f(\vartheta, m)\,\mathrm{d}m\,\mathrm{d}\tilde{m} &= p_\lambda(\vartheta|\mathcal{D}_n)\int_A \int q_2(\tilde{m}|\vartheta, m)\prod_{l=1}^{2}\varphi_{\nabla L_n(\vartheta)^l, s_l^2}(m_l)\,\mathrm{d}m\,\mathrm{d}\tilde{m} \\ &= p_\lambda(\vartheta|\mathcal{D}_n)\prod_{l=1}^{2}\int_{A_l}\int \varphi_{\beta_l m_l + (1-\beta_l)\nabla L_n(\vartheta)^l, \rho_l^2}(\tilde{m}_l)\varphi_{\nabla L_n(\vartheta)^l, s_l^2}(m_l)\,\mathrm{d}m_l\,\mathrm{d}\tilde{m}_l \\ &= p_\lambda(\vartheta|\mathcal{D}_n)\prod_{l=1}^{2}\mathbb{P}(\beta_l(\nabla L_n(\vartheta)^l + s_l Z_1) + (1-\beta_l)\nabla L_n(\vartheta)^l + \rho_l Z_2 \in A_l) \\ &= p_\lambda(\vartheta|\mathcal{D}_n)\prod_{l=1}^{2}\mathbb{P}(\nabla L_n(\vartheta)^l + \sqrt{\beta_l^2 s_l^2 + \rho_l^2}Z_1 \in A_l) \\ &= \int_A f(\vartheta, \tilde{m})\,\mathrm{d}\tilde{m},\end{aligned} \tag{12}$$

that is $\int q_2(\tilde{m}|\vartheta, m)f(\vartheta, m)\,\mathrm{d}m = f(\vartheta, \tilde{m})$. Therefore,

$$\begin{aligned}&\int\int_A q_k(\tilde{\vartheta}, \tilde{m}|\vartheta, m)f(\vartheta, m)\,\mathrm{d}(\tilde{\vartheta}, \tilde{m})\,\mathrm{d}(\vartheta, m) \\ &\quad= \int\int \mathbf{1}_A(\tilde{\vartheta}, \tilde{m})q_{1,k}(\tilde{\vartheta}|\vartheta, \tilde{m})\alpha_k(\tilde{\vartheta}|\vartheta, \tilde{m})q_2(\tilde{m}|\vartheta, m)f(\vartheta, m)\,\mathrm{d}(\tilde{\vartheta}, \tilde{m})\,\mathrm{d}(\vartheta, m) \\ &\qquad\quad + \int\int \mathbf{1}_A(\vartheta, \tilde{m})q_{1,k}(\tilde{\vartheta}|\vartheta, \tilde{m})\big(1-\alpha_k(\tilde{\vartheta}|\vartheta, \tilde{m})\big)q_2(\tilde{m}|\vartheta, m)f(\vartheta, m)\,\mathrm{d}(\tilde{\vartheta}, \tilde{m})\,\mathrm{d}(\vartheta, m) \\ &\quad= \int\int \mathbf{1}_A(\tilde{\vartheta}, \tilde{m})q_{1,k}(\tilde{\vartheta}|\vartheta, \tilde{m})\alpha_k(\tilde{\vartheta}|\vartheta, \tilde{m})f(\vartheta, \tilde{m})\,\mathrm{d}(\tilde{\vartheta}, \tilde{m})\,\mathrm{d}\vartheta \\ &\qquad\quad + \int\int \mathbf{1}_A(\vartheta, \tilde{m})q_{1,k}(\tilde{\vartheta}|\vartheta, \tilde{m})\big(1-\alpha_k(\tilde{\vartheta}|\vartheta, \tilde{m})\big)f(\vartheta, \tilde{m})\,\mathrm{d}(\tilde{\vartheta}, \tilde{m})\,\mathrm{d}\vartheta\end{aligned}$$

$$
\begin{aligned}
&= \int\int \mathbf{1}_A(\tilde\vartheta,\tilde m) q_{1,k}(\vartheta|\tilde\vartheta,\tilde m)\alpha_k(\vartheta|\tilde\vartheta,\tilde m) f(\tilde\vartheta,\tilde m)\,\mathrm{d}(\tilde\vartheta,\tilde m)\,\mathrm{d}\vartheta \\
&\qquad + \int \mathbf{1}_A(\vartheta,\tilde m)\int f(\vartheta,\tilde m)\,\mathrm{d}(\vartheta,\tilde m) \\
&\qquad - \int\int \mathbf{1}_A(\vartheta,\tilde m) q_{1,k}(\tilde\vartheta|\vartheta,\tilde m)\alpha_k(\tilde\vartheta|\vartheta,\tilde m) f(\vartheta,\tilde m)\,\mathrm{d}(\tilde\vartheta,\tilde m)\,\mathrm{d}\vartheta \\
&= \int \mathbf{1}_A(\vartheta,\tilde m) f(\vartheta,\tilde m)\,\mathrm{d}(\vartheta,\tilde m).
\end{aligned}
$$

Hence, if the distribution of $(\vartheta^{(k)}, m^{(k)})$ is f, then $(\vartheta^{(k+1)}, m^{(k+1)})$ is also distributed according to f which is the claimed stationarity. □

A.2 Proof of Theorem 2

We denote the joint density of $(\vartheta^{(k)}, m^{(k)})$ by $f^k(\vartheta, m)$. The density of marginal distribution of $\vartheta^{(k)}$ is denoted by $f^k(\vartheta)$ and, e.g. the conditional density of $m^{(k)}$ given $\vartheta^{(k)} = \vartheta$ by $f^k(m|\vartheta)$ and similarly for f. Throughout let $\tilde\vartheta, \vartheta \in \Omega, \tilde m, m = (\tilde m_1, \tilde m_2) \in (\mathbb{R}^P)^2$. The proof is organized in five steps.

Step 1: We show that the relative error $D^k(\tilde\vartheta,\tilde m) := \frac{f^k(\tilde\vartheta,\tilde m)}{f(\tilde\vartheta,\tilde m)} - 1$ remains bounded. Define

$$
\tilde f^k(\tilde\vartheta,\tilde m) := \int f^k(\tilde\vartheta, m) q_2(\tilde m|\tilde\vartheta, m)\mathrm{d}m \qquad \text{and} \qquad \tilde D^k(\tilde\vartheta,\tilde m) := \frac{\tilde f^k(\tilde\vartheta,\tilde m)}{f(\tilde\vartheta,\tilde m)} - 1.
$$

With the convention $0/0 = 0$ and the momentum adjusted detailed balance condition (11), we get

$$
\begin{aligned}
f^{k+1}(\tilde\vartheta,\tilde m) &= \int f^k(\vartheta, m) q_{1,k}(\tilde\vartheta|\vartheta,\tilde m)\alpha_k(\tilde\vartheta|\vartheta,\tilde m) q_2(\tilde m|\vartheta, m)\,\mathrm{d}(\vartheta, m) \\
&\qquad + \int f^k(\tilde\vartheta, m) q_{1,k}(\vartheta|\tilde\vartheta,\tilde m)\big(1-\alpha_k(\vartheta|\tilde\vartheta,\tilde m)\big) q_2(\tilde m|\tilde\vartheta, m)\,\mathrm{d}(\vartheta, m) \\
&= \int \tilde f^k(\vartheta,\tilde m) q_{1,k}(\tilde\vartheta|\vartheta,\tilde m)\alpha_k(\tilde\vartheta|\vartheta,\tilde m)\,\mathrm{d}\vartheta + \int \tilde f^k(\tilde\vartheta,\tilde m) q_{1,k}(\vartheta|\tilde\vartheta,\tilde m)\big(1-\alpha_k(\vartheta|\tilde\vartheta,\tilde m)\big)\,\mathrm{d}\vartheta \\
&= \tilde f^k(\tilde\vartheta,\tilde m) + \int f(\vartheta,\tilde m) q_{1,k}(\tilde\vartheta|\vartheta,\tilde m)\alpha_k(\tilde\vartheta|\vartheta,\tilde m)\frac{\tilde f^k(\vartheta,\tilde m)}{f(\vartheta,\tilde m)}\,\mathrm{d}\vartheta \\
&\qquad - \int f(\tilde\vartheta,\tilde m) q_{1,k}(\vartheta|\tilde\vartheta,\tilde m)\alpha_k(\vartheta|\tilde\vartheta,\tilde m)\frac{\tilde f^k(\tilde\vartheta,\tilde m)}{f(\tilde\vartheta,\tilde m)}\,\mathrm{d}\vartheta \\
&= \tilde f^k(\tilde\vartheta,\tilde m) - \int \Big(\frac{\tilde f^k(\tilde\vartheta,\tilde m)}{f(\tilde\vartheta,\tilde m)} - \frac{\tilde f^k(\vartheta,\tilde m)}{f(\vartheta,\tilde m)}\Big) h_k(\vartheta,\tilde\vartheta,\tilde m)\,\mathrm{d}\vartheta, \qquad (13)
\end{aligned}
$$

where

$$
h_k(\vartheta,\tilde\vartheta,\tilde m) := \big(f(\tilde\vartheta,\tilde m) q_{1,k}(\vartheta|\tilde\vartheta,\tilde m)\big) \wedge \big(f(\vartheta,\tilde m) q_{1,k}(\tilde\vartheta|\vartheta,\tilde m)\big).
$$

Setting $Q_k(\tilde\vartheta,\vartheta,\tilde m) := h_k(\vartheta,\tilde\vartheta,\tilde m)/f(\tilde\vartheta,\tilde m)$ this leads to

$$
\begin{aligned}
D^{k+1}(\tilde\vartheta,\tilde m) &= \tilde D^k(\tilde\vartheta,\tilde m) - \int \Big(\frac{\tilde f^k(\tilde\vartheta,\tilde m)}{f(\tilde\vartheta,\tilde m)} - 1\Big)\frac{h_k(\vartheta,\tilde\vartheta,\tilde m)}{f(\tilde\vartheta,\tilde m)}\,\mathrm{d}\vartheta + \int \Big(\frac{\tilde f^k(\vartheta,\tilde m)}{f(\vartheta,\tilde m)} - 1\Big)\frac{h_k(\vartheta,\tilde\vartheta,\tilde m)}{f(\tilde\vartheta,\tilde m)}\,\mathrm{d}\vartheta \\
&= \tilde D^k(\tilde\vartheta,\tilde m) - \int \tilde D^k(\tilde\vartheta,\tilde m) Q_k(\tilde\vartheta,\vartheta,\tilde m)\,\mathrm{d}\vartheta + \int \tilde D^k(\vartheta,\tilde m) Q_k(\tilde\vartheta,\vartheta,\tilde m)\,\mathrm{d}\vartheta \\
&= \tilde D^k(\tilde\vartheta,\tilde m)\Big(1 - \int Q_k(\tilde\vartheta,\vartheta,\tilde m)\,\mathrm{d}\vartheta\Big) + \int \tilde D^k(\vartheta,\tilde m) Q_k(\tilde\vartheta,\vartheta,\tilde m)\,\mathrm{d}\vartheta.
\end{aligned}
$$

Since $\int Q_k(\tilde\vartheta,\vartheta,\tilde m)\,\mathrm{d}\vartheta \le 1$, we conclude

$$
D^{k+1}(\tilde\vartheta,\tilde m) \le \|\tilde D^k\|_\infty - \int \big(\|\tilde D^k\|_\infty - \tilde D^k(\vartheta,\tilde m)\big) Q_k(\tilde\vartheta,\vartheta,\tilde m)\,\mathrm{d}\vartheta \le \|\tilde D^k\|_\infty. \qquad (14)
$$

Since (12) yields

$$\|\tilde{D}^k\|_\infty = \sup_{\tilde\vartheta,\tilde m}\Big|\frac{\int (f^k(\tilde\vartheta,m)-f(\tilde\vartheta,m))q_2(\tilde m|\tilde\vartheta,m)\,\mathrm{d}m}{f(\tilde\vartheta,\tilde m)}\Big| = \sup_{\tilde\vartheta,\tilde m}\Big|\frac{\int D^k(\tilde\vartheta,m)f(\tilde\vartheta,m)q_2(\tilde m|\tilde\vartheta,m)\,\mathrm{d}m}{f(\tilde\vartheta,\tilde m)}\Big| \le \|D^k\|_\infty,$$

we obtain for all $k\in\mathbb{N}$

$$\|D^{k+1}\|_\infty \le \|D^k\|_\infty \le \|D^0\|_\infty.$$

Step 2: We now study the relative error of the marginal distribution of ϑ. To this end we write

$$F^{k+1}(\vartheta) := \frac{\Delta f^{k+1}(\vartheta)}{f(\vartheta)}, \quad \Delta f^{k+1}(\vartheta) := f^{k+1}(\vartheta) - f(\vartheta), \quad \Delta f^{k+1}(\vartheta,m) := f^{k+1}(\vartheta,m) - f(\vartheta,m),$$
$$\Delta\tilde f^{k+1}(\vartheta,m) := \tilde f^{k+1}(\vartheta,m) - f(\vartheta,m).$$

To bound $\|F^{k+1}\|_\infty$, we start similarly to Step 1. Using (13), we have

$$\begin{aligned}
\Delta f^{k+1}(\tilde\vartheta,\tilde m) &= \Delta f^k(\tilde\vartheta)f(\tilde m|\tilde\vartheta) - \int \Big(\frac{\Delta f^k(\tilde\vartheta)f(\tilde m|\tilde\vartheta)}{f(\tilde\vartheta,\tilde m)} - \frac{\Delta f^k(\vartheta)f(\tilde m|\vartheta)}{f(\vartheta,\tilde m)}\Big)h_k(\vartheta,\tilde\vartheta,\tilde m)\,\mathrm{d}\vartheta \\
&\quad + \big(\Delta\tilde f^k(\tilde\vartheta,\tilde m) - \Delta f^k(\tilde\vartheta)f(\tilde m|\tilde\vartheta)\big) \\
&\quad - \int \Big(\frac{\Delta\tilde f^k(\tilde\vartheta,\tilde m) - \Delta f^k(\tilde\vartheta)f(\tilde m|\tilde\vartheta)}{f(\tilde\vartheta,\tilde m)} - \frac{\Delta\tilde f^k(\vartheta,\tilde m) - \Delta f^k(\vartheta)f(\tilde m|\vartheta)}{f(\vartheta,\tilde m)}\Big)h_k(\vartheta,\tilde\vartheta,\tilde m)\,\mathrm{d}\vartheta \\
&=: T_1(\tilde\vartheta,\tilde m) + T_2^k(\tilde\vartheta,\tilde m)
\end{aligned}$$

with

$$\begin{aligned}
T_1(\tilde\vartheta,\tilde m) &= \Delta f^k(\tilde\vartheta)f(\tilde m|\tilde\vartheta) - \int \Big(\frac{\Delta f^k(\tilde\vartheta)f(\tilde m|\tilde\vartheta)}{f(\tilde\vartheta,\tilde m)} - \frac{\Delta f^k(\vartheta)f(\tilde m|\vartheta)}{f(\vartheta,\tilde m)}\Big)h_k(\vartheta,\tilde\vartheta,\tilde m)\,\mathrm{d}\vartheta \\
&= \Delta f^k(\tilde\vartheta)f(\tilde m|\tilde\vartheta) - \int \Big(\frac{\Delta f^k(\tilde\vartheta)}{f(\tilde\vartheta)} - \frac{\Delta f^k(\vartheta)}{f(\vartheta)}\Big)h_k(\vartheta,\tilde\vartheta,\tilde m)\,\mathrm{d}\vartheta.
\end{aligned}$$

Hence,

$$F^{k+1}(\tilde\vartheta) = \frac{1}{f(\tilde\vartheta)}\int \Delta f^{k+1}(\tilde\vartheta,\tilde m)\,\mathrm{d}\tilde m = F^k(\tilde\vartheta) - \int \big(F^k(\tilde\vartheta) - F^k(\vartheta)\big)\tilde Q_k(\tilde\vartheta,\vartheta)\,\mathrm{d}\vartheta + R_k(\tilde\vartheta)$$

with $\tilde Q_k(\tilde\vartheta,\vartheta) := \int \frac{h_k(\vartheta,\tilde\vartheta,\tilde m)}{f(\tilde\vartheta)}\,\mathrm{d}\tilde m$ and $R_k(\tilde\vartheta) := \frac{1}{f(\tilde\vartheta)}\int T_2^k(\tilde\vartheta,\tilde m)\,\mathrm{d}\tilde m$.

Step 3: We first bound the main term in the above display. To this end we verify the momentum adjusted Doeblin-type condition

$$\tilde Q_k(\tilde\vartheta,\vartheta) \ge a f(\vartheta), \qquad \forall \tilde\vartheta,\vartheta\in\Omega \tag{15}$$

for some $a>0$.

It holds that

$$\begin{aligned}
\tilde Q_k(\tilde\vartheta,\vartheta) = \int \frac{h_k(\vartheta,\tilde\vartheta,\tilde m)}{f(\tilde\vartheta)}\,\mathrm{d}\tilde m &= \int \frac{1}{f(\tilde\vartheta)}\Big(\big(q_{1,k}(\vartheta|\tilde\vartheta,\tilde m)f(\tilde\vartheta,\tilde m)\big)\wedge\big(f(\vartheta,\tilde m)q_{1,k}(\tilde\vartheta|\vartheta,\tilde m)\big)\Big)\,\mathrm{d}\tilde m \\
&= \int \big(q_{1,k}(\vartheta|\tilde\vartheta,\tilde m)f(\tilde m|\tilde\vartheta)\big)\wedge\big(\tfrac{f(\vartheta)}{f(\tilde\vartheta)}f(\tilde m|\vartheta)q_{1,k}(\tilde\vartheta|\vartheta,\tilde m)\big)\,\mathrm{d}\tilde m.
\end{aligned}$$

Since Ω is bounded, $\det(\Sigma_k)$ as calculated in (5) is uniformly bounded from above and away from 0 and since $\Sigma_k^{-1}\le C_1$ in the ordering of positive definite matrices for some $C_1>0$, we have

$$\begin{aligned}
q_{1,k}(\vartheta|\tilde\vartheta,\tilde m)f(\tilde m|\tilde\vartheta) &= \varphi_{\tilde\vartheta-u_k(\tilde m),\Sigma_k}(\vartheta)\varphi_{\nabla L_n(\tilde\vartheta),\rho_1^2/(1-\beta_1^2)}(\tilde m_1)\varphi_{\nabla L_n(\tilde\vartheta)^2,\rho_2^2/(1-\beta_2^2)}(\tilde m_2) \\
&\gtrsim \exp(-C_1|u_k(\tilde m)|^2)\varphi_{\nabla L_n(\tilde\vartheta),\rho_1^2/(1-\beta_1^2)}(\tilde m_1)\varphi_{\nabla L_n(\tilde\vartheta)^2,\rho_2^2/(1-\beta_2^2)}(\tilde m_2) \\
&\ge \exp(-\tilde C_1|\tilde m_1|^2)\varphi_{\nabla L_n(\tilde\vartheta),\rho_1^2/(1-\beta_1^2)}(\tilde m_1)\varphi_{\nabla L_n(\tilde\vartheta)^2,\rho_2^2/(1-\beta_2^2)}(\tilde m_2).
\end{aligned}$$

With an analogous bound for $\frac{f(\vartheta,\tilde{m})}{f(\tilde{\vartheta})}q_{1,k}(\tilde{\vartheta}|\vartheta,\tilde{m})$, we obtain for some constant $\tilde{C}>0$

$$\begin{aligned}\tilde{Q}_k(\tilde{\vartheta},\vartheta) &\gtrsim \int \exp(-\tilde{C}|\tilde{m}_1|^2)\varphi_{\nabla L_n(\tilde{\vartheta}),\rho_1^2/(1-\beta_1^2)}(\tilde{m}_1)\varphi_{\nabla L_n(\tilde{\vartheta})^2,\rho_2^2/(1-\beta_2^2)}(\tilde{m}_2)\,\mathrm{d}\tilde{m}\\ &= \int \exp(-\tilde{C}|\tilde{m}_1|^2)\varphi_{\nabla L_n(\tilde{\vartheta}),\rho_1^2/(1-\beta_1^2)}(\tilde{m}_1)\,\mathrm{d}\tilde{m}_1. \qquad (16)\end{aligned}$$

This integral is equal to $\mathbb{E}[\exp(-|Z|^2)]$ for $Z=(Z_1,\dots,Z_P)\sim\mathcal{N}\big(\mu,\tilde{\sigma}^2 I_P\big)$ with $\mu=(\mu_1,\dots,\mu_P)^\top=\sqrt{\tilde{C}}\nabla L_n(\tilde{\vartheta})$ and $\tilde{\sigma}^2=\tilde{C}\rho_1^2/(1-\beta_1^2)$. It holds that

$$\begin{aligned}\mathbb{E}[\exp(-|Z|^2)] = \prod_{i=1}^P \mathbb{E}[\exp(-|Z_i|^2)] &= \frac{1}{(2\pi\tilde{\sigma}^2)^{P/2}}\prod_{i=1}^P\int \exp\big(-|z_i|^2\big)\exp\Big(-\frac{1}{2\tilde{\sigma}^2}(z_i-\mu_i)^2\Big)\,\mathrm{d}z_i\\ &= \frac{1}{(2\tilde{\sigma}^2+1)^{P/2}}\exp\Big(-\frac{|\mu|^2}{2\tilde{\sigma}^2+1}\Big). \qquad (17)\end{aligned}$$

Noting that $|\mu|$ is bounded, $\tilde{\sigma}^2$ is bounded and bounded away from 0 and f is bounded and bounded away from 0 on its support, (16) and (17) yield (15).

Note that $\int \tilde{Q}_k(\tilde{\vartheta},\vartheta)\,\mathrm{d}\vartheta\le 1$. Therefore, we can conclude from (15) that

$$\begin{aligned}F^{k+1}(\tilde{\vartheta}) &= F^k(\tilde{\vartheta})\Big(1-\int \tilde{Q}_k(\tilde{\vartheta},\vartheta)\,\mathrm{d}\vartheta\Big)+\int F^k(\vartheta)\tilde{Q}_k(\tilde{\vartheta},\vartheta)\,\mathrm{d}\vartheta+R_k(\tilde{\vartheta})\\ &\le \|F^k\|_\infty-\int\big(\|F^k\|_\infty-F^k(\vartheta)\big)\tilde{Q}_k(\tilde{\vartheta},\vartheta)\,\mathrm{d}\vartheta+R_k(\tilde{\vartheta})\\ &\le \|F^k\|_\infty-a\int\big(\|F^k\|_\infty-F^k(\vartheta)\big)f(\vartheta)\,\mathrm{d}\vartheta+R_k(\tilde{\vartheta})\\ &= (1-a)\|F^k\|_\infty+a\int(f^k(\vartheta)-f(\vartheta))\,\mathrm{d}\vartheta+R_k(\tilde{\vartheta})\\ &= (1-a)\|F^{k-1}\|_\infty+R_k(\tilde{\vartheta}).\end{aligned}$$

With an analogous bound for $-F^{k+1}$ we obtain

$$|F^{k+1}(\tilde{\vartheta})|\le(1-a)\|F^{k-1}\|_\infty+|R_k(\tilde{\vartheta})|.$$

Step 4: Finally we bound the remainder R_k. Recall that $\Delta f^k(\vartheta,m):=f^k(\vartheta,m)-f(\vartheta,m)$ and, similarly, write $\Delta f^k(m):=f^k(m)-f(m)$. Using (12), we have

$$\begin{aligned}\big|\Delta\tilde{f}^k(\vartheta,\tilde{m})-\Delta f^k(\vartheta)f(\tilde{m}|\vartheta)\big| &= \Big|\int \Delta f^k(\vartheta,m)\big(q_2(\tilde{m}|\vartheta,m)-f(\tilde{m}|\vartheta)\big)\,\mathrm{d}m\Big|\\ &= \Big|\int D^k(\vartheta,m)f(\vartheta,m)\Phi(\vartheta,\tilde{m},m)\,\mathrm{d}m\Big|\\ &\le \|D^k\|_\infty\int f(\vartheta,m)|\Phi(\vartheta,\tilde{m},m)|\,\mathrm{d}m \qquad (18)\end{aligned}$$

with

$$\Phi(\vartheta,\tilde{m},m):=q_2(\tilde{m}|\vartheta,m)-f(\tilde{m}|\vartheta)=\prod_{l=1}^2\varphi_{\beta_l,m_l+(1-\beta_l)\nabla L(\vartheta)^l,\rho_l^2}(\tilde{m}_l)-\prod_{l=1}^2\varphi_{\nabla L(\vartheta)^l,\rho_l^2/(1-\beta_l^2)}(\tilde{m}_l).$$

Therefore, we can use $h_k(\vartheta,\tilde{\vartheta},\tilde{m})/f(\vartheta,\tilde{m})\le q_{1,k}(\tilde{\vartheta}|\vartheta,\tilde{m})$ to obtain

$$\begin{aligned}|R_k(\tilde{\vartheta})| &\le \frac{1}{f(\tilde{\vartheta})}\int\big|\Delta\tilde{f}^k(\tilde{\vartheta},\tilde{m})-\Delta f^k(\tilde{\vartheta})f(\tilde{m}|\tilde{\vartheta})\big|\Big(1+\int\frac{h_k(\vartheta,\tilde{\vartheta},\tilde{m})}{f(\tilde{\vartheta},\tilde{m})}\,\mathrm{d}\vartheta\Big)\,\mathrm{d}\tilde{m}\\ &\qquad+\frac{1}{f(\tilde{\vartheta})}\iint\big|\Delta\tilde{f}^k(\vartheta,\tilde{m})-\Delta f^k(\vartheta)f(\tilde{m}|\vartheta)\big|\frac{h_k(\vartheta,\tilde{\vartheta},\tilde{m})}{f(\vartheta,\tilde{m})}\,\mathrm{d}\vartheta\,\mathrm{d}\tilde{m}\\ &\le \|D^k\|_\infty\Big(\frac{2}{f(\tilde{\vartheta})}\iint f(\tilde{\vartheta},m)|\Phi(\tilde{\vartheta},\tilde{m},m)|\,\mathrm{d}m\,\mathrm{d}\tilde{m}\end{aligned}$$

$$
\begin{aligned}
&\qquad + \frac{1}{f(\tilde{\vartheta})} \iiint f(\vartheta, m) |\Phi(\vartheta, \tilde{m}, m)| q_{1,k}(\tilde{\vartheta}|\vartheta, \tilde{m}) \, \mathrm{d}m \, \mathrm{d}\tilde{m} \, \mathrm{d}\vartheta \Big) \\
&=: C(\tilde{\vartheta}, \beta) \|D^k\|_\infty.
\end{aligned}
$$

From the explicit form of Φ and the boundedness of $f(\tilde{\vartheta})$ away from 0, we can now derive a constant $b > 0$ such that

$$
C(\tilde{\vartheta}, \beta) \leq b\beta \tag{19}
$$

for all $\tilde{\vartheta} \in \Omega$.

To this end, we employ Proposition 2.1 in Devroye et al. (2018) to obtain

$$
\begin{aligned}
\int |\Phi(\tilde{\vartheta}, \tilde{m}, m)| \, \mathrm{d}\tilde{m} &= 2\mathrm{TV}\Big(\bigotimes_{l=1}^{2} \mathcal{N}\big(\beta_l m_l + (1 - \beta_l)\nabla L(\tilde{\vartheta})^l, \rho_l^2 I_P\big), \bigotimes_{l=1}^{2} \mathcal{N}\big(\nabla L(\tilde{\vartheta})^l\big), \rho_l^2/(1 - \beta_l^2) I_P)\big)\Big) \\
&\leq \Big(\sum_{l=1}^{2} \beta_l \rho_l^{-1} |m_l - \nabla L_n(\tilde{\vartheta})^l|^2 + \sqrt{P}\big(\beta_l/(1 - \beta_l^2) + \big(\log(1 - \beta_l^2)\big)\big)\Big)^{1/2}.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&\int \int f(\tilde{\vartheta}, m) |\Phi(\tilde{\vartheta}, \tilde{m}, m)| \, \mathrm{d}m \, \mathrm{d}\tilde{m} \\
&= \int f(\tilde{\vartheta}, m) \Big(\int |\Phi(\tilde{\vartheta}, \tilde{m}, m)| \, \mathrm{d}\tilde{m} \Big) \mathrm{d}m \\
&\leq \sum_{l=1}^{2} \Big(\beta_l \rho_l^{-1} \int |m_l - \nabla L_n(\tilde{\vartheta})^l| f(\tilde{\vartheta}, m) \, \mathrm{d}m + \sqrt{P}\big(\beta_l/(1 - \beta_l^2)^{1/2} + \big|\log(1 - \beta_l)\big|^{1/2}\big) \int f(\tilde{\vartheta}, m) \, \mathrm{d}m \Big) \\
&= f(\tilde{\vartheta}) \sum_{l=1}^{2} \Big(\beta_l \rho_l^{-1} \int |m_l - \nabla L_n(\tilde{\vartheta})^l| \varphi_{\nabla L_n(\tilde{\vartheta}), \rho_1^2/(1-\beta_1^2)}(m_1) \varphi_{\nabla L_n(\tilde{\vartheta})^2, \rho_2^2/(1-\beta_2^2)}(m_2) \\
&\qquad\qquad + \sqrt{P}\big(\beta_l/(1 - \beta_l^2)^{1/2} + \big|\log(1 - \beta_l)\big|^{1/2}\big)\Big) \\
&\leq f(\tilde{\vartheta}) \sum_{l=1}^{2} \Big(\beta_l \rho_l^{-1} \int |m_l - \nabla L_n(\tilde{\vartheta})^l| \varphi_{\nabla L_n(\tilde{\vartheta})^l, \rho_l^2/(1-\beta_l^2)}(m_l) \, \mathrm{d}m_l \\
&\qquad\qquad + \sqrt{P}\big(\beta_l/(1 - \beta_l^2)^{1/2} + \big|\log(1 - \beta_l)\big|^{1/2}\big)\Big).
\end{aligned}
$$

Note that for fixed $l = 1, 2$, the integral term in the last display is equal to $\mathbb{E}[|Z|]$ with $Z \sim \mathcal{N}\big(0, (\rho_l^2/(1 - \beta_l^2) I_P\big)$. It holds that

$$
\mathbb{E}[|Z|] \leq \mathbb{E}[|Z|_1] \leq \frac{P\rho_l}{(1 - \beta_l^2)^{1/2}}.
$$

Hence,

$$
\begin{aligned}
\int \int f(\tilde{\vartheta}, m) |\Phi(\tilde{\vartheta}, \tilde{m}, m)| \, \mathrm{d}m \, \mathrm{d}\tilde{m} &\leq f(\tilde{\vartheta}) \sum_{l=1}^{2} \Big(\frac{P\beta_l}{(1 - \beta_l^2)^{1/2}} + \sqrt{P}\big(\beta_l/(1 - \beta_l^2)^{1/2} + \big|\log(1 - \beta_l^2)\big|^{1/2}\big)\Big) \\
&=: f(\tilde{\vartheta}) C(\beta).
\end{aligned} \tag{20}
$$

Further, note that $q_{1,k}(\tilde{\vartheta}|\vartheta, \tilde{m}) \leq (2\pi\sigma)^{-P/2}$. Similarly to (20), we have

$$
\begin{aligned}
\int \int \int f(\vartheta, m) |\Phi(\vartheta, \tilde{m}, m)| q_{1,k}(\tilde{\vartheta}|\vartheta, \tilde{m}) \, \mathrm{d}m \, \mathrm{d}\tilde{m} \, \mathrm{d}\vartheta &\leq (2\pi\sigma)^{-P/2} \int \Big(\int f(\vartheta, m) \int |\Phi(\vartheta, \tilde{m}, m)| \, \mathrm{d}\tilde{m} \, \mathrm{d}m \Big) \, \mathrm{d}\vartheta \\
&= (2\pi\sigma)^{-P/2} C(\beta) \int f(\vartheta) \, \mathrm{d}\vartheta \\
&= (2\pi\sigma)^{-P/2} C(\beta).
\end{aligned}
$$

Overall, (19) is verified, since f is bounded away from 0 on its support and $C(\beta) \leq b'\beta$ for some $b' > 0$.

Step 5: Putting everything together, we obtain

$$\|F^{k+1}\|_\infty \leq (1-a)\|F^k\|_\infty + b\|D^0\|_\infty\beta \leq (1-a)^{k+1}\|F^0\|_\infty + b\|D^0\|_\infty\beta\sum_{l=0}^{k}(1-a)^l$$
$$\leq (1-a)^{k+1}\|F^0\|_\infty + \frac{b}{a}\beta\|D^0\|_\infty.$$

Since our choice of the distribution for initializing the chain implies that F^0, D^0 are bounded, the proof is complete. □

Acknowledgments

SB is supported by the Helmholtz Information and Data Science Schools via DASHH (Data Science in Hamburg - HELMHOLTZ Graduate School for the Structure of Matter) with the grant HIDSS-0002. SB and GK acknowledge support by the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy – EXC 2121 Quantum Universe – 390833306. This research was supported in part through the Maxwell computational resources operated at Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany. MS and MT acknowledge support by the DFG through project TR 1349/3-1.

Code

The code for running `AdamMCMC` in general, as well es the presented experiments is provided at `https://github.com/sbieringer/AdamMCMC`

References

Alquier, P. (2021). User-friendly introduction to PAC-Bayes bounds.

Balan, A. K., Chen, Y., & Welling, M. (2014). Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In *Proceedings of the 31th International Conference on Machine Learning*, volume 32 of *JMLR Workshop and Conference Proceedings* (pp. 181–189).

Bardenet, R., Doucet, A., & Holmes, C. C. (2014). Towards scaling up Markov chain Monte Carlo: An adaptive subsampling approach. In *Proceedings of the 31th International Conference on Machine Learning*, volume 32 of *JMLR Workshop and Conference Proceedings* (pp. 405–413).

Bieringer, S., Diefenbacher, S., Kasieczka, G., & Trabs, M. (2024a). Calibrating Bayesian generative machine learning for Bayesiamplification. *Mach. Learn. Sci. Tech.*, 5(4), 045044.

Bieringer, S., Kasieczka, G., Kieseler, J., & Trabs, M. (2024b). Classifier surrogates: sharing AI-based searches with the world. *Eur. Phys. J. C*, 84(9), 972.

Bieringer, S., Kasieczka, G., Steffen, M. F., & Trabs, M. (2023). Statistical guarantees for stochastic Metropolis-Hastings.

Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural network. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research* (pp. 1613–1622).: PMLR.

Butter, A. et al. (2019). The Machine Learning landscape of top taggers. *SciPost Phys.*, 7, 014.

Chauveau, D. & Vandekerkhove, P. (2002). Improving convergence of the Hastings-Metropolis algorithm with an adaptive proposal. *Scandinavian Journal of Statistics*, 29(1), 13–29.

Chen, C., Carlson, D. E., Gan, Z., Li, C., & Carin, L. (2016). Bridging the gap between stochastic gradient MCMC and stochastic optimization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *JMLR Workshop and Conference Proceedings* (pp. 1051–1060).

Chen, T., Fox, E., & Guestrin, C. (2014). Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *JMLR Workshop and Conference Proceedings* (pp. 1683–1691).

Daxberger, E. A., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., & Hennig, P. (2021). Laplace redux - effortless bayesian deep learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021* (pp. 20089–20103).

Devroye, L., Mehrabian, A., & Reddad, T. (2018). The total variation distance between high-dimensional Gaussians with the same mean.

Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., & Neven, H. (2014). Bayesian sampling using stochastic gradient thermostats. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014* (pp. 3203–3211).

Franssen, S. & Szabó, B. (2022). Uncertainty quantification for nonparametric regression using empirical Bayesian neural networks.

Gal, Y. (2016). *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge.

Heek, J. & Kalchbrenner, N. (2019). Bayesian inference for large scale image classification.

Jones, G. L., Roberts, G. O., & Rosenthal, J. S. (2014). Convergence of conditional Metropolis-Hastings samplers. *Advances in Applied Probability*, 46(2), 422 – 445.

Karagiorgi, G., Kasieczka, G., Kravitz, S., Nachman, B., & Shih, D. (2022). Machine learning in the search for new fundamental physics. *Nature Reviews Physics*, 4(6), 399–412.

Kawasaki, E. & Holzmann, M. (2022). Data subsampling for Bayesian neural networks.

Kingma, D. P. & Ba, J. (2015). Adam: A method for stochastic optimization. In *The 3rd International Conference on Learning Representations*.

Li, C., Chen, C., Carlson, D. E., & Carin, L. (2016). Preconditioned stochastic gradient Langevin dynamics for deep neural networks. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence* (pp. 1788–1794).: AAAI Press.

Ludkin, M. & Sherlock, C. (2023). Hug and hop: a discrete-time, nonreversible markov chain monte carlo algorithm. *Biometrika*, 110(2), 301–318.

Ma, Y., Chen, T., & Fox, E. B. (2015). A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015* (pp. 2917–2925).

Patterson, S. & Teh, Y. W. (2013). Stochastic gradient riemannian langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems 26: Annual Conference on Neural Information Processing Systems 2013* (pp. 3102–3110).

Qu, H. & Gouskos, L. (2020). ParticleNet: Jet Tagging via Particle Clouds. *Phys. Rev. D*, 101(5), 056019.

Qu, H., Li, C., & Qian, S. (2022a). Jetclass: A large-scale dataset for deep learning in jet physics.

Qu, H., Li, C., & Qian, S. (2022b). Particle Transformer for Jet Tagging.

Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., & Köthe, U. (2020). Bayesflow: Learning complex stochastic models with invertible neural networks. *IEEE transactions on neural networks and learning systems*, 33(4), 1452–1466.

Ritter, H., Botev, A., & Barber, D. (2018). A scalable Laplace approximation for neural networks. In *6th International Conference on Learning Representations*.

Robert, C. P. & Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer, second edition.

Roberts, G. O. & Tweedie, R. L. (1996a). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4), 341–363.

Roberts, G. O. & Tweedie, R. L. (1996b). Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika*, 83(1), 95–110.

Seita, D., Pan, X., Chen, H., & Canny, J. F. (2018). An efficient minibatch acceptance test for Metropolis-Hastings. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (pp. 5359–5363).

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. (2019). Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5), 1–12.

Welling, M. & Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning* (pp. 681–688).: Omnipress.

Xifara, T., Sherlock, C., Livingstone, S., Byrne, S., & Girolami, M. (2014). Langevin diffusions and the metropolis-adjusted langevin algorithm. *Statistics & Probability Letters*, 91, 14–19.

Zhang, R., Cooper, A. F., & Sa, C. D. (2020). Asymptotically optimal exact minibatch metropolis-hastings. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*.

Statistical guarantees for stochastic Metropolis-Hastings

Sebastian Bieringer[1,*], Gregor Kasieczka[1,*], Maximilian F. Steffen[2,*] and Mathias Trabs[2,*]

[1]Universität Hamburg and [2]Karlsruhe Institute of Technology

Abstract

A Metropolis-Hastings step is widely used for gradient-based Markov chain Monte Carlo methods in uncertainty quantification. By calculating acceptance probabilities on batches, a stochastic Metropolis-Hastings step saves computational costs, but reduces the effective sample size. We show that this obstacle can be avoided by a simple correction term. We study statistical properties of the resulting stationary distribution of the chain if the corrected stochastic Metropolis-Hastings approach is applied to sample from a Gibbs posterior distribution in a nonparametric regression setting. Focusing on deep neural network regression, we prove a PAC-Bayes oracle inequality which yields optimal contraction rates and we analyze the diameter and show high coverage probability of the resulting credible sets. With a numerical example in a high-dimensional parameter space, we illustrate that credible sets and contraction rates of the stochastic Metropolis-Hastings algorithm indeed behave similar to those obtained from the classical Metropolis-adjusted Langevin algorithm.

Keywords: Stochastic neural network, optimal contraction rate, credible sets, oracle inequality, uncertainty quantification

MSC 2020: 68T07, 62F15, 62G08, 68T37

1 Introduction

An essential feature in modern data science, especially in machine learning as well as high-dimensional statistics, are large sample sizes and large parameter space dimensions. As a consequence, the design of methods for uncertainty quantification is characterized by a tension between numerically feasible and efficient algorithms and approaches which satisfy theoretically justified statistical properties. In this work we demonstrate that a Bayesian MCMC-based method with a stochastic Metropolis-Hastings step achieves both: It is scalable, i.e., it is computationally feasible for large samples, and we can prove an optimal bound for the prediction risk as well as uncertainty statements for the underlying posterior distribution.

Bayesian methods enjoy high popularity for quantifying uncertainties in complex models. The classical approach to sample from the posterior distribution are Markov Chain Monte Carlo methods (MCMC). For large parameter spaces gradient-based Monte Carlo methods are particularly useful, with e.g. Langevin dynamics serving as a prototypical example. State-of-the-art methods such as Metropolis adjusted Langevin (MALA) [11, 47] and Hamiltonian Monte Carlo [24, 41] equip a Metropolis-Hastings (MH) step to accept or reject the proposed next state of the chain. From the practical point of view, the MH step improves robustness with respect to the choice of the tuning parameters and in theory MH speeds up the convergence of the Markov chain.

[*]The authors would like to thank Botond Szabó and two anonymous referees for helpful comments. SB is supported by DASHH (Data Science in Hamburg - HELMHOLTZ Graduate School for the Structure of Matter) with the grant HIDSS-0002. SB and GK acknowledge support by the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy - EXC 2121 Quantum Universe - 390833306. MS and MT acknowledge support by the DFG through project TR 1349/3-1. The empirical studies were enabled by the Maxwell computational resources operated at Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany.

If the sample size is large, the computational costs of gradient-based MCMC methods can be reduced by replacing the gradient of the full loss over all observations by a stochastic gradient. This is standard in empirical risk minimization and has been successfully applied for Langevin dynamics as well [1, 36, 43, 55]. In this case, the MH steps remain as a computational bottleneck: Since the target distribution depends on the full dataset, we have to compute the loss on the full sample to calculate the acceptance probabilities. Among the approaches to circumvent this problem, see Bardenet et al. [9] for a review, a *stochastic MH* step is presumably the most natural one. There, the full loss in the acceptance probability is replaced by a (mini-)batch approximation which reduces the computational cost considerably.

Bardenet et al. [9, Section 6.1] have argued heuristically that the naive stochastic MH step reduces the effective sample size, which determines, for instance, contraction rates of the posterior distribution, to the size of the batch. To rigorously understand the statistical consequences of a stochastic MH step, we apply the pseudo-marginal Metropolis-Hastings perspective by Andrieu & Roberts [4] and Maclaurin & Adams [37]. It turns out that a Markov Chain with a stochastic MH step does not converge to the original target posterior distribution, but a different distribution, which we call *surrogate posterior* and whose statistical performance is indeed determined to the batch size only. However, we show that there is a simple correction term in the risk such that the resulting stochastic MH chain converges to a surrogate posterior which achieves the full statistical power in terms of optimal contraction rates.

In a nonparametric regression problem, we investigate the distance of the surrogate posteriors associated to the stochastic MH algorithm and the corrected stochastic MH algorithm to the original posterior distribution in terms of the Kullback-Leibler divergence. While these approximation results could be used to analyze the surrogate posteriors based on properties of the original posterior as done for variational Bayes methods, see Ray & Szabó [45], we will instead directly investigate the surrogate posteriors which will allow for sharp results.

We prove oracle inequalities for the surrogate posteriors of the stochastic MH method and its corrected modification in the context of deep neural networks. Based on that we can conclude contraction rates as well as rates of convergence for the surrogate posterior mean. Applied to Hölder regular hierarchical regression functions, the contraction rate of the corrected stochastic MH procedure coincides with the minimax rate by Schmidt-Hieber [51] (up to a logarithmic factor). While the latter paper has analyzed sparse deep neural networks with ReLU activation function, similar results for fully connected networks are given by Kohler & Langer [35] and we exploit their main approximation theorem. Moreover, we investigate size and coverage of credible balls from the surrogate posterior. A mixing approach, as e.g. in Alquier & Biau [3], allows for learning the optimal width of the network and leads to a fully adaptive method, see Section 4.

A simulation study demonstrates the merit of the correction term for sampling from a 10401 dimensional parameter space for a low-dimensional regression task. The samples from the surrogate posterior of our corrected stochastic MH algorithm, as well as their mean, show a significant improvement in terms of the empirical prediction risk and size of credible balls over those taken from the surrogate posterior of the naive stochastic MH algorithm. The correction term cancels the bias on the size of accepted batches introduced by the stochastic setting. The Python code of the numerical example is available on GitHub.[1]

Related literature. In view of possibly better scaling properties, variational Bayes methods have been intensively studied in recent years. Instead of sampling from the posterior distribution itself, variational Bayes methods approximate the posterior within a parametric distribution class which can be easily sampled from, see Blei et al. [13] for a review. The theoretical understanding of variational Bayes methods is a current research topic, see [57, 58, 45] and references therein.

Our oracle inequalities rely on PAC-Bayes theory which provides *probably approximately correct* error bounds and goes back to Shawe-Taylor & Williamson [52] and McAllester [39, 40]. We refer to the review papers by Guedj [29] and Alquier [2]. PAC-Bayes bounds in a regression setting have been studied, see e.g. Audibert [6, 7], Audibert & Catoni [8] and the references therein. Our analysis of the Bayesian procedure from a frequentist point of view embeds into the nonparametric Bayesian inference, see Ghosal

[1] `https://github.com/sbieringer/csMALA.git`

& van der Vaart [27]. Coverage of credible sets has been studied, for instance, by Szabó et al. [54] and Rousseau & Szabó [49] and based on the Bernstein-von Mises theorem in Castillo & Nickl [14] among others. While contraction rates for Bayes neural networks have been studied by Polson & Ročková [44] and Chérief-Abdellatif [17], the theoretical properties of credible sets are not well understood so far. Franssen & Szabó [25] have studied an empirical Bayesian approach where only the last layer of the network is Bayesian while the remainder of the network remains fixed.

For an introduction to neural networks, see e.g. Goodfellow et al. [28] and Schmidhuber [50]. While early theoretical foundations for neural nets are summarized by Anthony & Bartlett [5], the excellent approximation properties of deep neural nets, especially with the ReLU activation function, have been discovered in recent years, see e.g. Yarotsky [56] and the review paper DeVore et al. [23]. In addition to these approximation properties, an explanation of the empirical capabilities of neural networks has recently been given by Schmidt-Hieber [51] as well as Bauer & Kohler [10]: While classical regression methods suffer from the curse of dimensionality, deep neural network estimators can profit from a hierarchical structure of the regression function and a possibly much smaller intrinsic dimension.

Tailoring Markov Chains to the needs of current neural network application is an field of ongoing investigation. Different efforts to improve efficiency by improve mixing, that is transitioning between modes of the posterior landscape, exist. Zhang et al. [59] employ a scheduled step-size to help the algorithm move between different modes of the posterior, while contour stochastic gradient MCMC [22, 21] uses a piece-wise continuous function to flatten the posterior landscape which is itself determined through MCMC sampling or from parallel chains. Parallel chains of different temperature are employed by [20] at the cost of memory space during computation. Only limited research on scaling MCMC for large data has been done. Most recently, Cobb & Jalaian [18] introduced a splitting scheme for Hamiltonian Monte Carlo maintaining the full Hamiltonian.

Organization. The paper is organized as follows: In Section 2, we derive the stochastic MH procedure, introduce the stochastic MH correction and study the Kullback-Leibler divergences of the surrogate posterior from the Gibbs posterior. In Section 3, we state the oracle inequality and the resulting contraction rates and we investigate credible sets. In Section 4 we present a data-driven approach to choosing architecture of the network for our method. The numerical performance of the method is studied in Section 5. All proofs have been postponed to Section 6.

2 Stochastic Metropolis-adjusted Langevin algorithm

The aim is to estimate a regression function $f\colon \mathbb{R}^p \to \mathbb{R}$, $p \in \mathbb{N}$ based on a training sample $\mathcal{D}_n := (\mathbf{X}_i, Y_i)_{i=1,\dots,n} \subseteq \mathbb{R}^p \times \mathbb{R}$ given by $n \in \mathbb{N}$ i.i.d. copies of generic random variables $(\mathbf{X}, Y) \in \mathbb{R}^p \times \mathbb{R}$ on some probability space $(\Omega, \mathcal{A}, \mathbb{P})$ with $Y = f(\mathbf{X}) + \varepsilon$ and observation error ε satisfying $\mathbb{E}[\varepsilon \mid \mathbf{X}] = 0$ almost surely (a.s.). Equivalently, $f(\mathbf{X}) = \mathbb{E}[Y \mid \mathbf{X}]$ a.s. For any estimator $\widehat{f}$, the prediction risk and its empirical counterpart are given by

$$R(\widehat{f}) := \mathbb{E}_{(\mathbf{X},Y)}\big[\big(Y - \widehat{f}(\mathbf{X})\big)^2\big] \qquad \text{and} \qquad R_n(\widehat{f}) = \frac{1}{n}\sum_{i=1}^n \big(Y_i - \widehat{f}(\mathbf{X}_i)\big)^2,$$

respectively, where $\mathbb{E}$ denotes the expectation under $\mathbb{P}$ and $\mathbb{E}_Z$ is the (conditional) expectation only with respect to a random variable Z. The accuracy of the estimation procedure will be quantified in terms of the excess risk

$$\mathcal{E}(\widehat{f}) := R(\widehat{f}) - R(f) = \mathbb{E}_{\mathbf{X}}\big[\big(\widehat{f}(\mathbf{X}) - f(\mathbf{X})\big)^2\big] = \|\widehat{f} - f\|^2_{L^2(\mathbb{P}^{\mathbf{X}})},$$

where $\mathbb{P}^{\mathbf{X}}$ denotes the distribution of $\mathbf{X}$.

We consider a parametric class of potential estimators $\mathcal{F} = \{f_\vartheta : \vartheta \in [-B, B]^P\}$ for some fixed $B \geqslant 1$ and a potentially large parameter dimension $P \in \mathbb{N}$. For $f_\vartheta \in \mathcal{F}$ we abbreviate $R(\vartheta) = R(f_\vartheta)$ and

$$R_n(\vartheta) = R_n(f_\vartheta) = \frac{1}{n}\sum_{i=1}^n \ell_i(\vartheta) \qquad \text{with} \qquad \ell_i(\vartheta) = \big(Y_i - f_\vartheta(\mathbf{X}_i)\big)^2.$$

Throughout, $|x|_q$ denotes the ℓ^q-norm of a vector $x \in \mathbb{R}^p$, $q \in [1,\infty]$. For brevity, $|\cdot| := |\cdot|_2$ is the Euclidean norm. We write $a \vee b := \max\{a,b\}$ and $a \wedge b := \min\{a,b\}$ for $a,b \in \mathbb{R}$. The identity matrix in $\mathbb{R}^d$ is denoted by E_d.

2.1 Prior and posterior distribution

As a prior on the parameter set of the class $\mathcal{F}$ we choose a uniform distribution $\Pi = \mathcal{U}([-B,B]^P)$. The corresponding *Gibbs posterior* $\Pi_\lambda(\cdot \mid \mathcal{D}_n)$ is defined as the solution to the minimization problem

$$\inf_\nu \Big(\int R_n(\vartheta)\,\nu(\mathrm{d}\vartheta) + \frac{1}{\lambda}\,\mathrm{KL}(\nu \mid \Pi) \Big)$$

where the infimum is taken over all probability distributions ν on $\mathbb{R}^P$. Hence, $\Pi_\lambda(\cdot \mid \mathcal{D}_n)$ will concentrate at parameters ϑ with a small empirical risk $R_n(\vartheta)$, but it takes into account a regularization term determined by the Kullback-Leibler divergence (denoted by KL, see (6.2) for a definition) to the prior distribution Π and weighted via the *inverse temperature parameter* $\lambda > 0$. This optimization problem has a unique solution given by

$$\Pi_\lambda(\mathrm{d}\vartheta \mid \mathcal{D}_n) \propto \exp\big(-\lambda R_n(\vartheta)\big)\Pi(\mathrm{d}\vartheta), \tag{2.1}$$

see Alquier [2] or Lemma 15 below. While (2.1) coincides with the classical Bayesian posterior distribution if $Y_i = f_\vartheta(\mathbf{X}_i) + \varepsilon_i$ with i.i.d. $\varepsilon_i \sim \mathcal{N}(0, \frac{n}{2\lambda})$, the so-called tempered likelihood, see e.g. Bissiri et al. [12], Guedj [29], $\exp(-\lambda R_n(\vartheta))$ serves as a proxy for the unknown distribution of the observations given ϑ. As we will see, the method is indeed applicable under quite general assumptions on the regression model.

Based on the Gibbs posterior distribution the regression function can be estimated via a random draw from the posterior

$$\widehat{f}_\lambda := f_{\widehat{\vartheta}_\lambda} \qquad \text{for} \qquad \widehat{\vartheta}_\lambda \mid \mathcal{D}_n \sim \Pi_\lambda(\cdot \mid \mathcal{D}_n), \tag{2.2}$$

or via the posterior mean

$$\bar{f}_\lambda := \mathbb{E}\big[f_{\widehat{\vartheta}_\lambda} \,\big|\, \mathcal{D}_n\big] = \int f_\vartheta\,\Pi_\lambda(\mathrm{d}\vartheta \mid \mathcal{D}_n). \tag{2.3}$$

Another popular approach is to use the maximum a posteriori (MAP) estimator, but we will focus on the previous two estimators.

To apply the estimators $\widehat{f}_\lambda$ and $\bar{f}_\lambda$ in practice, we need to sample from the Gibbs posterior. The MCMC approach is to construct a Markov chain $(\vartheta^{(k)})_{k\in\mathbb{N}_0}$ with stationary distribution $\Pi_\lambda(\cdot \mid \mathcal{D}_n)$, see [46]. In particular, the *Langevin* MCMC sampler is given by

$$\vartheta^{(k+1)} = \vartheta^{(k)} - \gamma\nabla_\vartheta R_n(\vartheta^{(k)}) + sW_k, \tag{2.4}$$

where $\nabla_\vartheta R_n(\vartheta)$ denotes the gradient of $R_n(\vartheta)$ with respect to ϑ, $\gamma > 0$ is the learning rate and $sW_k \sim \mathcal{N}(0, s^2E_P)$ is i.i.d. white noise with noise level $s > 0$. This approach can also be interpreted as a noisy version of the gradient descent method commonly used to train neural networks. In practice this approach requires careful tuning of the procedural parameters and Langevin-MCMC suffers from relatively slow polynomial convergence rates of the distribution of $\vartheta^{(k)}$ to the target distribution $\Pi_\lambda(\cdot \mid \mathcal{D}_n)$, see [42, 16]. Only in special cases, the convergence rates are faster, see e.g. Freund et al. [26] for an overview and Dalalyan & Riou-Durand [19] for the case of log-concave densities. This convergence rate can be considerably improved by adding an MH step resulting in the *Metropolis-adjusted Langevin algorithm* (MALA), see [47].

Applying the generic MH algorithm to $\Pi_\lambda(\cdot \mid \mathcal{D}_n)$ and taking into account that the prior Π is uniform, we obtain the following iterative method: Starting with some initial choice $\vartheta^{(0)} \in \mathbb{R}^P$, we successively generate $\vartheta^{(k+1)}$ given $\vartheta^{(k)}$, $k \in \mathbb{N}_0$, by

$$\vartheta^{(k+1)} = \begin{cases} \vartheta' & \text{with probability } \alpha(\vartheta' \mid \vartheta^{(k)}) \\ \vartheta^{(k)} & \text{with probability } 1 - \alpha(\vartheta' \mid \vartheta^{(k)}) \end{cases},$$

where ϑ' is a random variable drawn from some conditional proposal density $q(\cdot \mid \vartheta^{(k)})$ and the *acceptance probability* is chosen as

$$\alpha(\vartheta' \mid \vartheta) = \exp\big(- \lambda R_n(\vartheta') + \lambda R_n(\vartheta)\big) \mathbb{1}_{[-B,B]^P}(\vartheta') \frac{q(\vartheta \mid \vartheta')}{q(\vartheta' \mid \vartheta)} \wedge 1. \tag{2.5}$$

In view of (2.4) the probability density q of the proposal distribution is given by

$$q(\vartheta' \mid \vartheta) = \frac{1}{(2\pi s^2)^{P/2}} \exp\Big(- \frac{1}{2s^2} \big|\vartheta' - \vartheta + \gamma \nabla_\vartheta R_n(\vartheta)\big|^2\Big). \tag{2.6}$$

The standard deviation s should not be too large as otherwise the acceptance probability might be too small. As a result the proposal would rarely be accepted, the chain might not be sufficiently randomized and the convergence to the invariant target distribution would be too slow in practice. On the other hand, s should not be smaller than the shift $\gamma \nabla_\vartheta R_n(\vartheta)$ in the mean, since otherwise $q(\vartheta \mid \vartheta')$ might be too small. The MH step ensures that $(\vartheta^{(k)})_{k \in \mathbb{N}_0}$ is a Markov chain with invariant distribution $\Pi_\lambda(\cdot \mid \mathcal{D}_n)$ (under rather mild conditions on q). The convergence to the invariant distribution follows from Roberts & Tweedie [48, Theorem 2.2] with geometric rate.

To calculate the estimators $\widehat{f}_\lambda$ and $\bar{f}_\lambda$ from (2.2) and (2.3), respectively, one chooses a *burn-in* time $b \in \mathbb{N}$ to let the distribution of the Markov chain stabilize at its invariant distribution and then sets

$$\widehat{f}_\lambda = f_{\vartheta^{(b)}} \qquad \text{and} \qquad \bar{f}_\lambda = \frac{1}{N} \sum_{k=1}^N f_{\vartheta^{(b+ck)}}.$$

A sufficiently large *gap length* $c \in \mathbb{N}$ ensures the necessary variability and reduced dependence between $\vartheta^{(b+ck)}$ and $\vartheta^{(b+c(k+1))}$, whereas $N \in \mathbb{N}$ has to be large enough for a good approximation of the expectation by the empirical mean.

2.2 Stochastic Metropolis-Hastings

The gradient has to be calculated only once in each MALA iteration. Hence, using the full gradient $\nabla_\vartheta R_n(\vartheta) = \frac{1}{n}\sum_{i=1}^n \nabla_\vartheta \ell_i(\vartheta)$, the additional computational price of MALA compared to training a standard neural network by empirical risk minimization only comes from a larger number of necessary iterations due to the rejection with probability $1 - \alpha(\vartheta' \mid \vartheta^{(k)})$. For large datasets however the standard training of a neural network would rely on a stochastic gradient method, where the gradient $\frac{1}{m}\sum_{i \in \mathcal{B}} \nabla_\vartheta \ell_i(\vartheta)$ is only calculated on (mini-)batches $\mathcal{B} \subseteq \{1, \dots, n\}$ of size $m < n$. While we could replace $\nabla_\vartheta R_n(\vartheta)$ in (2.6) by a stochastic approximation without any additional obstacle, the MH step still requires the calculation of the loss $\ell_i(\vartheta')$ for all $1 \leqslant i \leqslant n$ in (2.5).

To avoid a full evaluation of the empirical risk $R_n(\vartheta)$, a natural approach is to replace the empirical risks in $\alpha(\vartheta' \mid \vartheta)$ by a batch-wise approximation, too. To study the consequences of this approximation we follow a pseudo-marginal MH approach, see [4, 37, 9].

We augment our target distribution by a set of auxiliary random variables $Z_i \overset{\text{i.i.d.}}{\sim} \mathrm{Ber}(\rho)$ with some $\rho \in (0, 1]$ and aim for a reduction of the empirical risk $R_n(\vartheta)$ to the stochastic approximation

$$R_n(\vartheta, Z) := \frac{1}{n\rho} \sum_{i=1}^n Z_i \ell_i(\vartheta)$$

in the algorithm. Hence, we define the joint target distribution by

$$\begin{aligned}\bar{\Pi}_{\lambda,\rho}(\vartheta, z \mid \mathcal{D}_n) &\propto \prod_{i=1}^n \rho^{z_i}(1-\rho)^{1-z_i} \exp\big(- \lambda R_n(\vartheta, z)\big)\Pi(\mathrm{d}\vartheta) \\ &\propto \exp\Big(- \lambda R_n(\vartheta, z) + \log\Big(\frac{\rho}{1-\rho}\Big) \sum_{i=1}^n z_i\Big)\Pi(\mathrm{d}\vartheta), \qquad z \in \{0,1\}^n.\end{aligned}$$

The marginal distribution in ϑ is then given by

$$\bar{\Pi}_{\lambda,\rho}(\vartheta \mid \mathcal{D}_n) = \sum_{z\in\{0,1\}^n} \bar{\Pi}_{\lambda,\rho}(\vartheta, z \mid \mathcal{D}_n) \propto \prod_{i=1}^n \Big(\rho \mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)} + 1 - \rho\Big)\Pi(\mathrm{d}\vartheta). \tag{2.7}$$

As proposal for the MH algorithm we use

$$\begin{aligned} \bar{q}(\vartheta', z' \mid \vartheta, z) &= q_{\mathrm{s}}(\vartheta' \mid \vartheta, z) \prod_{i=1}^n \rho^{z_i'}(1-\rho)^{1-z_i'} \qquad \text{with} \qquad (2.8)\\ q_{\mathrm{s}}(\vartheta' \mid \vartheta, z) &= \frac{1}{(2\pi s^2)^{P/2}} \exp\Big(-\frac{1}{2s^2}\big|\vartheta' - \vartheta + \gamma\nabla_\vartheta R_n(\vartheta, z)\big|^2\Big). \end{aligned}$$

Hence, the proposed $Z' = z'$ is indeed a vector of independent $\mathrm{Ber}(\rho)$-random variables and $q_{\mathrm{s}}(\vartheta' \mid \vartheta, z)$ is the stochastic analogue to q from (2.6) with a stochastic gradient. The resulting acceptance probabilities are given by

$$\begin{aligned} \alpha(\vartheta', z' \mid \vartheta, z) &= \frac{\bar{q}(\vartheta, z \mid \vartheta', z')\bar{\Pi}_{\lambda,\rho}(\vartheta', z' \mid \mathcal{D}_n)}{\bar{q}(\vartheta', z' \mid \vartheta, z)\bar{\Pi}_{\lambda,\rho}(\vartheta, z \mid \mathcal{D}_n)} \wedge 1\\ &= \frac{q_{\mathrm{s}}(\vartheta \mid \vartheta', z')}{q_{\mathrm{s}}(\vartheta' \mid \vartheta, z)} \mathbb{1}_{[-B,B]^P}(\vartheta') e^{-\lambda R_n(\vartheta', z') + \lambda R_n(\vartheta, z)} \wedge 1. \end{aligned}$$

We observe that $\alpha(\vartheta', z' \mid \vartheta, z)$ corresponds to a stochastic MH step where we have to evaluate the loss $\ell_i(\vartheta')$ for the new proposal ϑ' only if $z_i' = Z_i' \sim \mathrm{Ber}(\rho)$ is one, i.e. with probability ρ. Calculating $\alpha(\vartheta', z' \mid \vartheta, z)$ thus requires only few evaluations of $\ell_i(\vartheta)$ for small values of ρ. The expected number of data points on which the gradient and the loss have to be evaluated is $n\rho$ and corresponds to a batch size of $m = n\rho$.

Generalizing (2.2), we define the stochastic MH estimator

$$\widehat{f}_{\lambda,\rho} := f_{\widehat{\vartheta}_{\lambda,\rho}} \qquad \text{for} \qquad \widehat{\vartheta}_{\lambda,\rho} \mid \mathcal{D}_n \sim \bar{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n). \tag{2.9}$$

For $\rho = 1$ we recover the standard MALA.

As discussed by Bardenet et al. [9], the previous derivation reveals that the stochastic MH step leads to a different invariant distribution of the Markov chain, namely (2.7) instead of the Gibbs posterior from (2.1). Writing

$$\bar{\Pi}_{\lambda,\rho}(\vartheta \mid \mathcal{D}_n) \propto \exp\big(-\lambda \bar{R}_{n,\rho}(\vartheta)\big)\Pi(\mathrm{d}\vartheta) \qquad \text{with} \qquad \bar{R}_{n,\rho}(\vartheta) := -\frac{1}{\lambda}\sum_{i=1}^n \log\big(\rho \mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)} + 1 - \rho\big), \tag{2.10}$$

we observe that $\bar{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ is itself a Gibbs posterior distribution, the *surrogate posterior*, corresponding to the modified risk $\bar{R}_{n,\rho}(\vartheta)$. Note that $\bar{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ coincides with $\Pi_\lambda(\cdot \mid \mathcal{D}_n)$ for $\rho = 1$ and thus $\widehat{f}_\lambda = \widehat{f}_{\lambda,1}$ and $\bar{f}_\lambda = \bar{f}_{\lambda,1}$ in distribution. Whether $\bar{\Pi}_{\lambda,\rho}(\vartheta \mid \mathcal{D}_n)$ also behaves as our original target distribution $\Pi_\lambda(\vartheta \mid \mathcal{D}_n)$ for $\rho < 1$ depends on the choice of λ and ρ:

Lemma 1. *If f and all f_ϑ are bounded by some constant $C > 0$, then we have*

$$\frac{1}{n\rho}\mathrm{KL}\big(\bar{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)\,\big|\,\Pi_\lambda(\cdot \mid \mathcal{D}_n)\big) \leqslant \Big(\frac{\lambda}{n\rho}\Big)^2\Big(64C^4 + \frac{4}{n}\sum_{i=1}^n \varepsilon_i^4\Big).$$

For $\rho < 1$ and the probability distribution $\varpi_{\lambda,\rho}(\vartheta \mid \mathcal{D}_n) :\propto \exp\big(\rho\sum_{i=1}^n \mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)}\big)\Pi(\mathrm{d}\vartheta)$ we moreover have

$$\frac{1}{n\rho}\mathrm{KL}\big(\bar{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)\,\big|\,\varpi_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)\big) \leqslant \frac{\rho}{1-\rho}.$$

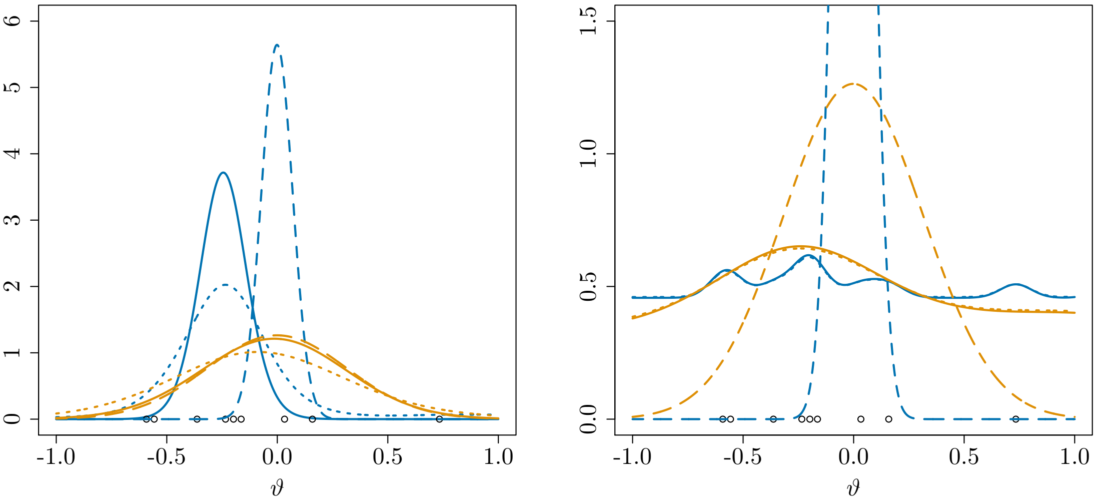


Figure 1: *Points*: $Y_1, \dots, Y_n \sim \mathcal{N}(0, 0.5)$ for $n = 10$. *Solid lines*: densities of $\bar{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ with $\lambda = 10n$ (blue) and $\lambda = n/2$ (orange) and $\rho = 0.9$ (left) and $\rho = 0.1$ (right). *Dashed lines*: corresponding densities of $\Pi_\lambda(\cdot \mid \mathcal{D}_n)$. Dotted lines: corresponding densities of $\varpi_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$.

On the one hand, if $\frac{\lambda}{n\rho}$ is sufficiently small, then the surrogate posterior $\bar{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ is indeed a good approximation for the Gibbs posterior $\Pi_\lambda(\cdot \mid \mathcal{D}_n)$. On the other hand, for $\rho \to 0$ the distribution $\bar{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ behaves as the distribution $\varpi_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ with density proportional to

$$\exp\Big(\rho \sum_{i=1}^{n} \mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)}\Big)\Pi(\mathrm{d}\vartheta).$$

For large $\frac{\lambda}{n\rho}$ the terms $\mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)}$ rapidly decay for all ϑ with $\ell_i(\vartheta) > 0$, i.e. $\varpi_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ emphasizes interpolating parameter choices. For all ϑ where $\frac{\lambda}{n\rho}\ell_i(\vartheta)$ is relatively large the density converges to a constant. Therefore, in the extreme case $\rho \to 0$ and $\frac{\lambda}{n\rho} \to \infty$ the distribution $\varpi_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ and thus $\bar{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ converge to the uninformative prior with interpolating spikes at parameters where $\ell_i(\vartheta)$ are zero.

We illustrate Lemma 1 in a simple setting where $Y_i = \mathcal{N}(0, 0.5)$ and $f_\vartheta(x) = \vartheta$ for $\vartheta \in [-1, 1]$. The densities of the measures $\Pi(\cdot \mid \mathcal{D}_n)$, $\bar{\Pi}(\cdot \mid \mathcal{D}_n)$ and $\varpi(\cdot \mid \mathcal{D}_n)$ are shown in Fig. 1 for different choices of λ and ρ. Fig. 1 confirms the predicted approximation properties: $\bar{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ behaves similarly to $\Pi_\lambda(\cdot \mid \mathcal{D}_n)$ if λ is not too large (violet lines) or ρ is not too small (left figure). Additionally, we observe that $\bar{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ is still informative if λ is in the order $n\rho$ even if it is not close to the Gibbs posterior at all.

The scaling of the Kullback-Leibler distance with $n\rho$ in Lemma 1 is quite natural in this setting. In particular, applying an approximation result from the variational Bayes literature by Ray & Szabó [45, Theorem 5] we obtain for the two reference measures $\mathbb{Q} \in \{\Pi_\lambda(\cdot \mid \mathcal{D}_n), \varpi_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)\}$ and a high probability parameter set Θ_n with $\mathbb{Q}(\Theta_n^{\mathsf{c}}) \leqslant C\mathrm{e}^{-n\rho}$ for some constant $C > 0$ that

$$\mathbb{E}\big[\bar{\Pi}_{\lambda,\rho}(\Theta_n \mid \mathcal{D}_n)\big] \leqslant \frac{2}{n\rho}\mathbb{E}\big[\,\mathrm{KL}\left(\bar{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n) \,\middle|\, \mathbb{Q}\right)\big] + C\mathrm{e}^{-n\rho/2}. \tag{2.11}$$

Hence, for $\frac{\lambda}{n\rho} \to 0$ we could analyze the surrogate posterior via the Gibbs posterior itself at the cost of the approximation error $\frac{1}{n\rho}\,\mathrm{KL}\left(\bar{\Pi}_{\lambda,\rho}(\vartheta \mid \mathcal{D}_n) \,\middle|\, \Pi_\lambda(\vartheta \mid \mathcal{D}_n)\right)$. Instead of this route, we will directly investigate $\bar{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ which especially allows for λ in the order of $n\rho$.

2.3 Corrected stochastic MALA

The computational advantage of the stochastic MH algorithm due to the reduction of the information parameter from n to ρn comes at the cost of a slower convergence rate, see Theorem 5.

To remedy this loss while retaining scalability, we define another joint target distribution as

$$\begin{aligned}\widetilde{\Pi}_{\lambda,\rho}(\vartheta, z \mid \mathcal{D}_n) &\propto \prod_{i=1}^{n} \big(\mathrm{e}^{-\frac{\lambda}{n}\ell_i(\vartheta)z_i}(1-\rho)^{1-z_i}\big)\Pi(\mathrm{d}\vartheta) \\ &\propto \exp\Big(-\frac{\lambda}{n}\sum_{i=1}^{n} z_i\ell_i(\vartheta) - \log(1-\rho)\sum_{i=1}^{n} z_i\Big)\Pi(\mathrm{d}\vartheta), \qquad z \in \{0,1\}^n,\end{aligned}$$

with marginal distribution in ϑ given by

$$\widetilde{\Pi}_{\lambda,\rho}(\vartheta \mid \mathcal{D}_n) = \sum_{z\in\{0,1\}^n} \widetilde{\Pi}_{\lambda,\rho}(\vartheta, z \mid \mathcal{D}_n) \propto \prod_{i=1}^{n}\Big(\rho\frac{\mathrm{e}^{-\frac{\lambda}{n}\ell_i(\vartheta)}}{\rho} + 1 - \rho\Big)\Pi(\mathrm{d}\vartheta) = \exp\big(-\lambda\widetilde{R}_{n,\rho}(\vartheta)\big)\Pi(\mathrm{d}\vartheta)$$

with

$$\widetilde{R}_{n,\rho}(\vartheta) := -\frac{1}{\lambda}\sum_{i=1}^{n}\log\big(\mathrm{e}^{-\frac{\lambda}{n}\ell_i(\vartheta)} + 1 - \rho\big). \tag{2.12}$$

Compared to $\bar{R}_{n,\rho}$ from (2.10) there is no ρ in the first term in the logarithm. In line with (2.2) and (2.3), we obtain the estimators

$$\widetilde{f}_{\lambda,\rho} := f_{\widetilde{\vartheta}_{\lambda,\rho}} \qquad \text{for} \qquad \widetilde{\vartheta}_{\lambda,\rho} \mid \mathcal{D}_n \sim \widetilde{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n) \tag{2.13}$$

and

$$\bar{f}_{\lambda,\rho} := \mathbb{E}\big[f_{\widetilde{\vartheta}_{\lambda,\rho}} \,\big|\, \mathcal{D}_n\big] = \int f_\vartheta \,\widetilde{\Pi}_{\lambda,\rho}(\mathrm{d}\vartheta \mid \mathcal{D}_n). \tag{2.14}$$

To sample from $\widetilde{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ the MH algorithm with proposal density $q(\vartheta', z' \mid \vartheta, z) = q_{\mathrm{s}}(\vartheta' \mid \vartheta, z)\prod_{i=1}^{n}\rho^{z_i'}(1-\rho)^{1-z_i'}$ as in (2.8) leads to the acceptance probabilities

$$\alpha(\vartheta', z' \mid \vartheta, z) = \frac{q_{\mathrm{s}}(\vartheta \mid \vartheta', z')}{q_{\mathrm{s}}(\vartheta' \mid \vartheta, z)}\mathbb{1}_{[-B,B]^P}(\vartheta')\exp\Big(-\sum_{i=1}^{n} z_i'\big(\tfrac{\lambda}{n}\ell_i(\vartheta') + \log\rho\big) + \sum_{i=1}^{n} z_i\big(\tfrac{\lambda}{n}\ell_i(\vartheta) + \log\rho\big)\Big) \wedge 1.$$

To take the randomized batches into account, we thus introduce a small *correction term* $\frac{\log\rho}{\lambda}|Z| = \mathcal{O}_{\mathbb{P}}(\frac{n}{\lambda}\rho\log\rho)$ in the empirical risks. The resulting surrogate posterior $\widetilde{\Pi}_{\lambda,\rho}(\vartheta \mid \mathcal{D}_n)$ achieves a considerably improved approximation of the Gibbs distribution $\Pi_\lambda(\cdot \mid \mathcal{D}_n)$:

Lemma 2. *If f and all f_ϑ are bounded by some constant $C > 0$, then we have*

$$\frac{1}{n}\mathrm{KL}\big(\widetilde{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n) \mid \Pi_{\lambda/(2-\rho)}(\cdot \mid \mathcal{D}_n)\big) \leqslant \Big(\frac{\lambda}{n}\Big)^2\Big(32C^4 + \frac{2}{n}\sum_{i=1}^{n}\varepsilon_i^4\Big).$$

Compared to Lemma 1, the approximation error of $\widetilde{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ in terms of the Kullback-Leibler distance is now determined by the full sample size n instead of the possibly much smaller batch size ρn as for the stochastic MH algorithm. The only price to pay is a reduction of the inverse temperature parameter λ by the factor $(2-\rho)^{-1} \in [\frac{1}{2}, 1]$. As already mentioned in (2.11), we can conclude contraction and coverage results for $\widetilde{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ by combining Ray & Szabó [45, Theorem 5] with Lemma 2 if $\lambda/n \to 0$. A direct analysis of $\widetilde{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ will even allow for λ of the order n in our main results and thus lead to results as good as we can hope for the Gibbs measure itself.

The corrected stochastic MALA (csMALA) is summarized in Algorithm 1. The implementation omits the restriction of the proposed network weights to $[-B, B]^P$ which is practically negligible for sufficiently large constant B and the correction term $\frac{\log\rho}{\lambda}|Z| = \mathcal{O}_{\mathbb{P}}(\frac{n}{\lambda}\rho\log\rho)$ in the empirical risk is weighted by some tuning parameter $\zeta \geqslant 0$. For $\zeta = 0$ we recover the uncorrected method. In theory we always set $\zeta = 1$, but in practice the flexibility gained from choosing ζ was beneficial.

```
Algorithm 1 csMALA - corrected stochastic Metropolis adjusted Langevin Algorithm
    Input: inverse temperature $\lambda > 0$, learning rate $\gamma > 0$, standard deviation $s > 0$,
    correction parameter $\zeta \geqslant 0$, batch size $m \in \{1,\dots,n\}$, burn-in $b \in \mathbb{N}$,
    gap length $c \in \mathbb{N}$, number of draws $N \in \mathbb{N}$.
      1. Initialize $\vartheta^{(0)} \in \mathbb{R}^P$ and $Z^{(0)} \sim \mathrm{Ber}(\frac{m}{n})^{\otimes n}$.
      2. Calculate $R_n^{(0)} = \frac{1}{n}\sum_{i=1}^n Z_i^{(0)}\ell_i(\vartheta^{(0)}) + \zeta\frac{\log\rho}{\lambda}|Z^{(0)}|$ and $\nabla R_n^{(0)} = \nabla_\vartheta R_n(\vartheta^{(0)}, Z^{(0)})$.
      3. For $k = 0,\dots,b+cN$ do:
         (a) Draw $Z' \sim \mathrm{Ber}(\frac{m}{n})^{\otimes n}$.
         (b) Draw $\vartheta' \sim \mathcal{N}(\vartheta^{(k)} - \gamma\nabla R_n^{(k)}, s^2)$ and calculate
             $R_n' = \frac{1}{n}\sum_{i=1}^n Z_i'\ell_i(\vartheta') + \zeta\frac{\log\rho}{\lambda}|Z'|$ and $\nabla R_n' = \nabla_\vartheta R_n(\vartheta', Z')$.
         (c) Calculate acceptance probability

             $\alpha^{(k+1)} = \exp\Big(\lambda R_n^{(k)} + \frac{1}{2s^2}\big|\vartheta' - \vartheta^{(k)} + \gamma\nabla R_n^{(k)}\big|^2 - \lambda R_n' - \frac{1}{2s^2}\big|\vartheta^{(k)} - \vartheta' + \gamma\nabla R_n'\big|^2\Big).$

         (d) Draw $u \sim \mathcal{U}([0,1])$. If $u \leqslant \alpha^{(k+1)}$,
             then set $\vartheta^{(k+1)} = \vartheta', R_n^{(k+1)} = R_n', \nabla R_n^{(k+1)} = \nabla R_n^{'}$,
             else set $\vartheta^{(k+1)} = \vartheta^{(k)}, R_n^{(k+1)} = R_n^{(k)}, \nabla R_n^{(k+1)} = \nabla R_n^{(k)}$.
    Output: $\widetilde{f}_{\lambda,\rho} = f_{\vartheta^{(b)}}$, $\bar{f}_{\lambda,\rho} = \frac{1}{N}\sum_{k=1}^N f_{\vartheta^{(b+ck)}}$
```

3 Oracle inequality and its consequences

In this section we state the statistical guarantees for the estimators defined in terms of the surrogate posterior distributions. It is worth noting that our analysis is independent of the choice of the proposal distribution. We derive oracle inequalities for the estimators $\widehat{f}_{\lambda,\rho}$ (Theorem 5) and $\widetilde{f}_{\lambda,\rho}$ (Theorem 3) and as a consequence an analogous oracle inequality for $\bar{f}_{\lambda,\rho}$ (Corollary 6), which verify that these estimators are not much worse than the optimal choice for ϑ. We also discuss the properties of credible balls.

In the sequel the estimator $\widehat{f}$ is chosen as a neural network. More precisely, we consider a *feedforward multilayer perceptron* with $p \in \mathbb{N}$ inputs, $L \in \mathbb{N}$ hidden layers and constant width $r \in \mathbb{N}$. The latter restriction is purely for notational convenience. The *rectified linear unit* (ReLU) $\phi(x) := \max\{x, 0\}, x \in \mathbb{R}$, is used as activation function. We write $\phi_v x := \big(\phi(x_i + v_i)\big)_{i=1,\dots,d}$ for vectors $x, v \in \mathbb{R}^d$. With this notation we can represent such neural networks as

$$g_\vartheta(\mathbf{x}) := W^{(L+1)}\phi_{v^{(L)}}W^{(L)}\phi_{v^{(L-1)}}\cdots W^{(2)}\phi_{v^{(1)}}W^{(1)}\mathbf{x} + v^{(L+1)}, \qquad \mathbf{x} \in \mathbb{R}^p,$$

where the parameter vector ϑ contains all entries of the weight matrices $W^{(1)} \in \mathbb{R}^{r\times p}, W^{(2)},\dots,W^{(L)} \in \mathbb{R}^{r\times r}, W^{(L+1)} \in \mathbb{R}^{1\times r}$ and the shift ('bias') vectors $v^{(1)},\dots,v^{(L)} \in \mathbb{R}^r, v^{(L+1)} \in \mathbb{R}$. The total number of network parameters is

$$P := (p+1)r + (L-1)(r+1)r + r + 1.$$

A layer-wise representation of g_ϑ is given by

$$\begin{aligned} \mathbf{x}^{(0)} &:= \mathbf{x} \in \mathbb{R}^p, \\ \mathbf{x}^{(l)} &:= \phi(W^{(l)}\mathbf{x}^{(l-1)} + v^{(l)}),\ l = 1,\dots,L, \\ g_\vartheta(\mathbf{x}) := \mathbf{x}^{(L+1)} &:= W^{(L+1)}\mathbf{x}^{(L)} + v^{(L+1)}, \end{aligned} \tag{3.1}$$

where the activation function is applied coordinate-wise. We denote the class of all such functions g_ϑ by $\mathcal{G}(p, L, r)$. For some $C \geqslant 1$, we also introduce the class of clipped networks

$$\mathcal{F}(p, L, r, C) := \big\{f_\vartheta = (-C) \vee (g_\vartheta \wedge C) \,\big|\, g_\vartheta \in \mathcal{G}(p, L, r)\big\}.$$

3.1 Oracle inequality

Our first main result compares the performance of the estimator $\widetilde{f}_{\lambda,\rho}$ from (2.13) to the best possible network f_{ϑ^*} for the *oracle choice*

$$\vartheta^* \in \underset{\vartheta\in[-B,B]^P}{\arg\min}\ R(\vartheta) = \underset{\vartheta\in[-B,B]^P}{\arg\min}\ \mathcal{E}(\vartheta). \tag{3.2}$$

The oracle is not accessible to the practitioner because $R(\vartheta)$ depends on the unknown distribution of $(\mathbf{X}, Y)$. A solution to the minimization problem in (3.2) always exists since $[-B,B]^P$ is compact and $\vartheta \mapsto R(\vartheta)$ is continuous. If there is more than one solution, we choose one of them. We need some mild assumption on the regression model.

Assumption A.

1. ***Bounded regression function:*** *For some $C \geqslant 1$ we have $\|f\|_\infty \leqslant C$.*
2. ***Second moment of inputs:*** *For some $K \geqslant 1$ we have $\mathbb{E}[|\mathbf{X}|^2] \leqslant K$.*
3. ***Conditional sub-Gaussianity of observation noise:*** *There are constants $\sigma, \Gamma > 0$ such that*

$$\mathbb{E}[|\varepsilon|^k \mid \mathbf{X}] \leqslant \frac{k!}{2}\sigma^2\Gamma^{k-2} \ a.s., \qquad \text{for all } k \geqslant 2.$$

4. ***Conditional symmetry of observation noise:*** *ε is conditionally on $\mathbf{X}$ symmetric.*

Note that neither the loss function nor the data are assumed to be bounded. We obtain the following non-asymptotic oracle inequality for our estimator $\widetilde{f}_{\lambda,\rho}$ from (2.13):

Theorem 3 (PAC-Bayes oracle inequality for csMALA)**.** *Under Assumption A there are constants $Q_0, Q_1 > 0$ depending only on C, Γ, σ such that for $\lambda = n/Q_0$ and sufficiently large n we have for all $\delta \in (0,1)$ with probability of at least $1-\delta$ that*

$$\mathcal{E}(\widetilde{f}_{\lambda,\rho}) \leqslant 12\mathcal{E}(f_{\vartheta^*}) + \frac{Q_1}{n}\big(PL\log(n) + \log(2/\delta)\big). \tag{3.3}$$

Remark 4. For $\rho = 1$ we do not need the conditional symmetry condition in Assumption A. An explicit admissible choice for λ is $\lambda = n/\big(2^5C(\Gamma \vee (2C)) + 2^7(C^2+\sigma^2) + 2^3(\sigma C + \sigma^2)\big)$. The dependence of Q_1 on C, Γ, σ is at most quadratic and $n \geqslant n_0 = 2 \vee B \vee K \vee L \vee r \vee p$ is sufficiently large.

The right-hand side of (3.3) can be interpreted similarly to the classical bias-variance decomposition in nonparametric statistics. The first term $\mathcal{E}(f_{\vartheta^*}) = \mathbb{E}[(f_{\vartheta^*}(\mathbf{X}) - f(\mathbf{X}))^2]$ quantifies the approximation error while second term is an upper bound for the stochastic error. Theorem 3 is in line with classical PAC-Bayes oracle inequalities, see Bissiri et al. [12], Guedj & Alquier [30], Zhang [60]. In particular, Chérief-Abdellatif [17] has obtained a similar oracle inequality for a variational approximation of the Gibbs posterior distribution. A main step in the proof of Theorem 3 is to verify the compatibility between the risk $\widetilde{R}_{n,\rho}$ from (2.12) and the empirical risk R_n as established in Proposition 13.

We obtain a similar result for $\widehat{f}_{\lambda,\rho}$ from (2.9). Note that here the stochastic error term is of order $\mathcal{O}(\frac{PL}{n\rho})$ instead of $\mathcal{O}(\frac{PL}{n})$ as in Theorem 3 (up to logarithms).

Theorem 5 (Oracle inequality for sMALA)**.** *Under Assumption A there are constants $Q_0', Q_1' > 0$ depending only on C, Γ, σ such that for $\lambda = n\rho/Q_0'$ and sufficiently large n we have for all $\delta \in (0,1)$ with probability of at least $1-\delta$ that*

$$\mathcal{E}(\widehat{f}_{\lambda,\rho}) \leqslant 4\mathcal{E}(f_{\vartheta^*}) + \frac{Q_1'}{n\rho}\big(PL\log(n) + \log(2/\delta)\big).$$

In view of Theorem 5 the following results are also true for the stochastic MH estimator if n is replaced by $n\rho$. However, we focus only on the analysis of $\widetilde{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)$ for the sake of clarity.

The $1-\delta$ probability in Theorem 3 takes into account the randomness of the data and of the estimate. Denoting

$$r_n^2 := 12\|f_{\vartheta^*} - f\|_{L^2(\mathbb{P}^{\mathbf{X}})}^2 + \frac{Q_1}{n} PL\log(n), \tag{3.4}$$

we can rewrite (3.3) as

$$\mathbb{E}\big[\widetilde{\Pi}_{\lambda,\rho}\big(\|f_{\widetilde{\vartheta}_{\lambda,\rho}} - f\|_{L^2(\mathbb{P}^{\mathbf{X}})}^2 > r_n^2 + t^2 \,\big|\, \mathcal{D}_n\big)\big] \leqslant 2\mathrm{e}^{-nt^2/Q_1}, \qquad t > 0,$$

which is a *contraction rate* result in terms of a frequentist analysis of the nonparametric Bayes method.

An immediate consequence is an oracle inequality for the posterior mean $\bar{f}_{\lambda,\rho}$ from (2.14).

Corollary 6 (Posterior mean). *Under the conditions of Theorem 3 we have with probability of at least $1-\delta$ that*

$$\mathcal{E}(\bar{f}_{\lambda,\rho}) \leqslant 12\mathcal{E}(f_{\vartheta^*}) + \frac{Q_2}{n}\big(PL\log(n) + \log(2/\delta)\big)$$

with a constant Q_2 only depending on C, Γ, σ from Assumption A.

Using the approximation properties of neural networks, the oracle inequality yields the optimal rate of convergence (up to a logarithmic factor) over the following class of hierarchical functions:

$$\begin{aligned}\mathcal{H}(q, \mathbf{d}, \mathbf{t}, \beta, C_0) := \Big\{ g_q \circ \dots \circ g_0 \colon [0,1]^p \to \mathbb{R} \,\Big|\, & g_i = (g_{ij})_j^\top : [a_i, b_i]^{d_i} \to [a_{i+1}, b_{i+1}]^{d_{i+1}}, \\ & g_{ij} \text{ depends on at most } t_i \text{ arguments}, \\ & g_{ij} \in \mathcal{C}_{t_i}^{\beta_i}([a_i, b_i]^{t_i}, C_0), \text{ for some } |a_i|, |b_i| \leqslant C_0 \Big\},\end{aligned}$$

where $\mathbf{d} := (p, d_1, \dots, d_q, 1) \in \mathbb{N}^{q+2}, \mathbf{t} := (t_0, \dots, t_q) \in \mathbb{N}^{q+1}, \beta := (\beta_0, ..., \beta_q) \in (0, \infty)^{q+1}$ and where $\mathcal{C}_{t_i}^{\beta_i}([a_i, b_i]^{t_i}, C_0)$ denote classical Hölder balls with Hölder regularity $\beta_i > 0$. For a detailed discussion of $\mathcal{H}(q, \mathbf{d}, \mathbf{t}, \beta, C_0)$, see [51]. Theorem 3 reveals the following convergence rate which is in line with the upper bounds by Schmidt-Hieber [51] and Kohler & Langer [35]:

Proposition 7 (Rates of convergence). *Let $\mathbf{X} \in [0,1]^p$. In the situation of Theorem 3, there exists a network architecture $(L, r) = (C_1 \log n, C_2(n/(\log n)^3)^{t^*/(4\beta^*+2t^*)})$ with $C_1, C_2 > 0$ only depending on upper bounds for $q, |\mathbf{d}|_\infty, |\beta|_\infty, C_0$ such that the estimators $\widetilde{f}_{\lambda,\rho}$ and $\bar{f}_{\lambda,\rho}$ satisfy for sufficiently large n uniformly over all hierarchical functions $f \in \mathcal{H}(q, \mathbf{d}, \mathbf{t}, \beta, C_0)$*

$$\begin{aligned}\mathcal{E}(\widetilde{f}_{\lambda,\rho}) &\leqslant Q_3\Big(\frac{(\log n)^3}{n}\Big)^{2\beta^*/(2\beta^*+t^*)} + Q_3\frac{\log(2/\delta)}{n} \qquad \textit{and} \\ \mathcal{E}(\bar{f}_{\lambda,\rho}) &\leqslant Q_4\Big(\frac{(\log n)^3}{n}\Big)^{2\beta^*/(2\beta^*+t^*)} + Q_4\frac{\log(2/\delta)}{n}\end{aligned}$$

with probability of at least $1-\delta$, respectively, where β^* and t^* are given by

$$\beta^* := \beta_{i^*}^*, \qquad t^* := t_{i^*}^* \qquad \textit{for} \qquad i^* \in \underset{i=0,\dots,q}{\arg\min} \frac{2\beta_i^*}{2\beta_i^* + t_i^*} \qquad \textit{and} \qquad \beta_i^* := \beta_i \prod_{l=i+1}^q (\beta_l \wedge 1).$$

The constants Q_3 and Q_4 only depend on upper bounds for $q, \mathbf{d}, \beta$ and C_0 as well as the constants from Assumption A.

Remark 8. Similarly, there exists a network architecture $(L, r) = (C_1(n/\log n)^{t^*/(4\beta^*+2t^*)}, C_2)$ such that we achieve the same rate of convergence just with $\log n$ instead of $(\log n)^3$.

It has been proved by Schmidt-Hieber [51] that this is the minimax optimal rate of convergence for the nonparametric estimation of f up to logarthmic factors. Studying the special case of classical Hölder balls $\mathcal{C}_p^\beta([0,1]^p, C_0)$, a contraction rate of order $n^{-2\beta/(2\beta+p)}$ has been derived by Polson & Ročková [44] and Chérief-Abdellatif [17].

3.2 Credible sets

In addition to the contraction rates, the Bayesian approach offers a possibility for uncertainty quantification. For this, we will assume that the distribution $\mathbb{P}^{\mathbf{X}}$ of $\mathbf{X}$ is known. We define the *credible ball*

$$\widehat{C}(\tau_\alpha) := \{h \in L^2 : \|h - \bar{f}_{\lambda,\rho}\|_{L^2(\mathbb{P}^{\mathbf{X}})} \leqslant \tau_\alpha\}, \qquad \alpha \in (0,1),$$

with critical values

$$\tau_\alpha := \operatorname*{arg\,inf}_{\tau>0} \big\{\widetilde{\Pi}_{\lambda,\rho}(\vartheta : \|f_\vartheta - \bar{f}_{\lambda,\rho}\|_{L^2(\mathbb{P}^{\mathbf{X}})} \leqslant \tau \mid \mathcal{D}_n) > 1-\alpha\big\}.$$

By construction $\widehat{C}(\tau_\alpha)$ is the smallest L^2-ball around $\bar{f}_{\lambda,\rho}$ which contains $1-\alpha$ mass of the surrogate posterior measure. Despite the posterior belief, it is not necessarily guaranteed that the true regression function is contained in $\widehat{C}(\tau_\alpha)$. More precisely, the posterior distribution might be quite certain, in the sense that the credible ball is quite narrow, but suffers from a significant bias. In general, it might happen that $\mathbb{P}(f \in \widehat{C}(\tau_\alpha)) \to 0$, see e.g. Knapik et al. [34, Theorem 4.2] in a Gaussian model. To circumvent this, Rousseau & Szabó [49] have introduced inflated credible balls where the critical value is multiplied with a slowly diverging factor. While they proved that this method works in several classical nonparametric models with a sieve prior, our neural network setting causes an additional problem. In order to prove coverage, we would like to compare norms in the intrinsic parameter space, i.e. the space of the network weights, with the norm of the resulting predicted regression function. While the fluctuation of f_ϑ can be controlled via the fluctuation of ϑ, more precisely we have $\|f_\vartheta - f_{\vartheta'}\|_{L^2(\mathbb{P}^{\mathbf{X}})} = \mathcal{O}(\Delta(L,r)\cdot|\vartheta-\vartheta'|_\infty)$ with $\Delta(L,r) := (2rB)^L$, see Lemma 17 below, the converse direction does not hold. Even locally around an oracle choice ϑ^* we cannot hope to control $|\vartheta|_\infty$ via $\|f_\vartheta\|_{L^2(\mathbb{P}^{\mathbf{X}})}$ in view of the ambiguous network parametrization. As a consequence, we define another critical value at the level of the parameter space

$$\tau_\alpha^\vartheta := \operatorname*{arg\,inf}_{\tau>0} \big\{\widetilde{\Pi}_{\lambda,\rho}(\vartheta : |\vartheta|_\infty \leqslant \Delta(L,r)^{-1}\tau \mid \mathcal{D}_n) > 1-\alpha\big\}.$$

Remark 9. The factor $\Delta(L,r)$ in the definition of τ_α^ϑ could be improved by a different geometry in the parameter space at the cost of a different approximation theory for the resulting network classes. For instance, we may assume that all weight matrices are bounded by B in the ℓ^2-operator norm $\|\cdot\|_2$, which is in line with the weight scaling employed in the theory of neural tangent spaces, cf. [32]. In this case a minor modification of Lemma 17 yields $\|f_\vartheta - f_{\vartheta'}\|_{L^2(\mathbb{P}^{\mathbf{X}})} = \mathcal{O}((2B)^L)\cdot\|\vartheta-\vartheta'\|$ where $\|\vartheta\|$ is defined as the maximal $\|\cdot\|_2$-norm of all weight matrices and all $|\cdot|_2$-norms of the biases. The resulting critical value is given by $\operatorname{arg\,inf}_{\tau>0}\big\{\widetilde{\Pi}_{\lambda,\rho}(\vartheta : \|\vartheta\| \leqslant (2B)^{-L}\tau \mid \mathcal{D}_n) > 1-\alpha\big\}$ avoiding the undesirable dependence on the network width r.

Both critical values measure the fluctuation of the posterior. The theoretical properties of the credible ball are summarized in the following theorem:

Theorem 10 (Credible balls). *Under Assumption A and with constants $Q_0, Q_1, Q_2 > 0$ from above we have for $\lambda = n/(2Q_0)$, r_n^2 from* (3.4) *and sufficiently large n that*

$$\mathbb{P}\Big(\operatorname{diam}\big(\widehat{C}(\tau_\alpha)\big) \leqslant 4\sqrt{2r_n^2 + \frac{4(Q_1\vee Q_2)}{n}\log\frac{2}{\alpha}}\Big) \geqslant 1-\alpha.$$

If the depth L and the width r are chosen such that $L\log(n)\mathcal{E}(f_{\vartheta^*}) = \mathcal{O}(PL\log(n)/\lambda)$, then we have for some constant $\xi > \sqrt{L\log n}$ depending on K, p and α that

$$\mathbb{P}\big(f \in \widehat{C}(\xi\tau_\alpha^\vartheta)\big) \geqslant 1-\alpha.$$

Therefore, the order of the diameter of $\widehat{C}(\tau_\alpha)$ is of the best possible size if L and r are chosen as in Proposition 7. On the other hand, the larger credible set $\widehat{C}(\xi\tau_\alpha^\vartheta)$ defines an honest confidence set for a fixed class $\mathcal{H}(q,\mathbf{d},\mathbf{t},\beta,C_0)$ of the regression function if ξ is chosen sufficiently large depending on the class parameters. That is, $f \in \mathcal{H}(q, \mathbf{d}, \mathbf{t}, \beta, C_0)$ is contained in $\widehat{C}(\xi\tau_\alpha^\vartheta)$ with probability of at least $1-\alpha$. In that sense ξ is a non-asymptotic version of the inflation factor by Rousseau & Szabó [49]. To circumvent the unknown constant ξ, we can conclude from Theorem 10 that for any sequence $a_n \uparrow \infty$, e.g. $a_n = \log n$, we have

$$\mathbb{P}\big(f \in \widehat{C}(a_n\tau_\alpha^\vartheta)\big) \geqslant 1-\alpha \qquad \text{for sufficiently large } n.$$

The condition $L\log(n)\mathcal{E}(f_{\vartheta^*}) = \mathcal{O}(PL\log(n)/\lambda)$ for the coverage result means that the rate is dominated by the stochastic error term and can be achieved with a slightly larger network compared to Proposition 7. This guarantees that the posterior is not underfitting and that the posterior's bias is covered by its dispersal.

4 Learning the width

To balance the approximation error term and the stochastic error term in (3.4), we have to choose an optimal network width. In this section we present a fully data-driven approach to this hyperparameter optimization problem which avoids evaluating competing network architectures on a validation set. To account for the model selection problem, we augment the approach with a mixing prior, which prefers narrower neural networks. Equivalently, this approach can be understood as a hierarchical Bayes method where we put a geometric distribution on the hyperparameter r. While this method has interesting theoretical properties, an efficient implementation is challenging and left for future research.

We set

$$\check{\Pi} = \sum_{r=1}^{n} 2^{-r}\Pi_r \Big/ (1-2^{-n}),$$

where $\Pi_r = \mathcal{U}([-B,B]^{P_r})$ with

$$P_r := (p+1)r + (L-1)(r+1)r + r + 1.$$

The basis 2 of the geometric weights is arbitrary and can be replaced by a larger constant to assign even less weight to wide networks, but the theoretical results remain the same up to constants.

We obtain our adaptive estimator $\check{f}_{\lambda,\rho}$ by drawing a parameter ϑ from the surrogate-posterior distribution with respect to this prior, i.e.

$$\check{f}_{\lambda,\rho} := f_{\check{\vartheta}_{\lambda,\rho}} \qquad \text{for} \qquad \check{\vartheta}_{\lambda,\rho} \mid \mathcal{D}_n \sim \check{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n) \qquad \text{with} \qquad \check{\Pi}_{\lambda,\rho}(\vartheta \mid \mathcal{D}_n) \propto \exp\big(-\lambda\widetilde{R}_{n,\rho}(\vartheta)\big)\check{\Pi}(\mathrm{d}\vartheta).$$

This modification allows the estimator to adapt to the optimal network width and we can compare its performance against that of the network corresponding the oracle choice of the parameter

$$\vartheta_r^* \in \underset{\vartheta\in[-B,B]^{P_r}}{\arg\min}\ R(\vartheta) \tag{4.1}$$

given any width r. We obtain the following adaptive version of Theorem 3:

Theorem 11 (Width-adaptive oracle inequality). *Under Assumption A there is a constant $Q_5 > 0$ depending only on C, Γ, σ such that for $\lambda = n/Q_0$ (with Q_0 from Theorem 3) and sufficiently large n we have for all $\delta \in (0,1)$ with probability of at least $1-\delta$ that*

$$\mathcal{E}(\check{f}_{\lambda,\rho}) \leqslant \min_{r=1,\dots,n}\Big(12\mathcal{E}(f_{\vartheta_r^*}) + \frac{Q_5}{n}\big(P_rL\log(n) + \log(2/\delta)\big)\Big).$$

Since the modified estimator mimics the performance of the optimal network choice regardless of width, we obtain the following width-adaptive version of Proposition 7 with no additional loss in the convergence rate:

Corollary 12 (Width-adaptive rates of convergence)**.** *Let* $\mathbf{X} \in [0,1]^p$*. In the situation of Theorem 11, there exists a network depth* $L = C_3 \log n$ *with* $C_3 > 0$ *only depending on upper bounds for* $q, |\mathbf{d}|_\infty, |\beta|_\infty, C_0$ *such that the estimator* $\check{f}_{\lambda,\rho}$ *satisfies for sufficiently large* n *uniformly over all hierarchical functions* $f \in \mathcal{H}(q, \mathbf{d}, \mathbf{t}, \beta, C_0)$

$$\mathcal{E}(\check{f}_{\lambda,\rho}) \leqslant Q_6 \Big(\frac{(\log n)^3}{n} \Big)^{2\beta^*/(2\beta^*+t^*)} + Q_6 \frac{\log(2/\delta)}{n}$$

with probability of at least $1 - \delta$*, where* β^* *and* t^* *are as in Proposition 7. The constant* Q_6 *only depends on upper bounds for* $q, |\mathbf{d}|_\infty, |\beta|_\infty$ *and* C_0 *as well as the constants* C, Γ, σ *from Assumption A.*

For sparse neural networks, contraction rates for hierarchical Bayes procedures have been analysed by Polson & Ročková [44] and Steffen & Trabs [53]. It has to be noted that we cannot hope to construct credible sets with coverage as in Theorem 10 based on the adaptive posterior distribution. It is well known that adaptive honest confidence sets are only possible under additional assumptions, e.g. self-similarity or polished tail conditions, on the regularity of the regression function, see Hoffmann & Nickl [31] and we remark that such conditions with respect to the network parametrization seem infeasible.

5 Numerical examples

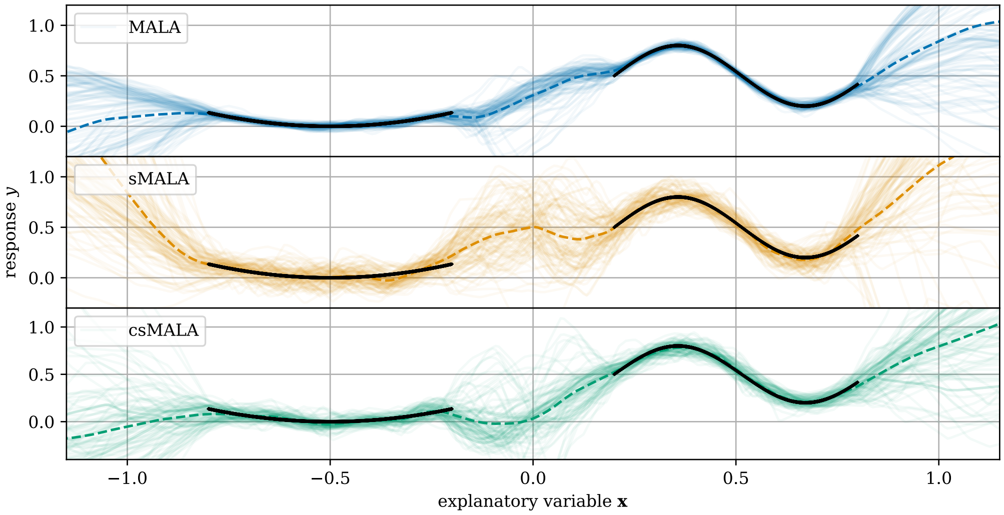


Figure 2: 50 samples drawn from the different MALA chains, given a training sample (black markers) of 10000 points. Random variables are drawn for $\rho = 0.1$. The dashed line shows the corresponding posterior mean $\bar{f}_\lambda$.

Section 2.3 introduces a correction to the batch-wise approximation of the empirical risk when calculating the MH step. In the following, we will show the merit of this correction for learning a one-dimensional regression function using a feed-forward neural network of $L = 2$ layers of $r = 100$ nodes each and ReLU activation. The neural network has a total number of 10401 parameters. The training sample of size 10000 consist of two equally populated intervals $[-0.8, -0.2]$ and $[0.2, 0.8]$ with $\mathbf{X}_i \sim \mathcal{U}([-0.8, -0.2] \cup [0.2, 0.8])$ and true regression function

$$f(x) = \begin{cases} 1.5(x+0.5)^2 & \text{for } x < 0 \\ 0.3 \sin(10x - 2) + 0.5 & \text{for } x \geqslant 0. \end{cases}$$

	MALA	sMALA	csMALA
λ	n	$n \cdot \rho$	$n \cdot (2-\rho)$
γ	10^{-4}	10^{-4}	$10^{-4}/\rho$
s		$0.2/\sqrt{P}$	
b		$100000/\rho$	
c		5000	
N		20	

Table 1: Parameter choice for the different MALA chains. For $\rho = 0.1$, we chose a burn-in of $b = 50000$ to keep computation costs low.

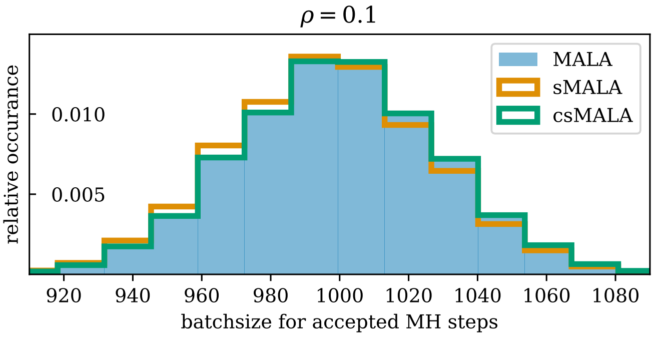


Figure 3: Histogram of the summed auxiliary variables, that is the number of training samples contributing to the stochastic risk, for all accepted steps. For MALA the the MH acceptance step is calculated on the full sample and the distribution of the samples contribution to the risk gradients is thus unbiased by the batch size.

We generate the trainng sample we generate $Y = f(\mathbf{X})+\varepsilon$ by adding an observation error $\varepsilon \sim \mathcal{N}(0, 0.02^2)$. In the interval between -0.2 and 0.2 no data is produced in order to illustrate whether the methods recover the resulting large uncertainty due to missing data. For a sufficiently flexible model we expect a large spread between samples from each Markov chain in this region. Fig. 2 depicts exactly this behaviour, as well as the training sample.

To compare the convergence of MALA, stochastic MALA (sMALA) and our corrected stochastic MALA (csMALA) within reasonable computation time, we initialize the chains with network parameters obtained through optimization of the empirical risk with stochastic gradient descent for 2000 steps. For this pre-training, we use a learning rate of 10^{-3}. The hyperparameters of the subsequent chains are listed in Table 1. The inverse temperature is chosen to counteract the different normalization terms of the risk for (s)MALA and csMALA, as well as the reduction of the learning rate by $(2-\rho)$ through the correction term from Section 2.3. The proposal noise level per parameter dimension is normalized with respect to the number of network parameters such that the total length of the noise vector is independent of the parameter space dimension.

To further improve the efficiency of the sampling, we restart Algorithm 1 with $\vartheta^{(0)}$ set to the last accepted parameters whenever no proposal has been accepted for 100 steps. Especially for small ρ and large ε, the stochastic MH algorithms exhibit the tendency to get stuck after accepting an outlier batch with low risk.

It is also important to adapt ζ such that

$$\zeta \frac{\log \rho}{\lambda} \approx \frac{1}{n} \sum_{i=1}^{n} \ell_i(\vartheta^{(k)}).$$

For ζ lower than this, a bias is introduced towards accepting updates where many points of the data sample contributed to the stochastic risk approximation due to the Bernoulli distributed auxiliary variables. Conversely, for higher values updates are preferably accepted for low amounts of points in the risk approximation. This bias to small batches, note the minus sign due to $\log \rho$, can also be observed for the uncorrected sMALA. It arises from the dependence of R_n on the sum of the drawn auxiliary variables Z_i. Fig. 3 shows a histogram of this sum for all accepted steps. A clear bias for sMALA towards small batches can be seen. To achieve a good correction, we update ζ every 100 steps to fulfill the preceding correspondence. Over the chain, the correction factor thus falls like the empirical risk with $\zeta \ll 1$ due to the proportionality to n^{-1}.

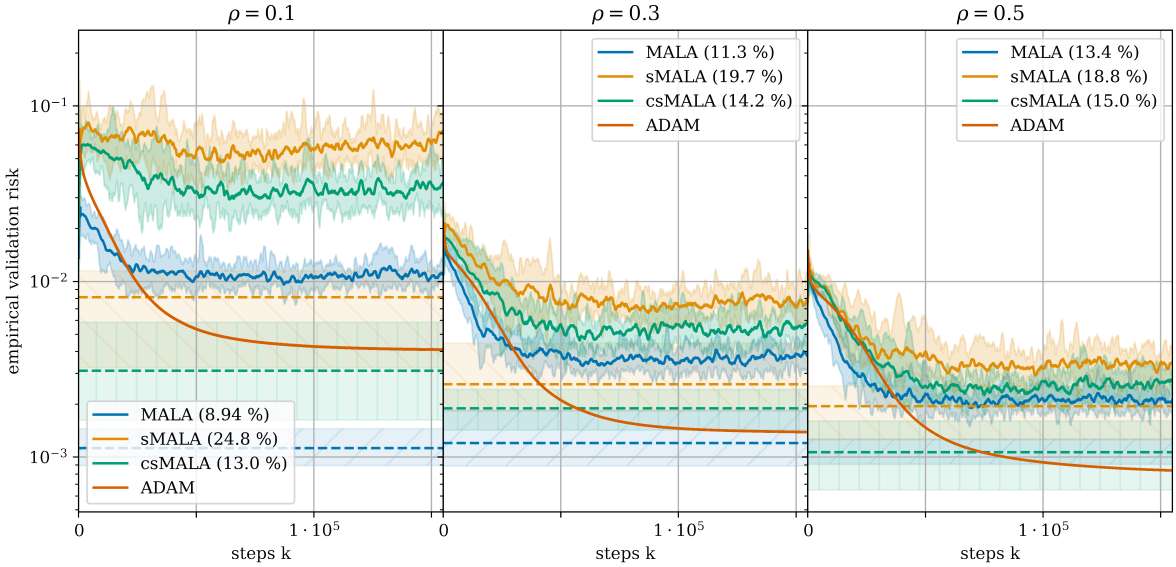


Figure 4: Average empirical risk on a validation set of 10000 points during running of the MALA chains. We show different batch probabilities ρ, as well as the values of the posterior mean (dashed lines). Uncertainties correspond to the minimum and maximum values of 10 identical chains. For clarity, a the simple moving average over 1501 steps is plotted. In the legend, the average acceptance probability over all 10 chains is given. For easier interpretation of the risk values, we also show the behaviour of a gradient-based optimization using `ADAM`.

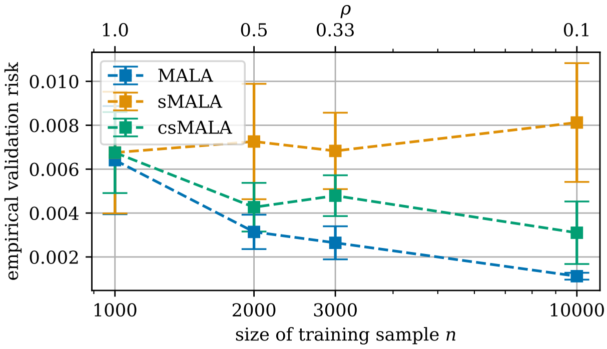
We quantify the performance of the estimators gathered from the different chains with an independent validation sample $\mathcal{D}^{\mathrm{val}}_{n_{\mathrm{val}}} := (\mathbf{X}^{\mathrm{val}}{}_i, Y_i^{\mathrm{val}})_{i=1,\dots,n_{\mathrm{val}}} \subseteq \mathbb{R}^p \times \mathbb{R}$ of size $n_{\mathrm{val}} = 10000$ drawn from the same intervals as the training sample and calculate the empirical validation risk

$$R_n(\widehat{f}) = \frac{1}{n_{\mathrm{val}}} \sum_{i=1}^{n_{\mathrm{val}}} \left(Y_i^{\mathrm{val}} - \widehat{f}(\mathbf{X}_\mathbf{i}^{\mathrm{val}})\right)^2$$

during running of the chain. Fig. 4 illustrates the behaviour of the empirical validation risk for the different MALA algorithms, as well as for a simple inference fit using `ADAM` [33] with a learning rate of 10^{-3}. For a fair comparison, we calculate the gradient updates for all algorithms, including MALA and `ADAM`, from Bernoulli drawn batches, and only calculate the MH step for MALA using the full training sample. We can see, the individual samples of MALA outperform those of the sMALA chains, while the samples from the corrected chain achieve substantially better values than those of the uncorrected stochastic algorithm. On a level of individual samples, all chains are outperformed by the gradient-based optimization using `ADAM`. Investigating the posterior means, MALA outperforms `ADAM` for small ρ where our corrected algorithm reaches similar risk values as the gradient-based optimization. For moderate values of ρ the corrected stochastic MALA restores the performance of the full MH step for both, posterior samples and posterior means, at a level similar to `ADAM`. While the acceptance rates of MALA decrease for low ρ and those of sMALA increase, the acceptance rates of the corrected algorithm are stable under variation of the average batch size.

To study the empirical coverage properties, we calculate 10 individual chains per algorithm and ρ and estimate the credible sets and their average radii. As radius of our credible balls, we approximate the 99.5% quantile $q_{1-\alpha}$ of the mean squared distance to the posterior mean via

$$\tau_{\alpha,n} = q_{1-\alpha}\big((h_1,\dots,h_N)\big) \qquad \text{with} \qquad h_k = \frac{1}{n_{\mathrm{val}}}\sum_{i=1}^{n_{\mathrm{val}}} \left|f_{\vartheta^{(b+ck)}}(\mathrm{X}_i^{\mathrm{val}}) - \bar{f}_{\lambda,\rho}(\mathrm{X}_i^{\mathrm{val}})\right|^2 .$$



ρ	MALA	sMALA	csMALA
0.1	1.42 ± 0.16	13.5 ± 1.4	7.72 ± 0.82
0.3	1.10 ± 0.15	3.70 ± 0.51	2.15 ± 0.23
0.5	1.28 ± 0.11	2.76 ± 0.19	1.91 ± 0.36

Figure 5: Scaling of the empirical risk of the posterior mean $\bar{f}$ on a 10000 point validation set with the size of the training sample. We scale ρ to keep the average batch size $n\rho = 1000$ constant. Errorbars report the standard deviation of 10 identical chains.

Table 2: Average radii $\tau_\alpha \cdot 10^3$ of credible sets for $\alpha = 0.005$ calculated from 10 Monte Carlo chains. All sets show a coverage probability $\widehat{C}(\tau_\alpha)$ of 100%.

To determine the coverage probability, we then calculate the number of chains with a mean squared distance of the posterior mean to the true regression function not exceeding this radius. The results are shown in Table 2. While the uncertainty estimates of all algorithms remain conservative, we find the correction term leads to considerably more precise credible sets.

To illustrate Theorem 3 and Theorem 5, we also investigate the scaling behavior of the empirical validation risk of the posterior means with the training sample size n while keeping $n\rho$ constant. We expect the risk of MALA to fall with growing n, while sMALA should not decay due to the constant $n\rho$. The numerical simulation of Fig. 5 coincides with the theoretical expectations. For our corrected algorithm, we regain the scaling behaviour of MALA as expected.

6 Proofs

We will start with proving the main theorems. Additional proofs of auxiliary results are postponed to Section 6.7 and Section 6.8.

6.1 Compatibility between $\widetilde{R}_{n,\rho}$ and the excess risk

The first step in our analysis is to verify that the empirical risk $\widetilde{R}_{n,\rho}$ which arises from the stochastic MH step is compatible with the excess risk $\mathcal{E}(\vartheta) = \mathbb{E}\big[\big(f(\mathbf{X}_1) - f_\vartheta(\mathbf{X}_1)\big)^2\big]$. More precisely, we require the following concentration inequality. A concentration inequality for the empirical risk $R_n(\vartheta) - R_n(f)$ follows as the special case where $\rho = 1$.

Proposition 13. *Grant Assumption A. Define*

$$\widetilde{\mathcal{E}}_n(\vartheta) \coloneqq \widetilde{R}_{n,\rho}(\vartheta) - \widetilde{R}_{n,\rho}(f).$$

and set $C_{n,\lambda} \coloneqq \frac{\lambda}{n}\frac{8(C^2+\sigma^2)}{1-w\lambda/n}$, $w \coloneqq 16C(\Gamma \vee 2C)$. *Then for all* $\lambda \in [0, n/w) \cap \big[0, \frac{n\log 2}{8(C^2+\sigma^2)}\big]$, $\rho \in (0,1]$ *and* $n \in \mathbb{N}$ *we have*

$$\mathbb{E}\big[\exp\big(\lambda\big(\widetilde{\mathcal{E}}_n(\vartheta) - \mathcal{E}(\vartheta)\big)\big)\big] \leqslant \exp\big(\big(C_{n,\lambda} + \tfrac{\lambda}{n}(\sigma C + \sigma^2)\big)\lambda\mathcal{E}(\vartheta)\big) \qquad \textit{and}$$
$$\mathbb{E}\big[\exp\big(-\lambda\big(\widetilde{\mathcal{E}}_n(\vartheta) - \mathcal{E}(\vartheta)\big)\big)\big] \leqslant \exp\big(\big(C_{n,\lambda} + \tfrac{3}{4} + \tfrac{\lambda}{n}(\sigma C + \sigma^2)\big)\lambda\mathcal{E}(\vartheta)\big).$$

Proof. Define $\psi_\rho(x) \coloneqq -\log\big(\mathrm{e}^{-x} + 1 - \rho\big)$ such that

$$\widetilde{\mathcal{E}}_n(\vartheta) = \frac{1}{\lambda}\sum_{i=1}^n \big(\psi_\rho\big(\tfrac{\lambda}{n}\ell_i(\vartheta)\big) - \psi_\rho\big(\tfrac{\lambda}{n}\ell_i(f)\big)\big).$$

We have

$$\widetilde{\mathcal{E}}_n(\vartheta) = \frac{1}{n}\sum_{i=1}^n \big(\ell_i(\vartheta) - \ell_i(f)\big)\psi_\rho'\big(\xi_i\tfrac{\lambda}{n}\ell_i(\vartheta) + (1-\xi_i)\tfrac{\lambda}{n}\ell_i(f)\big) \tag{6.1}$$

with some random variables $\xi_i \in [0,1]$. Using $\ell_1(\vartheta) - \ell_1(f) = \big(f(\mathbf{X}_1) - f_\vartheta(\mathbf{X}_1)\big)^2 + 2\varepsilon_1\big(f(\mathbf{X}_1) - f_\vartheta(\mathbf{X}_1)\big)$, we can decompose the expectation of (6.1):

$$\begin{aligned}\mathbb{E}\big[\widetilde{\mathcal{E}}_n(\vartheta)\big] &= \mathbb{E}\big[\big(f(\mathbf{X}_1) - f_\vartheta(\mathbf{X}_1)\big)^2\psi_\rho'\big(\xi_1\tfrac{\lambda}{n}\ell_1(\vartheta) + (1-\xi_1)\tfrac{\lambda}{n}\ell_1(f)\big)\big] \\ &\quad + 2\mathbb{E}\big[\varepsilon_1\big(f(\mathbf{X}_1) - f_\vartheta(\mathbf{X}_1)\big)\psi_\rho'\big(\xi_1\tfrac{\lambda}{n}\ell_1(\vartheta) + (1-\xi_1)\tfrac{\lambda}{n}\ell_1(f)\big)\big] \\ &\eqqcolon E_1 + E_2.\end{aligned}$$

We treat both terms separately. We have

$$\begin{aligned}1 \geqslant \psi_\rho'(x) &= (1 + (1-\rho)\mathrm{e}^x)^{-1} \\ &\geqslant \frac{1}{1 + 2(1-\rho)} \geqslant \frac{1}{3} \qquad \text{for } x \in [0, \log 2]\end{aligned}$$

and $\psi_\rho'(x) \in (0,1]$ for all $x \geqslant 0$. In particular, we observe

$$E_1 \leqslant \mathbb{E}\big[\big(f_\vartheta(\mathbf{X}_1) - f(\mathbf{X}_1)\big)^2\big] = \mathcal{E}(\vartheta).$$

If $|\varepsilon_1| \leqslant 2\sigma$, we have $\frac{\lambda}{n}\ell_1(\cdot) \leqslant \frac{\lambda}{n}8(C^2+\sigma^2) \leqslant \log 2$ for $\frac{\lambda}{n} \leqslant \frac{\log 2}{8(C^2+\sigma^2)}$. Hence,

$$\begin{aligned}E_1 &\geqslant \mathbb{E}\big[\big(f(\mathbf{X}_1) - f_\vartheta(\mathbf{X}_1)\big)^2\psi_\rho'\big(\xi_1\tfrac{\lambda}{n}\ell_1(\vartheta) + (1-\xi_1)\tfrac{\lambda}{n}\ell_1(f)\big)\mathbb{1}_{\{|\varepsilon_1| \leqslant 2\sigma\}}\big] \\ &\geqslant \frac{1}{3}\mathbb{E}\big[\big(f(\mathbf{X}_1) - f_\vartheta(\mathbf{X}_1)\big)^2\mathbb{P}(|\varepsilon_1| \leqslant 2\sigma \mid \mathbf{X}_1)\big] \\ &= \frac{1}{3}\mathbb{E}\big[\big(f(\mathbf{X}_1) - f_\vartheta(\mathbf{X}_1)\big)^2\big(1 - \mathbb{P}(|\varepsilon_1| > 2\sigma \mid \mathbf{X}_1)\big)\big] \\ &\geqslant \frac{1}{4}\mathbb{E}\big[\big(f(\mathbf{X}_1) - f_\vartheta(\mathbf{X}_1)\big)^2\big]\end{aligned}$$

where we used Chebyshev's inequality in the last estimate. Hence, $\frac{1}{4}\mathcal{E}(\vartheta) \leqslant E_1 \leqslant \mathcal{E}(\vartheta)$. For E_2 we use $\mathbb{E}[\varepsilon_1\psi_\rho'(\frac{\lambda}{n}\varepsilon_1^2) \mid \mathbf{X}_1] = 0$ by symmetry together with $\ell_1(f) = \varepsilon_1^2$ to obtain for some random $\xi_1' \in [0,1]$

$$\begin{aligned}E_2 &= 2\mathbb{E}\big[\varepsilon_1\big((f(\mathbf{X}_1) - f_\vartheta(\mathbf{X}_1)\big)\big(\psi_\rho'\big(\tfrac{\lambda}{n}\ell_1(f) + \xi_1\tfrac{\lambda}{n}\big(\ell_1(\vartheta) - \ell_1(f)\big)\big) - \psi_\rho'\big(\tfrac{\lambda}{n}\ell_1(f)\big)\big)\big] \\ &= \frac{2\lambda}{n}\mathbb{E}\big[\varepsilon_1\big(f(\mathbf{X}_1) - f_\vartheta(\mathbf{X}_1)\big)\xi_1\big(\ell_1(\vartheta) - \ell_1(f)\big)\psi_\rho''\big(\xi_1'\tfrac{\lambda}{n}\ell_1(\vartheta) + (1-\xi_1')\tfrac{\lambda}{n}\ell_1(f)\big)\big] \\ &= \frac{\lambda}{n}\mathbb{E}\big[2\xi_1\big(\varepsilon_1\big(f(\mathbf{X}_1) - f_\vartheta(\mathbf{X}_1)\big)^3 + 2\varepsilon_1^2\big(f(\mathbf{X}_1) - f_\vartheta(\mathbf{X}_1)\big)^2\big)\psi_\rho''\big(\xi_1'\tfrac{\lambda}{n}\ell_1(\vartheta) + (1-\xi_1')\tfrac{\lambda}{n}\ell_1(f)\big)\big].\end{aligned}$$

Since $\max_{y \geqslant 0}\frac{y}{(1+y)^2} = \frac{1}{4}$, we have

$$|\psi_\rho''(x)| = \frac{(1-\rho)\mathrm{e}^x}{(1+(1-\rho)\mathrm{e}^x)^2} \leqslant \frac{1}{4} \qquad \text{for } x \geqslant 0.$$

Therefore,

$$|E_2| \leqslant \frac{\lambda}{n}\big(\tfrac{1}{2}\mathbb{E}\big[|\varepsilon_1||f_\vartheta(\mathbf{X}_1) - f(\mathbf{X}_1)|^3 + 2\varepsilon_1^2\big(f(\mathbf{X}_1) - f_\vartheta(\mathbf{X}_1)\big)^2\big]\big)$$

$$\leqslant \frac{\lambda}{n}\big(\sigma C+\sigma^2\big)\mathcal{E}(\vartheta).$$

In combination with the bounds for E_1 we obtain

$$\big(\tfrac{1}{4}-\tfrac{\lambda}{n}(\sigma C+\sigma^2)\big)\mathcal{E}(\vartheta)\leqslant \mathbb{E}\big[\widetilde{\mathcal{E}}_n(\vartheta)\big]\leqslant \big(1+\tfrac{\lambda}{n}(\sigma C+\sigma^2)\big)\mathcal{E}(\vartheta).$$

Define $Z_i(\vartheta):=\frac{n}{\lambda}\big(\psi_\rho\big(\frac{\lambda}{n}\ell_i(\vartheta)\big)-\psi_\rho\big(\frac{\lambda}{n}\ell_i(f)\big)\big)$ such that $\widetilde{\mathcal{E}}_n(\vartheta)=\frac{1}{n}\sum_{i=1}^n Z_i(\vartheta)$. The previous bounds for $\mathbb{E}[\widetilde{\mathcal{E}}_n(\vartheta)]$ yield

$$\begin{aligned}\mathbb{E}\big[\exp\big(\lambda\widetilde{\mathcal{E}}_n(\vartheta)-\lambda\mathcal{E}(\vartheta)\big)\big]&=\mathbb{E}\big[\mathrm{e}^{\frac{\lambda}{n}\sum_{i=1}^n(Z_i(\vartheta)-\mathbb{E}[Z_i(\vartheta)])}\big]\mathrm{e}^{\lambda(\mathbb{E}[\widetilde{\mathcal{E}}_n(\vartheta)]-\mathcal{E}(\vartheta))}\\&\leqslant \mathbb{E}\big[\mathrm{e}^{\frac{\lambda}{n}\sum_{i=1}^n(Z_i(\vartheta)-\mathbb{E}[Z_i(\vartheta)])}\big]\mathrm{e}^{\frac{\lambda^2}{n}(\sigma C+\sigma^2)\mathcal{E}(\vartheta)}\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}\big[\exp\big(-\lambda\widetilde{\mathcal{E}}_n(\vartheta)+\lambda\mathcal{E}(\vartheta)\big)\big]&=\mathbb{E}\big[\mathrm{e}^{\frac{\lambda}{n}\sum_{i=1}^n(-Z_i(\vartheta)-\mathbb{E}[-Z_i(\vartheta)])}\big]\mathrm{e}^{\lambda(\mathcal{E}(\vartheta)-\mathbb{E}[\widetilde{\mathcal{E}}_n(\vartheta)])}\\&\leqslant \mathbb{E}\big[\mathrm{e}^{\frac{\lambda}{n}\sum_{i=1}^n(-Z_i(\vartheta)-\mathbb{E}[-Z_i(\vartheta)])}\big]\mathrm{e}^{(\frac{3\lambda}{4}+\frac{\lambda^2}{n}(\sigma C+\sigma^2))\mathcal{E}(\vartheta)}.\end{aligned}$$

To bound the centered exponential moments, we use Bernstein's inequality. The second moments are bounded by

$$\begin{aligned}\mathbb{E}[Z_i^2]&=\mathbb{E}\big[\big(\tfrac{n}{\lambda}\big(\psi_\rho\big(\tfrac{\lambda}{n}\ell_1(\vartheta)\big)-\psi_\rho\big(\tfrac{\lambda}{n}\ell_1(f)\big)\big)\big)^2\big]\\&=\mathbb{E}\big[\big(\big(\ell_1(\vartheta)-\ell_1(f)\big)\psi_\rho'\big(\xi_1\tfrac{\lambda}{n}\ell_1(\vartheta)+(1-\xi_1)\tfrac{\lambda}{n}\ell_1(f)\big)\big)^2\big]\\&=\mathbb{E}\big[\big((f_\vartheta(\mathbf{X}_1)-f(\mathbf{X}_1))^2+2\varepsilon_1(f_\vartheta(\mathbf{X}_1)-f(\mathbf{X}_1))\big)^2(\psi_\rho')^2\big(\xi_1\tfrac{\lambda}{n}\ell_1(\vartheta)+(1-\xi_1)\tfrac{\lambda}{n}\ell_1(f)\big)\big]\\&\leqslant 2\mathbb{E}\big[\big(f_\vartheta(\mathbf{X}_1)-f(\mathbf{X}_1)\big)^4+4\varepsilon_1^2\big(f_\vartheta(\mathbf{X}_1)-f(\mathbf{X}_1)\big)^2\big]\\&\leqslant 8\big(C^2+\sigma^2\big)\mathcal{E}(\vartheta)=:U.\end{aligned}$$

Moreover, we have for $k\geqslant 3$

$$\begin{aligned}\mathbb{E}\big[(Z_i)_+^k\big]&\leqslant \mathbb{E}\big[\big|\ell_1(\vartheta)-\ell_1(f)\big|^k\big|\psi_\rho'\big(\xi_1\tfrac{\lambda}{n}\ell_1(\vartheta)+(1-\xi_1)\tfrac{\lambda}{n}\ell_1(f)\big)\big|^k\big]\\&\leqslant \mathbb{E}\big[\big|\ell_1(\vartheta)-\ell_1(f)\big|^k\big]\\&=\mathbb{E}[|f(\mathbf{X}_1)-f_\vartheta(\mathbf{X}_1)+2\varepsilon_1|^k|f(\mathbf{X}_1)-f_\vartheta(\mathbf{X}_1)|^{k-2}(f(\mathbf{X}_1)-f_\vartheta(\mathbf{X}_1))^2]\\&\leqslant (2C)^{k-2}\mathbb{E}[|f(\mathbf{X}_1)-f_\vartheta(\mathbf{X}_1)+2\varepsilon_1|^k(f(\mathbf{X}_1)-f_\vartheta(\mathbf{X}_1))^2]\\&\leqslant (2C)^{k-2}2^{k-1}((2C)^k+k!2^{k-1}\sigma^2\Gamma^{k-2})\mathcal{E}(\vartheta)\\&\leqslant (2C)^{k-2}k!8^{k-2}\big((2C)^{k-2}\vee\Gamma^{k-2}\big)U\\&=k!Uw^{k-2}.\end{aligned}$$

Hence, Bernstein's inequality [38, inequality (2.21)] yields

$$\mathbb{E}\big[\mathrm{e}^{\frac{\lambda}{n}\sum_{i=1}^n(Z_i(\vartheta)-\mathbb{E}[Z_i(\vartheta)])}\big]\leqslant \exp\Big(\frac{U\lambda^2}{n(1-w\lambda/n)}\Big)=\exp\big(C_{n,\lambda}\lambda\mathcal{E}(\vartheta)\big)$$

for $C_{n,\lambda}$ as defined in Proposition 13. The same bound remains true if we replace Z_i by $-Z_i$. We conclude

$$\mathbb{E}\big[\exp\big(\lambda\widetilde{\mathcal{E}}_n(\vartheta)-\lambda\mathcal{E}(\vartheta)\big)\big]\leqslant \exp\big(\big(C_{n,\lambda}+\tfrac{\lambda}{n}(\sigma C+\sigma^2)\big)\lambda\mathcal{E}(\vartheta)\big)$$

and

$$\mathbb{E}\big[\exp\big(-\lambda\widetilde{\mathcal{E}}_n(\vartheta)+\lambda\mathcal{E}(\vartheta)\big)\big]\leqslant \exp\big(\big(C_{n,\lambda}+\tfrac{3}{4}+\tfrac{\lambda}{n}(\sigma C+\sigma^2)\big)\lambda\mathcal{E}(\vartheta)\big).\qquad\square$$

Remark 14. Replacing ψ_ρ by $\bar{\psi}_\rho(x) := -\log\big(\rho \mathrm{e}^{-x/\rho} + 1 - \rho\big)$, $x \geqslant 0$, and using

$$\begin{aligned}1 \geqslant \bar{\psi}_\rho'(x) &= (\rho + (1-\rho)\mathrm{e}^{x/\rho})^{-1} \\ &\geqslant \frac{1}{\rho + 3(1-\rho)} \geqslant \frac{1}{3} \qquad \text{for } x \in [0, \rho \log 3],\end{aligned}$$

we can analogously prove under Assumption A that $\bar{\mathcal{E}}_n(\vartheta) := \bar{R}_{n,\rho}(\vartheta) - \bar{R}_{n,\rho}(f)$ with $\bar{R}_{n,\rho}$ from (2.10) satisfies for all $\lambda \in [0, n/w) \cap \big[0, \frac{n \log 3}{8(C^2+\sigma^2)}\big]$, $\rho \in (0,1]$ and $n \in \mathbb{N}$:

$$\begin{aligned}\mathbb{E}\big[\exp\big(\lambda\big(\bar{\mathcal{E}}_n(\vartheta) - \mathcal{E}(\vartheta)\big)\big)\big] &\leqslant \exp\big(\big(C_{n,\lambda} + \tfrac{\lambda}{n\rho} 4(\sigma C + \sigma^2)\big)\lambda \mathcal{E}(\vartheta)\big) \qquad \text{and} \\ \mathbb{E}\big[\exp\big(-\lambda\big(\bar{\mathcal{E}}_n(\vartheta) - \mathcal{E}(\vartheta)\big)\big)\big] &\leqslant \exp\big(\big(C_{n,\lambda} + \tfrac{1}{4} + \tfrac{\lambda}{n\rho} 4(\sigma C + \sigma^2)\big)\lambda \mathcal{E}(\vartheta)\big).\end{aligned}$$

6.2 A PAC-Bayes bound

Let μ, ν be probability measures on a measurable space $(E, \mathscr{A})$. The *Kullback-Leibler divergence* of μ with respect to ν is defined via

$$\mathrm{KL}(\mu \mid \nu) := \begin{cases} \int \log\big(\frac{\mathrm{d}\mu}{\mathrm{d}\nu}\big)\,\mathrm{d}\mu, & \text{if } \mu \ll \nu \\ \infty, & \text{otherwise} \end{cases}. \tag{6.2}$$

The following classical lemma is a key ingredient for PAC-Bayes bounds, cf. Catoni [15, p. 159] or Alquier [2]. We include the short proof for the sake of completeness.

Lemma 15. *Let $h\colon E \to \mathbb{R}$ be a measurable function such that $\int \exp \circ h \,\mathrm{d}\mu < \infty$. With the convention $\infty - \infty = -\infty$ it then holds that*

$$\log\Big(\int \exp \circ h \,\mathrm{d}\mu\Big) = \sup_\nu \Big(\int h \,\mathrm{d}\nu - \mathrm{KL}(\nu \mid \mu)\Big), \tag{6.3}$$

where the supremum is taken over all probability measures ν on $(E, \mathscr{A})$. If additionally, h is bounded from above on the support of μ, then the supremum in (6.3) *is attained for $\nu = g$ with the Gibbs distribution g, i.e. $\frac{\mathrm{d}g}{\mathrm{d}\mu} :\propto \exp \circ h$.*

Proof. For $D := \int \mathrm{e}^h \,\mathrm{d}\mu$, we have $\mathrm{d}g = D^{-1}\mathrm{e}^h \mathrm{d}\mu$ and obtain for all $\nu \ll \mu$:

$$\begin{aligned}0 \leqslant \mathrm{KL}(\nu \mid g) = \int \log \frac{\mathrm{d}\nu}{\mathrm{d}g}\,\mathrm{d}\nu &= \int \log \frac{\mathrm{d}\nu}{\mathrm{e}^h \mathrm{d}\mu / D}\,\mathrm{d}\nu \\ &= \mathrm{KL}(\nu \mid \mu) - \int h \,\mathrm{d}\nu + \log\Big(\int \mathrm{e}^h \,\mathrm{d}\mu\Big).\end{aligned}$$ □

Note that no generality is lost by considering only those probability measures ν on $(E, \mathscr{A})$ such that $\nu \ll \mu$ and thus

$$\log\Big(\int \exp \circ h \,\mathrm{d}\mu\Big) = -\inf_{\nu \ll \mu}\Big(\mathrm{KL}(\nu \mid \mu) - \int h \,\mathrm{d}\nu\Big).$$

In combination with Proposition 13 we can verify a PAC-Bayes bound for the excess risk. The basic proof strategy is in line with the PAC-Bayes literature, see e.g. Alquier & Biau [3].

Proposition 16 (PAC-Bayes bound)**.** *Grant Assumption A. For any sample-dependent (in a measurable way) probability measure $\varrho \ll \Pi$ and any $\lambda \in (0, n/w)$ and $\rho \in (0,1]$ such that $C_{n,\lambda} + \frac{\lambda}{n}(\sigma C + \sigma^2) \leqslant \frac{1}{8}$, we have*

$$\mathcal{E}(\widetilde{\vartheta}_{\lambda,\rho}) \leqslant 9 \int \mathcal{E}\,\mathrm{d}\varrho + \frac{16}{\lambda}\big(\mathrm{KL}(\varrho \mid \Pi) + \log(2/\delta)\big) \tag{6.4}$$

with probability of at least $1 - \delta$.

Proof. Proposition 13 yields

$$\begin{aligned}&\mathbb{E}\big[\exp\big(\lambda\widetilde{\mathcal{E}}_n(\vartheta)-\big(1+C_{n,\lambda}+\tfrac{\lambda}{n}(\sigma C+\sigma^2)\big)\lambda\mathcal{E}(\vartheta)-\log\delta^{-1}\big)\big]\leqslant\delta \qquad \text{and}\\ &\mathbb{E}\big[\exp\big(\lambda\big(\tfrac14-C_{n,\lambda}-\tfrac{\lambda}{n}(\sigma C+\sigma^2)\big)\mathcal{E}(\vartheta)-\lambda\widetilde{\mathcal{E}}_n(\vartheta)-\log\delta^{-1}\big)\big]\leqslant\delta.\end{aligned}$$

Integrating in ϑ with respect to the prior probability measure Π and applying Fubini's theorem, we conclude

$$\begin{aligned}&\mathbb{E}\Big[\int\exp\big(\lambda\widetilde{\mathcal{E}}_n(\vartheta)-\big(1+C_{n,\lambda}+\tfrac{\lambda}{n}(\sigma C+\sigma^2)\big)\lambda\mathcal{E}(\vartheta)-\log\delta^{-1}\big)\,\mathrm{d}\Pi(\vartheta)\Big]\leqslant\delta \qquad \text{and} \qquad (6.5)\\ &\mathbb{E}\Big[\int\exp\big(\lambda\big(\tfrac14-C_{n,\lambda}-\tfrac{\lambda}{n}(\sigma C+\sigma^2)\big)\mathcal{E}(\vartheta)-\lambda\widetilde{\mathcal{E}}_n(\vartheta)-\log\delta^{-1}\big)\,\mathrm{d}\Pi(\vartheta)\Big]\leqslant\delta.\end{aligned}$$

The Radon-Nikodym density of the posterior distribution $\widetilde{\Pi}_{\lambda,\rho}(\cdot\mid\mathcal{D}_n)\ll\Pi$ with respect to Π is given by

$$\frac{\mathrm{d}\widetilde{\Pi}_{\lambda,\rho}(\vartheta\mid\mathcal{D}_n)}{\mathrm{d}\Pi}=\widetilde{D}_\lambda^{-1}\exp\Big(-\sum_{i=1}^n\psi_\rho\big(\tfrac{\lambda}{n}\ell_i(\vartheta)\big)\Big)$$

with

$$\widetilde{D}_\lambda\coloneqq\int\mathrm{e}^{-\lambda\widetilde{R}_{n,\rho}(\vartheta)}\,\Pi(\mathrm{d}\vartheta)=\int\exp\Big(-\sum_{i=1}^n\psi_\rho\big(\tfrac{\lambda}{n}\ell_i(\vartheta)\big)\Big)\,\Pi(\mathrm{d}\vartheta). \qquad (6.6)$$

We obtain

$$\begin{aligned}\delta&\geqslant\mathbb{E}_{\mathcal{D}_n}\Big[\int\exp\big(\lambda\big(\tfrac14-C_{n,\lambda}-\tfrac{\lambda}{n}(\sigma C+\sigma^2)\big)\mathcal{E}(\vartheta)-\lambda\widetilde{\mathcal{E}}_n(\vartheta)-\log\delta^{-1}\big)\,\mathrm{d}\Pi(\vartheta)\Big]\\ &=\mathbb{E}_{\mathcal{D}_n,\widetilde{\vartheta}\sim\widetilde{\Pi}_{\lambda,\rho}(\cdot|\mathcal{D}_n)}\Big[\exp\Big(\lambda\big(\tfrac14-C_{n,\lambda}-\tfrac{\lambda}{n}(\sigma C+\sigma^2)\big)\mathcal{E}(\widetilde{\vartheta})-\lambda\widetilde{\mathcal{E}}_n(\widetilde{\vartheta})\\ &\qquad\qquad -\log\delta^{-1}-\log\Big(\frac{\mathrm{d}\widetilde{\Pi}_{\lambda,\rho}(\widetilde{\vartheta}\mid\mathcal{D}_n)}{\mathrm{d}\Pi}\Big)\Big)\Big]\\ &=\mathbb{E}_{\mathcal{D}_n,\widetilde{\vartheta}\sim\widetilde{\Pi}_{\lambda,\rho}(\cdot|\mathcal{D}_n)}\Big[\exp\Big(\lambda\big(\tfrac14-C_{n,\lambda}-\tfrac{\lambda}{n}(\sigma C+\sigma^2)\big)\mathcal{E}(\widetilde{\vartheta})-\lambda\widetilde{\mathcal{E}}_n(\widetilde{\vartheta})\\ &\qquad\qquad -\log\delta^{-1}+\sum_{i=1}^n\psi_\rho\big(\tfrac{\lambda}{n}\ell_i(\widetilde{\vartheta})\big)+\log\widetilde{D}_\lambda\Big)\Big]\end{aligned}$$

Since $\mathbb{1}_{[0,\infty)}(x)\leqslant\mathrm{e}^{\lambda x}$ for all $x\in\mathbb{R}$, we deduce with probability not larger than δ that

$$\big(\tfrac14-C_{n,\lambda}-\tfrac{\lambda}{n}(\sigma C+\sigma^2)\big)\mathcal{E}(\widetilde{\vartheta})-\widetilde{\mathcal{E}}_n(\widetilde{\vartheta})+\frac{1}{\lambda}\sum_{i=1}^n\psi_\rho\big(\tfrac{\lambda}{n}\ell_i(\widetilde{\vartheta})\big)-\frac{1}{\lambda}\big(\log\delta^{-1}-\log\widetilde{D}_\lambda\big)\geqslant0.$$

Provided $C_{n,\lambda}+\frac{\lambda}{n}(\sigma C+\sigma^2)\leqslant\frac18$, we thus have for $\widetilde{\vartheta}\sim\widetilde{\Pi}_{\lambda,\rho}(\cdot\mid\mathcal{D}_n)$ with probability of at least $1-\delta$:

$$\begin{aligned}\mathcal{E}(\widetilde{\vartheta})&\leqslant8\Big(\widetilde{\mathcal{E}}_n(\widetilde{\vartheta})-\frac{1}{\lambda}\sum_{i=1}^n\psi_\rho\big(\tfrac{\lambda}{n}\ell_i(\widetilde{\vartheta})\big)+\frac{1}{\lambda}\big(\log\delta^{-1}-\log\widetilde{D}_\lambda\big)\Big)\\ &\leqslant8\Big(-\frac{1}{\lambda}\sum_{i=1}^n\psi_\rho\big(\tfrac{\lambda}{n}\ell_i(f)\big)+\frac{1}{\lambda}\big(\log\delta^{-1}-\log\widetilde{D}_\lambda\big)\Big)\end{aligned}$$

Lemma 15 with $h=-\sum_{i=1}^n\psi_\rho(\frac{\lambda}{n}\ell_i(\vartheta))$ yields

$$\log\widetilde{D}_\lambda=\log\Big(\int\exp\big(-\sum_{i=1}^n\psi_\rho\big(\tfrac{\lambda}{n}\ell_i(\vartheta)\big)\big)\,\mathrm{d}\Pi(\vartheta)\Big)=-\inf_{\varrho\ll\Pi}\Big(\mathrm{KL}(\varrho\mid\Pi)+\int\sum_{i=1}^n\psi_\rho\big(\tfrac{\lambda}{n}\ell_i(\vartheta)\big)\,\mathrm{d}\varrho(\vartheta)\Big). \qquad (6.7)$$

Therefore, we have with probability of at least $1-\delta$:

$$\begin{aligned}\mathcal{E}(\widetilde{\vartheta}) &\leqslant 8 \inf_{\varrho \ll \Pi} \Big(\int \frac{1}{\lambda} \sum_{i=1}^{n} \big(\psi_\rho\big(\tfrac{\lambda}{n}\ell_i(\vartheta)\big) - \psi_\rho\big(\tfrac{\lambda}{n}\ell_i(f)\big)\big)\,\mathrm{d}\varrho(\vartheta) + \frac{1}{\lambda}\big(\log\delta^{-1} + \mathrm{KL}(\varrho \mid \Pi)\big)\Big) \\ &\leqslant 8 \inf_{\varrho \ll \Pi} \Big(\int \widetilde{\mathcal{E}}_n(\vartheta)\,\mathrm{d}\varrho(\vartheta) + \frac{1}{\lambda}\big(\log\delta^{-1} + \mathrm{KL}(\varrho \mid \Pi)\big)\Big).\end{aligned}$$

In order to reduce the integral $\int \widetilde{\mathcal{E}}_n(\vartheta)\,\mathrm{d}\varrho(\vartheta)$ to $\int \mathcal{E}(\vartheta)\,\mathrm{d}\varrho(\vartheta)$, we use $C_{n,\lambda} + \frac{\lambda}{n}(\sigma C + \sigma^2) \leqslant \frac{1}{8}$, Jensen's inequality and (6.5) to obtain for any probability measure $\varrho \ll \Pi$ (which may depend on $\mathcal{D}_n$)

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}_n}\Big[\exp\Big(\int \big(\lambda\widetilde{\mathcal{E}}_n(\vartheta) - \tfrac{9}{8}\lambda\mathcal{E}(\vartheta)\big)\,\mathrm{d}\varrho(\vartheta) - \mathrm{KL}(\varrho \mid \Pi) - \log\delta^{-1}\Big)\Big] \\ &\quad= \mathbb{E}_{\mathcal{D}_n}\Big[\exp\Big(\int \lambda\widetilde{\mathcal{E}}_n(\vartheta) - \tfrac{9}{8}\lambda\mathcal{E}(\vartheta) - \log\Big(\frac{\mathrm{d}\varrho}{\mathrm{d}\Pi}(\vartheta)\Big) - \log\delta^{-1}\,\mathrm{d}\varrho(\vartheta)\Big)\Big] \\ &\quad\leqslant \mathbb{E}_{\mathcal{D}_n,\vartheta\sim\varrho}\Big[\exp\Big(\lambda\widetilde{\mathcal{E}}_n(\vartheta) - \tfrac{9}{8}\lambda\mathcal{E}(\vartheta) - \log\Big(\frac{\mathrm{d}\varrho}{\mathrm{d}\Pi}(\vartheta)\Big) - \log\delta^{-1}\Big)\Big] \\ &\quad\leqslant \mathbb{E}_{\mathcal{D}_n}\Big[\int \exp\big(\lambda\widetilde{\mathcal{E}}_n(\vartheta) - \big(1 + C_{n,\lambda} + \tfrac{\lambda}{n}(\sigma C + \sigma^2)\big)\lambda\mathcal{E}(\vartheta) - \log\delta^{-1}\big)\,\mathrm{d}\Pi(\vartheta)\Big] \leqslant \delta.\end{aligned}$$

Using $\mathbb{1}_{[0,\infty)}(x) \leqslant \mathrm{e}^{\lambda x}$ again, we conclude with probability of at least $1-\delta$:

$$\int \widetilde{\mathcal{E}}_n(\vartheta)\,\mathrm{d}\varrho(\vartheta) \leqslant \frac{9}{8}\int \mathcal{E}(\vartheta)\,\mathrm{d}\varrho(\vartheta) + \lambda^{-1}\big(\mathrm{KL}(\varrho \mid \Pi) + \log\delta^{-1}\big).$$

Therefore, we conclude with probability of at least $1-2\delta$

$$\mathcal{E}(\widetilde{\vartheta}) \leqslant 9\int \mathcal{E}(\vartheta)\,\mathrm{d}\varrho(\vartheta) + \frac{16}{\lambda}\big(\mathrm{KL}(\varrho \mid \Pi) + \log\delta^{-1}\big). \qquad \square$$

6.3 Proof of Theorem 3

We fix a radius $\eta \in (0,1]$ and apply Proposition 16 with $\varrho = \varrho_\eta$ defined via

$$\frac{\mathrm{d}\varrho_\eta}{\mathrm{d}\Pi}(\vartheta) \propto \mathbb{1}_{\{|\vartheta-\vartheta^*|_\infty \leqslant \eta\}}$$

with ϑ^* from (3.2). Note that indeed $C_{n,\lambda} + \frac{\lambda}{n}(\sigma C + \sigma^2) \leqslant \frac{1}{8}$ for Q_0 sufficiently large. In order to control the integral term, we decompose

$$\begin{aligned}\int \mathcal{E}\,\mathrm{d}\varrho_\eta &= \mathcal{E}(\vartheta^*) + \int \mathbb{E}\big[(f_\vartheta(\mathbf{X}) - f(\mathbf{X}))^2 - (f_{\vartheta^*}(\mathbf{X}) - f(\mathbf{X}))^2\big]\,\mathrm{d}\varrho_\eta(\vartheta) \\ &= \mathcal{E}(\vartheta^*) + \int \mathbb{E}\big[(f_{\vartheta^*}(\mathbf{X}) - f_\vartheta(\mathbf{X}))^2\big]\,\mathrm{d}\varrho_\eta(\vartheta) + 2\int \mathbb{E}\big[(f(\mathbf{X}) - f_{\vartheta^*}(\mathbf{X}))(f_{\vartheta^*}(\mathbf{X}) - f_\vartheta(\mathbf{X}))\big]\,\mathrm{d}\varrho_\eta(\vartheta) \\ &\leqslant \mathcal{E}(\vartheta^*) + \int \mathbb{E}\big[(f_{\vartheta^*}(\mathbf{X}) - f_\vartheta(\mathbf{X}))^2\big]\,\mathrm{d}\varrho_\eta(\vartheta) \\ &\qquad + 2\int \mathbb{E}\big[(f(\mathbf{X}) - f_{\vartheta^*}(\mathbf{X}))^2\big]^{1/2}\mathbb{E}\big[(f_{\vartheta^*}(\mathbf{X}) - f_\vartheta(\mathbf{X}))^2\big]^{1/2}\,\mathrm{d}\varrho_\eta(\vartheta) \\ &\leqslant \frac{4}{3}\mathcal{E}(\vartheta^*) + 4\int \mathbb{E}\big[(f_{\vartheta^*}(\mathbf{X}) - f_\vartheta(\mathbf{X}))^2\big]\,\mathrm{d}\varrho_\eta(\vartheta), \qquad (6.8)\end{aligned}$$

using $2ab \leqslant \frac{a^2}{3} + 3b^2$ in the last step. To bound the remainder, we use the Lipschitz continuity of the map $\vartheta \mapsto f_\vartheta(\mathbf{x})$ for fixed $\mathbf{x} \in \mathbb{R}^p$:

Lemma 17. *Let $\vartheta, \widetilde{\vartheta} \in [-B,B]^P$. Then we have for $\mathbf{x} \in \mathbb{R}^p$ that*

$$|f_\vartheta(\mathbf{x}) - f_{\widetilde{\vartheta}}(\mathbf{x})| \leqslant 4(2rB)^L(|\mathbf{x}|_1 \vee 1)|\vartheta - \widetilde{\vartheta}|_\infty.$$

We obtain

$$\int \mathcal{E}\,\mathrm{d}\varrho_\eta \leqslant \frac{4}{3}\mathcal{E}(\vartheta^*) + \frac{4}{n^2} \qquad \text{for} \qquad \eta = \frac{1}{8K(2rB)^L p\, n}. \tag{6.9}$$

It remains to bound the Kullback-Leibler term in (6.4) which can be done with the following lemma:

Lemma 18. *We have* $\mathrm{KL}(\varrho_\eta \mid \Pi) \leqslant P\log(2B/\eta)$.

Plugging (6.9) and the bound from Lemma 18 into the PAC-Bayes bound (6.4), we conclude

$$\begin{aligned}\mathcal{E}(\widetilde{\vartheta}_{\lambda,\rho}) &\leqslant 12\mathcal{E}(\vartheta^*) + \frac{36}{n^2} + \frac{16}{\lambda}\big(P\log\big(16BK(2rB)^L pn\big) + \log(2/\delta)\big).\\ &\leqslant 12\mathcal{E}(\vartheta^*) + \frac{Q_1}{n}\big(PL\log(n) + \log(2/\delta)\big)\end{aligned}$$

for some constant Q_1 only depending on C, σ, Γ. □

6.4 Proof of Theorem 5

Due to Remark 14 we can prove analogously to Proposition 16 the following PAC-Bayes bound under Assumption A: For any sample-dependent (in a measurable way) probability measure $\varrho \ll \Pi$ and any $\lambda \in (0, n/w)$ and $\rho \in (0,1]$ such that $C_{n,\lambda} + \frac{\lambda}{n\rho}4(\sigma C + \sigma^2) \leqslant \frac{1}{4}$, we have

$$\mathcal{E}(\widehat{\vartheta}_\lambda) \leqslant \frac{5}{2}\int \mathcal{E}\,\mathrm{d}\varrho + \frac{4}{\lambda}\big(\mathrm{KL}(\varrho \mid \Pi) + \log(2/\delta)\big)$$

with probability of at least $1-\delta$. From here we can continue as in Section 6.3. □

6.5 Proof of Theorem 10

Choosing $\lambda = \frac{n}{2Q_0}$, Theorem 3 and Corollary 6 yield

$$\min\big\{\mathbb{E}[\widetilde{\Pi}_{\lambda,\rho}(\vartheta : \|f_\vartheta - f\|_{L^2(\mathbb{P}^{\mathbf{X}})} \leqslant s_n \mid \mathcal{D}_n)], \mathbb{P}(\|f - \bar{f}_{\lambda,\rho}\|_{L^2(\mathbb{P}^{\mathbf{X}})} \leqslant s_n)\big\} \geqslant 1 - \frac{\alpha^2}{2}$$

with $s_n^2 \coloneqq 2r_n^2 + \frac{4(Q_1 \vee Q_2)}{n}\log\frac{2}{\alpha}$. We conclude

$$\begin{aligned}\mathbb{P}\big(\operatorname{diam}(\widehat{C}(\tau_\alpha)) \leqslant 4s_n\big) &= \mathbb{P}\Big(\sup_{g,h\in\widehat{C}(\tau_\alpha)} \|g-h\|_{L^2(\mathbb{P}^{\mathbf{X}})} \leqslant 4s_n\Big)\\ &\geqslant \mathbb{P}\Big(\sup_{g,h\in\widehat{C}(\tau_\alpha)} \|g-\bar{f}_{\lambda,\rho}\|_{L^2(\mathbb{P}^{\mathbf{X}})} + \|\bar{f}_{\lambda,\rho} - h\|_{L^2(\mathbb{P}^{\mathbf{X}})} \leqslant 4s_n\Big)\\ &\geqslant \mathbb{P}\big(\tau_\alpha \leqslant 2s_n\big)\\ &= \mathbb{P}\big(\widetilde{\Pi}_{\lambda,\rho}(\vartheta : \|f_\vartheta - \bar{f}_{\lambda,\rho}\|_{L^2(\mathbb{P}^{\mathbf{X}})} \leqslant 2s_n \mid \mathcal{D}_n) > 1-\alpha\big)\\ &\geqslant \mathbb{P}\big(\widetilde{\Pi}_{\lambda,\rho}(\vartheta : \|f_\vartheta - \bar{f}_{\lambda,\rho}\|_{L^2(\mathbb{P}^{\mathbf{X}})} > 2s_n \mid \mathcal{D}_n) < \alpha\big)\\ &= 1 - \mathbb{P}\big(\widetilde{\Pi}_{\lambda,\rho}(\vartheta : \|f_\vartheta - \bar{f}_{\lambda,\rho}\|_{L^2(\mathbb{P}^{\mathbf{X}})} > 2s_n \mid \mathcal{D}_n) \geqslant \alpha\big)\\ &\geqslant 1 - \alpha^{-1}\mathbb{E}\big[\widetilde{\Pi}_{\lambda,\rho}(\vartheta : \|f_\vartheta - \bar{f}_{\lambda,\rho}\|_{L^2(\mathbb{P}^{\mathbf{X}})} > 2s_n \mid \mathcal{D}_n)\big]\\ &\geqslant 1 - \alpha^{-1}\big(\mathbb{E}\big[\widetilde{\Pi}_{\lambda,\rho}(\vartheta : \|f_\vartheta - f\|_{L^2(\mathbb{P}^{\mathbf{X}})} > s_n \mid \mathcal{D}_n)\big] + \mathbb{P}\big(\|\bar{f}_{\lambda,\rho} - f\|_{L^2(\mathbb{P}^{\mathbf{X}})} > s_n\big)\big)\\ &\geqslant 1-\alpha.\end{aligned}$$

The first statement in Theorem 10 is thus verified.

For the coverage statement, we denote $\bar{\xi} \coloneqq \xi\Delta(L,r) = \xi(2rB)^L$ and bound

$$\mathbb{P}\big(f \in \widehat{C}(\xi\tau_\alpha^\vartheta)\big) = \mathbb{P}\big(\|f - \bar{f}_{\lambda,\rho}\|_{L^2(\mathbb{P}^{\mathbf{X}})} \leqslant \xi\tau_\alpha^\vartheta\big)$$

$$
\begin{aligned}
&\geqslant \mathbb{P}\big(\widetilde{\Pi}_{\lambda,\rho}(\vartheta : |\vartheta|_\infty \leqslant \bar{\xi}^{-1}\|f-\bar{f}_{\lambda,\rho}\|_{L^2(\mathbb{P}^{\mathbf{X}})} \mid \mathcal{D}_n) < 1-\alpha\big)\\
&\geqslant \mathbb{P}\big(\widetilde{\Pi}_{\lambda,\rho}(\vartheta : |\vartheta|_\infty \leqslant \bar{\xi}^{-1}s_n \mid \mathcal{D}_n) < 1-\alpha\big) - \alpha^2\\
&= 1-\alpha^2 - \mathbb{P}\big(\widetilde{\Pi}_{\lambda,\rho}(\vartheta : |\vartheta|_\infty \leqslant \bar{\xi}^{-1}s_n \mid \mathcal{D}_n) \geqslant 1-\alpha\big)\\
&\geqslant 1-\alpha^2-(1-\alpha)^{-1}\mathbb{E}\big[\widetilde{\Pi}_{\lambda,\rho}(B_n \mid \mathcal{D}_n)\big]
\end{aligned}
$$

with

$$B_n \coloneqq \big\{\vartheta : |\vartheta|_\infty \leqslant \bar{\xi}^{-1}s_n\big\}.$$

In terms of $\widetilde{\mathcal{E}}_n(\vartheta)=\widetilde{R}_{n,\rho}(\vartheta)-\widetilde{R}_{n,\rho}(f)$ and $\widetilde{D}_\lambda=\int\exp\big(-\lambda\widetilde{R}_{n,\rho}(\vartheta)\big)\,\Pi(\mathrm{d}\vartheta)$ the inequalities by Cauchy-Schwarz and Jensen imply

$$
\begin{aligned}
\mathbb{E}\big[\widetilde{\Pi}_{\lambda,\rho}(B_n \mid \mathcal{D}_n)\big] &= \mathbb{E}\Big[\widetilde{D}_\lambda^{-1}\int_{B_n}\mathrm{e}^{-\lambda\widetilde{R}_{n,\rho}(\vartheta)}\,\Pi(\mathrm{d}\vartheta)\Big]\\
&= \mathbb{E}\Big[\widetilde{D}_\lambda^{-1}\mathrm{e}^{-\lambda\widetilde{R}_{n,\rho}(f)}\int_{B_n}\mathrm{e}^{-\lambda\widetilde{\mathcal{E}}_n(\vartheta)}\,\Pi(\mathrm{d}\vartheta)\Big]\\
&\leqslant \mathbb{E}\big[\widetilde{D}_\lambda^{-2}\mathrm{e}^{-2\lambda\widetilde{R}_{n,\rho}(f)}\big]^{1/2}\mathbb{E}\Big[\Big(\int_{B_n}\mathrm{e}^{-\lambda\widetilde{\mathcal{E}}_n(\vartheta)}\,\Pi(\mathrm{d}\vartheta)\Big)^2\Big]^{1/2}\\
&\leqslant \mathbb{E}\big[\widetilde{D}_\lambda^{-2}\mathrm{e}^{-2\lambda\widetilde{R}_{n,\rho}(f)}\big]^{1/2}\mathbb{E}\Big[\Pi(B_n)\int_{B_n}\mathrm{e}^{-2\lambda\widetilde{\mathcal{E}}_n(\vartheta)}\,\Pi(\mathrm{d}\vartheta)\Big]^{1/2}.
\end{aligned}
$$

The smaller choice of $\lambda=n/(2Q_0)$ instead of n/Q_0 ensures $C_{n,2\lambda}+\frac{2\lambda}{n}(\sigma C+\sigma^2)\leqslant\frac{1}{8}$ allowing us to apply Proposition 13 with 2λ. With Fubini's theorem and the uniform distribution of the prior, the second factor can thus be bounded using

$$
\begin{aligned}
\mathbb{E}\Big[\int_{B_n}\mathrm{e}^{-2\lambda\widetilde{\mathcal{E}}_n(\vartheta)}\,\Pi(\mathrm{d}\vartheta)\Big] &= \int_{B_n}\mathbb{E}\big[\mathrm{e}^{-2\lambda\widetilde{\mathcal{E}}_n(\vartheta)}\big]\,\Pi(\mathrm{d}\vartheta)\\
&\leqslant \int_{B_n}\exp\big(2\big(C_{n,2\lambda}+\tfrac{3}{4}+\tfrac{2\lambda}{n}(\sigma C+\sigma^2)-1\big)\lambda\mathcal{E}(\vartheta)\big)\,\Pi(\mathrm{d}\vartheta)\\
&\leqslant \Pi(B_n)\\
&\leqslant \exp\big(P\log\frac{s_n}{B\bar{\xi}}\big).
\end{aligned}
$$

Based on (6.7), we conclude

$$
\begin{aligned}
\mathbb{E}\big[\widetilde{\Pi}_{\lambda,\rho}(B_n \mid \mathcal{D}_n)\big] &\leqslant \exp\big(P\log\frac{s_n}{B\bar{\xi}}\big)\mathbb{E}\big[\widetilde{D}_\lambda^{-2}\mathrm{e}^{-2\lambda\widetilde{R}_{n,\rho}(f)}\big]^{1/2}\\
&= \exp\big(P\log\frac{s_n}{B\bar{\xi}}\big)\mathbb{E}\Big[\exp\Big(\inf_{\varrho\ll\Pi}\Big(2\,\mathrm{KL}(\varrho\mid\Pi)+2\int\lambda\widetilde{R}_{n,\rho}(\vartheta)\,\mathrm{d}\varrho(\vartheta)\Big)-2\lambda\widetilde{R}_{n,\rho}(f)\Big)\Big]^{1/2}\\
&= \exp\big(P\log\frac{s_n}{B\bar{\xi}}\big)\mathbb{E}\Big[\exp\Big(\inf_{\varrho\ll\Pi}\Big(2\,\mathrm{KL}(\varrho\mid\Pi)+\int 2\lambda\widetilde{\mathcal{E}}_n(\vartheta)\,\mathrm{d}\varrho(\vartheta)\Big)\Big)\Big]^{1/2}.
\end{aligned}
$$

For $\varrho_{\eta'}$ defined via

$$\frac{\mathrm{d}\varrho_{\eta'}}{\mathrm{d}\Pi}(\vartheta)\propto\mathbb{1}_{\{|\vartheta-\vartheta^*|_\infty\leqslant\eta'\}},\qquad \eta'=\frac{s_n}{8K\Delta(L,r)p\sqrt{L\log n}}.$$

we can moreover estimate with (6.8), Lemma 17 and Lemma 18

$$
\begin{aligned}
\inf_{\varrho\ll\Pi}\Big(\mathrm{KL}(\varrho\mid\Pi)+\int\lambda\widetilde{\mathcal{E}}_n(\vartheta)\,\mathrm{d}\varrho(\vartheta)\Big) &\leqslant \mathrm{KL}(\varrho_{\eta'}\mid\Pi)+\frac{4}{3}\lambda\mathcal{E}(\vartheta^*)+3\lambda\int\mathbb{E}\big[\big(f_{\vartheta^*}(\mathbf{X})-f_\vartheta(\mathbf{X})\big)^2\big]\,\mathrm{d}\varrho_{\eta'}(\vartheta)\\
&\qquad+\lambda\int\big(\widetilde{\mathcal{E}}_n(\vartheta)-\mathcal{E}(\vartheta)\big)\,\mathrm{d}\varrho_{\eta'}(\vartheta)\\
&\leqslant P\log\frac{2B}{\eta'}+\frac{4}{3}\lambda\mathcal{E}(\vartheta^*)+3L^{-1}\lambda s_n^2+\lambda\int\big(\widetilde{\mathcal{E}}_n(\vartheta)-\mathcal{E}(\vartheta)\big)\,\mathrm{d}\varrho_{\eta'}(\vartheta).
\end{aligned}
$$

In the sequel $Q_i > 0, i = 7, 8, \dots,$ are numerical constants which may depend on C, Γ, σ, K, p and α. Since $L\log(n)\mathcal{E}(\vartheta^*) \leqslant s_n^2 \leqslant Q_7 PL\log(n)/\lambda$ by assumption, we obtain

$$\begin{aligned}\mathbb{E}\big[\widetilde{\Pi}_{\lambda,\rho}(B_n \mid \mathcal{D}_n)\big] &\leqslant \exp\big(-P\log\bar{\xi} + P\log\big(16K\Delta(L,r)p\sqrt{L\log n}\big) + 5Q_7P\big)\\ &\qquad\times \mathbb{E}\Big[\exp\Big(2\lambda\int\big(\widetilde{\mathcal{E}}_n(\vartheta) - \mathcal{E}(\vartheta)\big)\,\mathrm{d}\varrho_{\eta'}(\vartheta)\Big)\Big]^{1/2}\\ &\leqslant \exp\big(-P\log\xi + P(Q_8 + \log\sqrt{L\log n})\big)\mathbb{E}\Big[\int\exp\big(2\lambda\big(\widetilde{\mathcal{E}}_n(\vartheta) - \mathcal{E}(\vartheta)\big)\big)\,\mathrm{d}\varrho_{\eta'}(\vartheta)\Big]^{1/2}\end{aligned}$$

applying Jensen's inequality in the last line. To bound the expectation in the previous line, Fubini's theorem, Proposition 13 with $C_{n,2\lambda} + \frac{2\lambda}{n}(\sigma C + \sigma^2) \leqslant \frac{1}{8}$ and Lemma 17 imply

$$\begin{aligned}\mathbb{E}\Big[\int\exp\big(2\lambda\big(\widetilde{\mathcal{E}}_n(\vartheta) - \mathcal{E}(\vartheta)\big)\big)\,\mathrm{d}\varrho_{\eta'}(\vartheta)\Big] &= \int\mathbb{E}\big[\exp\big(2\lambda\big(\widetilde{\mathcal{E}}_n(\vartheta) - \mathcal{E}(\vartheta)\big)\big)\big]\,\mathrm{d}\varrho_{\eta'}(\vartheta)\\ &\leqslant \int\exp\big(2\lambda\big(C_{n,2\lambda} + \tfrac{2\lambda}{n}(\sigma C + \sigma^2)\big)\mathcal{E}(\vartheta)\big)\,\mathrm{d}\varrho_{\eta'}(\vartheta)\\ &\leqslant \int\exp\big(\tfrac{1}{2}\lambda\mathcal{E}(\vartheta)\big)\,\mathrm{d}\varrho_{\eta'}(\vartheta)\\ &\leqslant \int\exp\big(\lambda\big(\mathcal{E}(\vartheta^*) + \|f_\vartheta - f_{\vartheta^*}\|^2_{L^2(\mathbb{P}^{\mathbf{X}})}\big)\big)\,\mathrm{d}\varrho_{\eta'}(\vartheta)\\ &\leqslant \int\exp\big(\lambda\big(\mathcal{E}(\vartheta^*) + s_n^2/L\big)\big)\,\mathrm{d}\varrho_{\eta'}(\vartheta)\\ &\leqslant \mathrm{e}^{2Q_7P\log n}.\end{aligned}$$

We conclude

$$\mathbb{E}\big[\widetilde{\Pi}_{\lambda,\rho}(B_n \mid \mathcal{D}_n)\big] \leqslant \exp\big(-P\big(\log\xi - Q_8 - Q_7\log n - \log\sqrt{L\log n}\big)\big).$$

For a sufficiently large $\xi \geqslant \sqrt{L\log n}$, we obtain $\mathbb{E}\big[\widetilde{\Pi}_{\lambda,\rho}(B_n \mid \mathcal{D}_n)\big] \leqslant \alpha(1-\alpha)^2$ and thus

$$\mathbb{P}\big(f \in \widehat{C}(\xi r_\alpha^\vartheta)\big) \geqslant 1 - \alpha^2 - \alpha(1-\alpha) \geqslant 1 - \alpha. \qquad \square$$

6.6 Proof of Theorem 11

The outline of the proof is similar to that of Theorem 3. Note that the only property of the prior that we used in the proof of Proposition 16 is that Π is a probability measure on the space of network weights. Hence, it is straightforward to see that the analogous statement still holds when replacing Π with $\check{\Pi}$. We obtain with probability of at least $1 - \delta$

$$\mathcal{E}(\check{\vartheta}_{\lambda,\rho}) \leqslant 9\int\mathcal{E}\,\mathrm{d}\varrho + \frac{16}{\lambda}\big(\mathrm{KL}(\varrho \mid \check{\Pi}) + \log(2/\delta)\big). \tag{6.10}$$

For a width $r \in \mathbb{N}$ and some radius $\eta \in (0,1]$, we now choose $\varrho = \varrho_{r,\eta}$ defined via

$$\frac{\mathrm{d}\varrho_{r,\eta}}{\mathrm{d}\Pi_r}(\vartheta) \propto \mathbb{1}_{\{|\vartheta - \vartheta_L^*|_\infty \leqslant \eta\}}$$

with ϑ_r^* from (4.1). Replacing ϑ^* with ϑ_r^* in the arguments from before, we find

$$\int\mathcal{E}\,\mathrm{d}\varrho_{r,\eta} \leqslant \frac{4}{3}\mathcal{E}(\vartheta_r^*) + \frac{3}{n^2} \qquad \text{for} \qquad \eta = \frac{1}{8K(2rB)^Lpn}.$$

To bound the Kullback-Leibler term in (6.10), we employ the following modification of Lemma 18:

Lemma 19. *We have* $\mathrm{KL}(\varrho_{r,\eta} \mid \check{\Pi}) \leqslant P_r\log(2B/\eta) + r$.

Therefore, we have with probability $1-\delta$

$$\mathcal{E}(\check{\vartheta}_{\lambda,\rho}) \leqslant 12\mathcal{E}(f_{\vartheta_r^*}) + \frac{Q_5}{n}\big(P_r L\log(n) + \log(2/\delta)\big),$$

for some $Q_5 > 0$ only depending on C, Γ, σ. Choosing r to minimize the upper bound in the last display yields the assertion. □

6.7 Remaining proofs for Section 3

6.7.1 Proof of Lemma 1

Define

$$D_\lambda \coloneqq \int \exp\big(-\lambda R_n(\vartheta)\big)\,\Pi(\mathrm{d}\vartheta), \qquad \bar{D}_\lambda \coloneqq \int \exp\big(-\lambda \bar{R}_{n,\rho}(\vartheta)\big)\,\Pi(\mathrm{d}\vartheta).$$

For the first part of the lemma, we write

$$\begin{aligned}\mathrm{KL}\big(\bar{\Pi}_{\lambda,\rho}(\cdot\mid\mathcal{D}_n)\,\big|\,\Pi_\lambda(\cdot\mid\mathcal{D}_n)\big) &= \int \log\frac{\mathrm{d}\bar{\Pi}_{\lambda,\rho}(\vartheta\mid\mathcal{D}_n)}{\mathrm{d}\Pi_\lambda(\cdot\mid\mathcal{D}_n)}\,\bar{\Pi}_{\lambda,\rho}(\mathrm{d}\vartheta\mid\mathcal{D}_n)\\ &= \lambda\int S_n(\vartheta)\,\bar{\Pi}_{\lambda,\rho}(\mathrm{d}\vartheta\mid\mathcal{D}_n) + \log\frac{D_\lambda}{\bar{D}_\lambda} \qquad \text{with}\\ S_n(\vartheta) &\coloneqq R_n(\vartheta) - \bar{R}_{n,\rho}(\vartheta).\end{aligned}$$

By concavity of the logarithm we have

$$\frac{1}{\lambda}\sum_{i=1}^n \log\big(\rho\mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)} + 1-\rho\big) \geqslant \frac{1}{\lambda}\sum_{i=1}^n \rho\log\mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)} + (1-\rho)\log 1 = -\frac{1}{n}\sum_{i=1}^n \ell_i(\vartheta) = -R_n(\vartheta).$$

Hence, $S_n(\vartheta) \geqslant 0$ and $D_\lambda \leqslant \bar{D}_\lambda$. We conclude

$$\mathrm{KL}\big(\bar{\Pi}_{\lambda,\rho}(\cdot\mid\mathcal{D}_n)\,\big|\,\Pi_\lambda(\cdot\mid\mathcal{D}_n)\big) \leqslant \lambda\int S_n(\vartheta)\,\bar{\Pi}_{\lambda,\rho}(\mathrm{d}\vartheta\mid\mathcal{D}_n).$$

Moreover, $\log(x+1) \leqslant x$ for all $x > -1$ and a second order Taylor expansion of $x \mapsto \mathrm{e}^x$ yields

$$\begin{aligned}S_n(\vartheta) &= \frac{1}{\lambda}\sum_{i=1}^n \Big(\log\big(\rho(\mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)} - 1) + 1\big) + \frac{\lambda}{n}\ell_i(\vartheta)\Big)\\ &\leqslant \frac{\rho}{\lambda}\sum_{i=1}^n \big(\mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)} - 1 + \tfrac{\lambda}{n\rho}\ell_i(\vartheta)\big)\\ &\leqslant \frac{\rho}{2\lambda}\sum_{i=1}^n \big(\tfrac{\lambda}{n\rho}\ell_i(\vartheta)\big)^2\mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)}\\ &\leqslant \frac{\lambda}{n\rho}\cdot\frac{1}{2n}\sum_{i=1}^n |\ell_i(\vartheta)|^2.\end{aligned}$$

For $\ell_i(\vartheta) = |Y_i - f_\vartheta(\mathbf{X}_i)|^2 \leqslant 2|f(\mathbf{X}_i) - f_\vartheta(\mathbf{X}_i)|^2 + 2\varepsilon_i^2 \leqslant 8C^2 + 2\varepsilon_i^2$ we obtain

$$S_n(\vartheta) \leqslant \frac{\lambda}{n\rho}\Big(64C^4 + \frac{4}{n}\sum_{i=1}^n \varepsilon_i^4\Big)$$

and thus

$$\frac{1}{\lambda}\mathrm{KL}\big(\bar{\Pi}_{\lambda,\rho}(\cdot\mid\mathcal{D}_n)\,\big|\,\Pi_\lambda(\cdot\mid\mathcal{D}_n)\big) \leqslant \frac{\lambda}{n\rho}\Big(64C^4 + \frac{4}{n}\sum_{i=1}^n \varepsilon_i^4\Big)\int \bar{\Pi}(\mathrm{d}\vartheta\mid\mathcal{D}_n)$$

$$= \frac{\lambda}{n\rho}\Big(64C^4 + \frac{4}{n}\sum_{i=1}^n \varepsilon_i^4\Big).$$

In the regime $\rho \to 0$, define

$$T_n(\vartheta) \coloneqq -\rho n \frac{1}{n}\sum_{i=1}^n \mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)} \qquad \text{and} \qquad D_{\varpi,\lambda} \coloneqq \int \exp\big(-T_n(\vartheta)\big)\,\Pi(\mathrm{d}\vartheta)$$

such that

$$\mathrm{KL}\left(\bar{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)\,\middle|\,\varpi_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)\right) = \int \big(T_n(\vartheta) - \lambda\bar{R}_{n,\rho}(\vartheta)\big)\,\bar{\Pi}_{\lambda,\rho}(\mathrm{d}\vartheta \mid \mathcal{D}_n) + \log\frac{D_{\varpi,\lambda}}{\bar{D}_\lambda}.$$

We have

$$\begin{aligned}
\lambda\bar{R}_{n,\rho}(\vartheta) - T_n(\vartheta) &= -\sum_{i=1}^n \log\big(\rho\mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)} + 1 - \rho\big) - T_n(\vartheta)\\
&= -n\log(1-\rho) - \sum_{i=1}^n \Big(\log\big(\rho\mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)} + 1 - \rho\big) - \log(1-\rho)\Big) - T_n(\vartheta)\\
&= -n\log(1-\rho) - \sum_{i=1}^n \rho\mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)}\int_0^1 \big(t\rho\mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)} + 1 - \rho\big)^{-1}\,\mathrm{d}t - T_n(\vartheta)\\
&= -n\log(1-\rho) - \sum_{i=1}^n \rho\mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)}\int_0^1 \Big(\frac{1}{t\rho\mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)} + 1 - \rho} - 1\Big)\,\mathrm{d}t,
\end{aligned}$$

where $(t\rho\mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)} + 1 - \rho)^{-1} - 1 \in [0, \frac{\rho}{1-\rho}]$. Therefore,

$$-\frac{\rho^2}{(1-\rho)}\sum_{i=1}^n \mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)} \leqslant \lambda\bar{R}_{n,\rho}(\vartheta) - T_n(\vartheta) + n\log(1-\rho) \leqslant 0.$$

This implies $\log\frac{D_{\varpi,\lambda}}{\bar{D}_\lambda} \leqslant -n\log(1-\rho)$ and thus

$$\mathrm{KL}\left(\bar{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)\,\middle|\,\varpi_{\lambda,\rho}(\cdot \mid \mathcal{D}_n)\right) \leqslant \frac{\rho^2}{1-\rho}\int \sum_{i=1}^n \mathrm{e}^{-\frac{\lambda}{n\rho}\ell_i(\vartheta)}\,\bar{\Pi}_{\lambda,\rho}(\mathrm{d}\vartheta \mid \mathcal{D}_n) \leqslant \frac{\rho^2 n}{1-\rho}. \qquad \square$$

6.7.2 Proof of Lemma 2

Recall $\psi_\rho(x) = -\log(\mathrm{e}^{-x} + 1 - \rho)$, $\psi_\rho'(x) = \frac{1}{1+(1-\rho)\mathrm{e}^x}$ and $\psi_\rho''(x) = -\frac{(1-\rho)\mathrm{e}^x}{(1+(1-\rho)\mathrm{e}^x)^2} \in [-1/4, 0]$. Since

$$\begin{aligned}
\widetilde{R}_{n,\rho}(\vartheta) &= \frac{1}{\lambda}\sum_{i=1}^n \psi_\rho\big(\tfrac{\lambda}{n}\ell_i(\vartheta)\big)\\
&= \frac{n}{\lambda}\psi_\rho(0) + \frac{1}{\lambda}\sum_{i=1}^n \tfrac{\lambda}{n}\ell_i(\vartheta)\psi_\rho'\big(\xi_i\tfrac{\lambda}{n}\ell_i(\vartheta)\big)\\
&= \frac{n}{\lambda}\psi_\rho(0) + \frac{\psi_\rho'(0)}{n}\sum_{i=1}^n \ell_i(\vartheta) + \frac{1}{n}\sum_{i=1}^n \ell_i(\vartheta)\big(\psi_\rho'(\xi_i\tfrac{\lambda}{n}\ell_i(\vartheta)) - \psi_\rho'(0)\big)\\
&= -\frac{n}{\lambda}\log(2-\rho) + \frac{1}{2-\rho}R_n(\vartheta) + \frac{\lambda}{n^2}\sum_{i=1}^n \ell_i(\vartheta)^2\xi_i\psi_\rho''\big(\xi_i'\tfrac{\lambda}{n}\ell_i(\vartheta)\big),
\end{aligned}$$

we have

$$-\frac{\lambda^2}{4n^2}\sum_{i=1}^n \ell_i(\vartheta)^2 \leqslant \lambda\widetilde{R}_{n,\rho}(\vartheta) - \frac{\lambda}{2-\rho}R_n(\vartheta) + n\log(2-\rho) \leqslant 0.$$

Therefore, we have with $\widetilde{D}_\lambda$ from (6.6) that

$$\begin{aligned}\mathrm{KL}\left(\widetilde{\Pi}_{\lambda,\rho}(\cdot \mid \mathcal{D}_n) \,\middle|\, \Pi_{\lambda/(2-\rho)}(\cdot \mid \mathcal{D}_n)\right) &= \int \Big(\frac{\lambda}{2-\rho} R_n(\vartheta) - \lambda \widetilde{R}_{n,\rho}(\vartheta)\Big) \widetilde{\Pi}_{\lambda,\rho}(\mathrm{d}\vartheta \mid \mathcal{D}_n) + \log \frac{D_{\lambda/(2-\rho)}}{\widetilde{D}_\lambda} \\ &\leqslant \int \Big(\frac{\lambda}{2-\rho} R_n(\vartheta) - \lambda \widetilde{R}_{n,\rho}(\vartheta) - n\log(2-\rho)\Big) \widetilde{\Pi}_{\lambda,\rho}(\mathrm{d}\vartheta \mid \mathcal{D}_n) \\ &\leqslant \frac{\lambda^2}{4n} \int \frac{1}{n} \sum_{i=1}^n \ell_i(\vartheta)^2 \, \widetilde{\Pi}_{\lambda,\rho}(\mathrm{d}\vartheta \mid \mathcal{D}_n) \\ &\leqslant \frac{\lambda^2}{n}\Big(32C^4 + \frac{2}{n}\sum_{i=1}^n \varepsilon_i^4\Big). \qquad \square\end{aligned}$$

6.7.3 Proof of Corollary 6

Jensen's and Markov's inequality yield for r_n^2 from (3.4) that

$$\begin{aligned}\mathbb{P}\Big(\mathcal{E}(\bar{f}_{\lambda,\rho}) > r_n^2 + \frac{Q_1}{n} + \frac{Q_1}{n}\log(2/\delta)\Big) &= \mathbb{P}\Big(\|\mathbb{E}[f_{\widetilde{\vartheta}_{\lambda,\rho}} \mid \mathcal{D}_n] - f\|^2_{L^2(\mathbb{P}^{\mathbf{X}})} > r_n^2 + \frac{Q_1}{n} + \frac{Q_1}{n}\log(2/\delta)\Big) \\ &\leqslant \mathbb{P}\Big(\mathbb{E}\big[\|f_{\widetilde{\vartheta}_{\lambda,\rho}} - f\|^2_{L^2(\mathbb{P}^{\mathbf{X}})} \,\big|\, \mathcal{D}_n\big] > r_n^2 + \frac{Q_1}{n} + \frac{Q_1}{n}\log(2/\delta)\Big) \\ &= \mathbb{P}\Big(\int_{\frac{Q_1}{n}\log(2/\delta)}^\infty \widetilde{\Pi}_{\lambda,\rho}\big(\|f_{\widetilde{\vartheta}_{\lambda,\rho}} - f\|^2_{L^2(\mathbb{P}^{\mathbf{X}})} > r_n^2 + t \,\big|\, \mathcal{D}_n\big)\,\mathrm{d}t > \frac{Q_1}{n}\Big) \\ &\leqslant \frac{n}{Q_1}\int_{\frac{Q_1}{n}\log(2/\delta)}^\infty \mathbb{E}\big[\widetilde{\Pi}_{\lambda,\rho}\big(\|f_{\widetilde{\vartheta}_\lambda} - f\|^2_{L^2(\mathbb{P}^{\mathbf{X}})} > r_n^2 + t \,\big|\, \mathcal{D}_n\big)\big]\,\mathrm{d}t.\end{aligned}$$

Using Theorem 3, we thus obtain

$$\mathbb{P}\Big(\mathcal{E}(\bar{f}_{\lambda,\rho}) > r_n^2 + \frac{Q_1}{n} + \frac{Q_1}{n}\log(2/\delta)\Big) \leqslant \frac{2n}{Q_1}\int_{\frac{Q_1}{n}\log(2/\delta)}^\infty \mathrm{e}^{-nt/Q_1}\,\mathrm{d}t = \delta. \qquad \square$$

6.7.4 Proof of Proposition 7

We combine arguments from [51] with the approximation results from [35]. By rescaling, we can rewrite

$$f = f_q \circ \cdots \circ f_0 = h_q \circ \cdots \circ h_0$$

with $h_i = (h_{ij})_{j=1,\dots,d_{i+1}}$, where $\widetilde{h}_{0j} \in \mathcal{C}_{t_0}^{\beta_0}([0,1]^{t_0}, 1)$, $\widetilde{h}_{ij} \in \mathcal{C}_{t_i}^{\beta_i}\big([0,1]^{t_i}, (2C_0)^{\beta_i}\big)$ for $i = 1,\dots,q-1$ and $\widetilde{h}_{qj} \in \mathcal{C}_{t_q}^{\beta_q}\big([0,1]^{t_q}, C_0(2C_0)^{\beta_q}\big)$ and h_{ij} is $\widetilde{h}_{ij}$ understood as a function in d_i instead of t_i arguments.

We want to show that there exists a constant C_i such that for any $M_i \in \mathbb{N}$ we can find sufficiently large $L_i, r_i \in \mathbb{N}$ and a neural network $\widetilde{g}_{ij} \in \mathcal{G}(t_i, L_i, r_i)$ with $P_{L_i,r_i} = c_i M^{t_i}$ parameters and

$$\|\widetilde{h}_{ij} - \widetilde{g}_{ij}\|_{L^\infty([0,1]^{t_i})} \leqslant C_i M_i^{-2\beta_i}. \tag{6.11}$$

To construct such g_{ij}, we use Theorem 2(a) from [35]. Their conditions

1. $L_i \geqslant 5 + \lceil \log_4(M^{2\beta_i})\rceil\big(\lceil \log_2(\max\{\lfloor \beta_i \rfloor, t_i\} + 1\rceil + 1\big)$ and
2. $r_i \geqslant 2^{t_i+6}\binom{t_i+\lfloor\beta_i\rfloor}{t_i} t_i^2(\lfloor\beta_i\rfloor + 1)M_i^{t_i}$

can be satisfied for $L_i = C_i\log(M_i), r_i = C_i M_i^{t_i}$, where C_i only depends on upper bounds for t_i and β_i. Hence, there exists a neural network $\widetilde{g}_{ij} \in \mathcal{G}(t_i, L_i, r_i)$ with (6.11). Careful inspection of the proof of this theorem reveals, that the weights and shifts of $\widetilde{g}_{ij}$ grow at most logarithmically in M, whereas our result still holds for linearly growing weights and shifts. Since $t_i \leqslant d_i, r_i$, we can easily embed $\widetilde{g}_{ij}$ into the class

$\mathcal{G}(d_i, L_i, r_i)$ by setting $g_{ij} = \widetilde{g}_{ij}(W_{ij}\,\cdot)$, where the matrix $W_{ij} \in \mathbb{R}^{t_i \times d_i}$ is chosen such that g_{ij} depends on the same t_i many arguments as h_{ij}. Note that the approximation accuracy of $\widetilde{g}_{ij}$ carries over to g_{ij}, that is

$$\|h_{ij} - g_{ij}\|_{L^\infty([0,1]^{d_i})} \leqslant \|\widetilde{h}_{ij} - \widetilde{g}_{ij}\|_{L^\infty([0,1]^{t_i})} \leqslant C_i M_i^{-2\beta_i}. \tag{6.12}$$

Setting $g = g_q \circ \cdots \circ g_0$ with $g_i = (g_{ij})_j$ we obtain a neural network $g \in \mathcal{G}(p, L, r)$ with $r = \max_{i=0,\dots,q,} r_i d_{i+1}$ and $L = \sum_{i=0}^q L_i$.

Counting the number of parameters of g and using $L_i = C_i M_i^{t_i}$, we get

$$P_{L,r} \leqslant Q_9 \sum_{i=0}^q L_i r_i^2$$

for some $Q_9 > 0$.

It follows from Schmidt-Hieber [51, Lemma 3] and (6.12) that

$$\|f - g\|_{L^\infty([0,1]^p)} \leqslant C_0 \prod_{l=0}^{q-1} (2C_0)^{\beta_{l+1}} \sum_{i=0}^q \||h_i - g_i|_\infty\|_{L^\infty([0,1]^{d_i})}^{\prod_{l=i+1}^q \beta_l \wedge 1} \leqslant Q_{10} \sum_{i=0}^q M_i^{-2\beta_i},$$

for some $Q_{10} > 0$.

Applying Theorem 3 together with $\mathcal{E}(f_{\vartheta^*}) \leqslant \|f - g\|^2_{L^\infty([0,1]^p)}$ we now obtain

$$\mathcal{E}(\widetilde{f}_{\lambda,\rho}) \leqslant Q_{11} \sum_{i=0}^q M_i^{-4\beta_i} + \frac{Q_{11}}{n} \sum_{i=0}^q M_i^{2t_i} (\log n)^3 + Q_{11} \frac{\log(2/\delta)}{n} \tag{6.13}$$

with probability of at least $1 - \delta$. Choosing

$$M_i = \left\lceil \Big(\frac{n}{(\log n)^3}\Big)^{1/(4\beta_i + 2t_i)} \right\rceil$$

ensures $L, r \leqslant n$ for sufficiently large n, balances the first two terms in the upper bound (6.13) and thus yields the asserted convergence rate for $\widetilde{f}_{\lambda,\rho}$.

The convergence rate for the posterior mean can be proved analogously using Corollary 6. □

6.7.5 Proof of Corollary 12

The statement follows by choosing L in the upper bound from Theorem 11 as in the statement of Proposition 7 and then using the same approximation result to control excess-risk of the corresponding oracle choice ϑ^*_L. □

6.8 Proofs of the auxiliary results

6.8.1 Proof of Lemma 17

Set $\eta := |\vartheta - \widetilde{\vartheta}|_\infty$ and let $W^{(1)}, \dots, W^{(L+1)}, v^{(1)}, \dots, v^{(L+1)}$ and $\widetilde{W}^{(1)}, \dots, \widetilde{W}^{(L+1)}, \widetilde{v}^{(1)}, \dots, \widetilde{v}^{(L+1)}$ be the weights and shifts associated with ϑ and $\widetilde{\vartheta}$, respectively. Define $\widetilde{\mathbf{x}}^{(l)}$, $l = 0, \dots, L+1$, analogously to (3.1). We can recursively deduce from the Lipschitz-continuity of ϕ that for $l = 2, \dots, L$:

$$\begin{aligned}
|\mathbf{x}^{(1)}|_1 &\leqslant |W^{(1)}\mathbf{x}|_1 + |v^{(1)}|_1 \\
&\leqslant 2rB(|\mathbf{x}|_1 \vee 1), \\
|\mathbf{x}^{(1)} - \widetilde{\mathbf{x}}^{(1)}|_1 &\leqslant |W^{(1)}\mathbf{x}^{(0)} + v^{(1)} - \widetilde{W}^{(1)}\widetilde{\mathbf{x}}^{(0)} - \widetilde{v}^{(1)}|_1 \\
&\leqslant \eta 2r(|\mathbf{x}|_1 \vee 1), \\
|\mathbf{x}^{(l)}|_1 &\leqslant |W^{(l)}\mathbf{x}^{(l-1)}|_1 + |v^{(l)}|_1
\end{aligned}$$

$$
\begin{aligned}
&\leqslant 2rB(|\mathbf{x}^{(l-1)}|_1 \vee 1) \qquad \text{and}\\
|\mathbf{x}^{(l)} - \widetilde{\mathbf{x}}^{(l)}|_1 &\leqslant |W^{(l)}\mathbf{x}^{(l-1)} + v^{(l)} - \widetilde{W}^{(l)}\widetilde{\mathbf{x}}^{(l-1)} - \widetilde{v}^{(l)}|_1\\
&\leqslant |(W^{(l)} - \widetilde{W}^{(l)})\mathbf{x}^{(l-1)}|_1 + |\widetilde{W}^{(l)}(\mathbf{x}^{(l-1)} - \widetilde{\mathbf{x}}^{(l-1)})|_1 + |v^{(l)} - \widetilde{v}^{(l)}|_1\\
&\leqslant \eta 2r(|\mathbf{x}^{(l-1)}|_1 \vee 1) + rB|\mathbf{x}^{(l-1)} - \widetilde{\mathbf{x}}^{(l-1)}|_1.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
|\mathbf{x}^{(L)}|_1 &\leqslant (2rB)^{L-1}(|\mathbf{x}^{(1)}|_1 \vee 1)\\
&\leqslant (2rB)^{L}(|\mathbf{x}|_1 \vee 1) \qquad \text{and}\\
|\mathbf{x}^{(L)} - \widetilde{\mathbf{x}}^{(L)}|_1 &\leqslant \eta 2r \sum_{k=1}^{L-1} (rB)^{k-1}(|\mathbf{x}^{(L-k)}|_1 \vee 1) + (rB)^{L-1}|\mathbf{x}^{(1)} - \widetilde{\mathbf{x}}^{(1)}|_1\\
&\leqslant \eta 2^{(L+1)} r(|\mathbf{x}|_1 \vee 1)(rB)^{L-1}
\end{aligned}
$$

Since the clipping function $y \mapsto (-C) \vee (y \wedge C)$ has Lipschitz constant 1, we conclude

$$
\begin{aligned}
|f_\vartheta(\mathbf{x}) - f_{\widetilde{\vartheta}}(\mathbf{x})| &\leqslant |g_\vartheta(\mathbf{x}) - g_{\widetilde{\vartheta}}(\mathbf{x})|\\
&= |\mathbf{x}^{(L+1)} - \widetilde{\mathbf{x}}^{(L+1)}|\\
&= |W^{(L+1)}\mathbf{x}^{(L)} + v^{(L+1)} - \widetilde{W}^{(L+1)}\widetilde{\mathbf{x}}^{(L)} - \widetilde{v}^{(L+1)}|\\
&\leqslant |(W^{(L+1)} - \widetilde{W}^{(L+1)})\mathbf{x}^{(L)}| + |\widetilde{W}^{(L+1)}(\mathbf{x}^{(L)} - \widetilde{\mathbf{x}}^{(L)})| + |v^{(L+1)} - \widetilde{v}^{(L+1)}|\\
&\leqslant r|W^{(L+1)} - \widetilde{W}^{(L+1)}|_\infty|\mathbf{x}^{(L)}|_1 + r|\widetilde{W}^{(L+1)}|_\infty|\mathbf{x}^{(L)} - \widetilde{\mathbf{x}}^{(L)}|_1 + |v^{(L+1)} - \widetilde{v}^{(L+1)}|\\
&\leqslant \eta r(2rB)^L(|\mathbf{x}|_1 \vee 1) + \eta(rB)^L 2^{L+1}(|\mathbf{x}|_1 \vee 1) + \eta\\
&\leqslant \eta 4(2rB)^L(|\mathbf{x}|_1 \vee 1). \qquad \square
\end{aligned}
$$

6.8.2 Proof of Lemma 18

Since ϱ_η and Π are product measures, their KL-divergence is equal to the sum of the KL-divergences in each of the P factors. For each such factor, we are comparing

$$
\mathcal{U}([(\vartheta^*)_i - \eta, (\vartheta^*)_i + \eta] \cap [-B, B]) \qquad \text{with} \qquad \mathcal{U}([-B, B]),
$$

where $(\vartheta^*)_i$ denotes the i-th entry of ϑ^*. The KL-divergence of these distributions is equal to

$$
\log\Big(\frac{\lambda([-B, B])}{\lambda([(\vartheta^*)_i - \eta, (\vartheta^*)_i + \eta] \cap [-B, B])}\Big) \leqslant \log\Big(\frac{\lambda([-B, B])}{\lambda([0, \eta])}\Big) = \log(2B/\eta),
$$

where λ denotes the Lebesgue-measure. Thus,

$$
\mathrm{KL}(\varrho_\eta \mid \Pi) = \sum_{i=1}^{P} \mathrm{KL}\big(\mathcal{U}([(\vartheta^*)_i - \eta, (\vartheta^*)_i + \eta] \cap [-B, B]) \,\big|\, \mathcal{U}([-B, B])\big) \leqslant P\log(2B/\eta). \qquad \square
$$

6.8.3 Proof of Lemma 19

We will show that

$$
\frac{\mathrm{d}\varrho_{r,\eta}}{\mathrm{d}\check{\Pi}} = 2^r(1 - 2^{-n})\frac{\mathrm{d}\varrho_{r,\eta}}{\mathrm{d}\Pi_r}, \tag{6.14}
$$

from which we can deduce

$$
\mathrm{KL}(\varrho_{r,\eta} \mid \check{\Pi}) = \int \log\Big(\frac{\mathrm{d}\varrho_{r,\eta}}{\mathrm{d}\check{\Pi}}\Big)\,\mathrm{d}\varrho_{r,\eta} = \int \log\Big(\frac{\mathrm{d}\varrho_{r,\eta}}{\mathrm{d}\Pi_r}\Big)\,\mathrm{d}\varrho_{r,\eta} + \log(2^r(1 - 2^{-n})) \leqslant \mathrm{KL}(\varrho_{L,\eta} \mid \Pi_L) + r
$$

and since the arguments from the proof of Lemma 18 yield $\mathrm{KL}(\varrho_{r,\eta} \mid \Pi_r) \leqslant P_r \log(2B/\eta)$, the lemma follows.

For (6.14), note that $\rho_{r,\eta}$ can only assign a positive probability to subsets $A \subseteq [-B, B]^{P_r}$. Hence,

$$\varrho_{r,\eta}(A) = \int_A \frac{\mathrm{d}\varrho_{r,\eta}}{\mathrm{d}\check{\Pi}}\,\mathrm{d}\check{\Pi} = (1-2^{-n})^{-1}\sum_{l=1}^{n} 2^{-l}\int_A \frac{\mathrm{d}\varrho_{r,\eta}}{\mathrm{d}\check{\Pi}}\,\mathrm{d}\Pi_l = (1-2^{-n})^{-1}2^{-r}\int_A \frac{\mathrm{d}\varrho_{r,\eta}}{\mathrm{d}\check{\Pi}}\,\mathrm{d}\Pi_r. \qquad \square$$

References

[1] Alexos, A., Boyd, A. J., & Mandt, S. (2022). Structured stochastic gradient MCMC. In *International Conference on Machine Learning* (pp. 414–434).

[2] Alquier, P. (2021). User-friendly introduction to PAC-Bayes bounds. *arXiv preprint arXiv:2110.11216.*

[3] Alquier, P. & Biau, G. (2013). Sparse single-index model. *Journal of Machine Learning Research*, 14, 243–280.

[4] Andrieu, C. & Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2), 697–725.

[5] Anthony, M. & Bartlett, P. L. (1999). *Neural network learning: Theoretical foundations.* Cambridge University Press.

[6] Audibert, J.-Y. (2004). Aggregated estimators and empirical complexity for least square regression. *Annales de l'Institut Henri Poincaré. Probabilités et Statistiques*, 40(6), 685–736.

[7] Audibert, J.-Y. (2009). Fast learning rates in statistical inference through aggregation. *The Annals of Statistics*, 37(4), 1591–1646.

[8] Audibert, J.-Y. & Catoni, O. (2011). Robust linear least squares regression. *The Annals of Statistics*, 39(5), 2766–2794.

[9] Bardenet, R., Doucet, A., & Holmes, C. (2017). On Markov chain Monte Carlo methods for tall data. *Journal of Machine Learning Research*, 18(47).

[10] Bauer, B. & Kohler, M. (2019). On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *The Annals of Statistics*, 47(4), 2261–2285.

[11] Besag, J. (1994). Comments on "Representations of knowledge in complex systems" by U. Grenander and M.I. Miller. *Journal of the Royal Statistical Society. Series B. Methodological*, 56(4), 549–581.

[12] Bissiri, P. G., Holmes, C. C., & Walker, S. G. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, 78(5), 1103–1130.

[13] Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518), 859–877.

[14] Castillo, I. & Nickl, R. (2014). On the Bernstein-von Mises phenomenon for nonparametric Bayes procedures. *The Annals of Statistics*, 42(5), 1941–1969.

[15] Catoni, O. (2004). *Statistical learning theory and stochastic optimization.* Springer.

[16] Cheng, X. & Bartlett, P. (2018). Convergence of Langevin MCMC in KL-divergence. In *Proceedings of Algorithmic Learning Theory*, volume 83 (pp. 186–211).

[17] Chérief-Abdellatif, B.-E. (2020). Convergence rates of variational inference in sparse deep learning. In *International Conference on Machine Learning* (pp. 1831–1842).

[18] Cobb, A. D. & Jalaian, B. (2021). Scaling hamiltonian monte carlo inference for bayesian neural networks with symmetric splitting. *Uncertainty in Artificial Intelligence.*

[19] Dalalyan, A. S. & Riou-Durand, L. (2020). On sampling from a log-concave density using kinetic Langevin diffusions. *Bernoulli. Official Journal of the Bernoulli Society for Mathematical Statistics and Probability*, 26(3), 1956–1988.

[20] Deng, W., Feng, Q., Gao, L., Liang, F., & Lin, G. (2020a). Non-convex learning via replica exchange stochastic gradient MCMC. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research* (pp. 2474–2483).

[21] Deng, W., Liang, S., Hao, B., Lin, G., & Liang, F. (2022). Interacting contour stochastic gradient Langevin dynamics. In *The Tenth International Conference on Learning Representations.*

[22] Deng, W., Lin, G., & Liang, F. (2020b). A contour stochastic gradient Langevin dynamics algorithm for simulations of multi-modal distributions. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems.*

[23] DeVore, R., Hanin, B., & Petrova, G. (2021). Neural network approximation. *Acta Numerica*, 30, 327–444.

[24] Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. (1987). Hybrid Monte Carlo. *Physics letters B*, 195(2), 216–222.

[25] Franssen, S. & Szabó, B. (2022). Uncertainty quantification for nonparametric regression using empirical Bayesian neural networks. *arXiv preprint arXiv:2204.12735.*

[26] Freund, Y., Ma, Y.-A., & Zhang, T. (2022). When is the convergence time of Langevin algorithms dimension independent? A composite optimization viewpoint. *Journal of Machine Learning Research*, 23, 1–32.

[27] Ghosal, S. & van der Vaart, A. (2017). *Fundamentals of nonparametric Bayesian inference*, volume 44 of *Cambridge Series in Statistical and Probabilistic Mathematics.* Cambridge University Press.

[28] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* MIT Press.

[29] Guedj, B. (2019). A primer on PAC-Bayesian learning. *arXiv preprint arXiv:1901.05353.*

[30] Guedj, B. & Alquier, P. (2013). PAC-Bayesian estimation and prediction in sparse additive models. *Electronic Journal of Statistics*, 7, 264–291.

[31] Hoffmann, M. & Nickl, R. (2011). On adaptive inference and confidence bands. *The Annals of Statistics*, 39(5), 2383–2409.

[32] Jacot, A., Gabriel, F., & Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31.

[33] Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

[34] Knapik, B. T., van der Vaart, A. W., & van Zanten, J. H. (2011). Bayesian inverse problems with Gaussian priors. *The Annals of Statistics*, 39(5), 2626–2657.

[35] Kohler, M. & Langer, S. (2021). On the rate of convergence of fully connected deep neural network regression estimates. *The Annals of Statistics*, 49(4), 2231–2249.

[36] Li, C., Chen, C., Carlson, D. E., & Carin, L. (2016). Preconditioned stochastic gradient Langevin dynamics for deep neural networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (pp. 1788–1794).

[37] Maclaurin, D. & Adams, R. P. (2014). Firefly Monte Carlo: Exact MCMC with subsets of data. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence.*

[38] Massart, P. (2007). *Concentration inequalities and model selection*, volume 1896 of *Lecture Notes in Mathematics.* Springer.

[39] McAllester, D. A. (1999a). PAC-Bayesian model averaging. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory* (pp. 164–170).

[40] McAllester, D. A. (1999b). Some PAC-Bayesian theorems. *Machine Learning*, 37(3), 355–363.

[41] Neal, R. M. (2011). MCMC using Hamiltonian dynamics, In: *Handbook of Markov chain Monte Carlo.* (pp. 113–163).

[42] Nickl, R. & Wang, S. (2022). On polynomial-time computation of high-dimensional posterior measures by Langevin-type algorithms. *Journal of the European Mathematical Society.*

[43] Patterson, S. & Teh, Y. W. (2013). Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems 26* (pp. 3102–3110).

[44] Polson, N. G. & Ročková, V. (2018). Posterior concentration for sparse deep learning. *Advances in Neural Information Processing Systems*, 31, 938–949.

[45] Ray, K. & Szabó, B. (2022). Variational Bayes for high-dimensional linear regression with sparse priors. *Journal of the American Statistical Association*, 117(539), 1270–1281.

[46] Robert, C. P. & Casella, G. (2004). *Monte Carlo statistical methods.* Springer, second edition.

[47] Roberts, G. O. & Tweedie, R. L. (1996a). Exponential convergence of of Langevin distributions and their discrete approximations. *Bernoulli. Official Journal of the Bernoulli Society for Mathematical Statistics and Probability*, 2(4), 341–363.

[48] Roberts, G. O. & Tweedie, R. L. (1996b). Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika*, 83(1), 95–110.

[49] Rousseau, J. & Szabó, B. (2020). Asymptotic frequentist coverage properties of Bayesian credible sets for sieve priors. *The Annals of Statistics*, 48(4), 2155–2179.

[50] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85–117.

[51] Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4), 1875–1897.

[52] Shawe-Taylor, J. & Williamson, R. C. (1997). A PAC analysis of a Bayesian estimator. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory* (pp. 2–9).

[53] Steffen, M. F. & Trabs, M. (2023). A PAC-Bayes oracle inequality for sparse neural networks. *arXiv preprint arXiv:2204.12392.*

[54] Szabó, B., van der Vaart, A. W., & van Zanten, J. H. (2015). Frequentist coverage of adaptive nonparametric Bayesian credible sets. *The Annals of Statistics*, 43(4), 1391–1428.

[55] Welling, M. & Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning* (pp. 681–688).

[56] Yarotsky, D. (2017). Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94, 103–114.

[57] Zhang, A. Y. & Zhou, H. H. (2020). Theoretical and computational guarantees of mean field variational inference for community detection. *The Annals of Statistics*, 48(5), 2575–2598.

[58] Zhang, F. & Gao, C. (2020). Convergence rates of variational posterior distributions. *The Annals of Statistics*, 48(4), 2180–2207.

[59] Zhang, R., Li, C., Zhang, J., Chen, C., & Wilson, A. G. (2020). Cyclical stochastic gradient MCMC for Bayesian deep learning. In *8th International Conference on Learning Representations*.

[60] Zhang, T. (2006). Information-theoretic upper and lower bounds for statistical estimation. 52(4), 1307–1321.

Eur. Phys. J. C (2024) 84:972
https://doi.org/10.1140/epjc/s10052-024-13353-w

THE EUROPEAN
PHYSICAL JOURNAL C

Regular Article - Experimental Physics

Classifier surrogates: sharing AI-based searches with the world

Sebastian Bieringer[1,a], **Gregor Kasieczka**[1], **Jan Kieseler**[2], **Mathias Trabs**[3]

[1] Institut für Experimentalphysik, Universität Hamburg, Luruper Chaussee 149, 22761 Hamburg, Germany
[2] Institut für Experimentelle Teilchenphysik, Karlsruher Institut für Technologie, Wolfgang-Gaede-Str. 1, 76131 Karlsruhe, Germany
[3] Institut für Stochastik, Karlsruher Institut für Technologie, Englerstr. 2, 76131 Karlsruhe, Germany

Received: 27 February 2024 / Accepted: 9 September 2024

Abstract In recent years, neural network-based classification has been used to improve data analysis at collider experiments. While this strategy proves to be hugely successful, the underlying models are not commonly shared with the public and rely on experiment-internal data as well as full detector simulations. We show a concrete implementation of a newly proposed strategy, so-called Classifier Surrogates, to be trained inside the experiments, that only utilise publicly accessible features and truth information. These surrogates approximate the original classifier distribution, and can be shared with the public. Subsequently, such a model can be evaluated by sampling the classification output from high-level information without requiring a sophisticated detector simulation. Technically, we show that continuous normalizing flows are a suitable generative architecture that can be efficiently trained to sample classification results using conditional flow matching. We further demonstrate that these models can be easily extended by Bayesian uncertainties to indicate their degree of validity when confronted with unknown inputs by the user. For a concrete example of tagging jets from hadronically decaying top quarks, we demonstrate the application of flows in combination with uncertainty estimation through either inference of a mean-field Gaussian weight posterior, or Monte Carlo sampling network weights.

1 Introduction

Current experimental work in particle physics, for example by the ATLAS and CMS collaborations, uses deep learning-based taggers to great success [1–4]. Such models often define unique and essential quantities in the analysis chain, which are hard to understand in terms of physical quantities. While the performance benefit is apparent, best practices for sharing the analysis as for traditional cut-based analyses [5,6] are not yet established. This especially hinders the re-interpretation of experimental results. Recently, a first set of proposals on sharing neural network-based results has been published [7]. On the purely technical side, solutions exist for sharing serialized networks [8,9] and some first searches shared with serialized models have been made public [10–13].

However, when the model inputs contain features which are not available outside the collaborations or can only be simulated at high computational cost within the collaboration, the benefit of sharing the network weights is limited as results still can either not be reproduced at all, or are very expensive. Costly and unavailable input features include detector level quantities, such as hits, or highly detector dependent quantities, such as soft jet-substructure variables. For example, both b-taggers of ATLAS and CMS use detector dependent information [14,15] and current research shows the best classification performance is achieved when using detector-level data, rather than only high-level observables [4,16]. For these cases, sharing a surrogate model trained to reproduce the classification results from truth-, parton- or reconstruction-level inputs has recently been proposed in discussions at the LHC Reinterpretation Forum and the 2023 PhysTeV workshop at Les Houches [7]. We will follow the newly introduced terminology and refer to such models as *Classifier Surrogates*. In this work

- we demonstrate for a concrete example how such a Classifier Surrogate could be constructed and evaluated
- and present a novel combination of Continuous Normalizing Flows with Monte Carlo-based Bayesian Neural Networks (BNN) for this purpose.

[a] e-mail: sebastian.guido.bieringer@uni-hamburg.de (corresponding author)

Published online: 27 September 2024

Complementary to sharing the full likelihood or the full statistical model [17]

$$p(\text{data} \mid \mu),$$

a Classifier Surrogate can be used to model dependencies on parameters $\tilde{\mu}$ that were not explicitly included in the statistical model at the time of the release and are hard to model with public fast simulation tools like Delphes [18]. Altering the parameters requires that the released model includes intermediate information, for example distributions of observables that are used in a template fit. These might stem from the output x of a complex neural network classifier. For such distributions, the application of a classification surrogate can be beneficial.

In practice, a Classifier Surrogate

$$p(x \mid c)$$

can be used to predict classifier output from any single-event surrogate input

$$c \sim p(c \mid \tilde{\mu}).$$

This simulation of truth-, parton- or reconstruction-level data allows an arbitrary choice of parameters $\tilde{\mu}$. If the simulated event is out-of-distribution (OOD) of the training data of the classifier, the surrogate will predict large uncertainties and thus prevent the practitioner from interpreting the analysis where the classification can not be applied reliably. For simulated events within the classifiers input range, the surrogate predicts samples from the distribution of viable classifier output. This output prediction in turn can be used to estimate expectation values in histogram bins of derived observables in full analogy to the processing of the classification results from observed data. A statistical model for the new parameters

$$p(\text{data} \mid \tilde{\mu})$$

can again be derived from the processed and possibly histogrammed surrogate output, for example by assuming Poisson-distributed bin values. The surrogate strategy therefore is a truly "open-world" approach to sharing a classifier-aided analysis.

The uncertainties from the statistical limitation of the dataset, as well as the the smearing introduced by the detector simulation and reduced information of the input c may also be absorbed into an additional nuisance parameter of the new statistical model.

Depending on the nuisance handling strategy used for classifier training [19], the dependence on the nuisance parameters needs to be included in the surrogate

$$p(x \mid c) \rightarrow p(x \mid c, \vartheta)$$

for nuisance-parameterized classifiers or in the corresponding input model

$$p(c \mid \tilde{\mu}) \rightarrow p(c \mid \tilde{\mu}, \vartheta)$$

for nuisance-invariant approaches.

If trained on truth- or parton-level, generating surrogate input events $c \sim p(c|\tilde{\mu})$ does not require detector simulation and can thus significantly improve the computational cost of any re-interpretation. Furthermore, eliminating the detector simulation also removes a major bottle-neck for sharing the results with colleagues, that do not have access to collaboration internal simulation-settings.

We introduce the strategy on the concrete example of a classifier derived from the particle transformer [16]. This setup is introduced in Sect. 2. In Sect. 3, we then discuss why a Classifier Surrogate needs to employ a generative architecture and introduce a possible architecture in Sect. 4. To model increased uncertainty for unknown inputs, we develop two BNN implementations of the architecture in Sect. 5. In Sect. 6, we discuss the performance of the surrogate both for data within the distribution of the training data, as well as for data new to the model. We evaluate calibration and scaling to the tails of the distribution, as well as OOD indication.

2 Particle transformer and JetClass dataset

As internal taggers of the big collaborations are not available for public study, we choose to emulate the state-of-the-art jet tagger, the Particle Transformer (ParT) [16]. ParT is an attention-based model trained to distinguish 10 different types of jets using per-particle information and trained on the 100 M JetClass dataset [20]. The features include kinematics, particle identification, and trajectory displacement of every particle in the jet.

From the large initial JetClass dataset as stand-in for the internal collaboration datasets, we distill our toy dataset by calculating transverse momentum, pesudorapidity, scattering angle, jet energy, number of particles, soft drop mass [21] and N-subjettiness [22] for $N = 1, \ldots, 4$, as well as the output of the full ParT for the regarding event. For the first studies we will restrain the experiments to the first five jet-observables as well as the true top or QCD label as surrogate input.

While learning a surrogate of a multiclassifier is possible by using a generative architecture with a multidimensional output space, we restrict the setup to finding a surrogate for binary classification of top jets. The toy train and validation datasets contain 1M jet events each from Z-events and

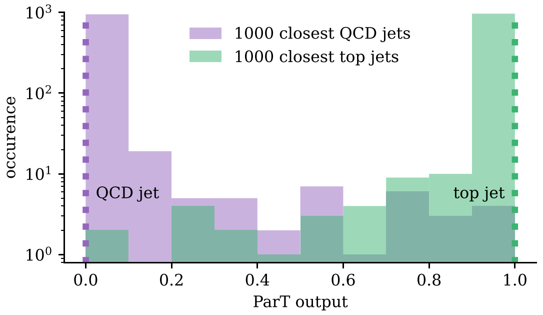


Fig. 1 Histograms of the ParT classification results for the 1000 jet events of the training data, closest to the two arbitrary jets indicated by the dotted lines. Although being classified with varying confidence from detector-level data, the high-level observables p_T, E_{jet} and n_{const} appear identical

hadronic decay of $t\bar{t}$. To reduce the 10-dimensional ParT output to a binary classification result, we rescale

$$p_{\text{t}\to\text{bqq}'} = \frac{p^{\text{ParT}}_{\text{t}\to\text{bqq}'}}{p^{\text{ParT}}_{\text{t}\to\text{bqq}'} + p^{\text{ParT}}_{\text{Z}\to\text{qq}'}}.$$

3 Detector smearing distribution

Due to the stochasticity of the detector simulation, jets with the same high-level observables can differ a lot on detector-level. Similarly, jets simulated for identical truth- or parton-level events, would vary significantly. These jets will thus result in different ParT outputs defining the likelihood per set of high-level observables

$$p(\underbrace{\text{ParT}}_{x} \mid \underbrace{p_T, \eta, \phi, E_{\text{jet}}, n_{\text{const}}, \dots}_{c}). \tag{1}$$

Based on its physical origin, we will also refer to this distribution as the detector smearing distribution.

We can generate a first approximation of this distribution by generating a histogram of the ParT output for the closest points in p_T, E_{jet} and n_{const}. In Fig. 1 we show this histogram for the 1000 nearest jet events in the training sample for two arbitrary jet events in the bulk of the transverse momentum distribution at $p_t \approx 530\,\text{GeV}$. The imperfect ParT classification introduces an output distribution with tails for events indistinguishable from the high-level features. Employing a generative architecture as introduced in Sect. 4, allows us to infer this distribution from the high-level observables.

For the toy setup, we assume the classifier to be constructed invariant for the relevant nuisance parameters [19]. Whenever a nuisance-parameterized classifier is applied, the nuisance parameters need to be included into the likelihood as well.

4 Neural density estimation

While all flavours of generative models have found numerous applications in high-energy physics, for example in [23] and [24], normalizing flows can easily and efficiently be applied to infer complex, low-dimensional conditional distributions [25,26]. For an early application to particle physics, see for example MadMiner [27] and Bayesflow [26,28]. In our tests, coupling block-based Normalizing Flows exhibit great performance for dense phase space regions, but larger deviations when modelling tails of distributions. To boost the performance of the model we employ Continuous Normalizing Flows (CNF), a generalization of coupling block Flows based on ordinary differential equations (ODE) introduced in Sect. 4.1.

In Classifier Surrogates, the deficiency of coupling block-based normalizing flows to model distribution tails is masked to large extend by the softmax-normalization employed on the classifier, and thus also surrogate, when calculating class probabilities. We do observe similar performance between both architectures. However, CNFs are also much more parameter efficient allowing us to reduce the number of parameters needed by a factor of ≈ 20 at the cost of slower inference time. As the weights of the surrogate are designed to be shared, and we do expect their use in case studies rather than evaluating on millions of jets, we believe that CNFs are best suited for the application.

4.1 Continuous Normalizing Flows and conditional flow matching

First introduced in [29], CNFs define a transformation $\phi_t : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ called *flow* dependent on a time variable t. The time variable is the continuous equivalent to the number of a coupling blocks in a coupling block-flow [30]. Instead of having multiple flow instances, the dependence of ϕ on t is defined through the ODE

$$\frac{\text{d}}{\text{d}t}\phi_t(x) = v_t(\phi_t(x)), \quad \phi_0(x) = x, \tag{2}$$

by the time dependent *vector-field* $v_t : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$, which itself is approximated by a deep neural network

$$\tilde{v}_t(\cdot, \theta) \approx v_t.$$

While this network can be arbitrarily complex, we stick to fully-connected architectures due to the low dimensionality of the task. In our case, the flow transforms data from a Gaussian distribution $\mathcal{N}(0, 1)$ for $t = 0$ into ParT output at $t = 1$. This choice sets the boundaries of the *probability path* $p_t : [0, 1] \times \mathbb{R}^d \to \mathbb{R}_{>0}$ induced by the vector-field trough

Equation (2) and

$$p_t(x) = p_0\left(\phi_t^{-1}(x)\right) \det\left(\frac{\partial \phi_t^{-1}(x)}{\partial x}\right). \tag{3}$$

A standard CNF is trained by solving the ODE Eq. (2) in reverse and minimizing the negative log-likelihood (NLL) of input data at $t = 1$. The computation of this loss objective is expensive, especially for higher dimensional models.

Thus, the authors of [31] introduce the conditional flow matching (CFM) objective

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p_t(x|x_1)} \|u_t(x|x_1) - \tilde{v}_t(x;\theta))\|^2 \tag{4}$$

It reduces the calculation of the optimization criterion to the calculation of a mean-squared error between the network output $\tilde{v}_t(x;\theta)$ and an analytical solution u_t for sampled $t \sim \mathcal{U}(0,1)$, $x_1 \sim q$ and $x \sim p_t(\cdot|x_1)$. Here, q is the probability distribution of the input data. A good choice of u_t and corresponding p_t is a Gaussian conditional probability path with mean and variance changing linear in time (optimal transport) [31]. The CFM-loss (4) then reduces even further

$$\begin{aligned}\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p(x_0)}\Bigg\| & (x_1 - (1-\sigma_{\min})\, x_0) \\ & - \tilde{v}_t(\sigma_t x_0 + \mu_t;\theta))\Bigg\|^2,\end{aligned} \tag{5}$$

where $\mu_t = t x_1$, $\sigma_t = 1 - (1-\sigma_{\min})t$, $p(x_0) = \mathcal{N}(0,1)$ and $\sigma_{\min}$ a small parameter, that can be chosen to match the noise level of the training data.

4.2 Conditional density estimation

Following the coupling-block flow based example of [26], we can extend CNFs to approximate a conditional density

$$p_t(x \mid c) = p_0\left(\phi_t^{-1}(x,c) \mid c\right) \det\left(\frac{\partial \phi_t^{-1}(x,c)}{\partial x}\right), \tag{6}$$

where the noise distribution is independent of the condition $p_0(\cdot \mid c) = p_0(\cdot)$, by appending the vector of conditions to every layer of the vector field model $\tilde{v}_t(x, c\,;\theta)$. For our surrogate, x will be the ParT output and c will be the vector of jet-observables.

5 Bayesian Neural Networks

To indicate the application of the surrogate on data not included in tagger and thus surrogate training, we employ Bayesian deep learning. Through modeling of (or sampling from) a posterior weight distribution

$$\pi\,(\theta \mid \mathcal{D}),$$

these methods give a large spread of predictions for data not included in the loss objective during training. This posterior distribution is the distribution of weights θ of the network $\tilde{v}_t(\cdot,\theta)$ given the training data

$$\mathcal{D} = \left\{(x^{(1)}, c^{(1)}), (x^{(2)}, c^{(2)}), \ldots\right\}.$$

Multiple instances from the weight posterior will form an ensemble of networks with differing weights. With both being conditional probability distributions, the weight posterior has to be distinguished from the likelihood of classifier output (1) that is to be inferred by every CNF in the ensemble. Sections 5.1 and 5.2 introduce two different approaches to connect both distributions.

5.1 Mean-field Gaussian variational inference (VIB)

A first way to relate the the weight posterior $\pi\,(\theta \mid \mathcal{D})$ to a CNF is to approximate it with an uncorrelated Normal distribution $\tilde{\pi}(\theta)$ [32]. This approximation is usually inferred during optimization of the network, by minimizing the Kullback–Leibler divergence (D_{KL}) between the posterior and its approximation

$$\begin{aligned}\mathcal{L}_{\mathrm{VIB}} &= D_{\mathrm{KL}}\left[\tilde{\pi}(\theta), \pi\,(\theta \mid \mathcal{D})\right] \\ &= -\int \mathrm{d}\theta\, \tilde{\pi}(\theta)\, \log \pi\,(\mathcal{D} \mid \theta) \\ &\quad + D_{\mathrm{KL}}\left[\tilde{\pi}(\theta), \pi(\theta)\right] + \text{ constant},\end{aligned} \tag{7}$$

where $\pi(\theta)$ is the prior imposed on the network weights. Following the construction in [33], we bridge the gap between the CFM-loss (5) and the log-likelihood of the data in (7) by introducing a factor k that can be optimized to account for the difference

$$\mathcal{L}_{\mathrm{VIB-CFM}} = \mathbb{E}_{\tilde{\pi}(\theta)}\mathcal{L}_{\mathrm{CFM}} + k D_{\mathrm{KL}}\left[\tilde{\pi}(\theta), \pi(\theta)\right]. \tag{8}$$

5.2 `AdamMCMC`

While the derivation of the loss (8) lacks theoretic backing and its optimization can take considerably longer than that of the CFM-loss (5) alone, the low dimensionality of the Classifier Surrogate problem allows us to directly sample the weight posterior distribution through Markov chain Monte Carlo (MCMC).

Full Hamiltonian Monte Carlo (HMC) is still often considered the gold-standard for inferring weight posteriors [34].

The large size of the training data however forces us to use stochastic MCMC algorithms. As one instance of this class, we choose `AdamMCMC` [35] due to its easy implementation. Competing algorithms, such as stochastic gradient HMC [36] or symmetric splitting HMC [37], will likely produce similar results.

We initialize the `AdamMCMC`-chain with a network optimized using the CFM-loss objective (5) and solve the ODE (2) to determine the negative log-likelihood $\mathcal{L}_{\mathrm{NLL}}$ of the data for every step of the MCMC from there on.

To ensure detailed balance we employ a Metropolis–Hastings (MH) correction with acceptance rate

$$\alpha = \frac{\exp\left(-\lambda \mathcal{L}_{\mathrm{NLL}}(\tau_i)\right) q(\theta_i \mid \tau_i)}{\exp\left(-\lambda \mathcal{L}_{\mathrm{NLL}}(\theta_i)\right) q(\tau_i \mid \theta_i)} \tag{9}$$

for all steps of the chain. Through the proportionality $\pi(\theta \mid \mathcal{D}) \propto -\mathcal{L}_{\mathrm{NLL}}$ (Bayes formula) the acceptance step guarantees sampling from the weight posterior. Here, the parameter λ gives the inverse temperature of the tempered posterior distribution sampled from. The proposed weights τ_i are drawn from a proposal distribution centered on a gradient descent step

$$\tilde{\theta}_{i+1} = \mathrm{Adam}(\theta_i, \mathcal{L}_{\mathrm{NLL}}(\theta_i)) \tag{10}$$

calculated using the `Adam` algorithm [38]. We can use the momentum terms of the update to ensure high acceptance rates by smearing the proposal distribution in the direction of the last update

$$\begin{aligned} \tau_i &\sim q(\cdot \mid \theta_i) \\ &= \mathcal{N}(\tilde{\theta}_{i+1}, \sigma^2 \mathbb{1} + (\tilde{\theta}_{i+1} - \theta_i)(\tilde{\theta}_{i+1} - \theta_i)^{\top}). \end{aligned} \tag{11}$$

To efficiently run this algorithm, we evaluate the NLL on batches of data. For proofs on convergence and invariant distribution of this algorithm, we refer to [35].

6 Results

To learn the detector smearing distribution from data, we found a CNF with only 3 multi-layer perceptrons (MLPs) with 3 layers of dimension 64 and ELU activation to be sufficient. The condition and time variable t are concatenated to every MLP input, totaling in 31617 network parameters. Converting to VIB as in [32], doubles the number of parameters. We train on a balanced set of 4M jets in batches of 131,072 for 4000 epochs using `Adam` [38] with a constant learning rate of 10^{-3}. As loss objective, we use the CFM-loss as introduced in Eqs. (5) and (8) respectively. To achieve good coverage, we choose $c = 100$ and $\lambda = 50$ from a course grid search over multiple orders of magnitude.

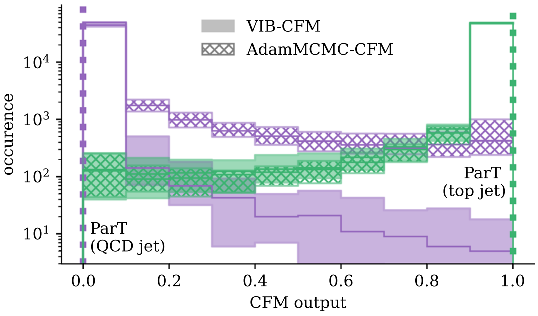


Fig. 2 Histograms of 50,000 samples drawn form the detector smearing distributions learned with a CFM-model. Uncertainties are generated by drawing the samples from 11 points sampled from the network posterior approximation or chain. The ParT-output for the arbitrary QCD and top jet used as condition is indicated with dotted lines. Both jet events are the same as for Fig. 1

We run the `AdamMCMC` chain for another 1000 epochs with the learning rate reduced to $5 \cdot 10^{-6}$ and $\sigma = 0.05$. For the sampled posterior we always report the results from CFM-optimization in solid lines and the uncertainty calculated as the min-max-envelope of 10 drawings and for the learned approximation (VIB) we give the mean and the min-max-envelope over 11 sets of weights.

Using a fully-connected architecture, the sampled networks, either from the VIB-approximation or MCMC, can be easily exported as a serialized file using ONNX [9] at only 0.3 MB per instance. The the ODE defined by the network remains to be solved at inference time.

6.1 In-distribution

We can use the trained CNFs to generate another approximation of the detector smearing distribution by performing the forward direction starting at different points in latent space but for the same high-level features. Figure 2 shows histograms of the generated data for the same arbitrary jet events as Fig. 1.

We can see similar distributions for the approximation with CNFs as for the histograms of the closest events. The biggest discrepancy occurs between the distribution for the QCD jet obtained using `AdamMCMC` and VIB. It can be attributed to the difference between the model at initialization of the `AdamMCMC` chain and the posterior mean output of VIB. The initialization can be adapted to accommodate desired behaviours, if well defined, by choosing between different epochs of the CFM-optimization. Furthermore, increasing the chain length decreases the dependence of the ensemble output on the initialization overall.

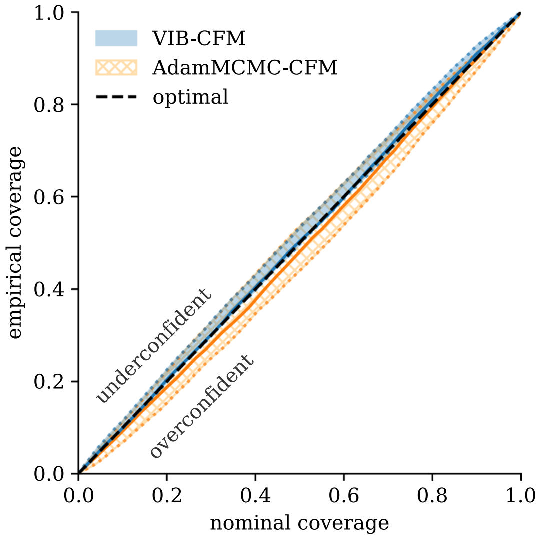


Fig. 3 Empirical over nominal coverage calculated by taking 1000 samples from the learned detector smearing distribution for 10,000 jet events each. Uncertainties again are calculated from 11 points of the network posterior

6.1.1 *Uncertainty calibration*

To find out whether surrogate predictions using `AdamMCMC` are in general conservative, we need to look at the calibration of the estimated detector smearing distributions for multiple events, here 10,000. Per event we take 1000 samples from the inferred distribution and calculate q-quantiles for 50 values of q (nominal coverage) linearly spaced between 0 and 1. We then evaluate the empirical coverage, that is the fraction of corresponding ParT output within the respective quantile of the inferred distribution. The calibration is perfect when nominal and empirical coverage agree. Figure 3 shows very good calibration for both methods, where `AdamMCMC` in fact tends to be slightly more confident than VIB approximations.

6.1.2 *Epistemic uncertainty*

In contrast to uncertainty due to noisy data resulting in the detector smearing distributions, epistemic uncertainty is the uncertainty encoded in the variations within the ensembles of network weights induced by data sparsity. For a further dive into the behaviour of the epistemic uncertainty, we calculate the mean distance of the maximum discrepancy between instances of the network posterior

$$\delta_{\text{epis}} = \frac{1}{n_{\text{stat}}} \sum_{i=0}^{n_{\text{stat}}} \max_{p(\theta|\mathcal{D})} \phi_{1,\theta}(x_i) - \min_{p(\theta|\mathcal{D})} \phi_{1,\theta}(x_i) \tag{12}$$

for a total of $n_{\text{stat}} = 1000$ points drawn from the Gaussian latent space $x_1, \ldots, x_{n_{\text{stat}}} \sim \mathcal{N}(0, 1)$. Ideally, this error estimate is large for sparsely populated areas of the high-level feature space and small in the bulk of the distribution. To investigate this behaviour, we plot a histogram of the high-level features of the training data as well as δ_{epis} for 10,000 jet events chosen at random from a test set for both methods in Fig. 4.

The most instructive panels show the dependence of the error estimate on the number of constituents in the jet n_{const}, which is the most descriptive input feature. We can see high uncertainties occurring in the regions where the distributions for QCD and top jets overlap in the training data. These are events that can not easily be attributed to one of the two classes by the five high-level observables alone, resulting in high uncertainties. These events also make up the high-error bulk when plotted over the other high-level features.

For every tailed distribution, we can also see an increase of the error estimate for top jet predictions towards the edges of the data. This behaviour is stronger for `AdamMCMC` than for VIB at the cost of higher uncertainties overall.

The same behaviour is not observed for QCD jets. This again can be traced back to the distribution of n_{const}. The distribution of the number of particles of top jets is fully within the support of the one for QCD jets inducing high epistemic uncertainties for both highly and lowly populated jets. On the other hand, the distribution of top jets does not include events with as few particles as for QCD jets, allowing a perfect classification of these jets that dominates the low uncertainty edge of the plotted cloud.

6.1.3 *Adding informative features*

Another measure for the informative value of a detector smearing distribution generated by a Classifier Surrogate is the predicted accuracy

$$\hat{a} = \frac{1}{n_{\text{stat}}} \sum_{i=0}^{n_{\text{stat}}} \begin{cases} \mathbf{1}_{[0.5,1]}\left(\phi_{1,\theta}(x_i)\right) & \text{for top jets} \\ \mathbf{1}_{[0,0.5)}\left(\phi_{1,\theta}(x_i)\right) & \text{for QCD jets} \end{cases} \tag{13}$$

per jet event, with $\mathbf{1}_A(x)$ the indicator function of set A. The cut value of 0.5 is arbitrary and can be chosen in line with the experimental analysis. Our choice reflects the requirement to yield symmetric output distributions in case of uninformative high-level input.

Figure 5 shows histograms of the predicted accuracy for 10,000 jet events chosen at random from the full balanced test set. The distributions are generated from surrogates using the five high-level jet features from before, as well as for surrogates including the soft drop mass m_{SD} and the N-subjettiness for $N \in \{1, .., 4\}$. Naively, we assume that adding more information will lead to more certain predictions and thus will shift the distributions towards high accuracy values. In the highest value bin, the information hierarchy is well reproduced, with the highest number of input

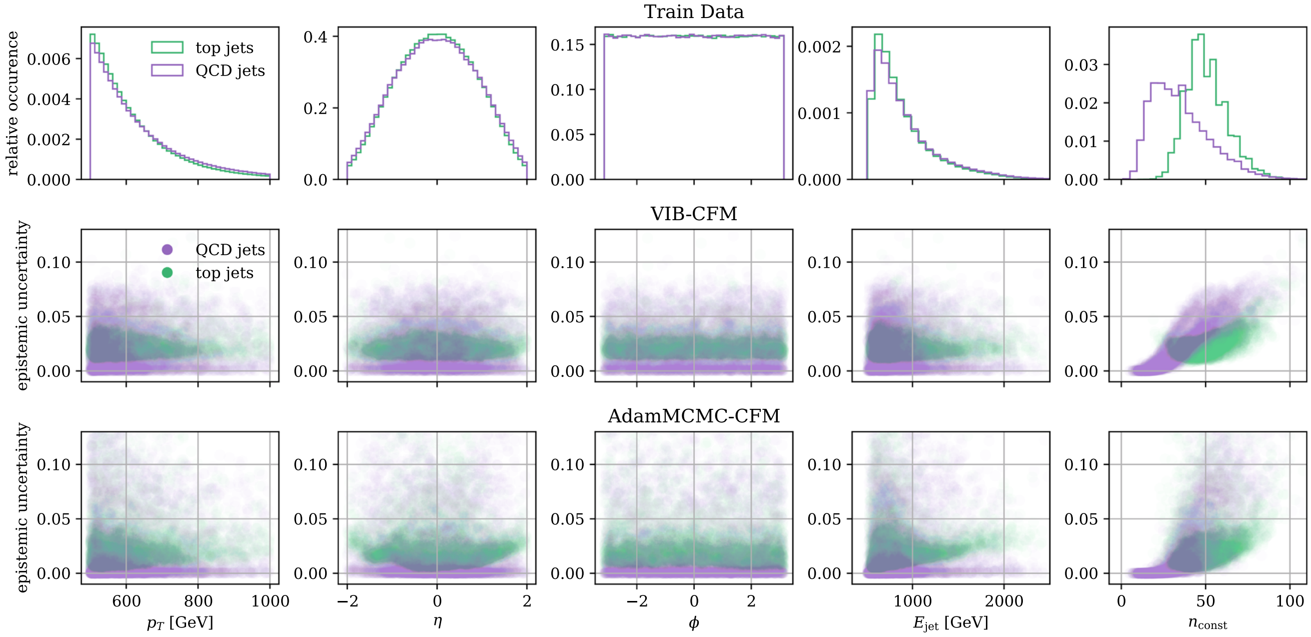


Fig. 4 Epistemic uncertainty calculated from the mean difference between 11 points from the network posterior over 1000 samples drawn from the respective learned detector smearing distribution for each event of the validation set. The uncertainty shows a clear scaling towards the edges of the train data, as well es in regions where n_{const} is uninformative

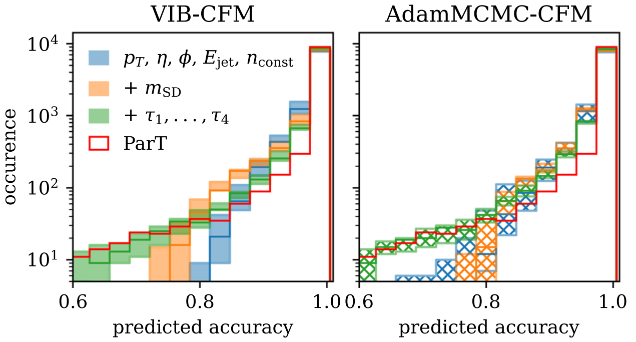


Fig. 5 Accuracy of 1000 ParT outputs predicted for each of 10,000 jet events. The different colors indicate the output of CNFs conditioned with increasing amount of features and thus provided with more information during inference. A histogram of the probabilistic ParT prediction itself is given in red

features leading to the highest number of certain outputs. In the range of 0.85 to 1, more informative input leads to fewer predictions in line with the naive assumption. For less certain predictions, a different effect can be observed. Increasing the information in the conditions allows the network to better model the ParT output, which features long tails of individual false positives and events predicted with low confidence. Thus, the Jensen-Shennon divergence between the histograms of surrogate and ParT output (Table 1) decreases with increasing number of input features.

Table 1 Jensen–Shannon-divergence between the histograms of predicted accuracies of Classifier Surrogates with different input features (Fig. 5) and the actual accuracy distribution of the ParT

JSD	VIB-CFM	AdamMCMC-CFM
p_T, η, ϕ, E_{jet}, n_{const}	0.174 ± 0.018	0.147 ± 0.036
+ m_{SD}	0.134 ± 0.023	0.160 ± 0.013
+ $\tau_1, \ldots, \tau_4$	0.080 ± 0.009	0.097 ± 0.007

6.2 Out-of-distribution

Although including an epistemic uncertainty into the evaluation this far is a nice feature to gauge uncertainties in the tail regions of the data, the true value of BNNs is indicating input that is outside the distribution of the training data by assigning high uncertainties. We use the introduced measures (12) and (13) to show the behaviour of the BNN surrogates for OOD data generated when artificially increasing the values for one jet-observable.

We produce OOD data by selecting 1000 jet events from the test set at random and increasing the values of a single jet-feature by adding a constant value. We perform this distortion for 3 dimensions, p_T, E_{jet} and n_{const}, and 10 values each. Again, we report the accuracy and error estimate calculated from $n_{\text{stat}} = 1000$ points of the learned detector smearing distribution.

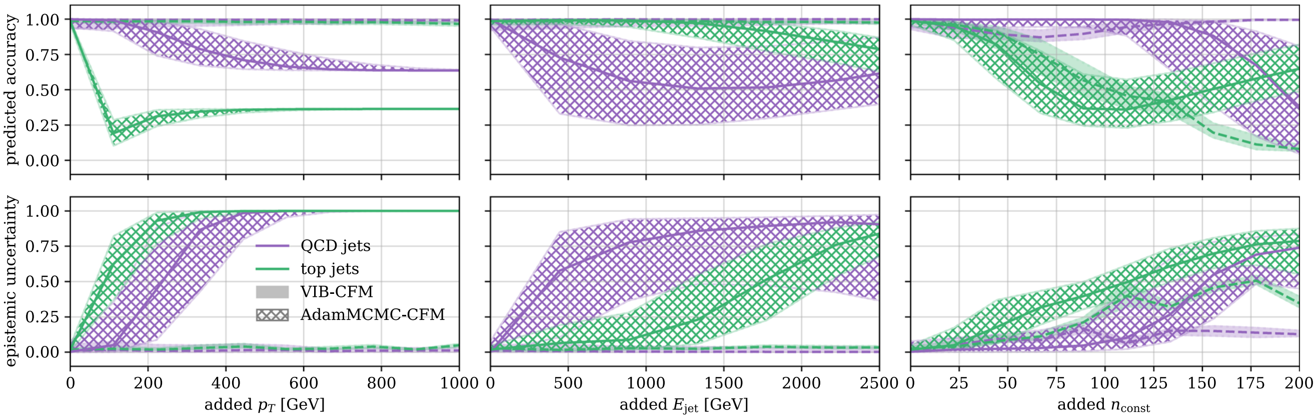


Fig. 6 Behaviour of the Classifier Surrogate predictions for distorted input over the size of the distortion artificially added to the given jet-observable for 1000 random events. The first row shows the accuracy predicted per event while the second row gives the mean epistemic uncertainty per event. Solid lines give the median of the set of events. The shaded and structured areas indicate the 10%–90%-quantile envelope of the VIB and `AdamMCMC` ensemble respectively

The first row of Fig. 6 shows the mean accuracy predicted for the OOD data by the different drawings from the weight posterior. The envelope and solid line give the 10%- and 90%-quantile and the median over the set of events. When adding an unphysical offset to the features, we can see the mean predicted accuracy of the `AdamMCMC` ensemble rapidly drops. Optimally, the network predicts 0.5 when all inputs are outside the training interval to indicate equal confidence of both classes. The ensemble seems to be able to detect most outliers, but only indicates large distortions of E_{jet} for top jets and of n_{const} for QCD jets.

The predicted accuracy of the VIB samples does not exhibit any dependence on the increasing offset in the OOD data. It is sensitive only to the number of jet constituents for top jets.

In the second row, we show the error estimate based on the difference between highest and lowest proposed output in the ensemble, see Eq. 12. This measure captures the differences in the output and thus the encoded uncertainty directly. We expect increasing uncertainties for increasing offset. Only the `AdamMCMC` ensemble shows this behaviour, for all three disturbed input dimensions, while VIB once again is only sensitive to OOD inputs in the particle number. While the predicted accuracy did not capture the decreasing confidence for distorted E_{jet} of top jets well, the error estimate clearly indicates the unknown inputs. Similarly, distortions in n_{const} of QCD jets appear earlier in this measure.

7 Conclusion

In this paper, we proposed a first architecture for training Classifier Surrogates, which are models describing the output of a deep neural network classification based on detector-level information from high-level jet-observables and truth information. We show that the resulting Classifier Surrogates are well calibrated and scale with the amount of information provided. A combination with Monte Carlo generated samples from the networks Bayesian weight posterior allows for stable uncertainty quantification, that incorporates the density of the training data towards the edges. The predicted uncertainty reliably indicates unknown inputs.

This approach should next be implemented by the large experimental collaborations to allow the statistical re-interpretation of analysis results.

Acknowledgements We thank the organisers and participants of the Les Houches – PhysTeV 2023 workshop for the stimulating environment which led to the idea for this work. Furthermore, we thank Joschka Birk for helping with data preparation in the early stages of the work.

Funding SB is supported by the Helmholtz Information and Data Science Schools via DASHH (Data Science in Hamburg – HELMHOLTZ Graduate School for the Structure of Matter) with the grant HIDSS-0002. SB and GK acknowledge support by the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy – EXC 2121 Quantum Universe – 390833306. JK is supported by the Alexander-von-Humboldt-Stiftung. This research was supported in part through the Maxwell computational resources operated at Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany.

Data Availability Statement response This manuscript has associated data in a data repository. [Authors' comment: The simulated data used in this work, is publicly available [20]. The exact handling of the data can be accessed at https://github.com/joschkabirk/jetclass-top-qcd.]

Code Availability Statement This manuscript has associated code/software in a data repository. [Authors' comment: The code generated during this study is available from https://github.com/sbieringer/ClassificationSurrogates.]

Declarations

Code The training and evaluation code is available from https://github.com/sbieringer/ClassificationSurrogates and an example on handeling the JetClass dataset is given in https://github.com/joschkabirk/jetclass-top-qcd.

Funded by SCOAP^3.

References

1. D. Guest, K. Cranmer, D. Whiteson, Deep learning and its application to LHC physics. Ann. Rev. Nucl. Part. Sci. **68**, 161–181 (2018). https://doi.org/10.1146/annurev-nucl-101917-021019. arXiv:1806.11484 [hep-ex]
2. K. Albertsson et al., Machine learning in high energy physics community white paper. J. Phys. Conf. Ser. **1085**(2), 022008 (2018). https://doi.org/10.1088/1742-6596/1085/2/022008. arXiv:1807.02876 [physics.comp-ph]
3. A. Radovic et al., Machine learning at the energy and intensity frontiers of particle physics. Nature **560**(7716), 41–48 (2018). https://doi.org/10.1038/s41586-018-0361-2
4. G. Karagiorgi, G. Kasieczka, S. Kravitz, B. Nachman, D. Shih, Machine learning in the search for new fundamental physics (2021). arXiv:2112.03769 [hep-ph]
5. S. Kraml et al., Searches for new physics: Les Houches recommendations for the presentation of LHC results. Eur. Phys. J. C **72**, 1976 (2012). https://doi.org/10.1140/epjc/s10052-012-1976-3. arXiv:1203.2489 [hep-ph]
6. W. Abdallah et al., Reinterpretation of LHC results for new physics: status and recommendations after Run 2. SciPost Phys. **9**(2), 022 (2020). https://doi.org/10.21468/SciPostPhys.9.2.022. arXiv:2003.07868 [hep-ph]
7. J.Y. Araz et al., Les Houches guide to reusable ML models in LHC analyses (2023). arXiv:2312.14575 [hep-ph]
8. D.H. Guest et al., Lwtnn/lwtnn: Version 2.13. https://doi.org/10.5281/zenodo.6467676
9. Open Neural Network Exchange. https://onnx.ai
10. ATLAS collaboration, Search for R-parity-violating supersymmetry in a final state containing leptons and many jets with the ATLAS experiment using $\sqrt{s} = 13 TeV$ proton–proton collision data. Eur. Phys. J. C **81**(11), 1023 (2021). https://doi.org/10.1140/epjc/s10052-021-09761-x. arXiv:2106.09609 [hep-ex]
11. ATLAS collaboration, Search for supersymmetry in final states with missing transverse momentum and three or more b-jets in 139 fb^{-1} of proton–proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector. Eur. Phys. J. C **83**(7), 561 (2023). https://doi.org/10.1140/epjc/s10052-023-11543-6. arXiv:2211.08028 [hep-ex]
12. ATLAS collaboration, Search for neutral long-lived particles in pp collisions at $\sqrt{s} = 13$ TeV that decay into displaced hadronic jets in the ATLAS calorimeter. JHEP **06**, 005 (2022). https://doi.org/10.1007/JHEP06(2022)005. arXiv:2203.01009 [hep-ex]
13. ATLAS collaboration, Anomaly detection search for new resonances decaying into a Higgs boson and a generic new particle X in hadronic final states using $\sqrt{s} = 13$ TeV pp collisions with the ATLAS detector. Phys. Rev. D **108**, 052009 (2023). https://doi.org/10.1103/PhysRevD.108.052009. arXiv:2306.03637 [hep-ex]
14. ATLAS collaboration, Performance of b-Jet Identification in the ATLAS Experiment. JINST **11**(04), 04008 (2016). https://doi.org/10.1088/1748-0221/11/04/P04008. arXiv:1512.01094 [hep-ex]
15. CMS collaboration, Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV. JINST **13**(05), 05011 (2018). https://doi.org/10.1088/1748-0221/13/05/P05011. arXiv:1712.07158 [physics.ins-det]
16. H. Qu, C. Li, S. Qian, Particle transformer for jet tagging, in *Proceedings of the 39th International Conference on Machine Learning*, pp. 18281–18292 (2022)
17. K. Cranmer et al., Publishing statistical models: getting the most out of particle physics experiments. SciPost Phys. **12**(1), 037 (2022). https://doi.org/10.21468/SciPostPhys.12.1.037. arXiv:2109.04981 [hep-ph]
18. J. Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, M. Selvaggi, Delphes 3: a modular framework for fast simulation of a generic collider experiment. J. High Energy Phys. (2014). https://doi.org/10.1007/jhep02(2014)057
19. T. Dorigo, P. De Castro Manzano, Dealing with Nuisance Parameters using Machine Learning in High Energy Physics: a Review (2020). arXiv:2007.09121 [stat.ML]
20. H. Qu, C. Li, S. Qian, JetClass: A Large-Scale Dataset for Deep Learning in Jet Physics. https://doi.org/10.5281/zenodo.6619768
21. A.J. Larkoski, S. Marzani, G. Soyez, J. Thaler, Soft drop. JHEP **05**, 146 (2014). https://doi.org/10.1007/JHEP05(2014)146. arXiv:1402.2657 [hep-ph]
22. J. Thaler, K. Van Tilburg, Identifying boosted objects with N-subjettiness. JHEP **03**, 015 (2011). https://doi.org/10.1007/JHEP03(2011)015. arXiv:1011.2268 [hep-ph]
23. S. Badger et al., Machine learning and LHC event generation. SciPost Phys. **14**(4), 079 (2023). https://doi.org/10.21468/SciPostPhys.14.4.079. arXiv:2203.07460 [hep-ph]
24. H. Hashemi, C. Krause, Deep generative models for detector signature simulation: an analytical taxonomy (2023). arXiv:2312.09597 [physics.ins-det]
25. C. Winkler, D.E. Worrall, E. Hoogeboom, M. Welling, Learning likelihoods with conditional normalizing flows. CoRR (2019). arXiv:1912.00042 [cs.lg]
26. S.T. Radev, U.K. Mertens, A. Voss, L. Ardizzone, U. Köthe, Bayesflow: learning complex stochastic models with invertible neural networks. IEEE Trans. Neural Netw. Learn. Syst. **33**(4), 1452–1466 (2020). arXiv:2003.06281 [stat.ML]
27. J. Brehmer, F. Kling, I. Espejo, K. Cranmer, MadMiner: machine learning-based inference for particle physics. Comput. Softw. Big Sci. **4**(1), 3 (2020). https://doi.org/10.1007/s41781-020-0035-2. arXiv:1907.10621 [hep-ph]
28. S. Bieringer, A. Butter, T. Heimel, S. Höche, U. Köthe, T. Plehn, S.T. Radev, Measuring QCD splittings with invertible networks. SciPost Phys. **10**(6), 126 (2021). https://doi.org/10.21468/SciPostPhys.10.6.126. arXiv:2012.09873 [hep-ph]
29. R.T. Chen, Y. Rubanova, J. Bettencourt, D.K. Duvenaud, Neural ordinary differential equations. Advances in neural information processing systems, vol. 31 (2018). arXiv:1806.07366 [cs.LG]
30. D. Rezende, S. Mohamed, Variational inference with normalizing flows, in *International Conference on Machine Learning*, pp. 1530–1538. PMLR (2015)
31. Y. Lipman, R.T.Q. Chen, H. Ben-Hamu, M. Nickel, M. Le, Flow matching for generative modeling, in *The Eleventh International Conference on Learning Representations* (2023)
32. C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural network, in *International Conference on Machine Learning*, pp. 1613–1622. PMLR (2015)
33. A. Butter et al., Jet diffusion versus JetGPT—modern networks for the LHC (2023). arXiv:2305.10475 [hep-ph]

34. P. Izmailov, S. Vikram, M.D. Hoffman, A.G.G. Wilson, What are Bayesian neural network posteriors really like? in *International Conference on Machine Learning*, pp. 4629–4640. PMLR (2021)
35. S. Bieringer, G. Kasieczka, M.F. Steffen, M. Trabs, AdamMCMC: Combining Metropolis adjusted Langevin with momentum-based optimization (2023). arXiv:2312.14027 [stat.ML]
36. T. Chen, E. Fox, C. Guestrin, Stochastic gradient Hamiltonian monte carlo, in *International Conference on Machine Learning*, pp. 1683–1691. PMLR (2014)
37. A.D. Cobb, B. Jalaian, Scaling hamiltonian monte carlo inference for Bayesian neural networks with symmetric splitting, in *Uncertainty in Artificial Intelligence*, pp. 675–685. PMLR (2021)
38. D.P. Kingma, J. Ba, Adam: A method for stochastic optimization. CoRR (2014). arXiv:1412.6980 [cs.LG]

MACHINE
LEARNING
Science and Technology

CrossMark

OPEN ACCESS

RECEIVED
5 August 2024

REVISED
29 October 2024

ACCEPTED FOR PUBLICATION
11 November 2024

PUBLISHED
20 November 2024

PAPER

Calibrating Bayesian generative machine learning for Bayesiamplification

S Bieringer[1,*], S Diefenbacher[2], G Kasieczka[1] and M Trabs[3]

[1] Institut für Experimentalphysik, Universität Hamburg, Luruper Chaussee 149, 22761 Hamburg, Germany
[2] Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States of America
[3] Department of Mathematics, Karlsruhe Institute of Technology, Englerstr. 2, 76131 Karlsruhe, Germany
[*] Author to whom any correspondence should be addressed.

E-mail: sebastian.guido.bieringer@uni-hamburg.de

Keywords: Bayesian neural networks, generative neural networks, data amplification, fast detector simulation

Abstract

Recently, combinations of generative and Bayesian deep learning have been introduced in particle physics for both fast detector simulation and inference tasks. These neural networks aim to quantify the uncertainty on the generated distribution originating from limited training statistics. The interpretation of a distribution-wide uncertainty however remains ill-defined. We show a clear scheme for quantifying the calibration of Bayesian generative machine learning models. For a Continuous Normalizing Flow applied to a low-dimensional toy example, we evaluate the calibration of Bayesian uncertainties from either a mean-field Gaussian weight posterior, or Monte Carlo sampling network weights, to gauge their behaviour on unsteady distribution edges. Well calibrated uncertainties can then be used to roughly estimate the number of uncorrelated truth samples that are equivalent to the generated sample and clearly indicate data amplification for smooth features of the distribution.

1. Introduction

The upcoming high-luminosity runs of the LHC will push the quantitative frontier of data taking to over 25-times its current rates. To ensure precision gains from such high statistics, this increase in experimental data needs to be met by an equal amount of simulation. The required computational power is predicted to outgrow the increase in budget in the coming years [1, 2]. One solution to this predicament is the augmentation of the expensive, Monte Carlo-based, simulation chain with generative machine learning. A special focus is often put on the costly detector simulation [3, 4].

This approach is only viable under the assumption that the generated data is not statistically limited to the size of the simulated training data. Previous studies have shown, for both toy data [5] and calorimeter images [6], that samples generated with generative neural networks can surpass the training statistics due to powerful interpolation abilities of the network in data space. These studies rely on comparing a distance measure between histograms of generated data and true hold-out data to the distance between smaller, statistically limited sets of Monte Carlo data and the hold-out set. The phenomenon of a generative model surpassing the precision of its training set is also known as amplification. While interesting in theory and crucial for the pursuit of the amplification approach, these studies can not be performed in experimental applications as they rely on large validation sets multiple orders of magnitude bigger than the training data.

Recently, generative architectures employing Bayesian network weight posteriors have been applied to event generation [7] allowing the generation of sets of data with a corresponding uncertainty on the generated data distribution. In the limit of large generated sets, this uncertainty is entirely based in the statistical limitations of the training data. For well calibrated uncertainty predictions, this raises the question whether an estimate of statistical power of the generated data can be formed from the uncertainty prediction itself. In this paper,

- we introduce a technique for quantifying the calibration of Bayesian uncertainties on generative neural networks based on the mean coverage of the prediction.
- We then develop an estimate of the number of simulated truth events matching the generated set in statistical power and validate this estimate.

For applications where the uncertainty calibration can be ensured, for example by evaluating on a validation region, this approach gives an inherent quantification of the significance of a generated set.

In Bayesian neural networks (BNNs) and beyond, calibrating uncertainty quantification is crucial for correct application of the prediction results [8]. While we prefer the uncertainties to align perfectly with the prediction error, overconfident predictions will lead to inflated significance values and false discoveries. Underconfident predictions on the other hand will obscure findings, but not lead to false results and can thus be tolerated to small extend.

Bayesian generative machine learning is inherently different from other BNNs in particle physics applications such as regression [9] or classification [10, 11]. Notably, in generative modeling, a low density region of data cannot be understood as low training statistics, but rather as a feature of the data that has to reproduced by the network. The uncertainty estimate thus behaves similarly to a low-dimensional, parameterized fit [12] introducing high error estimates at steep features of the data distribution or whenever the function class induced by the network architecture is not sufficient to reproduce the data. In a subsequent study of the quality of event generators [13], the authors also connect low uncertainty to good performance of the posterior mean in terms of a classifier test, but find that the weight distribution of a classifier is more sensitive to diverse failure modes than the Bayesian uncertainty.

In section 2, we will explain the basic concepts of BNNs, while the connection to generative machine learning will be made in section 3. We introduce the toy data, as well as the employed binning in section 4 and use them to evaluate the calibration of two different classes of BNNs in section 5. The idea of employing the Bayesian uncertainties for amplification is developed and deployed in section 6, before we conclude in section 7.

2. BNNs

In contrast to traditional, frequentist deep neural networks, in a Bayesian phrasing of deep learning, a distribution on the network weights is applied. This distribution encodes the belief in the occurrence of the weight configuration θ. This, so called *posterior* distribution

$$\pi\left(\theta|\mathcal{D}\right) = \frac{\pi\left(\mathcal{D}|\theta\right)\pi\left(\theta\right)}{\pi\left(\mathcal{D}\right)} \tag{2.1}$$

is formed from our *prior* beliefs $\pi(\theta)$ and the *likelihood* $\pi(\mathcal{D}|\theta)$ of the data $\mathcal{D}$ under the model. While the likelihood gives the probability of the data given its modelling through the network and thus encodes the data inherent distribution (aleatoric uncertainty), the posterior distribution provides the uncertainty due to a lack of data (epistemic uncertainty) [14].

Multiple methods of accessing the posterior distribution exist. For a broad overview over the existing techniques, we refer the readers to [8, 14–16]. They can mostly be classified as either approximating or sampling the posterior.

One popular option is approximating the posterior as an uncorrelated Gaussian distribution by learning a mean and a standard deviation per network weight. These parameters of the approximation are then inferred with (stochastic) variational inference. This technique is also referred to as 'Bayes-by-Backprop' [17] or within High-Energy Physics often understood as 'Bayesian Neural Networks'. We will refer to it as 'Variational Inference Bayes'(VIB).

For sampling the posterior, Markov Chain Monte Carlo (MCMC) methods are employed, with full Hamiltonian Monte Carlo (HMC) often considered the gold-standard [18]. To adapt this class of methods to the large datasets and high dimensional parameter spaces of deep learning stochastic and gradient-based chains have been developed. Most notably among them are stochastic gradient HMC [19] and its variations. Due to its easy application to different machine learning tasks and great performance on previous generative applications [20], we use `AdamMCMC` [21] as one instance of MCMC-based Bayesian inference of network weights.

With access to the posterior distribution of a neural network $f_\theta(x) = y$, we can generate the network prediction as the posterior mean prediction and its uncertainty prediction as

$$\hat{y} = \int \mathrm{d}\theta\,\pi\left(\theta|\mathcal{D}\right) f_\theta\left(x\right) \quad \text{and} \quad \sigma^2_{\hat{y}} = \int \mathrm{d}\theta\,\pi\left(\theta|\mathcal{D}\right)\left[f_\theta\left(x\right) - \hat{y}\right]^2. \tag{2.2}$$

Here, the integration is approximated as a summation over an ensemble of network weights obtained from the posterior directly via sampling or from its approximation.

For generative machine learning, a per-sample uncertainty cannot be evaluated due to the unsupervised setup of the problem. We thus generate sets of data with every network weight instance in the ensemble, calculate histograms for each set and report the mean and standard deviation per bin over all sets. This allows us to compare against the expected truth values in each histogram bin.

3. Bayesian continuous normalizing flows (CNFs)

Generative models of various flavours have been applied for fast simulation of detector effects [3, 4]. Meanwhile, Normalizing Flows, both block-based [22] and continuous [23], can be connected to Bayesian machine learning straight-forwardly, as the log-likelihood of the model is accessible. Due to the recent success of diffusion-style models in detector emulation [24–29] and their high data efficiency, we combine both and concentrate on CNFs in this study.

Let $x \in \mathbb{R}^d$ be a point in the data set $\mathcal{D}$. Following [30] , we first introduce the *flow* mapping $\phi_t : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ parameterized by a time parameter $t \in [0,1]$. In analogy to the application of multiple blocks in a coupling-block flow [22], the change of the flow mapping between target and latent space is determined by an ordinary differential equation (ODE)

$$\frac{\mathrm{d}}{\mathrm{d}t}\phi_t(x) = v_t(\phi_t(x)), \quad \phi_0(x) = x, \tag{3.1}$$

through a time dependent *vector-field* v_t: $[0,1] \times \mathbb{R}^d \to \mathbb{R}^d$. For Diffusion Models this differential equation is promoted to a stochastic differential equation through the addition of time-dependent noise. For both cases, the vector-field is approximated using a deep neural network

$$\tilde{v}_t(\cdot,\theta) \approx v_t.$$

By convention, the flow is constructed to model latent data from a standard Gaussian at $t = 0$ and detector/toy data at $t = 1$. This defines the the boundaries of the probability path induced by the flow mapping

$$p_t(x) = p_0\left(\phi_t^{-1}(x)\right) \det\left(\frac{\partial\phi_t^{-1}(x)}{\partial x}\right). \tag{3.2}$$

To circumvent solving the ODE to calculate the likelihood of the input data during training, we employ conditional flow matching (CFM) [30]. Instead of the arduous ODE solving, the CFM loss objective matches the neural network predictions $\tilde{v}_t(x;\theta)$ to an analytical solution u_t, by minimizing their respective mean-squared distance

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p_t(x|x_1)} \left\|u_t(x|x_1) - \tilde{v}_t(x;\theta)\right\|^2. \tag{3.3}$$

The expectation value is calculated by sampling $t \sim \mathcal{U}(0,1)$, $x_1 \sim q$ and $x \sim p_t(\cdot|x_1)$, with q the probability distribution of the detector/toy data. An efficient and powerful choice of u_t is the optimal transport path [30]. By applying a Gaussian conditional probability path the CFM loss objective reduces to

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p(x_0)} \left\|(x_1 - (1-\sigma_{\mathrm{min}})x_0) - \tilde{v}_t(\sigma_t x_0 + \mu_t;\theta)\right\|^2. \tag{3.4}$$

Here, we use the conventions $\mu_t = tx_1$ and $\sigma_t = 1-(1-\sigma_{\mathrm{min}})t$, as well as the Gaussian latent distribution $p(x_0) = \mathcal{N}(0,1)$ and a small parameter σ_{min}, that mimics the noise level of the training data.

3.1. VIB

The parameters of an approximation $\tilde{\pi}(\theta)$ of the posterior distribution $\pi(\theta|\mathcal{D})$ can be inferred, by minimizing their Kullback–Leibler (KL) divergence using stochastic gradient descent methods [17]. As the posterior is not analytically accessible, Bayes' theorem (2.1) is employed to rewrite the KL divergence in terms of the log-likelihood and the distance to the prior

$$\mathcal{L}_{\mathrm{VIB}} = D_{\mathrm{KL}}\left[\tilde{\pi}(\theta),\pi(\theta|\mathcal{D})\right] = -\int \mathrm{d}\theta\,\tilde{\pi}(\theta)\log\pi(\mathcal{D}|\theta) + D_{\mathrm{KL}}\left[\tilde{\pi}(\theta),\pi(\theta)\right] + \mathrm{constant}. \tag{3.5}$$

The log-likelihood of the data under the CNF can be directly employed here. However, calculating the log-likelihood of a CNF is costly as the ODE (3.1) needs to be solved for every point in the training data. The authors of [7] thus propose, to substitute the log-likelihood with the CFM loss (3.4) and attribute for the difference by a tunable factor k

$$\mathcal{L}_{\mathrm{VIB-CFM}} = \mathbb{E}_{\tilde{\pi}(\theta)}\mathcal{L}_{\mathrm{CFM}} + kD_{\mathrm{KL}}\left[\tilde{\pi}\left(\theta\right), \pi\left(\theta\right)\right]. \tag{3.6}$$

Similar to changing the width of the prior $\pi(\theta)$, varying k adjusts the balance of the CFM-loss to the prior and thus both the bias and variance of the predicted distributions. Trainings at low values of k produce better fits at smaller uncertainties, while higher values impact the fit performance by imposing higher smoothness at the trade-off of higher estimated uncertainties. In our experience, promoting a CFM model to a BNN this way increases the training time considerably, due to the low impact and thus slow convergence of the KL-loss term. Possible ways to mitigate this include initiating the prior distribution and the variational parameters from the a pretrained deterministic neural network [31].

3.2. MCMC

A competing approach to variational inference-based Bayesian deep learning is MCMC sampling. Our approach to MCMC sampling for neural networks, `AdamMCMC` [21], uses the independence of the sampled invariant distribution to the starting point to initiate the sampling from CFM-trained model parameters θ_0. This drastically reduces the optimization time over the joint optimization of section 3.1, and makes employing the costly log-likelihood for the consequent uncertainty quantification feasible.

For every step of the chain, the ODE (3.1) is solved to determine the negative log-likelihood $\mathcal{L}_{\mathrm{NLL}}$ of the data to construct a chain drawn from a proposal distribution around an `Adam` [32] step

$$\tilde{\theta}_{i+1} = \mathtt{Adam}\left(\theta_i, \mathcal{L}_{\mathrm{NLL}}\left(\theta_i\right)\right). \tag{3.7}$$

In combination with a proposal distribution that is elongated in the direction of the step

$$\tau_i \sim q\left(\cdot|\theta_i\right) = \mathcal{N}\left(\tilde{\theta}_{i+1}, \sigma^2 1 + \sigma_\Delta\left(\tilde{\theta}_{i+1} - \theta_i\right)\left(\tilde{\theta}_{i+1} - \theta_i\right)^\top\right). \tag{3.8}$$

This algorithm handles high dimensional sampling for neural networks very efficiently and results in a high acceptance rate in a subsequent stochastic Metropolis–Hastings correction with acceptance probability

$$\alpha = \frac{\exp\left(-\lambda\mathcal{L}_{\mathrm{NLL}}\left(\tau_i\right)\right)q\left(\theta_i|\tau_i\right)}{\exp\left(-\lambda\mathcal{L}_{\mathrm{NLL}}\left(\theta_i\right)\right)q\left(\tau_i|\theta_i\right)}, \tag{3.9}$$

for a large range of noise parameter settings. If the added noise σ is low, the results remain close to the stochastic optimization without error estimates close to zero, but if the noise levels are high, the random walk through parameter space dominates and the algorithm does not converge to a sensible parameter values. This behaviour is masked by diminishing acceptance probabilities for very low and very high σ [21].

Both the inverse temperature parameter λ and the noise parameter σ tune the predicted uncertainties. In theory small λ and high σ will result in high error estimates, albeit in practice the dependence on the inverse temperatures is very weak. We thus limit ourselves to adapting the noise parameter to align the generated uncertainties.

After an initial burn-in period, which can be skipped when initializing from a pretrained model, repeatedly saving the network parameters after gaps of length l ensures approximately independent parameter samples. The set of sampled parameters

$$\boldsymbol{\Theta}_{\mathrm{MCMC}} = \left\{\theta^{(1)}, \ldots, \theta^{(n_{\mathrm{MCMC}})}\right\} := \left\{\theta_{1\cdot l}, \ldots, \theta_{n_{\mathrm{MCMC}}\cdot l}\right\}. \tag{3.10}$$

Follows the tempered posterior distribution, due to Bayes' theorem and the resulting proportionality

$$\tilde{\pi}_\lambda\left(\theta|\mathcal{D}\right) \propto \exp\left(-\lambda\mathcal{L}_{\mathrm{NLL}}\left(\theta\right)\right)\pi\left(\theta\right). \tag{3.11}$$

4. Toy setup

4.1. Gamma function ring

Similar to previous studies on data amplification [5], we employ the CNF on a low-dimensional ring distribution. Generative Architectures often struggle with changes in the topology between latent space, typically Normal distributed, and data space [33]. The ring structure reflects this 'topological worst case'. A generalization of the results from a similar, topologically complicated, but low-dimensional toy to high-dimensional, simulated, and topologically less problematic calorimeter images was performed in [6]. In

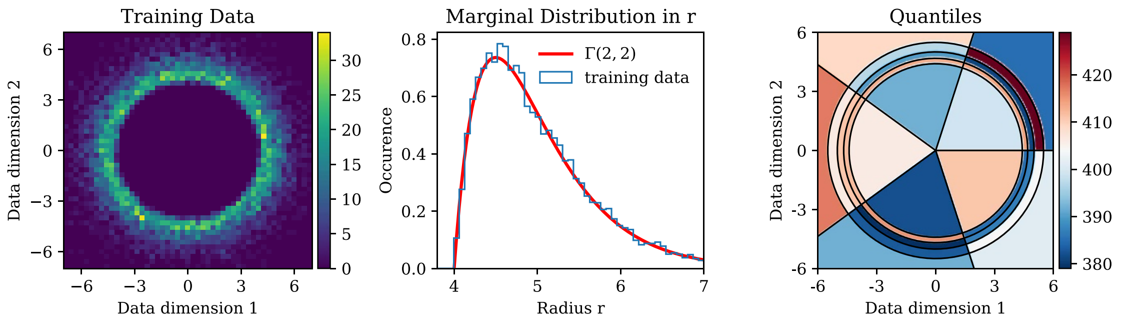


Figure 1. Left: Histogram of one training data set (10 000 points). The data follows a ring structure with a sharp edge at $r = 4$ and a long tail to higher radii. Mid: Marginal distribution of the training data in radial direction. Right: 5 × 5 quantiles generated from a data set of 10M points and filled with the training data. The quantiles are constructed with equal probability of the truth data to fall into every quantiles.

this paper, we focus on the calibration of generative uncertainties and draw a connection to data amplification. We thus limit the study to two dimensions for illustrative purposes and to reduce computational costs. Nevertheless, the calibration can be executed analogously for higher dimensional distributions. We generate samples from a ring distribution with an unsteady edge at a radius of $r = 4$, by sampling in spherical coordinates from

$$\phi \sim \text{uniform}\,(0, 2\pi) \quad \text{and} \quad r - 4 \sim \Gamma\,(\alpha, \beta)\,,$$

with parameters $\alpha = \beta = 2$ for the Gamma distribution. Per training, we use an independent sample of $N = 10\,000$ points. Before passing the data to the CNF, we transform into Cartesian coordinates to obtain the ring shape shown in figure 1. This construction allows us to estimate the behaviour of the uncertainties at distribution edges and simultaneously prevents divergences of the probability distribution in $(x, y) = (0, 0)$.

4.2. Hyperparameter choices

Due to the low dimensionality of the toy example, we do not need to employ complicated architectures to obtain a good approximation of the vector-field $\tilde{v}_t(\cdot, \theta)$. Based on a small grid search, a Multi-Layer Perceptron with 3 layers of 32 nodes and ELU activation is sufficient to reproduce the training data well. Each of the 3 layers takes the time variable t as an additional input. The neural network part of the CNF thus totals a mere 2498 parameters.

When parameterizing the weight posterior approximation $\tilde{\pi}(\theta)$ as an uncorrelated Normal distribution, as is standard in VIB [17], the number of parameters consequently doubles. For VIB we train using the `Adam` optimizer [32] at a learning rate of 10^{-3} for up to 250k epochs of 10 batches of 1000 datapoints each. To prevent overfitting, we evaluate the model at the earliest epoch after convergence of the KL-loss term. This point depends on the choice of k and varies between 75k for $k = 50$ and 250k for $k = 1$. We do 5 runs each for multiple values of $k \in [1, 5, 10, 50]$ to regulate the uncertainty quantification. For this range of k we have previously found sensible density estimation and optimization convergence trough performing a log-linearly spaced scan in $k \in [10^{-4}, 10^{5}])$ with only one run per parameter choice.

For the `AdamMCMC` sampling, we start the chain from a pretrained model. The model is first optimized for 2500 epochs (`Adam` with learning rate of 10^{-3}) using only the CFM-loss (3.4). For the deterministic model, this is enough to converge. We then run the sampling at a the same learning rate as the optimization with $\sigma_\Delta \approx 50$ and $\lambda = 1.0$. This choice of σ_Δ ensures high acceptance rates, while the choice of λ reflects sampling from the untempered posterior distribution, as per (3.11). We add a sample to the collection at intervals of 100 epochs, to ensure the independence of the sampled weights. To adjust the calibration, we scan the noise value at four points $\sigma \in [0.01, 0.05, 0.1, 0.5]$. This parameter span is based on a log-linearly spaced scan in $\sigma \in [10^{-4}, 10])$. Once again, we calculate 5 chains per noise parameter setting.

4.3. Quantiles

As in [5], we evaluate the generated data in histogram bins of equal probability mass. We will refer to these bins as quantiles Q_j, their count as q_j and the set of all quantiles as $\mathbf{Q} = \{Q_1, \ldots, Q_{n_Q}\}$. To construct bins with the same expected occupancy, we use spherical coordinates. In angular direction, the space can simply be divided into linearly spaced quantiles, while in radial direction we use the quantiles of a 10M generated truth dataset to gauge the boundaries of the quantiles. To guaranty even population, we always choose the same number of quantiles in both dimensions. Figure 1 illustrates the construction and occupancy for 5 × 5 quantiles in Cartesian coordinates.

For correlated data, quantiles can be constructed by iteratively dividing a truth set into sets of equal size [6]. The binning is however not relevant for the discussion of calibration and analogous arguments can be made for arbitrary histograms. The advantage of quantiles over other binning schemes is the clear definition of the number of bins without an offset by an arbitrary amount of insignificant bins in the sparsely or unpopulated areas of the data space. This allows us to show the behaviour of calibration and amplification over the number of bins in sections 5 and 6.

5. Calibration

To align the uncertainty quantification, for `AdamMCMC` we generate 10M points from the CNF for the $n_{\text{MCMC}} = 10$ parameter samples in $\mathbf{\Theta}_{\text{MCMC}}$. We obtain a set of points $\mathbf{G}^{(i)}$ per parameter sample $\theta^{(i)}$, with the corresponding count

$$g_j^{(i)} = \#\left\{x' \in Q_j \mid x' \in \mathbf{G}^{(i)}\right\}$$

in quantile Q_j. Each count corresponding to a parameter sample thus constitutes one drawing of a random variable G_j whose distribution is induced by the posterior.

Analogously, for VIB we draw a set $\mathbf{\Theta}_{\text{VIB}}$ of parameters from the posterior approximation $\tilde{\pi}(\theta)$, generate 10M samples from each and calculate the quantile counts to generate drawings of G_j. As the training cost does not depend on the number of draws for VIB, we use $n_{\text{VIB}} = 50$ samples for better accuracy.

Using the quantile values $g_j^{(i)}$, we approximate the cumulative distribution function (CDF)

$$\hat{F}_{G_j,\mathbf{\Theta}}\left(g_j\right) \approx F_{G_j}\left(g_j\right) = P\left(G_j \leqslant g_j\right), \tag{5.1}$$

from its empirical counterpart using linear interpolation. We leave the set $\mathbf{\Theta}$ general, without a subscript, for now. From the approximated CDF, we construct symmetric confidence intervals for a given confidence level c from its inversion

$$I_{j,\mathbf{\Theta}}\left(c\right) = \left[\hat{F}^{-1}_{G_j,\mathbf{\Theta}}\left(0.5 - \frac{c}{2}\right), \hat{F}^{-1}_{G_j,\mathbf{\Theta}}\left(0.5 + \frac{c}{2}\right)\right]. \tag{5.2}$$

The chosen confidence level c corresponds to the expected or *nominal coverage*.

To evaluate the observed coverage, we draw 5 different training sets from the Gamma ring distribution and calculate a VIB- and `AdamMCMC`-CNF ensemble each

$$\mathbf{\Theta}^s_{\text{MCMC}}\text{and } \mathbf{\Theta}^s_{\text{VIB}}\text{for } s \in \{1,\ldots,5\}.$$

For every model, we construct a confidence interval and evaluate the number of intervals containing the expected count of the truth distribution, i.e. $1/n_Q$. The ratio of models with an interval containing the truth value over the total number of models gives the *empirical coverage* per bin

$$\hat{c}_j = \frac{\#\left\{1/n_Q \in I_{j,\mathbf{\Theta}^s}\left(c\right) \mid s \in \{1,\ldots,5\}\right\}}{5}, \tag{5.3}$$

where we again keep the subscript on the set of parameters unspecified. For one quantile this coverage estimate is very coarse as it can only take on one of six values. Since we want to check the agreement of nominal and empirical coverage for multiple nominal coverage values, we report the mean empirical coverage

$$\bar{c} = \left\langle \hat{c}_j \right\rangle_{j \in \{1,\ldots,n_Q\}} \tag{5.4}$$

over all quantiles. The range of possible mean values is big enough to compare to a fine spacing in nominal coverage.

This also allows us to judge the agreement of nominal and empirical coverage in the full data space in a single figure. However, it also introduces the possibility for over- and underconfident areas to cancel each other out. This issue will be treated in more detial in sections 5.1 and 5.2.

Figure 2 shows the mean empirical coverage over all quantiles for 50 values of the nominal coverage linearly spaced between 0 and 1 and over three different numbers of quantiles. For a well calibrated uncertainty estimation, the empirical estimate closely follows the nominal coverage and the resulting curve is close to the diagonal of the plot. For figure 2 we can see that high noise levels in the MCMC chain lead to overestimated errors and a prediction that is underconfident on average. Inversely, low noise levels lead to overconfident predictions. From our chosen grid, $\sigma = 0.1$ shows the best agreement.

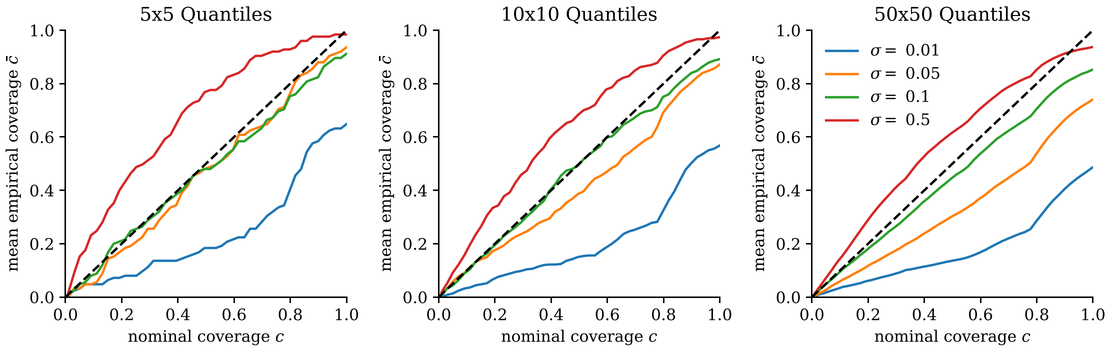


Figure 2. Mean empirical coverage for confidence intervals calculated from 10 samples of the Bayesian weight posterior drawn with `AdamMCMC` using 4 different hyperparameter settings. Higher σ will generally result in larger uncertainties. The empirical coverage is calculated from 5 independent runs and averaged over all quantiles. The panels show a clear dependence of the calibration on the number of quantiles increasing from left to right.

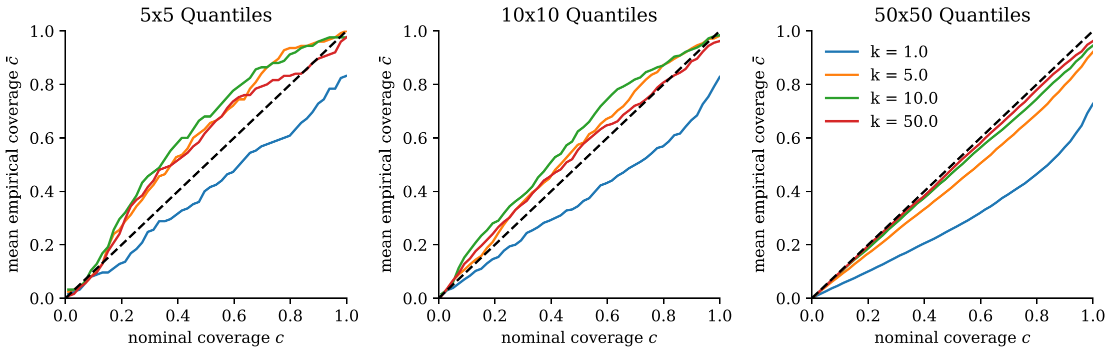


Figure 3. Mean empirical coverage for confidence intervals calculated from 50 drawings from the VIB approximation of the Bayesian weight posterior with 4 different hyperparameter settings. Larger k increases the dependence of the fit on the prior. The empirical coverage is calculated from 5 independent runs and averaged over all quantiles. The panels again show a clear dependence of the calibration on the number of quantiles increasing from left to right.

It further becomes apparent that the calibration depends on the number of quantiles. For lower numbers of quantiles, the fluctuations in the generated distribution average out and both the mean prediction and error estimation are more precise, while for higher numbers of quantiles good calibration becomes challenging while limited to 10 posterior samples.

For VIB in figure 3, where we evaluate 50 posterior samples, calibration seems to improve for high n_Q. While at lower numbers only a very small prior trade-off k leads to overconfident intervals and larger values result in underconfident predictions, at higher numbers of quantiles previously underconfident predictions appear well calibrated.

5.1. Scaling with the number of quantiles

To further investigate the calibration of our Bayesian generative neural networks, we pick the seemingly best calibrated parameter settings for both methods. For `AdamMCMC` this is $\sigma = 0.1$ and for VIB $k = 10$. We generate $n_{\mathrm{MCMC}} = n_{\mathrm{VIB}} = 50$ samples from the posterior for both methods now and evaluate the scaling with the number of quantiles in more detail.

As we do not want to evaluate one calibration plot for each quantile, we reduce the diagonal calibration plots by calculating the mean (absolute) deviation between empirical and nominal coverage

$$\mathrm{MD} = \langle \bar{c} - c \rangle_{c \in [0,1]} \quad \text{and} \quad \mathrm{MAD} = \langle |\bar{c} - c| \rangle_{c \in [0,1]} , \tag{5.5}$$

where the mean empirical coverage still depends on the nominal coverage $\bar{c} = \bar{c}(c)$. The composition of the mean on the quantiles, the absolute value, and the mean on the nominal coverage allows for under- and overestimation in individual quantiles to cancel out.

To gauge this we promote the index over all quantiles j to a tuple of indices (j_r, j_ϕ). We write $\hat{c}_{(j_r, j_\phi)}$ for the empirical coverage in the j_rth radial and j_ϕth angular bin. By limiting the average over the empirical

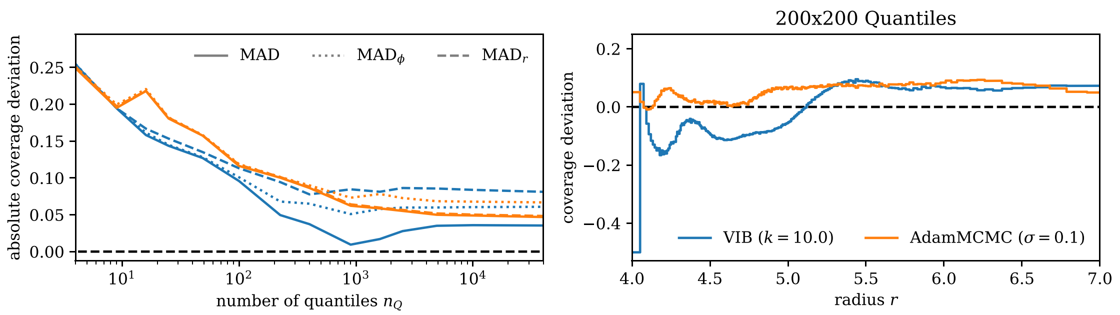


Figure 4. Left: Mean absolute deviation between the nominal and the empirical coverage (5 runs) for 50 posterior samples from both, VIB at $k = 10$ and `AdamMCMC` at $\sigma = 0.1$. The panel shows a strong dependence on the number of quantiles. From evaluating the calibration plots for all numbers of quantiles individually, we know both methods are undercertain at low numbers. With increasing n_Q, the calibration mean exhibits a strong dependence on the order of the absolute and average operations. When only the over- and underdensities along radial or angular dimension can cancel out, the mean calibration of the VIB-CNFs drastically decreases. Right: Difference between nominal and empirical coverage (mean over angular direction) for the radial direction of 200 × 200 quantiles. Values below 0 indicate overcertainty, while values above indicate undercertain predictions. While the predictions of the `AdamMCMC`-CNFs are well calibrated to slightly undercertain along the radius, the prediction of the VIB-CNFs starts out overcertain due to the oversmoothing caused by the strong prior dependence. When averaging over all radii, the VIB-CNF predictions cancel exactly and the BNN seems well calibrated.

coverage (5.4) to one of the dimensions

$$\bar{c}_{j_r} = \left\langle \hat{c}_{(j_r, j_\phi)} \right\rangle_{j_\phi} \quad \text{and} \quad \bar{c}_{j_\phi} = \left\langle \hat{c}_{(j_r, j_\phi)} \right\rangle_{j_r} \tag{5.6}$$

we construct marginal coverage distributions in the remaining directions. We can then again calculate the mean absolute deviation to the nominal coverage and suspend the average in the remaining direction until the very end

$$\text{MAD}_r = \left\langle \left\langle |\bar{c}_{j_r} - c| \right\rangle_{c\in[0,1]} \right\rangle_{j_r} \quad \text{and} \quad \text{MAD}_\phi = \left\langle \left\langle |\bar{c}_{j_\phi} - c| \right\rangle_{c\in[0,1]} \right\rangle_{j_\phi}$$

By switching the order, only quantile counts in direction of the first mean (5.6) can even out. When starting out with the average in the angular dimension, we end up with an estimate where the fluctuations in the radial direction are preserved in the absolute mean and vice versa.

The left panel of figure 4 shows the dependence of the three different coverage deviation averages on the number of quantiles. To keep the effect of statistic fluctuations per bin to a minimum, we generate sets of $1000 \cdot n_Q$ artificial points with the CNFs and evaluate between 2 × 2 and 200 × 200 quantiles.

We find a clear dependence of the coverage means on the number of quantiles. At low numbers, the mean prediction averages over large areas of the data space increasing the quality of the mean prediction. The uncertainty estimation is thus underconfident for both methods and the mean absolute deviations are high and do not depend on the order of averaging. For low numbers of quantiles, calibration is much better,i.e. the absolute deviation is closer to 0. While for `AdamMCMC` we cannot see big changes depending on the averaging order. This indicates a calibration independent of the dimension. At the same time, we find large discrepancies for VIB. This variation can be understood from the marginal calibrations.

5.2. Calibration at sharp features in radial direction

The right panel of figure 4 displays the the marginal empirical coverage in radial direction $\bar{c}_r$ for 200 × 200 quantiles and both BNN methods. We can see a distinct difference in the uncertainty quantification.

While the VIB prediction seems very well calibrated in total, in the radial direction, the VIB underestimates its bias for the steeply rising part of the data distribution between $r \in [4.0, 5.0]$. For the same interval, the MCMC prediction is well calibrated and less underconfident than for higher radii. For $r > 5$ both models slightly overestimate the uncertainty and show very similar calibration.

In terms of absolute uncertainty, both methods actually predict very similar results. However, the mean prediction of the VIB-CNF is strongly biased by the prior KL-loss term, resulting in large underpopulation of the generated density due to oversmoothing for $r < 4.5$ and a corresponding overpopulation in $r \in [4.5, 5.0]$. We have tested the predictions for $k = 50$ and the behaviour is magnified at higher values of k. For lower values ($k = 5$), the oversmoothing is reduced to the area below $r = 4.3$ at the cost of an overestimated tail. The `AdamMCMC`-CNF shows signs of oversmoothing as well, but only very close to the start of the radial distribution.

6. Bayesiamplification

Based on the previous discussion of both the total and marginal calibration, we can confidently say that our AdamMCMC-CNF is well calibrated, albeit slightly underconfident for some areas of data space and small numbers of bins. It is, however, important to note that truth information was needed to evaluate the calibration of the BNN. In a practical application, this would require either a validation region or a large hold-out set, the latter of which would partially defeat the purpose of data amplification in fast detector simulation. However, for applications with validations regions, such as generative anomaly detection [34, 35], precision improvements through data amplification can be realized.

With a well calibrated BNN, we can try and develop a measure of the statistical power of the generated set from the uncertainties. We do so by relating the uncertainty to the statistics of an uncorrelated set of points **T** from the truth distribution. For n_{bins} arbitrary bins, we expect the count in the jth bin to be approximately Poisson distributed with mean and variance t_j. For the same bin, the set of $n_{\mathrm{MCMC}} = 50$ AdamMCMC-CNF posterior samples gives a mean prediction

$$\bar{g}_j = \left\langle g_j^{(i)} \right\rangle_{i\in\{1,\ldots,n_{\mathrm{MCMC}}\}} \text{ and variance } \sigma^2_{\bar{g}_j} = \left\langle \left(g_j^{(i)} - \bar{g}_j \right)^2 \right\rangle_{i\in\{1,\ldots,n_{\mathrm{MCMC}}\}} .$$

We will now use the posterior mean and variance to construct an estimator $\hat{t}_j$ of the Poisson equivalent to the per-bin predictions. Using only the mean $\hat{t}_j := \bar{g}_j$, the equivalent will simply be the generated statistics. Thereby, we would disregard the correlations in the generated data through limited training data completely.

By instead equating the variance of the BNN to that of the equivalent uncorrelated set $\hat{t}_j := \sigma^2_{\bar{g}_j}$, we would introduce an unwanted dependence on the uncertainty prediction. Overestimated uncertainties would lead to an overestimation of the statistical power.

As we do not want to overestimate the generative performance, we aim to have undercertain predictions to lead to an underestimation of the uncorrelated equivalent. Such a behaviour can be constructed using the coefficient of variation

$$\frac{1}{\sqrt{\hat{t}_j}} := \frac{\sigma_{\bar{g}_j}}{\bar{g}_j} \quad \Longleftrightarrow \quad \hat{t}_j = \frac{\bar{g}_j^2}{\sigma^2_{\bar{g}_j}} . \tag{6.1}$$

The equivalent uncorrelated statistics now decreases for overestimated $\sigma_{\bar{g}_j}$. Both the predictions from the absolute and from the relative error give the similar estimates for well calibrated errors in our tests.

We calculate the equivalent truth set size for both the VIB-CNF and AdamMCMC-CNF and the quantiles from section 5. In figure 5, we report the *amplification* as the ratio of the sum over all bin estimates and the training statistics

$$\sum_{j=1}^{n_{\mathrm{bins}}} \hat{t}_j / N$$

in the left panel, as well as the mean estimate over all bins on the right.

Since the amplification contains the sum over all quantiles of our setup and $\hat{t}_j$ depends on the fluctuations of the individual predictions $g_j^{(i)}$ around the posterior mean prediction only, we expect it to scale linearly in the number of bins. This seems in good agreement with the figure. For large numbers of quantiles, where the BNNs are best calibrated, the average amplification per bin converges to a constant value. Fitting a exponential linear function $\exp(a + b \cdot \log(x)) = a' \cdot x^b$ to these last 8 points of figure 5 using least squares, we indeed find no significant deviations from $b = 1$. We estimate $a' = (4.3 \pm 2.9) \cdot 10^{-3}$ and $b = 0.99 \pm 0.06$ for the VIB-CNF and $a' = 0.012 \pm 0.004$ and $b = 0.99 \pm 0.04$ for the AdamMCMC-CNF. At lower numbers, the deviations of the model output for different parameters in the Bayesian set integrate over large intervals of the data space leading to smaller error estimates and increased amplification per bin.

This behaviour is consistent with the previous studies [5, 6] and the observation that one can not improve the estimation of low moments of the distribution, like the distribution mean, by oversampling with a generative neural network. From figure 5, we can also estimate the minimum amount of bins to leverage the amplification. For the MCMC sample, evaluating at 100 bins is expected to yield an improved density estimation over using only the training set. This number could decrease for a less underconfident model. For highly granular binning, we find amplification estimates of more than a factor 100.

For smaller training statistics, we expect a higher initial amplification at low numbers of bins, while the corresponding larger uncertainty estimate will result in a flatter slope. The number of quantiles where an amplification larger than 1 first occurs will be smaller in such a case. Higher training statistics on the other

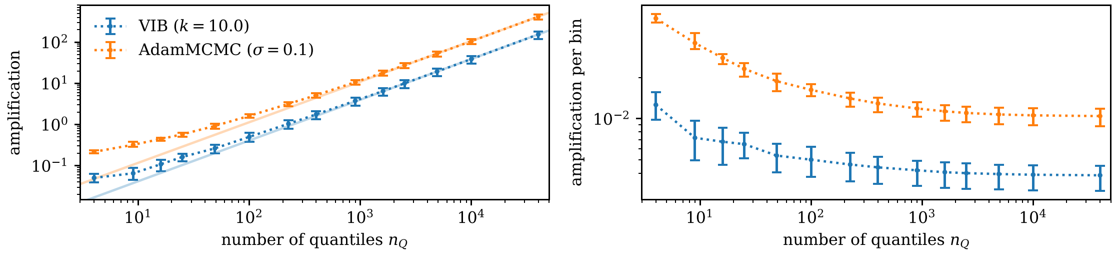


Figure 5. Left: Amplification estimate generated by equating the error prediction per bin for both BNNs to the Poisson error of an independent data set. Higher uncertainty results in lower amplification. Errorbars are calculated from the ensemble of 5 runs done per BNN method. Again, we use 50 samples of the weight posterior (approximation) for both methods. The faint solid lines show the result of exponential linear fit (least squares) to the last 8 points. Right: The mean estimate over all quantiles converges to a constant value at high numbers of quantiles resulting in linear scaling of the amplification with the number of quantiles.

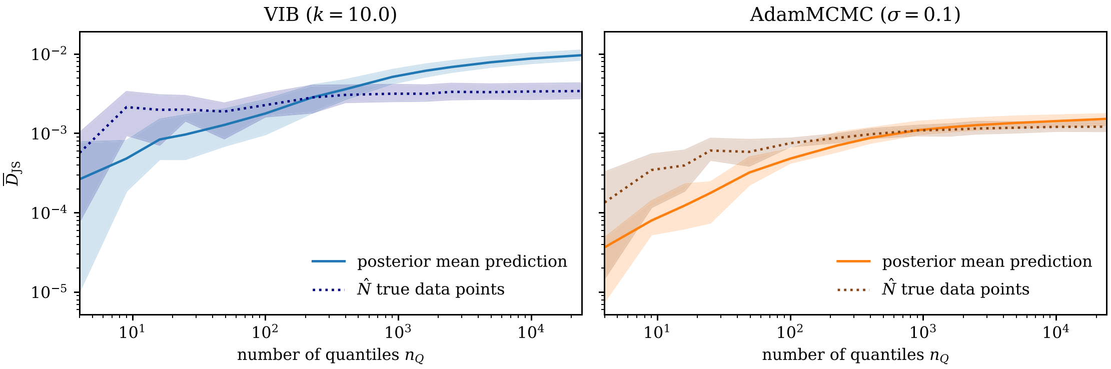


Figure 6. Jensen–Shannon divergence between the mean prediction by the Bayesian CNFs and the true data distribution (solid line) and between a uncorrelated data set of the size predicted by the BNN error estimate and the true data (dashed line). Both divergences align only for well calibrated uncertainties (AdamMCMC at high numbers of quantiles) and indicate over- and underestimated errors otherwise. Uncertainties are reported from the ensemble of 5 independent runs done for estimation of the empirical coverage.

hand will lead to a steeper slope and a later trade-off point. The results of [5] imply, that amplification effects are stronger in larger data spaces, due to the reduced density of the training data.

Similar calculations can be done for arbitrary binnings to justify the use of generative machine learning in a specific analysis. The evaluation of the Bayesian uncertainty prediction however requires the calculation of multiple sets of fast-simulation data points. This reduces the speed benefits of applying generative machine learning over more classical tools like MCMC simulation or inference.

6.1. Checking amplification with Jensen–Shannon (JS) divergence

To test how well the sum over all bin estimates

$$\hat{N} = \sum_{j=1}^{n_{\mathrm{bins}}} \hat{t}_j$$

actually gauges the size of an equivalent independent data set, we calculate the JS divergence

$$\bar{D}_{\mathrm{JS}}(p,q) = \frac{1}{2}\sum_{j=1}^{n_{\mathrm{bins}}}\left(p_j \log\frac{p_j}{\frac{1}{2}(p_j+q_j)} + q_j \log\frac{q_j}{\frac{1}{2}(p_j+q_j)}\right), \tag{6.2}$$

between the histogram estimation of the density in our quantiles and the known data distribution. The JS divergence is bounded by 0 and $\log 2$, with smaller values indicating similarity between the compared distributions.

In our toy setup, the bins are constructed as quantiles. We evaluate the JS divergence for $p_j = \frac{\bar{g}_j}{1000 \cdot n_Q}$, the mean prediction of the BNN relative to total number generated, and $q_j = 1/n_Q$ the probability per quantile when sampling from the data distribution. In figure 6, we compare it to the JS divergence for $p_j = t_j/\hat{N}$, the relative population of the quantiles for a set of $\hat{N}$ points drawn from the truth distribution, and the true quantile count $q_j = 1/n_Q$ for a large range of n_Q.

Where the BNN is well calibrated, i.e. for AdamMCMC and $n_Q > 10^3$, the quality of the mean prediction lines up with the results of the uncorrelated set drawn to the size of the BNN errors. The Bayesian coefficient of variation correctly predicts the equivalent uncorrelated statistics. At lower numbers of quantiles, the error is overestimated. Consequently, the statistical equivalent is underestimated. This can also be observed for the VIB-CNF. However, for large number of quantiles where the uncertainty at low radii is underestimated, see section 5.2, the performance of the mean prediction is worse than anticipated by the BNN. Good calibration on the full data space therefore is important for a reliable prediction of $\hat{N}$.

7. Conclusion

In the previous chapters, we present a novel evaluation of the uncertainty provided by a Bayesian generative neural network in a histogram. To this end, we propose constructing confidence intervals per histogram bin and compare the nominal coverage of the constructed interval to the empirical coverage obtained from a small ensemble of BNNs.

We observe a strong dependence of the calibration on the parameters of both a VIB-CNF and an MCMC-sampled CNF. Furthermore, we find a strong tendency to oversmooth with strong priors leading to underestimation of the data density and corresponding error at the non-differentiable inner edge of our toy distribution. While present in both approaches, this behavior was predominantly displayed by the VIB-CNFs.

We further use the calibrated errors to estimate the statistical power of the generated data in terms of the size of an equivalent independently sampled data set. This estimate correctly quantifies the performance of the BNNs mean prediction when the errors are well calibrated and assigns a concrete number to the data amplification in dependence of the employed binning. For a correct amplification estimate, it is crucial that the errors are well calibrated in the full data space.

Similar calibration checks can be applied wherever a generative neural network is used for inference or generation with a sufficient validation set or for interpolation into hold-out regions of the data.

Data availability statement

No new data were created or analysed in this study.

Acknowledgments

SB is supported by the Helmholtz Information and Data Science Schools via DASHH (Data Science in Hamburg—HELMHOLTZ Graduate School for the Structure of Matter) with the Grant HIDSS-0002. SB and GK acknowledge support by the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy—EXC 2121 Quantum Universe—390833306 and via the KISS consortium (05D23GU4) funded by the German Federal Ministry of Education and Research BMBF in the ErUM-Data action plan. This research was supported in part through the Maxwell computational resources operated at Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany. SD is supported by the U.S. Department of Energy (DOE), Office of Science under Contract DE-AC02-05CH11231.

Code

https://github.com/sbieringer/Bayesiamplify provides the code for simulating the toy example and conducting this analysis.

ORCID iDs

S Bieringer https://orcid.org/0000-0002-2615-5639
S Diefenbacher https://orcid.org/0000-0003-4308-6804
G Kasieczka https://orcid.org/0000-0003-3457-2755
M Trabs https://orcid.org/0000-0001-8104-4467

References

[1] Albrecht J *et al* HEP Software Foundation 2019 A roadmap for HEP software and computing R&D for the 2020s *Comput. Softw. Big Sci.* **3** 7
[2] Boehnlein A *et al* 2022 HL-LHC software and computing review panel (*2nd Report. Technical Report*) (CERN, Geneva)
[3] Butter A *et al* 2023 Machine learning and LHC event generation *SciPost Phys.* **14** 079
[4] Hashemi H and Krause C 2024 Deep generative models for detector signature simulation: an analytical taxonomy *Rev. Phys.* **12** 100092

[5] Butter A, Diefenbacher S, Kasieczka G, Nachman B and Plehn T 2021 GANplifying event samples *SciPost Phys.* **10** 139

[6] Bieringer S, Butter A, Diefenbacher S, Eren E, Gaede F, Hundhausen D, Kasieczka G, Nachman B, Plehn T and Trabs M 2022 Calomplification—the power of generative calorimeter models *J. Instrum.* **17** 09028

[7] Butter A, Huetsch N, Palacios Schweitzer S, Plehn T, Sorrenson P and Spinner J 2023 Jet diffusion versus JetGPT–Modern networks for the LHC (arXiv:2305.10475)

[8] Chen T Y, Dey B, Ghosh A, Kagan M, Nord B and Ramachandra N 2022 Interpretable uncertainty quantification in AI for HEP *Snowmass 2021* (https://doi.org/10.2172/1886020)

[9] Kronheim B S, Kuchera M P, Prosper H B and Karbo A 2021 Bayesian neural networks for fast SUSY predictions *Phys. Lett.* B **813** 136041

[10] Bollweg S, Haußmann M, Kasieczka G, Luchmann M, Plehn T and Thompson J 2020 Deep-learning jets with uncertainties and more *SciPost Phys.* **8** 006

[11] Araz J Y and Spannowsky M 2021 Combine and conquer: event reconstruction with Bayesian ensemble neural networks *J. High Energy Phys.* JHEP04(2021)296

[12] Bellagente M, Haussmann M, Luchmann M and Plehn T 2022 Understanding event-generation networks via uncertainties *SciPost Phys.* **13** 003

[13] Das R, Favaro L, Heimel T, Krause C, Plehn T and Shih D 2024 How to understand limitations of generative networks *SciPost Phys.* **16** 031

[14] Jospin L V, Laga H, Boussaïd F, Buntine W L and Bennamoun M 2022 Hands-on bayesian neural networks—A tutorial for deep learning users *IEEE Comput. Intell. Mag.* **17** 29–48

[15] Mena J, Pujol O and Vitrià J 2022 A survey on uncertainty estimation in deep learning classification systems from a bayesian perspective *ACM Comput. Surv.* **54** 1–35

[16] Goan E and Fookes C 2020 Bayesian neural networks: an introduction and survey *Case studies in applied Bayesian data science* ed K Mengersen, P Pudlo and C Robert (*Lecture notes in mathematics* vol 2259) (Springer) pp 45–87

[17] Blundell C, Cornebise J, Kavukcuoglu K and Wierstra D 2015 Weight uncertainty in neural networks *Int. Conf. on Machine Learning* (PMLR) pp 1613–22

[18] Izmailov P, Vikram S, Hoffman M D and Wilson A G G 2021 What are bayesian neural network posteriors really like? *Int. Conf. on Machine Learning* (PMLR) pp 4629–40

[19] Chen T, Fox E and Guestrin C 2014 Stochastic gradient hamiltonian monte carlo *Int. Conf. on Machine Learning* (PMLR) pp 1683–91

[20] Bieringer S, Kasieczka G, Kieseler J and Trabs M 2024 Classifier surrogates: sharing AI-based searches with the world *Eur. Phys. J.* C **84** 972

[21] Bieringer S, Kasieczka G, Steffen M F and Trabs M 2023 AdamMCMC: combining Metropolis adjusted Langevin with momentum-based optimization (arXiv:2312.14027)

[22] Rezende D and Mohamed S 2015 Variational inference with normalizing flows *Int. Conf. on Machine Learning* (PMLR) pp 1530–8

[23] Chen R T Q, Rubanova Y, Bettencourt J and Duvenaud D K 2018 Neural ordinary differential equations *Advances in Neural Information Processing Systems* **31** pp 6572–83

[24] Mikuni V, Nachman B and Pettee M 2023 Fast point cloud generation with diffusion models in high energy physics *Phys. Rev.* D **108** 036025

[25] Mikuni V and Nachman B 2024 CaloScore v2: single-shot calorimeter shower simulation with diffusion models *J. Instrum.* **19** 02001

[26] Leigh M, Sengupta D, Andrew Raine J, Quétant G and Golling T 2024 Faster diffusion model with improved quality for particle cloud generation *Phys. Rev.* D **109** 012010

[27] Buhmann E, Gaede F, Kasieczka G, Korol A, Korcari W, Krüger K and McKeown P 2024 CaloClouds II: ultra-fast geometry-independent highly-granular calorimeter simulation *J. Instrum.* **19** 04020

[28] Buhmann E, Ewen C, Faroughy D A, Golling T, Kasieczka G, Leigh M, Quétant G, Andrew Raine J, Sengupta D and Shih D 2023 EPiC-ly fast particle cloud generation with flow-matching and diffusion (arXiv:2310.00049)

[29] Kobylianskii D, Soybelman N, Dreyer E and Gross E 2024 Graph-based diffusion model for fast shower generation in calorimeters with irregular geometry *Phys. Rev. D* **110** 072003

[30] Lipman Y, Chen R T Q, Ben-Hamu H, Nickel M and Le. M 2023 Flow matching for generative modeling *The* 11th *Int. Conf. on Learning Representations*

[31] Krishnan R, Subedar M and Tickoo O 2020 Specifying weight priors in bayesian deep neural networks with empirical bayes *Proc. of the AAAI Conf. on Artificial Intelligence* (AAAI Press) pp 4477–84

[32] Kingma D P and Ba. J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)

[33] Winterhalder R, Bellagente M and Nachman B 2021 Latent space refinement for deep generative models (arXiv:2106.00792)

[34] Hallin A, Isaacson J, Kasieczka G, Krause C, Nachman B, Quadfasel T, Schlaffer M, Shih D and Sommerhalder M 2022 Classifying anomalies through outer density estimation *Phys. Rev.* D **106** 055006

[35] Golling T, Kasieczka G, Krause C, Mastandrea R, Nachman B, Andrew Raine J, Sengupta D, Shih D and Sommerhalder M 2024 The interplay of machine learning-based resonant anomaly detection methods *Eur. Phys. J.* C **84** 241

Eidesstattliche Versicherung / Declaration on oath

Hiermit versichere ich an Eides statt, die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt zu haben.

Sofern im Zuge der Erstellung der vorliegenden Dissertationsschrift generative Künstliche Intelligenz (gKI) basierte elektronische Hilfsmittel verwendet wurden, versichere ich, dass meine eigene Leistung im Vordergrund stand und dass eine vollständige Dokumentation aller verwendeten Hilfsmittel gemäß der Guten wissenschaftlichen Praxis vorliegt. Ich trage die Verantwortung für eventuell durch die gKI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate.

Illingen, den 14.01.2025

Ort, Datum

S. Bieringer

Unterschrift