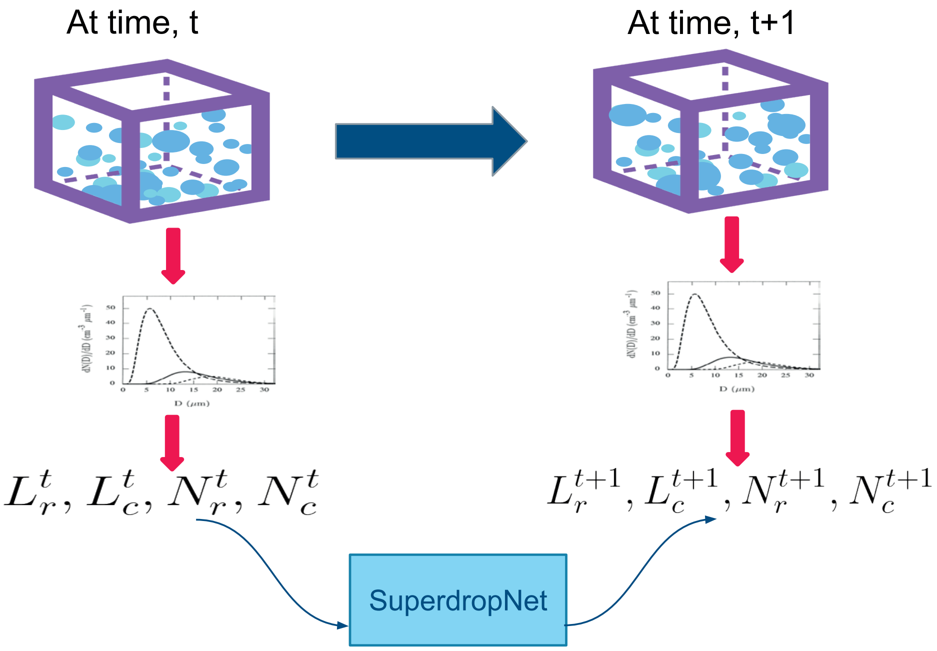
MAX-PLANCK-INSTITUT
FÜR METEOROLOGIE

Parameterizing Lagrangian Cloud Microphysics using Machine Learning


At time, t
At time, t+1
$L_r^t, L_c^t, N_r^t, N_c^t$
$L_r^{t+1}, L_c^{t+1}, N_r^{t+1}, N_c^{t+1}$
SuperdropNet
D (μm)

Shivani Sharma

Hamburg 2025

Berichte zur Erdsystemforschung
Reports on Earth System Science

289
2025

Hinweis

Die Berichte zur Erdsystemforschung werden vom Max-Planck-Institut für Meteorologie in Hamburg in unregelmäßiger Abfolge herausgegeben.

Sie enthalten wissenschaftliche und technische Beiträge, inklusive Dissertationen.

Die Beiträge geben nicht notwendigerweise die Auffassung des Instituts wieder.

Die "Berichte zur Erdsystemforschung" führen die vorherigen Reihen "Reports" und "Examensarbeiten" weiter.

Anschrift / Address

Max-Planck-Institut für Meteorologie
Bundesstrasse 53
20146 Hamburg
Deutschland

Tel./Phone: +49 (0)40 4 11 73 - 0
Fax: +49 (0)40 4 11 73 - 298

name.surname@mpimet.mpg.de
www.mpimet.mpg.de

Notice

The Reports on Earth System Science are published by the Max Planck Institute for Meteorology in Hamburg. They appear in irregular intervals.

They contain scientific and technical contributions, including PhD theses.

The Reports do not necessarily reflect the opinion of the Institute.

The "Reports on Earth System Science" continue the former "Reports" and "Examensarbeiten" of the Max Planck Institute.

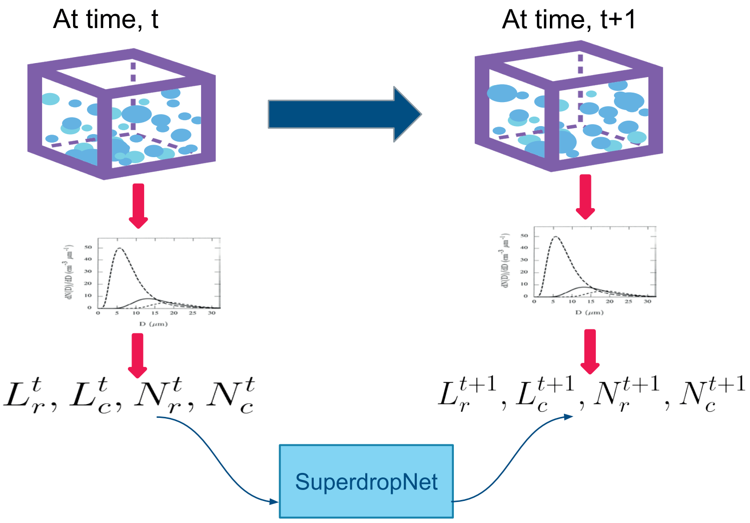
Layout

Bettina Diallo and Norbert P. Noreiks
Communication

Copyright

Photos below: ©MPI-M
Photos on the back from left to right:
Christian Klepp, Jochem Marotzke,
Christian Klepp, Clotilde Dubois,
Christian Klepp, Katsumasa Tanaka

Parameterizing Lagrangian Cloud Microphysics using Machine Learning


At time, t
At time, t+1
$L_r^t, L_c^t, N_r^t, N_c^t$
$L_r^{t+1}, L_c^{t+1}, N_r^{t+1}, N_c^{t+1}$
SuperdropNet
D (μm)

Shivani Sharma

Hamburg 2025

Shivani Sharma
aus Delhi, Indien

Helmholtz-Zentrum Hereon
Max-Planck-Straße 1
21502 Geesthacht

Max-Planck-Institut für Meteorologie
The International Max Planck Research School on Earth System Modelling
(IMPRS-ESM)
Bundesstrasse 53
20146 Hamburg

Tag der Disputation: 25. Oktober 2024

Folgende Gutachter empfehlen die Annahme der Dissertation:
Prof. Dr. Nedjeljka Žagar
Dr. David S. Greenberg

Vorsitzender des Promotionsausschusses:
Prof. Dr. Hermann Held

Dekan der MIN-Fakultät:
Prof. Dr.-Ing. Norbert Ritter

Berichte zur Erdsystemforschung / Max-Planck-Institut für Meteorologie 289
Reports on Earth System Science / Max Planck Institute for Meteorology 2025

ISSN 1614-1199 - doi: 10.17617/2. 3639728

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede.

https://bitbucket.org/amiede/classicthesis/

ABSTRACT

Parameterizations in weather and climate models are essential for representing sub-grid scale processes that cannot be explicitly resolved due to computational constraints. However, they introduce significant uncertainties in the prediction of key atmospheric variables such as precipitation, cloud cover, and the estimation of the radiative balance. As kilometer-scale models eliminate the need for many parameterization schemes, micro-scale phenomena such as cloud microphysical processes emerge as important sources of model error. Cloud microphysical processes are represented via bulk schemes, which, while computationally efficient, rely on simplified assumptions that can lead to substantial errors in forecasts.

This thesis presents the development of machine learning (ML) emulators aimed at improving the accuracy of these parameterizations by leveraging Lagrangian models, specifically the superdroplet method, which provides a more physically consistent representation of cloud microphysical processes. I introduce SuperdropNet, an ML-based emulator trained on superdroplet simulations, designed to predict the evolution of bulk moments while bypassing the assumptions of the bulk schemes. I introduce new training techniques and employ methods such as autoregressive training to develop a physics-informed ML emulator. SuperdropNet demonstrates a stable performance across a wide variety of initial conditions and outperforms previously available ML emulators for droplet collisions.

The emulator was also evaluated in an online scenario by coupling it with the ICOsahedral Nonhydrostatic (ICON) model. Coupling ML emulators to an atmospheric model is usually a labourious task that impedes the process of developing an effective emulator. I solved the technically challenging task of coupling a Python-based ML model to the FORTRAN codebase of ICON. I tested multiple coupling mechanisms for flexibility, ease of use, and speed, and found a C-based interface to be the most suitable. Encouragingly, when coupled to a warm bubble test scenario, SuperdropNet demonstrated long-term stability and predicted physically plausible quantities.

This research underscores the potential of ML emulators to bridge the gap between the computational efficiency of bulk moment schemes and the physical accuracy of Lagrangian models. Such an approach could be extended to other ill-represented cloud microphysical processes in atmospheric modeling, such as sedimentation and cloud-aerosol interactions, ultimately leading to more reliable weather and climate predictions.

ZUSAMMENFASSUNG

Parametrisierungen in Wetter- und Klimamodellen sind unerlässlich, um Prozesse auf Subgitter-Skalen darzustellen, die aufgrund rechnerischer Einschränkungen nicht explizit aufgelöst werden können. Sie führen jedoch zu erheblichen Unsicherheiten bei der Vorhersage wichtiger atmosphärischer Variablen wie Niederschlag, Wolkenbedeckung und der Abschätzung der Strahlungsbilanz. Da Kilometer-Skalen-Modelle den Bedarf an vielen Parametrisierungsschemata eliminieren, treten mikroskalige Phänomene wie Wolkenmikrophysikprozesse als wichtige Quellen von Modellfehlern hervor. Wolkenmikrophysikalische Prozesse werden über Bulk-Schemata dargestellt, die zwar recheneffizient sind, aber auf vereinfachten Annahmen beruhen, die zu erheblichen Fehlern in Vorhersagen führen können.

Diese Dissertation stellt die Entwicklung von Machine-Learning (ML)-Emulatoren vor, die darauf abzielen, die Genauigkeit dieser Parametrisierungen zu verbessern, indem Lagrangesche Modelle, insbesondere die Supertröpfchen-Methode, genutzt werden, die eine physikalisch konsistentere Darstellung von Wolkenmikrophysikprozessen bietet. Ich stelle SuperdropNet vor, einen ML-basierten Emulator, der auf Supertröpfchen-Simulationen trainiert wurde und entwickelt wurde, um die Entwicklung von Massenmomenten vorherzusagen, während die Annahmen der Bulk-Schemata umgangen werden. Ich führe neue Trainingstechniken ein und wende Methoden wie autoregressives Training an, um einen physikalisch fundierten ML-Emulator zu entwickeln. SuperdropNet zeigt eine stabile Leistung über eine Vielzahl von Anfangsbedingungen hinweg und übertrifft bisher verfügbare ML-Emulatoren für Tröpfchenkollisionen.

Der Emulator wurde auch in einem Online-Szenario evaluiert, indem er mit dem ICOsahedral Nonhydrostatic (ICON)-Modell gekoppelt wurde. Die Kopplung von ML-Emulatoren an ein atmosphärisches Modell ist in der Regel eine mühsame Aufgabe, die den Entwicklungsprozess eines effektiven Emulators behindert. Ich habe die technisch anspruchsvolle Aufgabe gelöst, ein Python-basiertes ML-Modell mit der FORTRAN-Codebasis von ICON zu koppeln. Ich habe mehrere Kopplungsmechanismen hinsichtlich Flexibilität, Benutzerfreundlichkeit und Geschwindigkeit getestet und festgestellt, dass eine C-basierte Schnittstelle am besten geeignet ist. Erfreulicherweise zeigte SuperdropNet in einem Warmblase-Test-Szenario langfristige Stabilität und sagte physikalisch plausible Größen vorher.

Diese Forschung unterstreicht das Potenzial von ML-Emulatoren, die Lücke zwischen der rechnerischen Effizienz von Bulk-Moment-Schemata und der physikalischen Genauigkeit von Lagrangeschen Modellen zu schließen. Ein solcher Ansatz könnte auf andere unzurei-

chend dargestellte wolkenmikrophysikalische Prozesse in der atmosphärischen Modellierung, wie Sedimentation und Wolken-Aerosol-Wechselwirkungen, ausgeweitet werden und letztlich zu zuverlässigeren Wetter- und Klimavorhersagen führen.

PUBLICATIONS

The following publications are part of this thesis:

APPENDIX A

Sharma, S., & Greenberg, D. S. (2024). SuperdropNet: A stable and accurate machine learning proxy for droplet-based cloud microphysics. *under review at the Journal of Advances in Modeling Earth Systems*. https://doi.org/arXiv:2402.18354

APPENDIX B

Arnold, C., Sharma, S., Weigel, T., & Greenberg, D. S. (2024). Efficient and stable coupling of the SuperdropNet deep-learning-based cloud microphysics (v0.1.0) with the ICON climate and weather model (v2.6.5). *Geoscientific Model Development*, *17*(9), 4017–4029. https://doi.org/10.5194/gmd-17-4017-2024

CONTENTS

I THE UNIFYING ESSAY 1

1 INTRODUCTION 3

- 1.1 Parameterization in Weather and Climate models 4
- 1.2 Should we wait for high resolution models? . . . 4
- 1.3 Bridging the gap through machine learning 6
- 1.4 The research questions 8

2 MACHINE LEARNING AND EARTH SYSTEM SCIENCE 11

- 2.1 Terminology as used in the field 12

3 SUPERDROPNET: LEARNING DROPLET COLLISIONS 15

- 3.1 The quest to find the best match 16
- 3.2 The Superdroplet Setup 16
- 3.3 Targetting the Bulk Moments 17
- 3.4 Stochastic Simulations, deterministic outputs . . . 17
- 3.5 Developing a physics-informed machine learning emulator 18
- 3.6 Did SuperdropNet improve upon the bulk moment scheme? 19

4 COUPLING SUPERDROPNET WITH ICON 25

- 4.1 The need for online coupling 25
- 4.2 Coupling ML-based parameterizations 25
- 4.3 The warm bubble test case 27
- 4.4 Lessons from coupling 27

5 SUMMARY AND OUTLOOK 31

- 5.1 Results 31
- 5.2 Outlook 32

II APPENDIX 35

A SUPERDROPNET: A STABLE AND ACCURATE MACHINE LEARNING PROXY FOR DROPLET-BASED CLOUD MICROPHYSICS 37

- A.1 Introduction 40
- A.2 Warm rain microphysical simulations 42
- A.3 Problem statement 44
- A.4 Data Generation with Droplet Simulations 44
- A.5 Deep learning of warm rain microphysics 48
- A.6 Results 50
- A.7 Discussion 62
- A.8 Open Research 63

B EFFICIENT AND STABLE COUPLING OF THE SUPERDROPNET DEEP-LEARNING-BASED CLOUD MICROPHYSICS (V0.1.0) WITH THE ICON CLIMATE AND WEATHER MODEL (V2.6.5) 65

- B.1 Introduction 68

B.2 Methods 70
B.3 Integrating SuperdropNet in ICON 73
B.4 Results 75
B.5 Conclusions 82

BIBLIOGRAPHY 85

Part I

THE UNIFYING ESSAY

1

INTRODUCTION

It is always encouraging to find parallels in the most unexpected areas. Inge Dick became fascinated with capturing the images of the sky and realized that capturing them with different camera lenses could reveal the beautiful kaleidoscopic palette of the individual clouds present. The digital image (Fig.1.1, right) reveals the constellation of colors as pixels that are only visible in a blurred continuity through an analog camera (Fig.1.1, left).

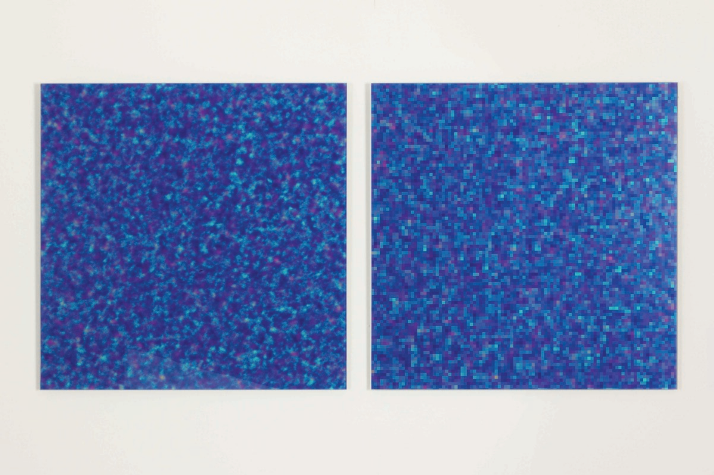


Figure 1.1: Inge Dick: Bleu du ciel, analog-digital 2001/2004/34, 2001/2004. Photo: Andreas Bestle; © VG Bild-Kunst, Bonn 2023. Left:with analog, right: with digital camera.

The scientific parallel to this photograph lies in the representation of any atmospheric model, where underneath a model's cohesive picture are the variety of small-scale phenomena which emerge only as we employ more sophisticated methods of representation. It is not always possible to get the best, most accurate representations of phenomena occurring at finer scales, due to limitation in our understanding or the computational resources available to simulate them. When equations of motion are discretized to solve for prognostic variables on a grid, processes that occur at scales below the grid spacing are represented by their approximate effect on the coarser model (Palmer, 2001). These approximations are what we refer to as *parameterizations*. Finer grids and shorter time steps improve accuracy by explicitly representing processes that would otherwise occur at sub-grid scales, but they impose huge memory and computing requirements that push against the limits of available hardware (Palmer, 2020). Hence, what we are often left with are models that merely hint at all the complex phenomena that lies beneath the surface.

1.1 PARAMETERIZATION IN WEATHER AND CLIMATE MODELS

All parameterizations are approximations and the errors introduced by them accumulate over time, causing inaccuracies in day-to-day weather prediction. For example, precipitation which is an extremely valuable variable on a regional scale, is often poorly estimated (Suzuki et al., 2011). On global scale, it is a general consensus that the parameterized representation of aerosols, clouds and cloud processes introduces uncertainties in climate projections (Boucher et al., 2013). These misrepresentations create a significant variation amongst climate models in the estimation of regional precipitation as well (Bony et al., 2013; O'Gorman & Dwyer, 2018; Wyant et al., 2012). There are often cumulative effects on other parameterized processes as well. For instance, poor representation of clouds and cloud processes, in turn, can throw off the radiative balance on a global scale (Lynn & Khain, 2007; Stephens et al., 2012).

Over the past decades, numerical weather prediction has improved significantly, corresponding to a better understanding of the physical processes, an improvement in the quality of observations as well as an increase in computational power, allowing the use of high-resolution models (Bauer et al., 2015). On a regional scale, Large-eddy simulations (LES) running at scales of a few meters, can resolve many cloud processes and represent energy fluxes with greater accuracy (Fig. 1.2). However, they are computationally expensive and hence, cannot be used on a global scale. The high resolution Cloud Resolving Models (CRM), running at approximately 1km scale can resolve many convective processes that previously required parameterization. Despite their spatial scales, LES and CRMs still require various other processes such as cloud microphysical processes, turbulent energy fluxes and radiation to be parameterized. For operational weather forecasting and long term climate projections, where models are often even coarser than LES and CRMs, it is still standard practise to parameterize sub-grid scale processes with varying levels of sophistication (Gross et al., 2018; Palmer, 2020; Stevens & Bony, 2013).

1.2 SHOULD WE WAIT FOR HIGH RESOLUTION MODELS?

As kilometer scale modelling becomes the norm, convective parameterizations can be expected to become obsolete. However, processes cloud microphysical processes represent an interesting challenge. Cloud microphysical processes refer to the various physical mechanisms that influence the formation, growth, and interaction of cloud droplets, including raindrops and ice crystals, within clouds. Due to high droplet counts even at small grid sizes and our incomplete understanding of processes that occur at the molecular level in clouds (Morrison et al., 2020), it is unlikely that in the near future, we would have access to atmospheric models at grid sizes of micrometers that would make parameterization of cloud microphysics obsolete.

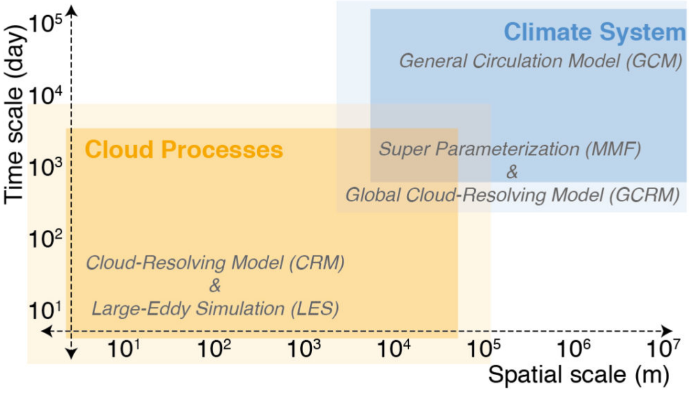

Time scale (day)
10^5
10^4
10^3
10^2
10^1
Climate System
General Circulation Model (GCM)
Cloud Processes
Super Parameterization (MMF)
&
Global Cloud-Resolving Model (GCRM)
Cloud-Resolving Model (CRM)
&
Large-Eddy Simulation (LES)
10^1
10^2
10^3
10^4
10^5
10^6
10^7
Spatial scale (m)

Figure 1.2: Variation in scales between the cloud processes and global circulation models (taken from *Clouds and Aerosols*, Fifth Assessment Report of the Intergovernmental Panel on Climate Change Boucher et al., 2013)

1.2.1 *Representation of Cloud Microphysical Processes: A Misadventure*

Since the only way to represent droplet interactions is to parameterize them, the parameterization schemes closer to the real physics offer a more realistic estimate. These are the Lagrangian models that track the evolution of cloud droplets in time and space. Lagrangian models developed as a means to study particle interactions in a more detailed manner, particularly effects of aerosols on cloud droplet growth (Flossmann, 1998; Flossmann et al., 1985). However, due to their high computational overhead, their use remains limited to research and development. One great advantage of Lagrangian schemes, lies in their ability to represent the complete distribution of droplets as they evolve in the simulation. This provides a complete picture and is as close as we can get to droplet interactions in a real atmosphere.

The reduction in computational overhead, results in modifications that simplify the representation at the cost of fidelity. One way to achieve this is to simplify the size distribution of droplets into a finite number of bins (Clark, 1974). Unsurprisingly, higher number of bins implies a better representation that comes at the cost of higher computational load, while a reduction in the number of bins, implies a coarser representation yet, faster computation.

The coarsification of the droplet size distribution (DSD) is most evident in the commonly used, bulk moment schemes (Kessler, 1969). Here, the DSD is reduced to its zeroth and first moments, and subsequently, only the evolution of these moments is simulated. The zeroth moment represents the total droplet concentration while the first moment represents the water mass. A warm rain scenario, with only clouds and rain present, is fully described by a density function $f(x)$ over droplets with mass x. This density function is used to define 4 bulk moments:

$$\begin{aligned} L_c &= \int_0^{x^*} x f(x) dx & N_c &= \int_0^{x^*} f(x) dx \\ L_r &= \int_{x^*}^{\infty} x f(x) dx & N_r &= \int_{x^*}^{\infty} f(x) dx \end{aligned} \tag{1}$$

Here $x^* = 2.6 \times 10^{-10}$kg is the mass threshold dividing cloud and rain droplets (Beheng & Doms, 1986). The zeroth moments, N_c and N_r, are the number density of cloud and rain droplets respectively. The first moments, L_c and L_r, represent the total cloud and rain water mass.

The shift from detailed Lagrangian schemes to the bulk moment schemes, imposes a few strong assumptions. One of the foremost assumptions is the belief that throughout the simulation, the DSD evolves in a unimodal fashion as for each hydrometeor, only one or two moments are calculated. Another set of assumptions are imposed by the introduction of artificial limits on the size of which droplets are to be considered cloud or rain. These assumptions serve as the basis for empirical equations that govern the evolution of the bulk moments. Even in simplified scenarios of warm rain formation, bulk moments schemes do not stand the scrutiny of their assumptions (Igel et al., 2022). Additionally, they struggle to represent more complex microphysical scenarios with multiple hydrometeors such as snow and graupel (Khain et al., 2015; Morrison et al., 2020). A concerning issue is also the inability of the bulk schemes to converge to more accurate results with the increase in the model resolution. This imposes a limitation to the representation of cloud microphysics even in high resolution models.

1.3 BRIDGING THE GAP THROUGH MACHINE LEARNING

While it remains challenging to directly implement Lagrangian schemes in operational weather prediction models, the Lagrangian schemes are often used within smaller domains to study various aspects of cloud microphysical processes. They are often used within LES models to study the effect of aerosols on droplet growth (Andrejczuk et al., 2010; Grabowski et al., 2017), including on ice particles (Sölch & Kärcher, 2010). They have also been used to study condensational growth (Sardina et al., 2017) and effects of turbulent mixing on droplet growth (Abade et al., 2018). The usage of Lagrangian models is supported by studies proving their agreement with observations (Arabas

& Shima, 2012) as well as with the analytical solutions for estimating the DSDs (Unterstrasser et al., 2017).

Perhaps, more relevant to our purpose, standalone models for Lagrangian cloud microphysics exist (Bartman et al., 2022; Brdar & Seifert, 2018; Naumann & Seifert, 2015). Since the schemes themselves are highly optimized, they can be used as a toy model to generate large amounts of data. This is a fertile ground for the data-hungry machine learning (ML) models that rely on huge amounts of data for supervised training.

1.3.1 *The Lagrangian Model: Superdroplet Method*

The stochastic Lagrangian scheme developed in Shima et al., 2009, called the superdroplet scheme, is considered one of the best matches to the DSDs obtained from the stochastic collection equation (Unterstrasser et al., 2017). It introduces the idea of a *superdroplet* which represents many individual droplets of similar size. This assumption reduces computational load of tracking each individual droplet. The scheme is also computationally efficient due to the sampling algorithm used for calculating collision efficiency. However, they are never as computationally efficient as the traditionally used bulk moment schemes, which limits their use in operational weather and climate models.

1.3.2 *Learning droplet representations*

While it is possible to learn the entire evolution of DSDs using machine learning algorithms, it would not be possible to directly test them within an operational numerical weather prediction(NWP) model. Since the representation within NWPs allows only for tracking the number concentrations and total mass, replacing this with a distribution would require the complete overhaul of the cloud microphysics routines. This is a work in progress as many research groups work towards coupling the superdroplet scheme into a complete atmospheric model. Specifically, our target atmospheric model ICON (ICOsahedral Nonhydrostatic), uses the bulk moment scheme based on Seifert and Beheng, 2006, and usually only allows only a mass moment based representation for cloud microphysics.

Instead of directing our focus towards rewriting entire routines, we focus on improving the currently used bulk moment schemes. Normally, the total mass and number concentration are calculated from an assumed DSD at the beginning of the simulation. With each timestep, these bulk moments evolve due to various processes such as collisions, condensation, nucleation and more. Using the thermodynamic variables such as temperature and pressure, as well the bulk moments themselves, the rate of these processes are determined.

These process rates are then used to update the bulk moments with each timestep. Our approach differs in that we never calculate the intermediate process rates. Instead, we learn the characteristic evolution of bulk moments based on the superdroplet simulations.

This approach has two advantages. Firstly, since we are learning from a more accurate and realistic approach, we have the advantage of estimating the changes more accurately. Secondly, we forego the calculation of intermediate process rates which are a by-product of the bulk moment scheme's assumptions and directly calculate bulk moments in the future from the bulk moments in the present.

1.4 THE RESEARCH QUESTIONS

The update to the bulk moments comprises of many different individual processes whose effects are accumulated and then used to determine the bulk moment tendencies at every timestep. In ICON, these processes are arranged in individual submodules, each submodule isolating the effect of one process. Since our ultimate goal is to couple the ML emulator to ICON, we proceed by creating an ML emulator for the collision-coalescence process.

We start by individually picking and choosing the appropriate modelling scenario in superdroplet setting, generating training data for a variety of distributions, training the ML emulator, testing its performance on modeling scenarios not encountered during training and then, coupling the ML emulator with ICON. The specific research questions that this thesis tries to address are:

- Can we improve the representation of warm rain collisions in the bulk moment scheme by learning ML-based representations from the more accurate, superdroplet simulations?
- Can we impose the relevant physical laws on the ML emulator such that it predicts accurate and plausible bulk moments?
- How challenging is the task of coupling an ML emulator to ICON? Once coupled, what changes are observed as we move from the default bulk moment scheme to an ML emulator for the superdroplet scheme?

In in Study A, I developed an ML-based parameterization for the collision-coalescence process, SuperdropNet, using superdroplet simulations. SuperdropNet is evaluated against the bulk moment scheme in its ability to match closely the evolution of bulk moments, as obtained form the superdroplet simulations. To accomplish this, I introduced physical constraints (Yuval et al., 2021), specialized objective functions and incorporated self-iteration through recurrence (Brenowitz & Bretherton, 2018) to capture the temporal structure of the problem. SuperdropNet conserves mass, does not produce infeasible outputs such as negative moments and remains stable for all sets of initial conditions used in the study.

One of the pitfalls of machine learning parameterizations lies in their poor performance when coupled to atmospheric dynamics (Brenowitz & Bretherton, 2018; Gentine et al., 2018; O'Gorman & Dwyer, 2018). These are often caused by ML emulators that violate physical laws, thereby destabilizing the simulation. Hence, to test SuperdropNet's stability in the presence of other processes, it is coupled to a simple warm bubble scenario in ICON in Study B. This required the development of an effective bridge between the two interfaces to enable coupling. We developed a C-based interface for coupling Python based SuperdropNet to FORTRAN based ICON. Performance of SuperdropNet was evaluated based on accurate evolution of the bulk moments, precipitation and heat fluxes, while using bulk moment-based ICON simulations as comparison.

I proceed with highlighting some of the applications of ML in Earth Science and introduce important terminology unique to the field in chapter 2. In chapters 3 and 4, I provide detailed explanations of the studies mentioned above. In chapter 5, I summarize the answers to all the research questions and discuss the implications of this work and the way forward.

2

MACHINE LEARNING AND EARTH SYSTEM SCIENCE

In recent years, the abundance of data in Earth Science, coupled with unprecedented improvements in machine learning methods have led to the creation of an altogether new field, i.e, application of machine learning in Earth Science. Various problems in Earth Science have benefited from data-driven solutions and here I present some of the well-researched areas:

- **Data-driven Weather Prediction**: Completely data-driven weather prediction models, such as AURORA (Bodnar et al., 2024) and AtmoRep (Lessig et al., 2023), have been developed. These models are trained on standard climate datasets such as ERA5 and are able to produce accurate predictions for a time window of 5-10 days. These datasets are often used to initialize weather and climate models but the inference time in data-driven models is significantly reduced, in comparison to running a weather or climate simulation. It is an active area of research, especially as researchers try to stabilize these models to obtain long term predictions.

- **Data Assimilation**: Corrective methods such as data assimilation methods can benefit from the statistical nature of machine learning (Bonavita and Laloyaux, 2020; Farchi et al., 2021). The idea of using observations to course correct models is extremely relevant for operational weather prediction.

- **Physical Parameterizations**: Machine learning can be used in a more process oriented way such that instead of learning the input-to-output projection for a weather prediction model, it is instead used for improving the representation of target processes.

For the purpose of improving the representation of sub-grid scale processes in Earth Science, many studies have used high-resolution simulations for training machine learning models. This approach has been applied to convection (Brenowitz & Bretherton, 2018; Gentine et al., 2018; O'Gorman & Dwyer, 2018; Rasp et al., 2018; Yuval et al., 2021), cloud cover(Grundner et al., 2022), radiative transfer (Belochitski & Krasnopolsky, 2021a; Veerman et al., 2021), gravity wave drag (Chantry et al., 2021) and gravity wave simulation (Dong et al., 2023), atmospheric chemistry (Kelp et al., 2020, 2022) and turbulence (Leufen & Schädler, 2019). Several studies have also focused on developing ML-based proxies for computationally expensive bin cloud microphysics scheme (Gettelman et al., 2021) as well as similar to our chosen problem, superdroplet scheme (Seifert & Rasp, 2020).

Generally, the ML models developed in studies learn the high-resolution tendencies of the target variables, on the assumption that a high-resolution simulation provides a more accurate picture of the evolution of the process. These trained ML models are then used within a low-resolution climate or weather simulation to mimic the behaviour of a high-resolution simulation, without the additional computational overhead. Such an approach is insufficient in the case of cloud microphysics as the modeling scenario considered to be the *more* accurate, is not just a high-resolution version of the coarser model, but rather a completely different modeling approach of the Lagrangian models.

2.1 TERMINOLOGY AS USED IN THE FIELD

In this section I introduce the relevant terminology as has been used in the thesis as well as the research articles in the appendix.

2.1.1 *Terminology in Machine Learning*

Machine learning models are mathematical and statistical tools that allow machines to learn patterns from data and make predictions or decisions without being explicitly programmed to perform specific tasks.

In this thesis, The term *neural network* is used interchangeably with *machine learning model*. Neural networks have several advantages over other ML methods:

- **Handling Spatio-Temporal Data**: Neural networks can process large amounts of structural data which is extremely useful when using weather or climate data in gridded formats. Traditional ML methods like decision trees might struggle to handle the complexity and high dimensionality of such data without extensive manual tuning.

- **Learning non-linear Dependencies**: Other ML methods like Support Vector Machines (SVMs) or k-Nearest Neighbors (k-NN) may perform well on simpler, more linear problems, but they struggle with more complex data. A neural network is made up of multiple *layers* of interconnected *neurons*. The outputs from each layer of neurons is also processed through non-linear functions called *activation functions*. This equips them to decipher abtract patterns in data.

- **Scalability with Large Datasets**: Neural networks are particularly effective when trained on large datasets. As the amount of data increases, neural networks typically improve in performance, while other methods might plateau or even degrade.

2.1.1.1 *The Learning Process*

A neural network is composed of layers of interconnected nodes, where each connection between neurons has an associated *weight*. Mathematically, it is just a matrix which when processed with tinputs arrives at the approximated output. During *training*, the network is presented with a set of known data pairs: inputs (which could be numerical data, images, etc.) and corresponding desired outputs.

- *Forward Propagation*: The input data is fed through the network layer by layer. Each neuron processes the input it receives by applying a weighted sum and an activation function, ultimately producing an output. This output is the network's prediction.
- *Loss Calculation*: The prediction is then compared to the actual desired output using a loss function, which quantifies how far off the prediction is from the true value. Common loss functions include mean squared error or mean absolute error for regression tasks.
- *Backpropagation*: To minimize this error, the network undergoes a process called backpropagation, where the error is propagated back through the network. This involves computing the gradient of the loss function with respect to each weight in the network using the chain rule from calculus
- *Weight Update*: Once the gradients are computed, an optimization algorithm (often gradient descent) is used to adjust the weights in the direction that reduces the error. This step is iteratively repeated over many examples in the training data set, gradually improving the network's accuracy.
- *Convergence*: The training continues over multiple iterations (called epochs) until the network's performance stabilizes, ideally converging to a set of weights that minimizes the error across the entire training data set.

After the neural network has been trained for a particular task, it can then be used for *inference*. During *inference*, the ability of the trained network is put to test as now it must look at the input and produce the output without access to the correct answer. Throughout this thesis, we have used the terms *inference* and *offline testing* interchangeably.

2.1.2 *Specific to ML applications in earth science*

- A *rollout* refers to the process of using a trained ML model to predict the future evolution of a physical system over time. This involves the iterative process of generating a sequence of predictions by using the system state at a time step as the input for the next, allowing the model to simulate the evolution of the solution over time in an autoregressive manner.

- *Offline* testing refers to testing the ML parameterization in an isolated state, without being coupled to an atmospheric model. This usually implies a lack of dynamics and the effects of other physical parameterizations.

- *Online* testing refers to testing the ML parameterization while being coupled to the atmospheric model, allowing the parameterization to encounter the effects of dynamics. The modeling scenario usually reflects the data on which the ML model was trained. Due to the difference in codebases (ML model in Python and atmospheric models in FROTRAN mostly), online testing requires the development of special coupling mechanisms that provide flexibility without sacrificing speed of data transfer.

3

SUPERDROPNET: LEARNING DROPLET COLLISIONS

Collision-coalescence amongst droplets is one the most fundamental processes that contributes to breaking up and formation of new droplets. The treatment of collisions within a parameterization scheme is dependent upon the sophistication of the scheme itself. In Lagrangian schemes, including the superdroplet method, droplets are sampled from an assumed distribution (more detail in Section 3.1) and probabilities of collision are calculated.

However, the bulk moment schemes usually treat collisions in a more artificial manner. Under a warm rain scenario (where only rain and cloud droplets are present), collisions amongst rain and cloud droplets are often broken down into three individual processes, namely *autoconversion*, *accretion* and *self-collection*. Self collection of cloud droplets refers to the interactions where cloud droplets collide with other cloud droplets, forming enlarged droplets which are not large enough to be considered raindrops. This is an important phenomena that leads to that precedes any rain formation. Next, the process of autoconversion predominates as the cloud droplets collide and coalesce to form raindrops. Over the course of the simulation, accretion becomes dominant as the concentration of raindrops exceeds that of the cloud droplets. Accretion refers to the process where existing raindrops collect additional cloud droplets as they fall, leading to further growth of raindrops. As all cloud droplets are depleted, collisions between raindrops leading to the formation of larger raindrops predominate and is referred to as the self-collection of raindrops. Often, $x^* = 2.6\times10^{-10}$kg is the mass threshold dividing cloud and rain and droplets (Beheng & Doms, 1986). This approach of defining droplet interactions somewhat simplifies the collisions and provides target process rates to calculate. The bulk moment scheme used within ICON, uses empirical equations to calculate the rates of these processes as defined in Seifert and Beheng, 2001.

In this study I explore the possibility of bridging the gap between the crude bulk moment representation and more realistic superdroplet representations using machine learning. While preserving the bulk moment representation, is it possible to create an ML emulator that evolves the bulk moments just as they would in a superdroplet simulation?

3.1 THE QUEST TO FIND THE BEST MATCH

With the prevalence of Doppler radars since the 1940s, there had been a significant interest in observing the size of various hydrometeors. Raindrops in particular occur in a huge variety. It was found that an exponential size distribution is generally a good match (Marshall & Palmer, 1948). However, a more precise representation should be more general so that a wider range of droplets sizes can be taken into account and in Willis, 1984, it was shown, using droplets size measurements from spectrometers that there exist smaller droplets that can often be left out when using an exponential distribution and suggested instead the use of a gamma distributions. These smaller droplets, over time grow in size and hence, need to be accounted for. Similar results were observed in Ulbrich, 1983 and Ferrier, 1994, and it was noted that droplets less than 1mm in diameter were often observed in low clouds. For the purpose of this study, in line with the current scientific consensus, we have used a form of generalized gamma distribution of cloud/rain droplet size to initialize the superdroplet simulations. This form of DSD is derived in Seifert and Beheng, 2006.

3.2 THE SUPERDROPLET SETUP

To simplify the representation of droplet swarms, Shima et al., 2009 proposed the superdroplet method. This method reduces the individual droplets in a swarm to many "superdroplets". A superdroplet is defined as a representative droplet that informs about various individual droplets that are similar in size and close together in space. Multiplicity defines the total number of individual droplets represented by a single superdroplet. At every point of the simulation, the evolution of superdroplets is tracked instead of individual droplets. A Monte-Carlo algorithm is used to sample droplet pairs and then probability of collision is determined. To isolate the effect of collisions and learn an emulator for it, we only simulate collisions in a zero dimensional control volume of 2500 m^3 with multiplicity of 25600. We chose a box-model setup because collisions in ICON are parameterized such that they depend only upon the droplet concentrations within the grid box.

We assume a warm rain scenario. Such a setup provides an ideal setting for testing the evolution of collision-coalescence of droplets by reducing the amount of interactions that can take place: cloud and rain collisions, rain-rain and cloud-cloud collisions.

3.3 TARGETTING THE BULK MOMENTS

In Seifert and Rasp, 2020 the authors used the superdroplet simulations to train an ML emulator to predict the process rates(autoconversion, accretion and self-collection) corresponding to collisions. In this approach we go a step further and completely eliminate the need for calculating the process rates. Our bulk moments evolve with time as if they were calculated at every timestep from a corresponding droplet distribution. This simple approach also allows our machine learning model to be easily coupled to a suitable atmospheric model, which in our case is ICON.

Since we choose to train neural networks to directly estimate the bulk moment tendencies $(y_{t+1} - y_t)/\Delta t$ (assuming explicit Eulerian integration, as used in ICON for warm rain collisions), so that the learned time stepping function is

$$\mathcal{M}(y_t) = y_t + h_\theta(y_t, \phi)\Delta t \approx y_{t+1} \tag{2}$$

where,

$$y_t = [L_c(t), \quad L_r(t), \quad N_c(t), \quad N_r(t)] \tag{3}$$

θ are trainable parameters of the network, h_θ its input-output function and ϕ are additional inputs.

3.4 STOCHASTIC SIMULATIONS, DETERMINISTIC OUTPUTS

The superdroplet simulations are stochastic in nature. The stochasticity arises from the process of sampling of droplet pairs for determining the probability of collisions. In our superdroplet model where other processes such as evaporation and condensation do not exist, the stochasticity in simulations is especially exaggerated. The absence of evaporation, for instance, can amplify random fluctuations in the simulation. Since there's no mechanism to counterbalance the random growth events, the outcome of the simulation becomes more sensitive to initial conditions and random coalescence events.From the perspective of our machine learning model, with deterministic targets, learning from stochastic simulations posses a challenge. It mean that give a y_t as an input, there can be multiple possible values of y_{t+1}.

We carry out multiple experiments to control the stochasticity of the simulations by lowering multiplicity and increasing the control volume. In Unterstrasser et al., 2017, the authors found that the superdroplet method yields the best matches to the stochastic collection equation's analytical solutions when for every set of initial DSD, the outcome of 50 superdroplet simulations are averaged. We instead used the average of 100 superdroplet simulations for training SuperdropNet.

3.5 DEVELOPING A PHYSICS-INFORMED MACHINE LEARNING EMULATOR

In the sections below, I highlight some special features of SuperdropNet that enable it to emulate collisions from a superdroplet simulation in a physically-consistent manner.

3.5.1 *Enforcing Conservation Laws*

There are various ways in which the outputs of the neural network can be forced to be within a certain range. In Beucler et al., 2021, the authors explore different approaches for enforcing the conservation of energy and momentum while developing a convective parameterization. They claim that only predicting the "independent" variables from the neural network and using them to determine other "dependent" variables yielded the best results, instead of learning to predict all of the variables together. The small deviations in the neural network output can violate conservation laws, which might be extremely detrimental to maintaining stability over long time integration windows. We used a similar approach where due to the presence of only cloud and rain droplets, we only predict the change in the cloud water mass and then calculate the rain water mass from the difference in the total water mass. This ensures that SuperdropNet never violates conservation of mass.

3.5.2 *Autoregressive predictions*

A network trained to predict the future state of the system, at time $t+1$, from the current state at t can quickly run into out-of-distribution values during inference. This is because the network never predicts the system state at $t+1$ with 100% accuracy. For a long term rollout, the model's output at $t+1$ is then used to predict the state of the system at the next timestep. Over time, this can deviate the trajectory of the system thereby destabilizing it and generating unrealistic states. For ensuring long term stability, which often is a pre-requisite for stable *online* coupling, we introduce autoregressive training, also known as multi-step training (Brenowitz and Bretherton, 2018; Kelp et al., 2020, 2022; Um et al., 2020). In autoregressive training, the output of the network is fed into it iteratively for a set number of steps ,k, to output the next set of bulk moments. This training mode mimics the long rollouts that the network needs to output during inference. SuperdropNet is trained to iteratively predict 25 timesteps into the future. The training loss is given by:

$$\mathcal{L}_k = \sum_{j=1}^{k} \lambda_j \left\| y_{t+j} - \mathcal{M}^{(j)}(y_t) \right\| \tag{4}$$

The loss is calculated over the predicted moments through iterative application of $\mathcal{M}$ and the moments derived from the superdroplet simulations at the same time step. The scalar weights λ_j emphasize different prediction horizons while training. Initial experiments showed best results when setting $\lambda_k = 1$ and all other $\lambda_j = 0$. We use the pushforward trick mentioned in Brandstetter et al., 2022 for stabilizing the training process by only calculating the loss based on the $k-$th timestep. We find that autoregressive training significantly contributes to stable long term predictions and increasing the value of k during training massively improves the quality of predictions.

3.5.3 *Curriculum Learning*

The idea of curriculum learning is based on progressively increasing difficulty, whether through the training data or the training method, for a machine learning model (Bengio et al., 2009; Ionescu et al., 2016). This allows the network to learn the *easier* tasks first, before embarking on more challenging ones. In learning dynamical systems, this idea is often leveraged to stabilize autoregressive learning, where predicting for small number of time steps ahead can be considered an *easy* task while predicting multiple timesteps ahead can be categorized as *hard*. During training, we found that directly learning to predict 25 steps ahead is very challenging for a neural network as the loss blow up very quickly (even at 8 steps ahead), complicating the learning process. To remedy this, we first train a neural network to first predict 1 step ahead, with $k = 1$. After training ceases and the model has converged with very low errors for 1-step ahead predictions, we use this model's weights to jumpstart training for a new neural network for which $k = 2$. This process is continued until we reach $k = 25$, as at this point, the network does not learn further.

Hence, SuperdropNet is a collection of 25 individually trained neural networks. Not only does this stabilize training loss, but also makes intuitive sense to inherit weights from a network that already mastered an easier task.

3.6 DID SUPERDROPNET IMPROVE UPON THE BULK MOMENT SCHEME?

We tested SuperdropNet's long and short term performace against the bulk moment scheme(SB2001) as well as the ML model developed in Seifert and Rasp, 2020 (PRNet) which predicts the process rates as described in the beginning of this section.

Instead of looking at individual simulations, we use a metric $t_1 0$ for quantifying the performance of SuperdropNet and other models. The quantity t_{10} for cloud water refers to the time in the simulation where 10% of either the cloud water has converted to rain water. The t_{10} for cloud number concentration refers to the time when cloud number

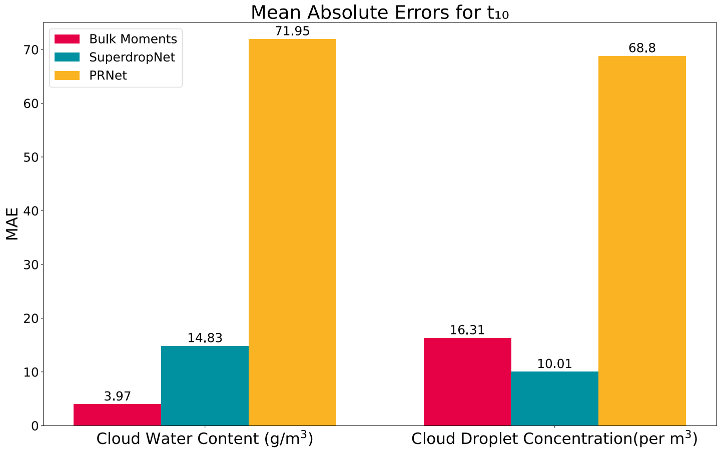


Figure 3.3: Mean Absolute Errors in the estimation of t_{10} for cloud water content(L_c) and cloud number concentration(N_c). SB2001 (pink) refers to the bulk momnet scheme from Seifert and Beheng, 2001, PRNet refers to the ML model developed in Seifert and Rasp, 2020. Taken from Study A.

concentration has decreased by 10% from its value at the beginning of the simulation.

A model that provides accurate estimates for t_{10} on average for all simulations, is a closer match to the superdroplet simulations. t_{10} also provides an insight into models which may suffer from delayed or rapid conversion of cloud into rain, compared to the superdroplet simulations. I summarize the results below:

- On average, SuperdropNet demonstrates a superior capability in estimating number concentrations compared to the bulk moment scheme (Fig. 3.3). This disparity highlights a fundamental issue with the bulk moment scheme: it is tuned primarily to produce accurate estimates of water mass, but it falls short in accurately estimating number concentrations

- SuperdropNet (blue solid line in Fig. 3.4 and 3.5) is a closer match to the superdroplet simulations (black line in Fig. 3.4 and 3.5) in comparison to the previous ML emulator developed in (Seifert & Rasp, 2020) (yellow dashed line in 3.4 and 3.5).

- Figures 3.4 and 3.5 also highlight the advantage of autoregressive training as SuperdropNet (blue solid) is a closer match to the superdroplet simulations (black) than the 1 step neural network(blue dashed).

- Finally, we found that SuperedropNet's performance was closely related to the initial conditions of the distribution itself. Simulations with low water content (Fig. 3.5) turned out to be more challenging. These were also the simulations benefiting most from autoregressive training. For slow evolving, low water simulations, SuperdropNet must learn extremely small tendencies over time. Even small errors in prediction, derails such simulations very quickly. Encouragingly, SuperdropNet excelled at simulations with sufficient water that evolve relatively faster. These simulations, with a run time of approximately 120 minutes (Fig. 3.4), correspond to warm rain cumulus in an atmospheric model.

Usage of superdroplet simulations allows for the exploration of conditions where only a very small amount of total cloud water converts to rain over extended periods, sometimes upwards of 12 hours. Such conditions are unlikely in real-world scenarios, as evaporation would typically prevent rain formation in these water-scarce environments. It is in these unlikely and challenging scenarios that SuperdropNet performs the worst, struggling to maintain accuracy over long periods with minimal water content. With this in mind, we next coupled SuperdropNet to a running simulation in ICON to evaluate its stability and to finally answer the question: What changes when SuperdropNet predicts warm rain collisions in ICON?

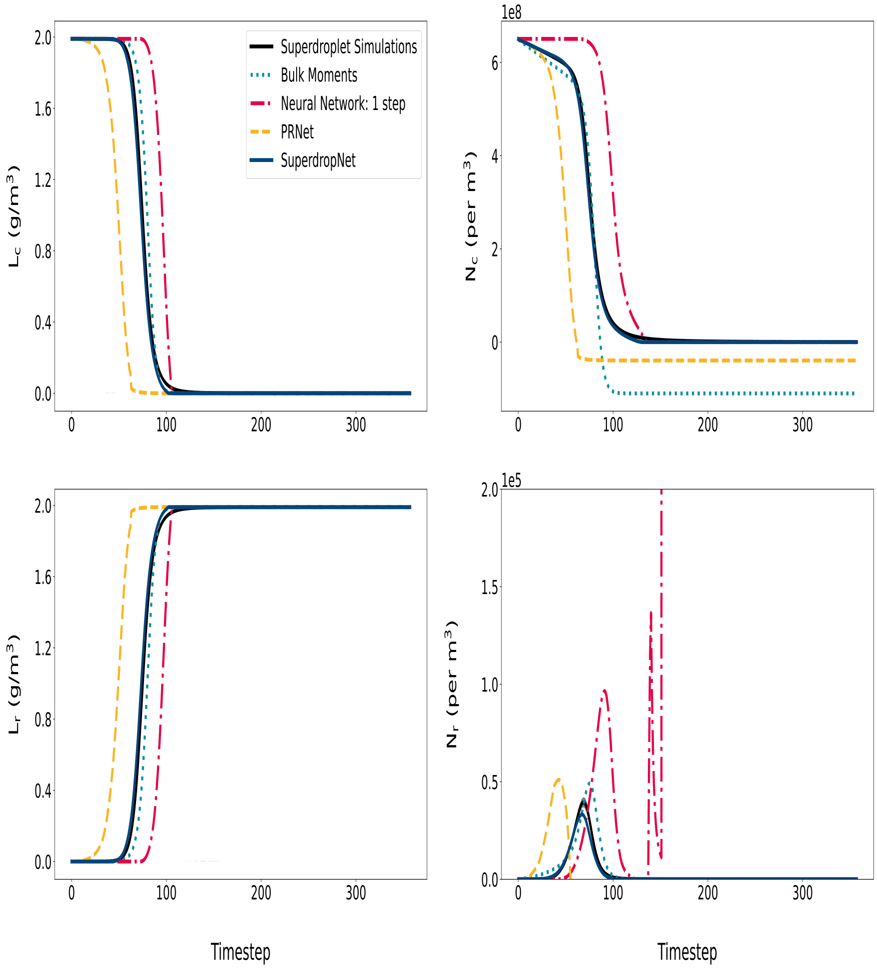


Figure 3.4: Superdroplet-derived bulk moments (black lines) compared to rollouts from a neural network trained to predict 1 step into the future (red-dashed lines), SuperdropNet (blue solid line), PRNet (yellow-dashed lines) and from a classical bulk moment scheme (blue-dotted lines). Results are shown for a simulation with $L_0 = 2\,\text{g/m}^3$, $r_0 = 9\mu\text{m}$, $\nu = 0$. Shaded region indicates +/- 1 standard deviation over 100 superdroplet simulations. A single time step corresponds to 20 s of simulation time.Taken from Study A

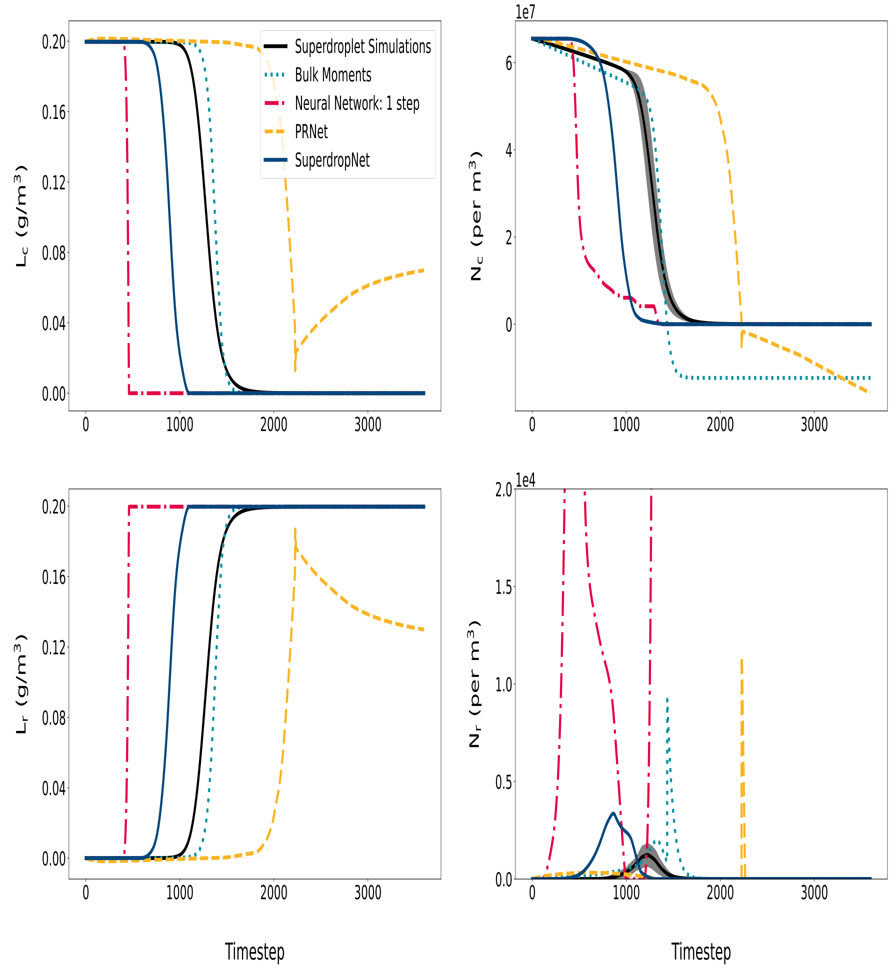


Figure 3.5: Superdroplet-derived bulk moments (black lines) compared to rollouts from a neural network trained to predict 1 step into the future (red-dashed lines), SuperdropNet (blue solid line), PRNet (yellow-dashed lines) and from a classical bulk moment scheme (blue-dotted lines). Results are shown for a simulation with $L_0 = 0.2$ g/m^3, $r_0 = 9\mu$m, $\nu = 2$. Shaded region indicates +/- 1 standard deviation over 100 superdroplet simulations. A single time step corresponds to 20 s of simulation time. Taken from Study A

4

COUPLING SUPERDROPNET WITH ICON

4.1 THE NEED FOR ONLINE COUPLING

The ultimate test for any ML parameterization(or for that matter, any parameterization scheme) is its ability to perform well when coupled to a running climate or weather model. It is standard procedure in most fields of earth science to develop parameterization schemes in isolated or simplified scenarios and then couple them to the target numerical model.

Development of any parameterization scheme is an iterative process. In its very fundamental form, this iterative process can be broken down into the following steps:

1. Isolate the target process such that it can be modelled outside of the outer model
2. Use mathematical models and/or observations for developing a new representation of the target process
3. Replace the old representation of the target process with this new representation in the outer model in a very rudimentary setting. This is done to see if coupling with simplest dynamics results in any inconsistencies before the added computational burden and complexity of a full NWP or climate run
4. If the previous step is a success, further coupling to an outer model with increased complexity
5. If Step 3 is not successful then go back to step 2, until step 3 results in a positive outcome

For the creation of a robust parameterization scheme, coupling to an outer model during the developmental stage is imperative. We can be drawn to believe that if a new parameterization scheme is theoretically and mathematically more descriptive than the older version, it is naturally going to result in better accuracy for the outer model as well. However, this is not always the case. The new parameterization needs to cooperate with various other processes that have been parameterized. Adding complexity to the outer model progressively during testing ensures that the point of failure can be easily detected.

4.2 COUPLING ML-BASED PARAMETERIZATIONS

Since machine learning models are developed in frameworks written in Python, coupling to the outer model can be technically challenging and often requires a *bridge*. However, success in *offline* performance does not always translate into good *online* performance (Brenowitz et al., 2020a; Pal et al., 2019; Rasp et al., 2018; Yuval et al., 2021). There exist various methods, with varying degrees of sophistication that can be used for coupling. However we found them to be lacking due to the following reasons:

- The earliest studies have used a rather crude approach of training an ML model and then writing specific routines of matrix multiplication in FORTRAN for the learned set of weights and biases. Apart from being prone to errors, this approach is time consuming, which prevents the ability to immediately couple a different ML models as for every new ML model, a separate multiplication routine would be needed.
- There are various bridges that are package specific, in that they can couple a specific ML framework with FORTRAN (Elafrou et al., 2023a; Ott et al., 2020). These are widely used for their convenience and speed but have limited capabilities as they do not allow for using other aspects of Python and sometimes, only allow the coupling of a a small variety of standard ML architectures.

4.2.1 *The bridges*

Ultimately, the choice of a bridge is dependent the target atmospheric model as well as the available hardware. Due to ICON's unique icosahedron grids, certain parameterizations might be challenging to couple. Fortunately for us, warm rain collisions in ICON work on the level of individual grids (Fig. 4.6), which also analogous to the 0-D training data for SuperdropNet. From our experience, the following bridges turned out the the most suitable for our application:

- We create an external Python library and used the C foreign function interface (CFFI) (Rigo and Fijalkowski, 2018) for communication between the Python library and ICON. There is inbuilt functionality in ICON that allows for addition of external libraries. We found this method to be extremely flexible as we had complete control over the Python codebase. However, during development, it required writing specific modules. Nevertheless, it also turned out to be the fastest means of coupling, out of the ones we tested.
- Another noteworthy candidate is the ironically named Yet-another-Coupler, or YAC (Hanke et al., 2023, 2016). YAC is often used within ICON for coupling different components, and also allows for calls to external Python libraries. While YAC turned out to be reliable and flexible, it increased runtime by a factor

of 2. Unlike CFFI, YAC runs two different jobs (one for Python and one for ICON), which adds to the runtime.

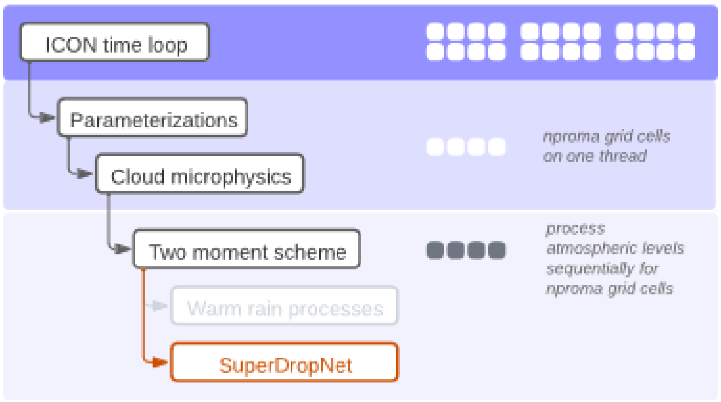


Figure 4.6: ICON flow control, where we replace the warm rain collisions with an external call to SuperdropNet(taken from Study B)

4.3 THE WARM BUBBLE TEST CASE

A warm bubble scenario refers to an experimental setup where the temperature is high enough to prevent formation of any snow or ice particles (hence, the term *warm*) and a single convective cloud (hence, the term *bubble*) is simulated. Especially for testing warm rain parameterization schemes, a warm rain bubble is a natural choice. The simplicity of a single convective cloud enables coupling to the dynamics while allowing detailed monitoring of prognostics variables such as precipitation, heat fluxes and wind velocities. Additionally, we carry out experiments in a cold bubble as well as a mixed-phase scenario.

4.4 LESSONS FROM COUPLING

In the warm bubble test case, we test SuperdropNet's performance against the default two-moment bulk scheme. One key piece of information to note is that a direct comparison against a superdorplet scheme coupled with ICON is not possible, as ICON does not allow for DSD-based representations. Hence, it is impossible to gauge SuperdropNet's deviation from a superdroplet scheme.

Coupling SuperdropNet, even with CFFI, increases runtime when compared against an ICON simulation with the default scheme. This is to be expected as adding an external library requires an interchange of large data arrays. SuperdropNet can never be faster than the two-moment bulk scheme but is always faster than superdroplet simulations from McSnow. Unfortunately, this too cannot be yet tested in an online setting.

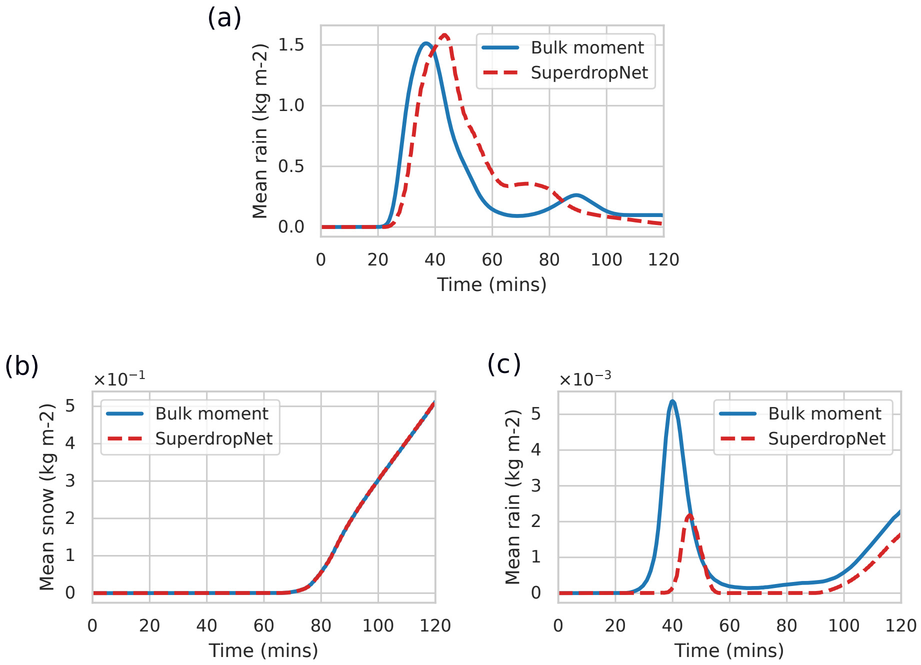


Figure 4.7: Grid-averaged quantities for the bulk moment scheme and SuperdropNet under (a) warm bubble scenario, (b) cold bubble scenario and (c) mixed phases scenario. Taken from Study B

We look at various prognostic variables such as precipitation flux, heat fluxes and profiles of specific humidity and rain droplet mass. Here are some the key findings:

- First of all, SuperdropNet remains stable throughout the length of the simulation and never produces physically implausible results such as excessive rain or unnatural looking profiles of humidity and rain water mass. For example, precipitation befalls in an expected manner such that initially, there is none and as the simulation progresses, it reaches a peak and then slowly rescinds without any discontinuities (Fig. 4.7-(a)).

- When compared against the two-moment bulk scheme, the onset of precipitation is delayed and it reaches a slightly higher peak.

- Overall the bulk moment scheme exhibits a higher evaporative flux near the surface, stronger winds and correspondingly, bigger raindrops(Fig. 4.8) near the surface. All of these findings are coherent as evaporative flux is proportional to the amount of raindrops and stronger winds point to increased exchange of energy. We also find that the vertical profiles of rain droplet

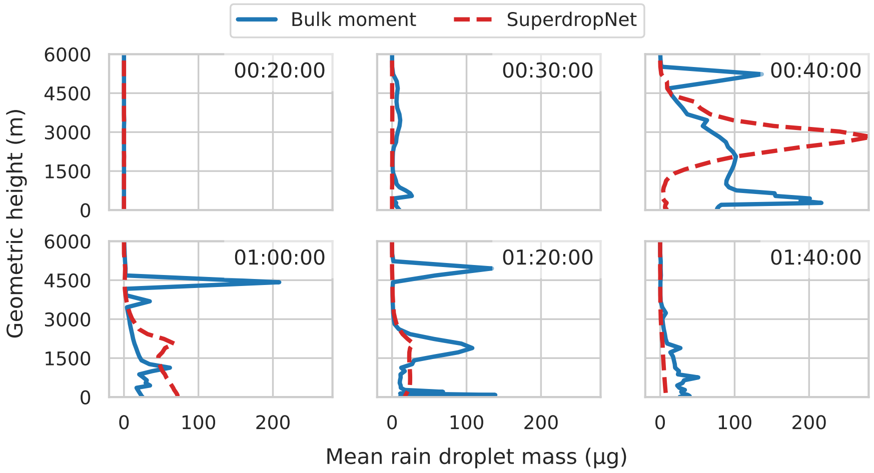


Figure 4.8: Vertical profile of the rain droplet mass, calculated as the ratio of the specific rain content and the number concentration of raindrops at different times for the bulk moment scheme and for SuperdropNet.Taken from Study B

mass obtained via SuperdropNet are smoother while those from the bulk moment scheme are more ragged. We infer from this that SuperdropNet *smoothes* out collisions in such a way that we can follow the descent of the rain water through the simulation (at the end of the simulation, most of the rainwater settles at the bottom and none is left at the top). On the other hand, the bulk scheme leaves large amounts of raindrops all over the column.

- In the case of cold bubble scenario (very low ambient temperature, such that clouds only convert to snow) we find that coupling to SuperdropNet makes no difference due the absence of cloud-to-rain conversion (Fig. 4.7-(b)). This served as a sanity check for the coupling mechanism and proved that SuperdropNet does not interfere with an ICON simulation unless the modeling scenario activates the formation of raindrops.
- A mixed-phase modeling scenario is perhaps the closest to a real atmospheric simulation where heterogeneity in hydrometeors is to be expected. Coupling with SuperdropNet delays the onset of rain more than in the case of a warm bubble scenario (Fig. 4.7-(a)), but contrary to the increased precipitation flux in a warm bubble, here it is significantly suppressed (Fig. 4.7-(c)). Warm rain collisions are simulated before processes related to ice and graupel formation. The suppression of rain formation leaves excess water mass to be processed by the subsequent submodules, resulting in an excess of snow and graupel.

SuperdropNet's stability during online testing is an encouraging result. It is possible to couple SuperdropNet to a complete NWP run in ICON, however, it would be computationally expensive due to the increased runtime. As the efforts to run ICON on GPUs continue, I believe that it would allow for faster coupling to ML-based parameterizations, which are trained to be run on GPUs. This would reduce the inference time spent within the Python module, making speed of data transfer the only problem to solve.

5

SUMMARY AND OUTLOOK

5.1 RESULTS

Can we improve the representation of warm rain collisions in the bulk moment scheme by learning ML-based representations from the more accurate, superdroplet simulations?

In Study A, I found that SuperdropNet could emulate the evolution of the superdroplet derived bulk moments better than the bulk moment scheme for rapidly evolving simulations (simulations with a high initial water content). However, for slow evolving simulations (with low initial water content), the bulk moment scheme was extremely hard to defeat. These results point to the nature of the problem, whereby in a slow evolving system, the ML model has to learn extremely small tendencies. This can be very challenging for the network as any noticeable deviation from the predictions can severely derail the simulation, as the magnitude of error can very easily surpass the value itself. Since these studies consider only a warm rain scenario, for realistic cases involving simulations where cloud-to-rain conversion occurs within 2 hours, the ML emulators provide better estimates of the bulk moments that the bulk moment scheme.

Can we impose the relevant physical laws on the ML emulator such that it predicts accurate and plausible bulk moments?

Unlike numerical solutions based on physical laws, machine learning methods are not inherently designed to follow the law of physics. Many studies have pointed out that using ML-based emulators can often violate laws of conservation of mass, energy and momentum (Rasp et al., 2018; Yuval et al., 2021). Therefore, to create reliable emulators, it is necessary to explicitly enforce appropriate restrictions that ensure that the ML model does not violate the laws of physics. The emulator developed in Study A conserves the total mass and remains stable for the duration of the simulations. These attributes are essential to a successful online coupling of ML-based emulators.

How challenging is the task of coupling an ML emulator to ICON? Once coupled, what changes are observed as we move from the default bulk moment scheme to an ML emulator for the superdroplet scheme?

In Study B, different bridges for ICON and SuperdropNet were tested (more generally for FORTRAN and Python). It was found that there are multiple ways of bridging the two codebases, depending upon one's specific requirements such as speed, ease of use and flexibility. We found that a C-based bridge worked the best for this application as it provided us with flexibility, which was crucial for immediate testing of different emulators. Ultimately, SuperdropNet remained stable in a coupled warm bubble scenario while generating physically plausible results. It was interesting to observe the variation in the warm bubble simulation when the collision-coalescence with the default bulk moment scheme is replaced with SuperdropNet. Overall, a higher precipitation is observed along with a delay in the onset of precipitation, a comparative decrease in the evaporative flux and smoother vertical profiles of moisture.

While it would have been to my great satisfaction to conclude that coupling with SuperdropNet provides more accurate results, the absence of a suitable baseline makes this impossible. An accurate *ground truth* could only be established by using a superdroplet-based parameterization in ICON, which remains impossible to implement. Many research groups are actively working to accomplish this. Only if coupling with SuperdropNet yielded results closer to the *ground truth*, compared to the bulk moment scheme (i.e., if the results from online coupling resembled Fig. 3.4), could I assert that SuperdropNet offers more accurate results in an online setting. I also found no conclusive evidence to suggest that a superdroplet-based representation consistently leads to higher rain formation in warm rain scenarios, similar to my findings in Study B. Therefore, I recommend interpreting the results of this study cautiously until the *ground truth* can be established.

5.2 OUTLOOK

An obvious future direction is to target other microphysical processes that are poorly represented in the bulk schemes. Even within the warm rain scenario, representation of processes such as sedimentation, for example, can cause instability and introduce shock waves (Wacker, 2000; Wacker & Seifert, 2001). Unlike the case of collisions, the gap between the droplet and bulk scheme based representations is more significant, which implies an even greater need for an ML emulator. In terms of the modeling scenario, this would introduce additional spatial complexity with the usage of a single column superdroplet model.

While it is possible to simulate only sedimentation in a superdroplet model and develop an ML emulator for it, an alternative modeling approach can also be adopted. For example, the superdroplet simulations can include multiple processes such as collisions, sedimentation, and evaporation. Depending on the need, either a single emulator can be developed or multiple emulators can be developed to isolate the effect of an individual process, splitting the different operations into separate ML models.

Fortunately, we already have the appropriate tools to couple these emulators to ICON (Study B). Depending on the processes, the ML emulators can then be added to ICON with calls to them replacing the submodules within ICON's microphysics subroutine.

Besides warm rain microphysics, a very interesting case is that of ice microphysics where the bulk moment scheme struggles with representing riming, nucleation and melting. Similarly, cloud-aerosol interactions are hard to parameterize and remain a source of uncertainty in climate models. A droplet based representation allows for a more realistic interaction and can be explored through superdroplet based methods.

Another direction involves the perhaps more challenging task of overhauling the representations of cloud microphysics within ICON. The main contribution of this thesis lies in using superdroplet simulations to improve the bulk moment scheme using machine learning. However, as long as we use the two-moment representation, there are limitations to the improvements that can be made. In a real atmosphere, the droplet distributions are often multi-modal in nature, or rather evolve into a multi-modal distribution over time. Use of a single representative set of moments fails to capture this diversity.

There are various generative machine learning methods that enable learning properties of complete distributions. Such a method would be able to learn the evolution of the complete DSDs from a superdroplet simulation for every timestep. However, as of now, this would only serve as a faster superdroplet emulator, with limited operational application since operational NWPs do not allow a distribution based representation of cloud microphysical process.

There remains a lot to be gained from parallel areas of research. In this thesis, the evolution of the bulk moments with time is treated simply as a dynamical system where the future state depends upon itself. Many of the techniques I employ were borrowed (Section 3.5) from ML models trained for completely different dynamical systems. These emulators serve as faster numerical solvers, similar to the way I have utilized SuperdropNet for superdroplet simulations.

ML emulators for cloud microphysics can benefit from using more complexity in the structure of the ML model, such as addition of convolution layers (LeCun et al., 1998) for spatial grids. Since SuperdropNet is trained on box-model simulations, convolution layers were not needed. Newer techniques such as diffusion models have shown promise in predicting turbulent flow (Kohl et al., 2024) and would be an interesting method to test when learning DSDs. Some studies have gained from using multiple time steps as an input to the ML model and then predicting multiple timesteps forward in time (Brandstetter et al., 2022). This approach is challenging to adopt in Earth Science as during online coupling, the state of the system is modified at every single timestep, limiting the utility of an ML emulator that predicts tendencies multiple steps forward in time for a particular process.

Benefits of autoregressive training have made it popular. However, many of its aspects are contested. There are studies disputing the usefulness of the pushforward trick in stabilizing training (Lippe et al., 2023). The ideas of curriculum learning are applied differently in every study (Geneva & Zabaras, 2020; Kohl et al., 2024; List et al., 2024). For me, development of SuperdropNet was an experimental process in terms of finding suitable techniques that work for my particular problem. Given the evolving nature of these ML techniques, future research could reveal both strengths and potential shortcomings of SuperdropNet. I look forward to revisit and refine the model as new insights emerge.

As computational power continues to advance and high-resolution models become the standard, processes like convection, which once required parameterization, can now be fully resolved. Over time, I expect fine-scale processes, such as cloud interactions and turbulence, to become the main source of model uncertainty. These processes are often not well understood, and stochastic methods, such as the superdroplet scheme, can help provide better estimates despite our gaps in understanding. Combined with airborne and ground-based observations, stochastic models can serve as data assimilation methods, providing corrective values at regular intervals. Improved representation of droplets would also enable better estimates of Earth's radiative balance, both through the direct estimation of cloud cover and the improved representation of cloud-aerosol interactions. Until distributions of droplets in atmospheric models become standard, using ML to bridge the gap remains a valuable endeavor.

Part II

APPENDIX

A

SUPERDROPNET: A STABLE AND ACCURATE MACHINE LEARNING PROXY FOR DROPLET-BASED CLOUD MICROPHYSICS

This work is currently under review at the Journal of Advances in Modeling Earth Systems:

Sharma, S., & Greenberg, D. S. (2024). SuperdropNet: A stable and accurate machine learning proxy for droplet-based cloud microphysics. *under review at the Journal of Advances in Modeling Earth Systems*. https://doi.org/arXiv:2402.18354

AUTHOR CONTRIBUTIONS

D. Greenberg and I conceptualized the study. I carried out data generation and model development. D. Greenberg and I prepared the manuscript and contributed to the interpretation of the results.

SUPERDROPNET: A STABLE AND ACCURATE MACHINE LEARNING PROXY FOR DROPLET-BASED CLOUD MICROPHYSICS

Authors:
Shivani Sharma[1,2,3,*], David S. Greenberg[2]
Affiliations:
[1]Helmholtz-Zentrum Hereon, Institute of Coastal Systems,
Model-Driven Machine Learning, Geesthacht, Germnay
[2]International Max Planck Research School on Earth System
Modeling, Hamburg, Germany
[3]Helmholtz AI

[*]**Corresponding author:** Shivani Sharma, shivani.sharma@hereon.de

KEYPOINTS

- We train SuperdropNet, a warm-rain emulator that learns from superdroplet simulations while operating on bulk moments
- For stable and accurate predictions, we employ autoregressive training, simplify statistical assumptions, and impose physical constraints
- SuperdropNet significantly outperforms other ML-based parameterizations over a broad range of conditions

ABSTRACT

Cloud microphysics has important consequences for climate and weather phenomena, and inaccurate representations can limit forecast accuracy. While atmospheric models increasingly resolve storms and clouds, the accuracy of the underlying microphysics remains limited by computationally expedient bulk moment schemes based on simplifying assumptions. Droplet-based Lagrangian schemes are more accurate but are underutilized due to their large computational overhead. Machine learning (ML) based schemes can bridge this gap by learning from vast droplet-based simulation datasets, but have so far struggled to match the accuracy and stability of bulk moment schemes. To address this challenge, we developed SuperdropNet, an ML-based emulator of the Lagrangian superdroplet simulations. To improve accuracy and stability, we employ multi-step autoregressive prediction during training, impose physical constraints, and carefully control

stochasticity in the training data. Superdropnet predicted hydrometeor states and cloud-to-rain transition times more accurately than previous ML emulators, and matched or outperformed bulk moment schemes in many cases. We further carried out detailed analyses to reveal how multistep autoregressive training improves performance, and how the performance of SuperdropNet and other microphysical schemes hydrometeors' mass, number and size distribution. Together our results suggest that ML models can effectively emulate cloud microphysics, in a manner consistent with droplet-based simulations.

A.1 INTRODUCTION

In early versions of weather and climate models, low spatial resolution was the primary source of model errors (Manabe & Bryan, 1969). Over time, finer grids and shorter time steps improved accuracy by explicitly representing processes that would otherwise occur at subgrid scales (Bauer et al., 2015), but the resulting computational costs, which scale with the 4^{th} power of relative increases in spatial resolution, make further improvements unsustainable on existing and foreseeable computing hardware (Palmer, 2020).

For operational weather forecasting and long term climate projections, it is still standard practice to parameterize sub-grid scale processes (Gross et al., 2018; Palmer, 2020) such as convection, radiation and cloud microphysics. All parameterization schemes involve some form of approximation which, while speeding up calculations, lead to errors and uncertainties in simulations and forecasts (Gross et al., 2018). In numerical weather prediction (NWP) models, systemic errors accumulate over time and model errors tend to be of the same magnitude as the predicted signals after forecast lead times of 7-10 days (Palmer, 2020). Parameterizations contribute significantly to these model errors (Gross et al., 2018; Palmer, 2020).

In recent years, a data driven approach has been applied in many instances to replace the traditional parameterization schemes. Essentially, simulation routines too computationally intensive for use in operational forecasting are used to generate training data for optimizing a machine learning (ML) model. This approach has been applied to convection (Brenowitz & Bretherton, 2018; Gentine et al., 2018; O'Gorman & Dwyer, 2018; Rasp et al., 2018; Yuval et al., 2021), radiative transfer (Belochitski & Krasnopolsky, 2021a; Veerman et al., 2021), gravity wave drag (Chantry et al., 2021) and gravity wave simulation (Dong et al., 2023), atmospheric chemistry (Kelp et al., 2020, 2022) and turbulence (Leufen & Schädler, 2019). Many ML models perform well in predicting single time steps but exhibit instability when run iteratively for many time steps. The primary source of instability here is the accumulation of error over longer integration times. In(Brenowitz & Bretherton, 2018) a multi-step loss was applied during training for developing a unified physics based parameterization for weather prediction time scales in an aquaplanet whereas in (Lam et al., 2023), similar approach was carried out to obtain stable results for medium-range weather forecasting. Both of these studies point to the gain in performance and long-term stability when using an autoregressive loss function.

Here we focus on the task of parameterizing cloud microphysics, and specifically the coalescence of liquid cloud droplets into rain. Weather prediction models typically employ bulk moment schemes (Seifert & Beheng, 2001) that simplify particle size distributions into the total mass and number densities of droplets above and below a chosen cloud/rain threshold size, and rely on approximate physics and statistical assumptions to update these quantities over time. While bulk moment schemes are fairly consistent with droplet-based simulations in simplified scenarios, they struggle in the presence of mixed-phase clouds, particularly in the representation of ice microphysical processes(Khain et al., 2015; Morrison et al., 2020). Ultimately, inaccurate cloud microphysics schemes manifest as inaccurate precipitation forecasts (Lynn & Khain, 2007) and errors in radiative transfer calculations.

In (Gettelman et al., 2021) an ML emulator for a bin microphysics scheme was developed and coupled to a general circulation model for an improved representation of cloud and rain particle distributions in a warm rain scenario. Another study showed that neural networks can be trained for computing bulk moment dynamics to match superdroplet simulations (Seifert & Rasp, 2020). This approache combines the low memory and compute requirements of bulk moment schemes with the accuracy, simplicity and physical consistency of superdroplet simulations. It also offers straightforward compatibility with existing atmospheric models that rely on bulk moment representations to simulate radiation transfer, convection and other moisture-dependent processes. However, while the networks accurately predicted instantaneous rates of various coalescence events, they were far less accurate at predicting the evolution of bulk moments over the longer time scales of cloud-to-rain transitions. Thus, for simulating cloud microphysics over the time scale of warm rain generation, ML currently exhibits a major performance gap compared to classical bulk moment parameterizations.

To address this challenge we developed SuperdropNet, an emulator for superdroplet simulations in a warm rain scenario. We introduce several innovations in designing and training our network that improve accuracy and stability over long time integration windows for diverse initial conditions. These include autoregressive prediction of multiple time steps during training (Grzeszczuk et al., 1998; Kelp et al., 2020, 2022; Um et al., 2020), mass conservation as a hard constraint, relaxing some assumptions of bulk moment schemes and carefully controlling stochasticity when generating training data. We also systematically analyze how the length of autoregressive rollouts during training affect forecast accuracy at various time horizons. We compare SuperdropNet to a traditional warm rain bulk moment scheme (Seifert & Beheng, 2001) commonly used in the ICON (Icosahedral Nonhydrostatic) model, and to a previously described ML-based parameterization (Seifert & Rasp, 2020). Our results show that SuperdropNet significantly closes the accuracy gap between ML parameterizations and bulk moment schemes, and even outperforms classical schemes for some initial conditions. We further examine performance of these schemes, and identify how their accuracy can depend on various factors in the simulated scenario.

A.2 WARM RAIN MICROPHYSICAL SIMULATIONS

A.2.1 *Droplet schemes*

Simulating droplets in a numerical simulation provides the most accurate estimation of droplet interactions that contribute to the cloud microphysical processes. However, individually simulating droplets can be computationally expensive even for small domain sizes. In Shima et al., 2009 this problem is simplified by using 'superdroplets' to represent multiple individual droplets of same size that are close to each other. The motion of the droplets is simulated along with collision-coalescence, condensation/evaporation and sedimentation processes. This method uses a Monte Carlo scheme for estimating the collision-coalescence process. The droplet size distribution is initially assumed to be gamma, exponential or a log-normal (Marshall & Palmer, 1948), and is evolved in time by sampling pairs of colliding droplets from it.

A.2.2 *Bulk moment schemes*

In most atmospheric models, cloud microphysical processes are represented using bulk moment schemes. Instead of simulating droplets and tracking collision probabilities, bulk moment schemes track only the evolution of the first and sometimes the zeroth moment of the droplet distribution. A warm rain scenario, with only clouds and rain present, is fully described by a density function $f(x)$ over droplets with mass x. This density function is used to define 4 bulk moments:

$$\begin{aligned} L_c &= \int_0^{x^*} x f(x) dx & N_c &= \int_0^{x^*} f(x) dx \\ L_r &= \int_{x^*}^{\infty} x f(x) dx & N_r &= \int_{x^*}^{\infty} f(x) dx \end{aligned} \tag{5}$$

Here $x^* = 2.6\times10^{-10}$kg is the mass threshold dividing cloud and rain and droplets (Beheng & Doms, 1986). The zeroth moments, N_c and N_r, are the number density of cloud and rain droplets respectively. The first moments, L_c and L_r, represent the total cloud and rain water mass.

To provide a baseline when evaluating ML-based microphysical schemes, we employ here the two-moment bulk scheme of Seifert and Beheng, 2001. The time evolution of cloud water, rain water, cloud droplet concentration and rain droplet concentration is estimated through the ordinary differential equation system given by:

$$\frac{dL_c}{dt} = -AU - AC \tag{6}$$

$$\frac{dL_r}{dt} = +AU + AC \tag{7}$$

$$\frac{dN_c}{dt} = -2AU_n - AC_n - SC_c = \frac{-2}{x^*}AU - \frac{1}{\overline{x_c}}AC - SC_c \tag{8}$$

$$\frac{dN_r}{dt} = +AU_n + AC_n - SC_r = \frac{1}{x^*}AU - SC_r \tag{9}$$

The autoconversion rate AU is the rate at which cloud mass converts to rain mass due to collisions between the cloud droplets, while the accretion rate AC describes the mass flux associated with collisions between rain and cloud droplets. The mean cloud droplet mass $\overline{x_c} = L_c/N_c$ is the average mass of cloud droplets. AU_n and AC_n are autoconversion and accretion rates corresponding to the number concentrations. They are calculated by assuming that autoconversion events involve droplets with an average mass of x^* and that accretion events lead to the formation of cloud droplets with an average mass $\overline{x_c}$. Self collection rates SC_c, SC_r describe the rate of collisions that do not convert cloud droplets to rain.

Following standard practice for cloud microphysics in atmospheric models such as ICON (Zängl et al., 2015), we carry out explicit (Euler) integration of these differential equations with fixed time step Δt. In practice, Δt used for microphysical processes may be shorter than the time step used for the dynamical core in an atmospheric model, and can vary from a few minutes to a few seconds.

The scheme in Seifert and Beheng, 2001 employs several approximations and statistical assumptions to derive formulas for calculating the process rates in a warm rain scenario. This approach of approximating process rates to compute changes in bulk moments is almost universally applied in operational weather and climate models and formed the basis for the ML approach formulated in Seifert and Rasp, 2020.

A.3 PROBLEM STATEMENT

This study aims to develop a scheme for warm-rain microphysics that efficiently computes bulk moment dynamics consistent with superdroplet simulations. Having described droplet-based and bulk moment schemes, we can now give a concrete formal description of this task. We use y_t to denote the vector of bulk moments at time t in a superdroplet simulation.

$$y_t = [L_c(t), \quad L_r(t), \quad N_c(t), \quad N_r(t)] \tag{10}$$

A classical or ML-based scheme is defined by a time-stepping function $\mathcal{M}$, which can be applied to y_t to compute $\mathcal{M}(y_t) \approx y_{t+1}$. We can also iteratively apply $\mathcal{M}$ k times, feeding the outputs back in as inputs to generate an autoregressive prediction. We denote this repeated application by $\mathcal{M}^{(k)}$, and can use it to predict y, the vector of bulk moments, k steps into the future.

$$\mathcal{M}^{(k)}(y_t) = \overbrace{\mathcal{M} \circ \mathcal{M} \circ \cdots \circ \mathcal{M}}^{\text{k times}}(y_t) \approx y_{t+k} \tag{11}$$

We refer to sequence of bulk moment vectors $\left\{\mathcal{M}(y_t), \mathcal{M}^{(2)}(y_t), \ldots, \mathcal{M}^{(k)}(y_t)\right\}$ computed by k repeated applications of $\mathcal{M}$ as a length-k *rollout*.

Our goal is then to obtain an $\mathcal{M}$ that can evolve bulk moments, starting from initial conditions y_0, to match the full course of a superdroplet simulation:

$$\mathcal{M}^{(t)}(y_0) \approx y_t, \quad \forall t \tag{12}$$

After giving further details on the superdroplet simulations used for training and evaluation (section A.4), we will describe how we use neural networks to define $\mathcal{M}$, and how we design and minimize a loss function to achieve our stated aims (section A.5).

A.4 DATA GENERATION WITH DROPLET SIMULATIONS

For generating the training data, we simulate superdroplets in a warm rain scenario in a zero-dimensional box. The superdroplet simulations were carried out using the McSnow software (Brdar & Seifert, 2018; Seifert & Rasp, 2020), and the output is recorded every $\Delta t = 20$ s. This time step was chosen to be consistent with previous work (Seifert & Rasp, 2020), and to fall within the typical range for atmospheric modeling.

Superdroplet simulations were used to compute bulk moments as follows:

$$\begin{aligned} L_c &= \sum_{\forall i : x_i < x^*} x_c \xi_c & \qquad N_c &= \sum_{\forall i : x_i < x^*} \xi_c \\ L_r &= \sum_{\forall i : x_i \geqslant x^*} x \xi_r & \qquad N_r &= \sum_{\forall i : x_i \geqslant x^*} \xi_r \end{aligned} \tag{13}$$

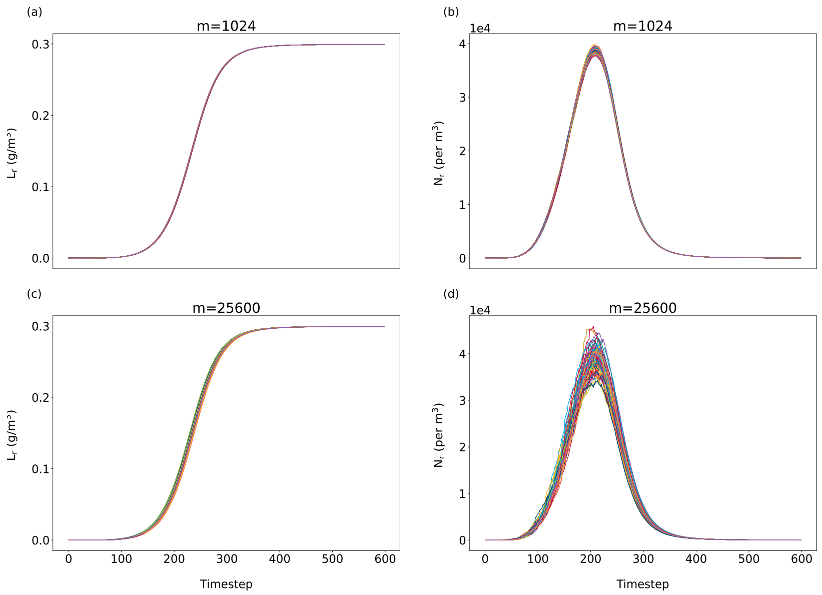


Figure A.1: Stochasticity of superdroplet simulations with box size 25m^3. (a) and (b) correspond to 100 simulations at multiplicity of 1024. (c) and (d) correspond to 100 simulations at multiplicity of 25600. For both sets of simulations, rain water mass (L_r) and rain number concentration(N_r) are compared. Other initial conditions for both simulations are the same with L_0=0.3 g/m^3, r_0=13 µm, ν=0.5.

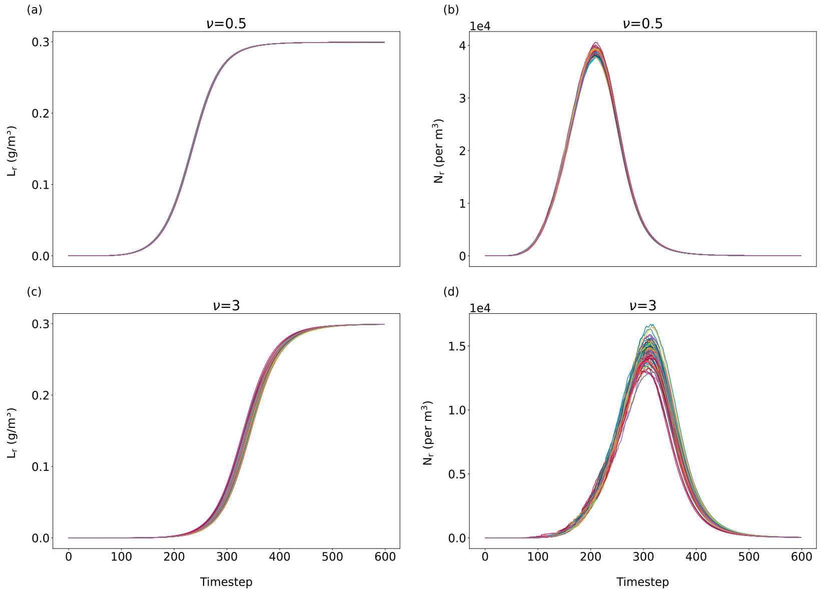


Figure A.2: Stochasticity in simulations at a box size of 2500 m^3 and multiplicity is 25600. (a) and (b) correspond to 100 simulations at ν=0.5. (c) and (d) correspond to 100 simulations at ν=3. For both sets of simulations, Rain water mass(L_r) and Rain number concentration(N_r) are compared. Other initial conditions for both simulations are the same with L_0=0.3 g/m^3, r_0=13 µm.

Here mass moments (L_c, L_r) measure the total mass of superdroplets above and below the cloud-rain threshold, while number counts (N_c, N_r) count the total number to cloud and rain droplets.

A.4.1 *Initial conditions*

For all simulations the box volume and multiplicity are fixed to be 2500 m^3 and 26500, respectively. In the context of superdroplet simulations, multiplicity refers to the number of individual droplets represented by a single superdroplet. Superdroplet simulations are initialized by sampling droplets with total mass L_0 from a gamma distribution, with shape parameter ν and mean droplet radius r_0. For a fixed mean droplet size, higher values of ν indicate a narrower distribution around a peak, while $\nu = 0$ gives an exponential distribution peaking at zero. This form of generalized gamma distribution is commonly used for describing the distribution of hydrometeors and further explanation and derivations can be found in (Seifert & Beheng, 2006). We sample from a wide range of parameter values as described in Table 1. In total, these combinations of parameter values lead to 819 distributions.

Table 1: Initial Conditions for box-model simulations

Quantity	Range
L_0	{0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,1.2,1.5,1.6,2.0} g/m^3
r_0	{9, 10, 11, 12, 13, 14, 15} μm
ν	{0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4}

The complete set of ICs was the same as in Seifert and Rasp, 2020.

A.4.2 *Measuring and controlling stochasticity*

As stated in section A.3, our ultimate aim is to train a neural network to update bulk moments, using data from superdroplet simulations. A challenge is posed by the fact that superdroplet simulations are inherently stochastic, and the same initial conditions can produce different results with in repeated simulations. The randomness in simulations is a feature of the zero-dimensional setup and in a real atmosphere this stochastic behavior is greatly reduced. In initial experiments, we found that randomness in training data, and in particular large infrequent jumps in the bulk moments, lead to overfitting of individual stochastic events, and instability in the optimization process used to train the network.

For fixed mean droplet radius and total water mass, randomness can be reduced by increasing the box volume or decreasing the multiplicity while keeping the total droplet count fixed. Since this increases the total number of simulated superdroplets, with more superdroplets, the effects of random collisions tend to better 'average out'. However, computation time also increases quadratically in the number of droplets. Reducing the shape parameter of the initial distribution also limits stochasticity by reducing the relative contribution of extremely large or small droplets.

We measured these effect of stochasticity by computing bulk moments from repetitive superdroplet simulation runs under the same set of initial conditions. We first examined the role of multiplicity using simulations with a low volume (25 m^3), moderate density ($L_0 = 0.3\ g/m^3$) and mean initial droplet size of 13 μm. For over 100 simulations with high multiplicity ($m = 25600$), we observed considerable variation in the height and timing of the peak raindrop count(Fig 1-(d)), and noticeable variation in the time evolution of the rain droplet mass (Fig. 1-(c)). Simulations with a lower multiplicity ($m = 1024$) exhibited noticeably less variation in these quantities (Fig. 1-(a,b)), suggesting that this stochasticity is an artifact of the superdroplet technique and exceeds the stochasticity of the original droplet collision rules.

We next examined the effect of the shape parameter ν in additional simulations, using the same L_0 and r_0, multiplicity 25600, a larger box (2500 m^3) and ν=0.5 or 3. As previously observed (Seifert & Rasp, 2020), we found that higher ν values produced greater variation in bulk moments over repeated runs (Fig. 2). Similarly, variability was higher for smaller box sizes (Fig. 1, lower vs. Fig. 2, upper).

To limit stochasticity when generating training data for deep learning (see below), we used the larger box size of 2500 m^3. To limit computation time and allow better comparison with previous work, we kept the multiplicity value of 25600 from Seifert and Rasp, 2020. To further limit variability, for each unique set of initial conditions (r_0, ν, L_0) we averaged bulk moments from 100 superdroplet simulations. We found that training on all individual simulations resulted in overfitting, poorer performance and longer computation times.

A.4.3 *Data preparation*

We carried out superdroplet simulations with 819 unique initial conditions (Table 1). Bulk moments were calculated from the collection of superdroplets every 20 seconds, and averaged over 100 repeated simulations of each set of initial conditions. We randomly assigned 100 superdroplet simulations to testing and the remaining 719 were assigned for training and validation. From those 719 simulations, the data was randomly chunked and 90% of it was assigned for training and 10% for validation. We z-scored each dimension of the inputs to the neural network.

A.5 DEEP LEARNING OF WARM RAIN MICROPHYSICS

A.5.1 *Time stepping with learned moment updates*

We chose to train neural networks to directly estimate the bulk moment tendencies $(y_{t+1} - y_t)/\Delta t$, so that the learned time stepping function is

$$\mathcal{M}(y_t) = y_t + h_\theta(y_t, \phi)\Delta t \approx y_{t+1} \tag{14}$$

θ are trainable parameters of the network, h_θ its input-output function and ϕ are additional inputs (details in sec. A.5.4).

This contrasts with the strategy presented in (Seifert & Rasp, 2020) which instead estimates process rates, then uses the same approximations as the original bulk moment scheme to impute droplet number-based process rates from droplet mass-based rates (sec. A.2.2). We compare our results to one such network and refer to it as PRNet. This network receives the bulk moments as the input and predicts the process rates which are then used to calculate the changes in the bulk moments as given in Equations 2-5. Our approach can potentially provide a closer match to superdroplet schemes by avoiding the approximation of droplet number-based process rates from droplet mass-based process rates, since it directly outputs the updates to the bulk moments.

A.5.2 *Physically constrained deep learning*

To improve accuracy and physical consistency, we enforced constraints that hold in superdroplet and bulk moment simulations on our neural networks.

- **Mass conservation** To ensure the total mass of rain and cloud droplets is conserved, we trained the neural network to predict cloud mass updates $\Delta\widehat{L_c} = L_c(t + \Delta t) - L_c(t)$, and defined $\widehat{L_r} = L_0 - \hat{L}_c$.

- **Irreversibility** In superdroplet and bulk moment simulations the total mass and count of cloud droplets can only decrease over time. To enforce this, we set any positive updates to cloud mass and droplet counts to zero at each time step (before calculating $\widehat{\Delta L_r}$). Initial experiments showed that this constraint interfered with learning by preventing backpropagation of loss gradients, so we used it only with trained models.

- **Positivity** Mass and droplet counts cannot be negative. We enforced this in a postprocessing step after moment prediction for all time steps, with negative moments set to zero while maintaining mass conservation. We did not use this constraint during training.

A.5.3 *Autoregressive Multi-step Training*

Our overall aim (sec. A.3) is to predict the evolution of bulk moments through repeated application of a trained network. Clearly, if $\mathcal{M}(y_t) = y_{t+1}$ precisely for all t, then also $\mathcal{M}^{(j)}(y_t) = y_{t+j}$ for all t and j. This suggests a simple 'offline training' strategy of minimizing 1-step prediction error $\mathcal{L}_1 = \|y_{t+1} - \hat{y}_{t+1}\|$.

However, this fails in practice since $\mathcal{L}_1$ does not reflect the rate at which errors grow over a rollout. Thus when a network trained offline generates a multistep rollout, it inevitably makes at least some small errors and encounters inputs outside the training set. This can lead to inaccurate results or divergence to infinity as rollout length increases. This problem is particularly severe for simulations with lower water content, which require a greater number of time steps to produce rain. The inadequacy of offline training has been noted for several other parameterization tasks (Kelp et al., 2020, 2022; Kochkov et al., 2021; Um et al., 2020).

A.5.3.1 *Loss Function for Autoregressive Rollouts*

To address this limitation, we predict k future time steps during training. Starting with superdroplet-derived bulk moments y_t, we use our network iteratively k times:

$$\mathcal{L}_k = \sum_{j=1}^{k} \lambda_j \left\| y_{t+j} - \mathcal{M}^{(j)}(y_t) \right\| \tag{15}$$

The loss is calculated over the moments as predicted by the iterative application of $\mathcal{M}$ and the moments as calculated from the superdroplet simulations at the same time step. The scalar weights λ_j allow emphasis of different prediction horizons during the training process. Initial experiments showed best results when setting $\lambda_k = 1$ and all other $\lambda_j = 0$. We also used the 'pushforward trick', in which $\mathcal{M}^(k-1)$ is treated as a fixed variable, and the loss gradients are not backpropagated to times earlier than $t + k - 1$. Previous studies reported that this improved performance in some cases (Brandstetter et al., 2022), though this seems to depend on the particular application (List et al., 2024) and the involved mechanism remains an open question.

A.5.3.2 *Increasing rollout length during training*

A challenge for optimizing functions such as $\mathcal{L}_k$, with repeated self-iteration of neural network, is that gradients can vanish towards zero or explode towards infinity over repeated iterations (Goodfellow et al., 2016). This problem is particularly severe for networks at the start or early phase of training, since these tend to produce large errors and have not yet encountered inputs resembling their own erroneous outputs. To avoid instability during training, we begin in offline mode with $k = 1$, and then gradually increase the value of k during training. We train until convergence (sec. A.5.5) on $\mathcal{L}_k$, then use the resulting network parameters to initialize training on $\mathcal{L}_{k+1}$, for $k \leqslant 25$.

While this procedure required longer computation time than a single optimization procedure, we found that initializing the neural network from the weights of the previous k value greatly reduced the number of optimization steps required. This is equivalent to a 'warm start' in training as the neural network at $k+1$ receives pre-trained weights from k-th model instead of randomly initialized weights.

A.5.4 *Network Architecture*

We use a fully-connected network with 3 hidden layers of 200 neurons each with ReLU (Rectified Linear Unit) activation functions. The total number of trainable network parameters was 83,200. In addition to y_t the network receives 5 additional inputs ϕ: the liquid water time scale ($\tau = L_r/L_0$), mean cloud droplet mass $\overline{x_c} = L_c/N_c$, r_0, ν and L_0. The outputs of the neural network are the tendencies for the four moments.

A.5.5 *Optimization Procedure*

We minimized $\mathcal{L}_k$ (eqn. 15) using the ADAM optimizer(Kingma & Ba, 2017) with initial learning rate 2e-4 for $k = 1$ and the batch size as 256. As we subsequently increased k, the learning rate was decreased as the updates to the network weights also decreased in magnitude. The learning rate was halved if at the subsequent value of k, training ended before 10 epochs. This procedure to update the learning rate was followed until $k = 24$. For $k = 25$, the learning rate was set to 2e-8 and decreasing it further did not yield continued reduction of the loss.

Optimization for each k used a maximum of 500 epochs. Training was cut short using early stopping when the validation loss did not decrease for 50 consecutive epochs. Bulk moments and network parameters were represented using 32-bit floating point.

A.6 RESULTS

A.6.1 *Accuracy of Single Step Training*

We first trained a neural network to predict superdroplet-derived bulk moments one time step into the future. After convergence, this '1-step' network closely predicted all 4 bulk moments one time step ahead (Fig. A.3-top panel). We calculated the mean absolute percentage errors(MAPE) between the predicted tendencies(direct output of

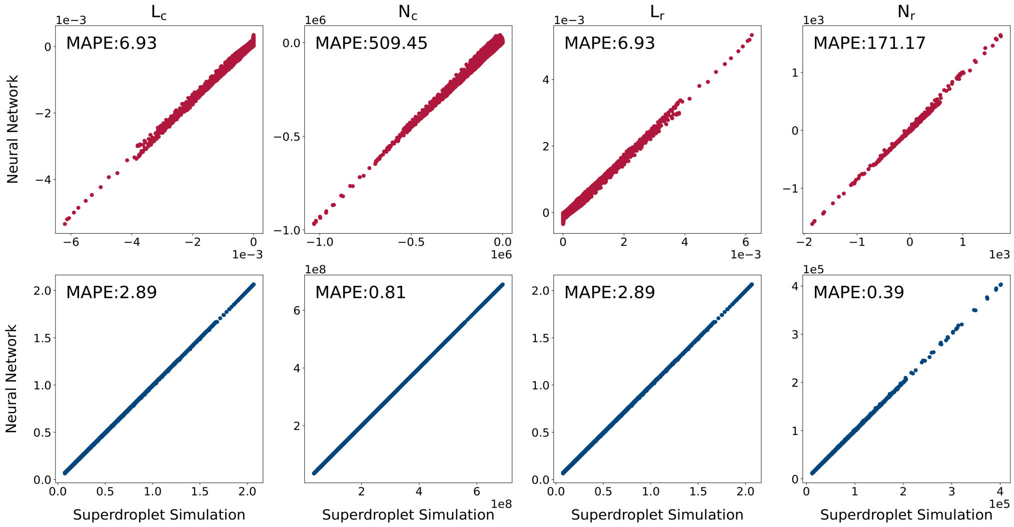


Figure A.3: **Upper row** changes over single time steps ($\Delta t = 20s$) in bulk moments derived from superdroplet simulations (x-axes) vs. change predicted by a network trained to predict one step into the future (y-axes). **Lower row** values of bulk moments predicted one step ahead by the same network (y-axes) vs. actual superdroplet-derived bulk moments one step ahead. Results are shown only for held-out testing data. Mean absolute percentage errors (MAPE) are reported.

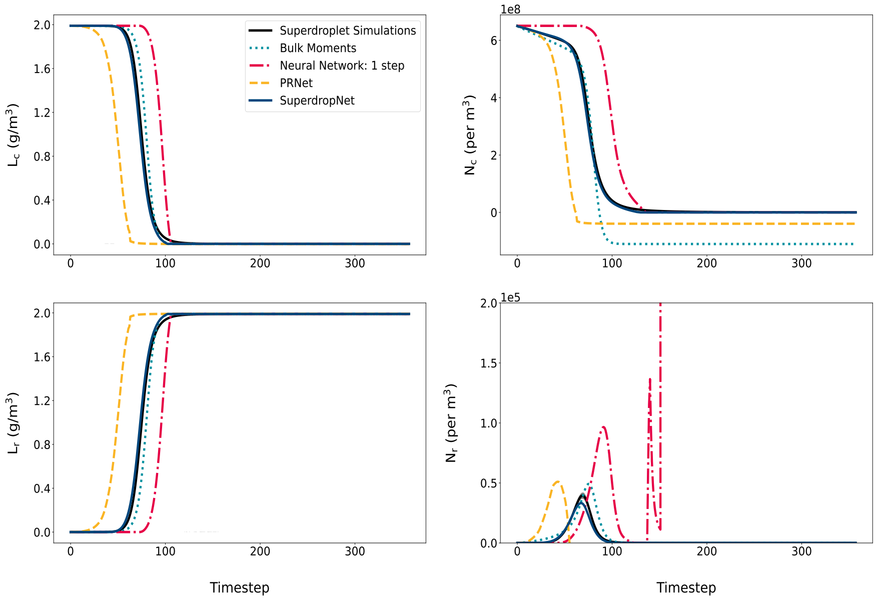


Figure A.4: Superdroplet-derived bulk moments (black lines) compared to rollouts from a neural network trained to predict 1 step into the future (red-dashed lines), SuperdropNet (blue solid line), PRNet (yellow-dashed lines) and from a classical bulk moment scheme (blue-dotted lines). Results are shown for a simulation with $L_0 = 2$ g/m^3, $r_0 = 9\mu$m, $\nu = 0$. Shaded region indicates +/- 1 standard deviation over 100 superdroplet simulations. A single time step corresponds to 20 s of simulation time.

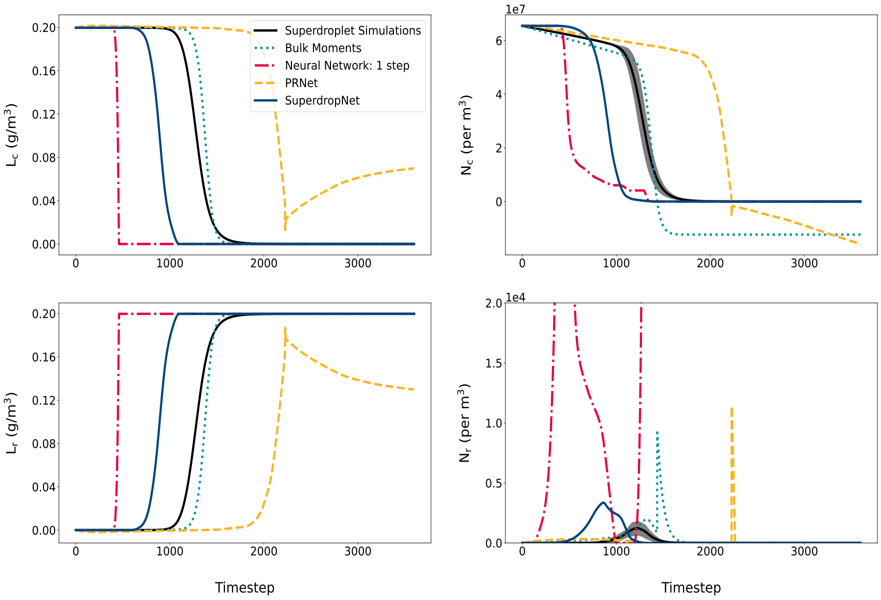


Figure A.5: Superdroplet-derived bulk moments (black lines) compared to rollouts from a neural network trained to predict 1 step into the future (red-dashed lines), SuperdropNet (blue solid line), PRNet (yellow-dashed lines) and from a classical bulk moment scheme (blue-dotted lines). Results are shown for a simulation with $L_0 = 0.2$ g/m^3, $r_0 = 9\mu$m, $\nu = 2$. Shaded region indicates +/- 1 standard deviation over 100 superdroplet simulations. A single time step corresponds to 20 s of simulation time.

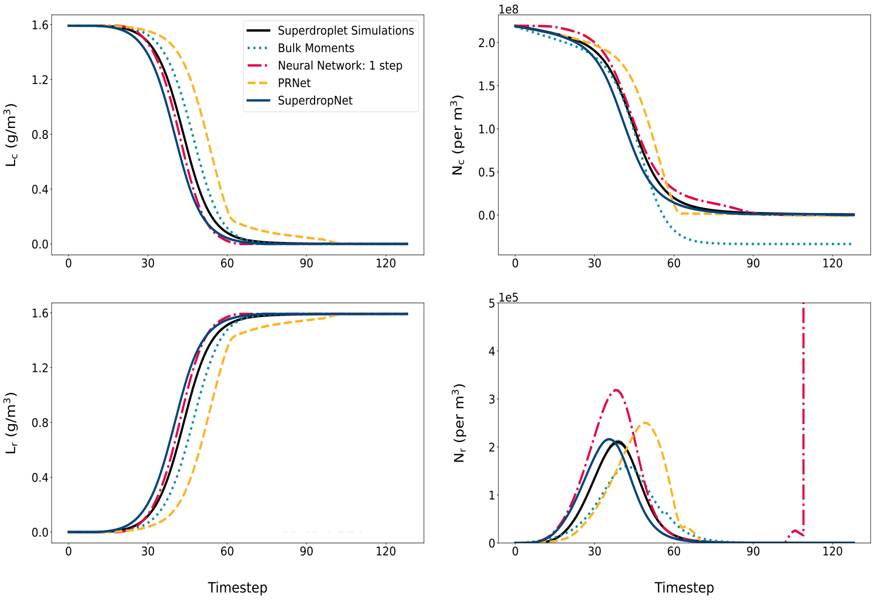


Figure A.6: Superdroplet-derived bulk moments (black lines) compared to rollouts from a neural network trained to predict 1 step into the future (red-dashed lines), SuperdropNet (blue solid line), PRNet (yellow-dashed lines) and from a classical bulk moment scheme (blue-dotted lines). Results are shown for a simulation with $L_0 = 1.6\ g/m^3$, $r_0 = 12\mu m$, $\nu = 0$. Shaded region indicates +/- 1 standard deviation over 100 superdroplet simulations. A single time step corresponds to 20 s of simulation time.

the neural network) and superdroplet-derived bulk moment tendencies as well as between the predicted moments and the superdroplet-derived bulk moments. MAPE between the predictions P, of the actual values A, is given by:

$$\mathrm{MAPE}(A,P) = \frac{100}{N}\sum_{i=0}^{N}\frac{A_i - P_i}{A_i} \tag{16}$$

where N is the total number of samples. Bulk moments at the next time step were inferred with mean absolute errors ranging from 0-3% (Fig. A.3-bottom panel). Since L_r and L_c sum to L_0, their mean absolute percentage errors (MAPEs) are comparable, while MAPE is lower for droplet number concentrations. These results confirmed that our network architecture can closely predict how bulk moments will evolve over a single time step.

We next examined whether the 1-step network could predict bulk moments further into the future. Starting from the initial conditions of each simulation in our dataset, we iteratively applied the network to generate rollouts over the full course of the simulation. For example, in a simulation with high water content and an initial exponential droplet size distribution ($\nu = 0$, Fig. A.4) the 1-step network (red-dashed line) roughly reproduced the bulk moments dynamics of superdroplet simulations (black line) except for N_r, where it diverged halfway through the simulation.

We also examined a more challenging case (Fig. A.5), where low initial water content necessitated a longer rollout and $\nu = 2$ produced a 'wider' droplet size distribution. In this case, the 1-step network converted cloud to rain faster than superdroplet simulations. In a third case with intermediate water content, higher initial droplet radius and $\nu = 0$ (Fig. A.6), the network matched superdroplet simulations more closely than the bulk moment parameterization for cloud and rain water content and cloud droplet concentration. For all three simulations (Fig. A.4-A.6), the 1-step network failed to produce stable and accurate predictions of rain droplet concentration.

A.6.2 *Accuracy of Multistep Training*

Given that a network trained 'offline' ($k = 1$, eq. 15) could accurately predict the next time step, but not long-term evolution of moments, we increased rollout length during training to a maximum value of $k = 25$. In general, we expected to observe an improvement in performance for predictions many time steps in the future, along with at least some degradation for single time step predictions as there will no longer be the only contributing factor our loss function. In fact, MAPE for tendencies over single time steps decreased for all moments except for N_r, for which it increased from 171.17 to 242.61 (Fig. A.7-top panel). This did not strongly impact prediction errors for bulk

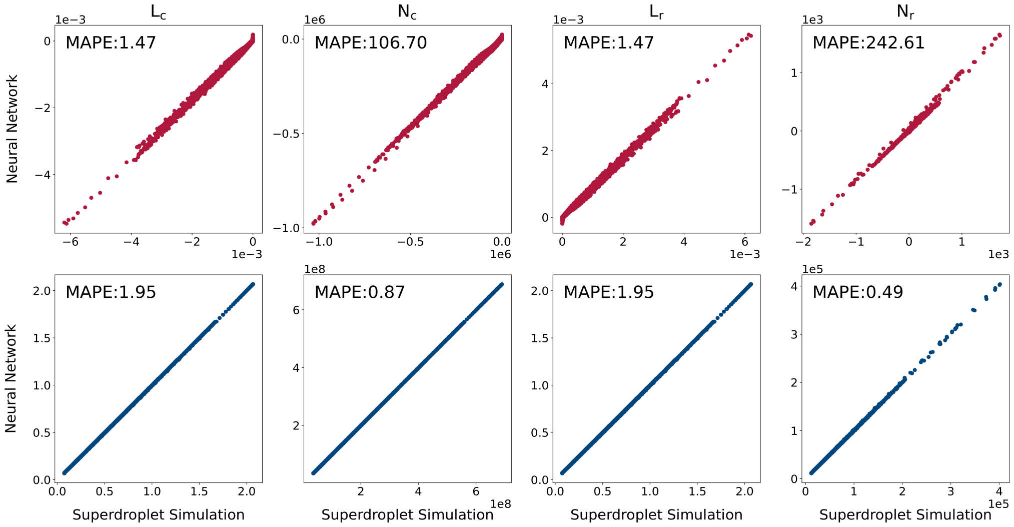


Figure A.7: Top panels, red – Superdroplet-derived changes in bulk moments over single time steps ($\Delta t = 20s$) vs. changes over single time steps predicted by SuperdropNet, a neural network trained to predict 25 time steps into the future. Bottom panels, blue – Superdroplet-derived bulk moments vs. predictions on time step ahead by SuperdropNet. Results are shown only for initial conditions in the held-out testing data. Mean absolute percentage errors (MAPE) are shown for each comparison.

moments one time step ahead (Fig. A.7-bottom panel). MAPE for bulk moments increased slightly with $k = 25$ for N_c (0.81 to 0.87) and N_r (0.39 to 0.49) compared to the network trained with $k = 1$ (Fig. A.3). For the mass moments, MAPE decreased from 2.89 to 1.95 when the rollout length was increased to 25. Given that multistep training did not lead to major or consistent degradation of the results for single time step predictions, we next examined its effects on longer rollouts.

A.6.3 *Rollout Lengths in Training and Evaluation*

The optimal rollout length for the training was not obvious: too low a k value could reduce accuracy for longer prediction horizons, while too high could make optimization unstable or reduce accuracy for shorter horizons. To investigate this further, we calculated accuracy as a function of both k and the prediction horizon (Fig. A.8). We calculated the mean MAE over normalized values of the four moments for all forecast horizons, from 1 to 20 steps, both on training/validation data (Fig. A.8-left panel) and on held-out testing data (Fig. A.8-right panel).

For $10 \leqslant k \leqslant 20$ changes were smaller and less consistent, while for $21 \leqslant k \leqslant 25$ we again observed reduction of errors by several

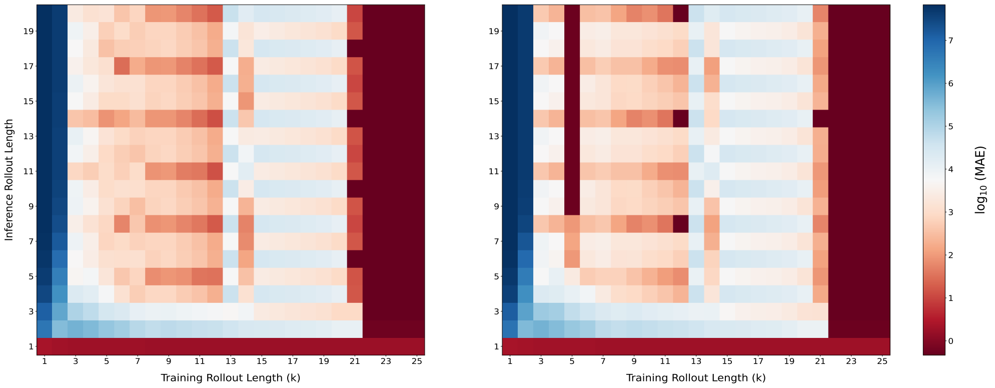


Figure A.8: Log_{10} MAEs corresponding to model training steps and inference steps. The left panel includes all 719 training simulations and the right panel includes all 100 testing simulations.

orders of magnitude for all prediction horizons with the exception of single time steps. This reduction of errors for prediction horizons longer than the training rollout length k was only observed when using the pushforward trick ($\lambda_j = 0, \forall j < k$, sec. A.5.3), and not when setting all $\lambda_j = 1$. Surprisingly, for $k > 21$ we achieved better predictions for 2 or more time steps into the future than for single time steps, suggesting the network is correcting its own errors. Overall, accuracy on training/validation data strongly resembled accuracy on test data, suggesting that overfitting is negligible for this combination of network architecture, data and training procedure, and that our trained network can generalize to initial conditions not observed during training. In light of these results, we used the network trained with $k = 25$ for all subsequent analyses, and termed it SuperdropNet. The network's ability to correct it's own errors can be seen clearly in Fig. A.9. While the prediction errors for the 1-step trained network (blue line) accumulate with longer rollouts, SuperdropNet's errors decrease over time (Fig. A.9-b).

A.6.4 *Rollout Accuracy Depends on Droplet Distribution Shape and Water Content*

We further investigated how SuperdropNet's accuracy over long rollouts depended on the conditions of the simulated warm rain process. Fig. A.4 -A.6 show SuperdropNet rollouts (dark-blue solid lines) over the full length of 3 simulations, starting from initial conditions. We also show rollouts for PRNet, which is the ML model developed to predict the process rates as in (Seifert & Rasp, 2020). SuperdropNet produced stable output in each case which match the superdroplet simulation (black lines) better than the 1-step network (red-dashed lines), but with varying accuracy.

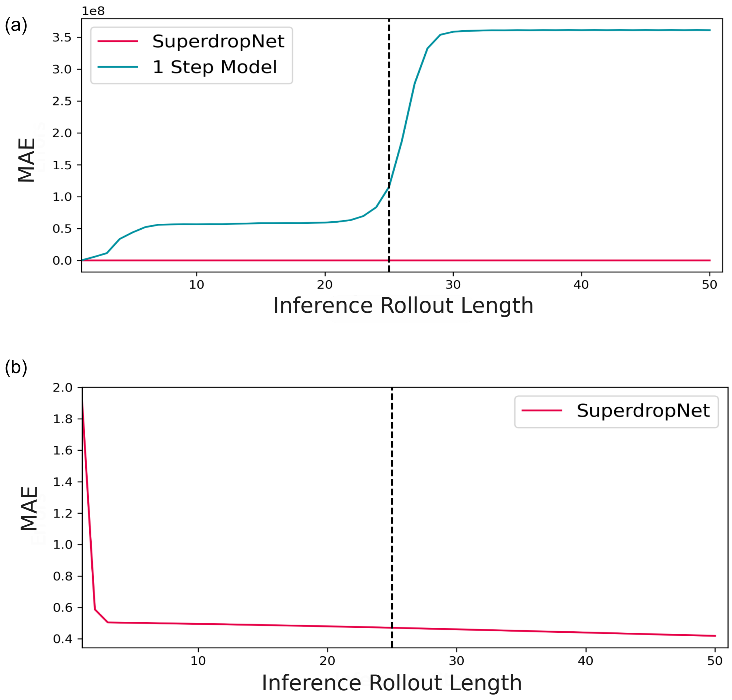


Figure A.9: Time evolution of errors as a function of rollout length during inference. (a) Mean absolute error as a function of rollout length using model trained to predict only one time step ahead (blue) vs. SuperdropNet. Errors are computed by averaging over 100 rollouts starting from randomly selected simulations and time points in the test set. The dashed black line shows SuperdropNet's training rollout length ($k = 25$). (b) As in 'a,' but showing only SuperdropNet.

Fig. A.4 shows a scenario with high water content, in which SuperdropNet predictions match the superdroplet simulations better than the bulk moment scheme (dotted blue lines) while PRNet (yellow-dotted lines) converts clouds to rain faster than in superdroplet simulations. In Fig. A.5, a simulation with low water content and a higher shape parameter, SuperdropNet predictions show a significant improvement over the 1-step network's predictions. SuperdropNet and the 1-step network convert the cloud droplets to rain faster than superdroplet simulations, while the bulk moment scheme is a more accurate match. However, the bulk moment scheme overestimates rain droplet concentrations, while SuperdropNet does not. In Fig. A.6, the 1-step network matches the superdroplet simulations better than SuperdropNet for the mass moments but SuperdropNet is a better match for the droplet concentrations. While the bulk moment scheme converts the cloud water to rain water faster than the superdroplet simulations, the bulk moment scheme takes longer than the superdroplet simulations. This is a case with relatively higher water content and a low value of shape parameter. PRNet overestimates the conversion time of cloud to rain in Fig. A.5 and A.6 .

We observe a general pattern of SuperdropNet struggling with cases where the initial water content is low and the distribution of droplet sizes is wider (higher values of ν). The PRNet model (Seifert & Rasp, 2020) also struggled on simulations with a low water content. In general the bulk moment scheme closely predicts L_c and L_r but struggles with N_r.

A.6.5 *Comparison of* t_{10} *and Mean Absolute Errors*

An important test of any representation of warm rain coalescence is whether it accurately captures the timing of cloud-to-rain transitions. We therefore evaluated the match between SuperdropNet and alternative methods in the time t_{10} at which 10% of total water mass had converted to rain. In order to obtain a full picture over all initial conditions, and as previous analyses showed a lack of overfitting (Fig. A.8), we conducted this analysis on all 819 simulations.

True t_{10} values showed overall agreement with t_{10} values computed from SuperdropNet rollouts (Fig. A.10, blue, MAE=14.83 g/m^3) but tended to underestimate transition times. The classical bulk moment scheme exhibited a lower MAE (Fig.A.10-(a), red, MAE = 3.97 g/m^3) but tended to overestimate transition times. PRNet underestimated many t_{10} values to a greater extent than SuperdropNet (Fig. A.10, yellow, MAE=71.95 g/m^3), but also frequently failed to reach 10% mass conversion before the end of the simulation (shown as negative values). For these simulations, the MAE was calculated by assuming the end of the simulation as the t_{10} value.

We further examined the accuracy with which the timing of droplet number dynamics were represented, by calculating the time t_{10} at which cloud droplet count had decreased by 10% of its initial value. Here we observed a closer match to superdroplet simulation for SuperdropNet than for the bulk moment parameterization (Fig. A.10-(b)). We also observe an overall lower MAE for SuperdropNet than the bulk moment scheme and PRNet (Fig. A.10-(c)).

Given our finding that the accuracy of all schemes' bulk moment predictions depended on the simulation's initial conditions, we further examined how these initial conditions impacted the accuracy of t_{10} values based on L_c and N_c (Fig. A.11). Consistent with our previous findings, SuperdropNet and PRnet performed best at higher water content. SuperdropNet's predictions also improved at higher initial droplet sizes (Fig. A.11-(b),(e)) and lower values of the shape parameter (Fig. A.11-(c-f)). For PRNet, the shape parameter affected the accuracy of prediction the most with an increased accuracy at higher values of ν (Fig. A.11-(c-f)). The bulk moment scheme's timing accuracy was largely unaffected by L_0 and r_0, but higher values of ν increased the magnitude of errors (Fig. A.11-(c-f)). Superdrop-

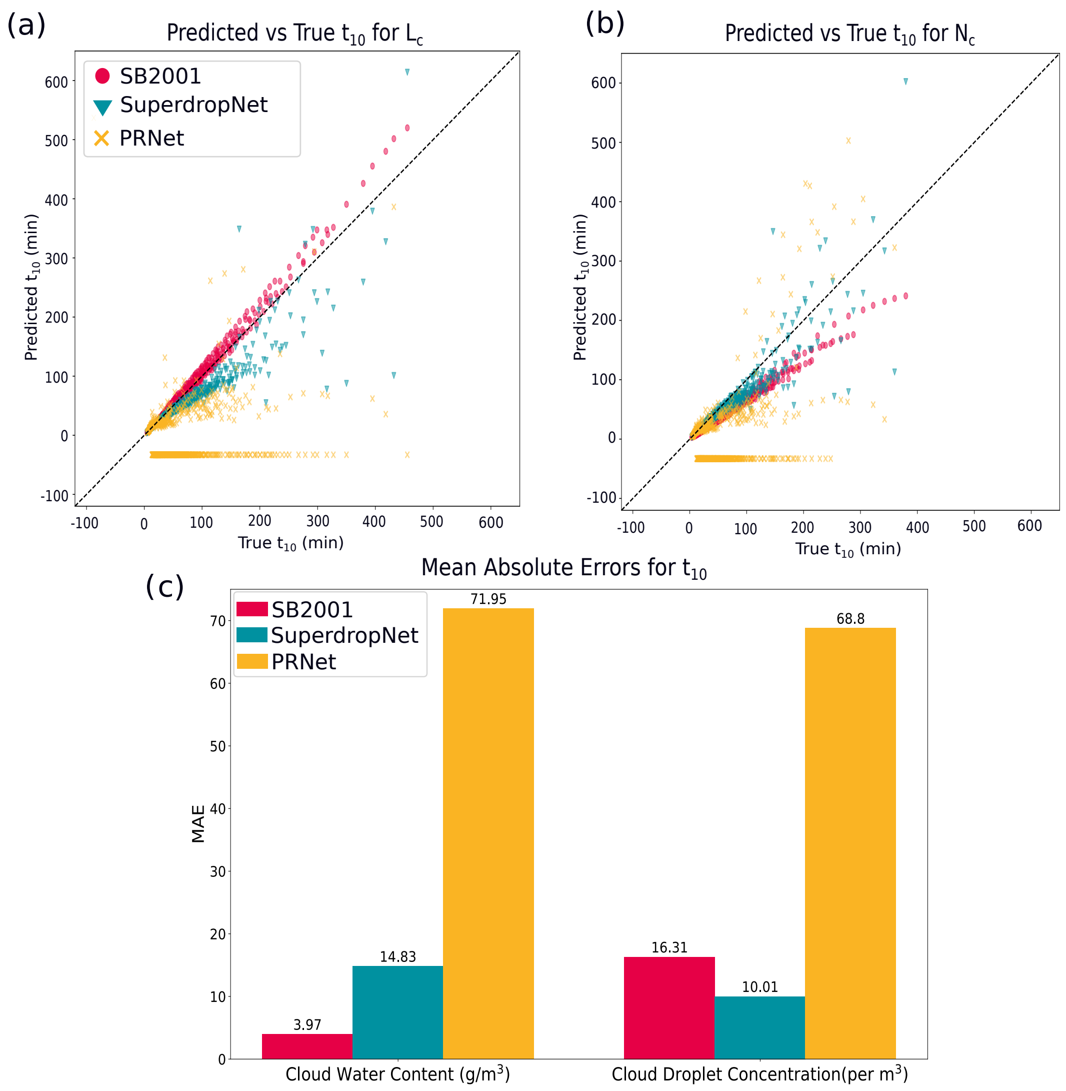


Figure A.10: The t_{10} values for L_c (a) and (b) N_c as calculated for 819 simulations using SB2001, SuperdropNet and the ML model from (Seifert & Rasp, 2020) based on estimating process rates, referred to as PRNet. Each point is one of the 819 simulations. True t_{10} on the x axis corresponds to the values from the superdroplet simulations. The negative values on the y-axis represent simulations for which the 10% of initial water never converted to rain during the length of the simulation. (c) Mean Absolute Errors (MAE) in approximation of t_{10} values for cloud water content (L_c) and cloud number Concentration (N_c)

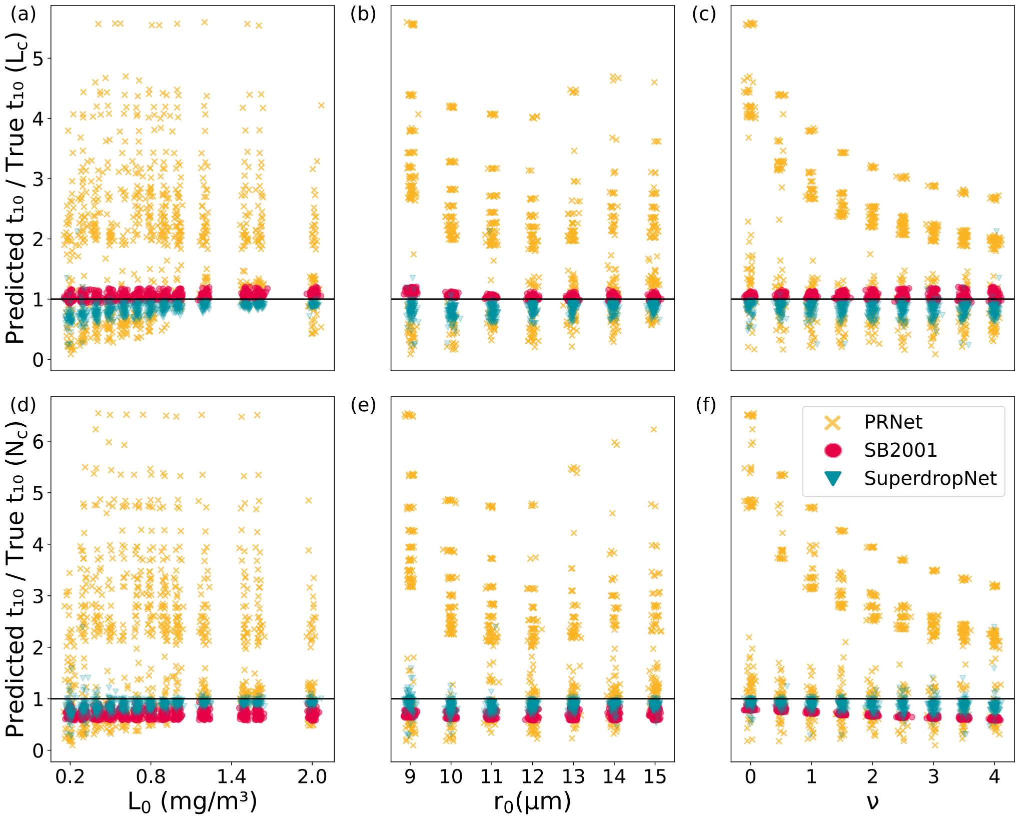


Figure A.11: (a)-(c) Ratio of predicted/true t_{10} values for L_c as a function of initial conditions of the superdroplet simulation. Here 'true' refers to the superdroplet simulation. Each point is one of the 819 simulations. (d)-(f) same as (a)-(c) but for t_{10} predictions for N_c.

Net yielded more accurate transition times for N_c across all initial conditions compared to the bulk moment scheme.

Together, these results show that SuperdropNet improves significantly upon the state of the art in ML-based parameterization of warm rain microphysics. SuperdropNet closes most of the existing performance gap between previous ML-based representations and classical bulk moment schemes, and exceeds the accuracy of those classical schemes in some cases.

A.7 DISCUSSION

We developed SuperdropNet, a neural network emulator of the warm rain formation process, and trained it on data from stochastic box-model simulations. SuperdropNet significantly advances the state of the art for ML-based emulation of warm-rain processes. For predicting the timing of cloud-to-rain transitions in droplet-based simulations, it closed most of the performance gap between neural networks and classical bulk moment parameterizations, and in some cases exceeded the accuracy of classical approaches. This performance gain can be attributed to several novel aspects of our approach: multistep rollouts, the pushforward trick, measurement and averaging out of stochasticity, and discarding the approximations used to link process rates for mass and droplet count. The striking improvements we observed with multistep training (Fig. A.8) can be considered a case of properly aligning our objective function with the ultimate task we wish to accomplish: emulating the warm rain dynamics accurately over many time steps.

We found SuperdropNet's accuracy to depend strongly on the conditions of the warm rain process, with better performance for high water content and a less-dispersed droplet distribution. In general these conditions make for an easier prediction task, as they lead to shorter but smoother cloud-to-rain transitions. On the whole, we found that SuperdropNet tends to underestimate transition times calculated using rain vs. cloud mass, but predicts transitions of droplet counts more accurately than a classical bulk moment scheme. In case of shallow convection where the t_{10} is usually less than 60 min, SuperdropNet estimates both cloud water mass and droplet concentrations accurately (Fig. A.10). Many simulations with extremely low initial water content would never lead to formation of rain droplets in a real atmosphere due to the effects of evaporation.

Particle-based schemes allow for better approximation to the stochastic collection equation by estimating pure stochastic growth (Unterstrasser et al., 2017). These properties can be leveraged to improve representation of processes such as sedimentation where the bulk moment schemes are known to introduce numerical discontinuities (Seifert & Beheng, 2006; Wacker & Seifert, 2001). Additionally, initial successes in a warm rain scenario with only rain and cloud open the possibility of emulating more complex microphysical phenomena involving additional hydrometeors such as ice, snow and graupel, as proposed in a previous study (Seifert & Rasp, 2020).

We further plan to couple SuperdropNet to atmospheric fluid dynamics and other physical processes. This will require solving the technical challenge of bidirectional communication between Python/Pytorch-based deep learning and FORTRAN based atmospheric simulation, as well as developing new training algorithms that encourage stability and accuracy of the coupled dynamics. We believe that fast, accurate microphysics emulators that can be used as modular simulation components could offer significant benefits in predicting and understanding weather, climate and air quality phenomena.

A.8 OPEN RESEARCH

The code for training and developing SuperdropNet can be found at https://github.com/m-dml/warm-rain-emulator. The training data is available at: https://zenodo.org/records/10054101. The data was generated using McSnow which is a lagrangian cloud microphysics model. It is part of the ICON modelling framework which is now available under BSD-3-C license at https://www.icon-model.org/. Access to McSnow can be granted by the developers (Brdar & Seifert, 2018) upon request.

ACKNOWLEDGMENTS

We thank Axel Seifert for providing access to McSnow which was used for generating the training data. This work was funded by Helmholtz Association's Initiative and Networking Fund through Helmholtz AI. We also thank Ann-Kristin Naumann for insightful scientific discussions.

B

EFFICIENT AND STABLE COUPLING OF THE SUPERDROPNET DEEP-LEARNING-BASED CLOUD MICROPHYSICS (V0.1.0) WITH THE ICON CLIMATE AND WEATHER MODEL (V2.6.5)

This work has been published as:

Arnold, C., Sharma, S., Weigel, T., & Greenberg, D. S. (2024). Efficient and stable coupling of the SuperdropNet deep-learning-based cloud microphysics (v0.1.0) with the ICON climate and weather model (v2.6.5). *Geoscientific Model Development*, 17(9), 4017–4029. https://doi.org/10.5194/gmd-17-4017-2024

AUTHOR CONTRIBUTIONS

C. Arnold developed the embedded Python and YAC coupling software, performed the experiments, provided the visualizations, and led the writing of the paper as a whole. I developed SuperdropNet. C. Arnold and I defined and evaluated the experiments, curated the software and data, and wrote the original draft. T. Weigel developed the pipe coupling software and contributed to the original draft. D. Greenberg helped conceive the project and define the coupling task and contributed to the original draft. All authors reviewed and edited the final paper

EFFICIENT AND STABLE COUPLING OF THE SUPERDROPNET DEEP LEARNING-BASED CLOUD MICROPHYSICS (V0.1.0) TO THE ICON CLIMATE AND WEATHER MODEL (V2.6.5)

Authors:
Caroline Arnold[1,2,3,*], Shivani Sharma[1,2,4,*]
Tobias Weigel[1,2,3], David S. Greenberg[2,3]
Affiliations:
[1]German Climate Computing Center DKRZ, Hamburg, Germany
[2]Helmholtz-Zentrum Hereon, Institute of Coastal Systems,
Model-Driven Machine Learning, Geesthacht, Germnay
[3]Helmholtz AI
[4]International Max Planck Research School on Earth System
Modeling, Hamburg, Germany

[*]**Corresponding authors:**
Caroline Arnold, arnold@dkrz.de
Shivani Sharma, shivani.sharma@hereon.de

ABSTRACT

Machine learning (ML) algorithms can be used in Earth System models (ESMs) to emulate sub-grid-scale processes. Due to the statistical nature of ML algorithms and the high complexity of ESMs, these hybrid ML-ESMs require careful validation. Simulation stability needs to be monitored in fully coupled simulations, and the plausibility of results needs to be evaluated in suitable experiments.

We present the coupling of SuperdropNet, a machine learning model for emulating warm rain processes in cloud microphysics, into ICON (Icosahedral Nonhydrostatic) 2.6.5. SuperdropNet is trained on computationally expensive droplet based simulations and can serve as an inexpensive proxy within weather prediction models. SuperdropNet emulates the collision-coalescence of rain and cloud droplets in a warm rain scenario and replaces the collision-coalescence process in the two-moment cloud microphysics scheme. We address the technical challenge of integrating SuperdropNet, developed in Python and PyTorch, into ICON, written in Fortran, by implementing three different coupling strategies: embedded Python via the C Foreign Function Interface, pipes, and coupling of program components via YetAnotherCoupler (YAC). We validate the emulator in the warm bubble scenario and find that SuperdropNet runs stable within the experiment.

In comparing experiment outcomes from the bulk moment scheme and SuperdropNet, we find that the results are physically consistent, and discuss differences that are observed for several diagnostic variables.

In addition, we provide a quantitative and qualitative computational benchmark for three different coupling strategies—embedded Python, coupler YAC, and pipes—and find that embedded Python is a useful software tool for validating hybrid ML-ESMs.

B.1 INTRODUCTION

Machine learning (ML) is increasingly used in Earth system models (ESMs) to emulate sub-grid-scale processes that are typically parameterized or neglected due to their high computational cost (Christensen & Zanna, 2022; Dueben et al., 2021; Gentine et al., 2018; Irrgang et al., 2021). ML algorithms are statistical algorithms that are trained on data. Neural networks are a widely used class of ML algorithms. They contain trainable parameters, the weights and biases, that are learned from data by minimizing a cost function. The trained algorithm can then be used for inference, i.e. application on unseen data of the same kind. When sub-grid-scale processes are replaced by ML algorithms, the improvement can aim at speeding up the overall simulation by emulating the existing parameterization. This was first established by using neural networks to emulate long-wave radiative transfer (Chevallier et al., 2000; Krasnopolsky et al., 2005). Recent examples include the emulation of gravitational wave drag (Chantry et al., 2021), cloud microphysics (Brenowitz et al., 2022), the ocean in a coupled climate model (Sonnewald et al., 2021), and cloud radiative effects (Meyer et al., 2022).

Other studies aim to improve the overall description of the Earth system by providing a better parameterization. ML algorithms can be trained on high-resolution ESM output or even on separately simulated processes to emulate resolved processes in a low-resolution simulation, e.g. for gravity waves (Dong et al., 2023), cloud cover parameterizations (Grundner et al., 2022), general parameterizations (Brenowitz & Bretherton, 2018), sub-grid-scale momentum transport (Yuval & O’Gorman, 2023), effects of cloud resolving simulations (Rasp et al., 2018), ozone distributions (Nowack et al., 2018), and radiative transfer (Belochitski & Krasnopolsky, 2021b).

Many parameterizations in ESMs can be removed at higher resolutions if the process can be completely resolved, such as the convective parameterizations. On the other hand, some others would need to be parameterized even for 1-km scale weather models. Cloud microphysical processes fall in this category. Processes dealing with the droplet interactions that lead to precipitation are lumped together and referred to as cloud microphysical processes. Due to high particle counts even at small grid sizes and our incomplete understanding of processes that occur at a molecular level in clouds (Morrison et al., 2020), we cannot expect cloud microphysical parameterizations to become obsolete in the near future for high resolution models.

The parameterization of these processes suffers from a unique accuracy/speed trade-off. The most accurate droplet based Lagrangian schemes such as the superdroplet method (Shima et al., 2009) are computationally expensive. The commonly used bulk moment schemes represent the complex particle size distributions as only the first two moments, referring to the total droplet concentration and the total water content of the hydrometeors. For modelling the droplet collisions in a warm-rain scenario, ICON uses the well studied bulk moment scheme developed in Seifert and Beheng, 2001. To bridge this gap and to make the use of more complex microphysical schemes feasible within operational models, a data-driven approach can be employed. We present here the integration of SuperdropNet (Sharma & Greenberg, 2024), an ML algorithm for emulating warm rain processes in cloud microphysics, into ICON 2.6.5. SuperdropNet is trained on zero dimensional box model superdroplet simulations from McSnow 1.1.0 (Brdar & Seifert, 2018), a superdroplet based cloud microphysics model,

in a warm rain scenario and replaces the warm rain processes in the two-moment scheme available in ICON 2.6.5 (Seifert & Beheng, 2006).

Due to the statistical nature of ML algorithms and the complex nonlinear interactions in ESMs, hybrid systems of numerical ESMs and ML algorithms require careful validation and verification (Brenowitz & Bretherton, 2019; Dueben et al., 2022). Stand-alone ML algorithms are first trained on a dataset and then validated on a hold-out test dataset that is not seen during training. This test set is within the distribution of the training data. When an ML algorithm is coupled to an ESM, it may encounter conditions outside of the range of the training data, and the required extrapolation could lead to instabilities (Yuval et al., 2021). Thus, the so-called *offline* performance of an ML algorithm is often not a good indicator of its *online* performance (Brenowitz et al., 2020c; Rasp, 2020). Stability is a major concern when introducing ML emulators into ESMs. It can be improved by adapting the training procedure (Brenowitz & Bretherton, 2018; Brenowitz et al., 2020b; Qu & Shi, 2023; Rasp, 2020) or by fulfilling physical constraints in the network architecture (Beucler et al., 2021; Yuval et al., 2021). Careful validation setups can help the scientific community to build trust in so-called black box ML algorithms (McGovern et al., 2019).

To avoid devoting resources to the development of ML algorithms that fail in contact with reality, we encourage incorporating online testing at an early stage. ML algorithms are developed iteratively, and new versions should be tested quickly in their final place of application in the Earth system model.

The popular software libraries for ML algorithm development, such as PyTorch (Paszke et al., 2019), Keras (Chollet et al., 2023), or Tensorflow (Abadi et al., 2016), are based on the Python language. On the other hand, ICON is written in Fortran. Online testing requires either rewriting the ML emulator in Fortran, or integrating the two programming languages with one another (Brenowitz & Bretherton, 2019). Since ML algorithm development is an iterative process, frequent rewrites of the ML algorithm would be required in the former case. In order to save developer resources, we recommend coupling Python and Fortran at least during the stage of algorithm development.

In Sect. B.2.1 we introduce the warm bubble scenario, which serves as a test case for SuperdropNet. The ML algorithm itself is described in B.2.3. Different strategies for integrating SuperdropNet into ICON are discussed in Sect. B.3. The results and the impact of SuperdropNet on atmospheric processes and prognostic variables are presented in Sect. B.4.2A computational and qualitative benchmark of three different strategies is included in Sect. B.4.3.

B.2 METHODS

B.2.1 *Warm bubble scenario*

We validate SuperdropNet in the warm bubble scenario, a test case for cloud microphysics available in ICON 2.6.5. It describes an atmosphere temperature profile with a warm air bubble at the bottom that rises vertically. The test case operates on a torus grid. This grid is created by a domain of 22×20 cells where periodic boundary conditions are applied in x and y direction. The horizontal resolution is 5 km, and there are 70 vertical levels in z direction. The simulation time step is 20 s with a total simulation time of 120 min. The experiment is computationally lightweight and runs on a single compute node. We test SuperdropNet in a warm atmosphere with no ice particle formation, as well as in a mixed-phase and a cold atmosphere that both allow ice formation. All simulation parameters are summarized in Table 2. We transport the tracers required for two-moment cloud microphysics, i.e. first and second moment of the hydrometeors cloud water, cloud ice, rain, snow, graupel, and hail.

B.2.2 *Bulk moment scheme for cloud microphysics*

In our test case, a two-moment bulk scheme is employed to compute the number concentration and total mass for all hydrometeors involved. In ICON, the bulk moment scheme used for warm

Parameter	Description	Warm bubble	Mixed-phase bubble	Cold bubble
L_D	Torus domain length	5000 m		
t_{dyn}	Dynamical time step	20 s		
t_{2mom}	Two-moment scheme time step	20 s		
z_{lev}	Atmospheric levels	70		
p_{srfc}	Surface pressure	1013.25 hPa		
T_0	Cold point of atmosphere	303.15 K	273.15 K	268.15 K
γ_0	Vertical temperature lapse rate	0.006 K/m		0.009 K/m
z_0	Altitude up to which γ_0 applies	3000 m		4000 m
γ_1	Lapse rate above z_0	0.00001 K/m		0.0001 K/m
$T_{perturb}$	Temperature perturbation	10 K		5 K
ϕ_{bg}	Background relative humidity	0.7		
ϕ_{mx}	Maximum relative humidity	0.9		0.95
ξ	Half-width of temperature perturbation in x	12500 m		
ζ	Half-width of temperature perturbation in z	200 m		250 m
x_0	Center of temperature perturbation in x	0 m		

Table 2: Experiment parameters for the warm bubble, mixed-phase bubble, and the cold bubble test case. Note that t_{dyn} and t_{2mom} reflect the time step used for training SuperdropNet.

rain cloud microphysics is based on Seifert and Beheng, 2001. To account for collision-coalescence, the number concentration and total mass for both cloud and rain are determined by calculating the rates of collision-coalescence processes, including autoconversion, accretion, and self-collection. Here autoconversion refers to the process by which cloud droplets coalesce to form rain droplets while accretion accounts for collisions between rain and cloud droplets. Self-collection rates for cloud and rain droplets account for collisions that do not convert cloud droplets into rain. These process rates rely solely on the droplets themselves and are subsequently utilized to update the bulk moments for the following time step using a set of ordinary differential equations.

B.2.3 *SuperdropNet cloud microphysics model*

SuperdropNet is a machine learning emulator for superdroplet simulations in a warm rain scenario. It is a neural network consisting of fully connected layers and is trained to predict updates of the bulk moments for cloud and rain over different droplet size distributions. SuperdropNet is detailed in (Sharma & Greenberg, 2024); therefore, we will provide only a brief summary of the training procedure here.

The superdroplet simulations used for training are generated with McSnow (Brdar & Seifert, 2018). In (Brdar & Seifert, 2018) McSnow was used for simulating ice particles, while in (Seifert & Rasp, 2020) it was simulating a warm rain scenario. Similar to (Seifert & Rasp, 2020), the training data for SuperdropNet is generated in a warm rain scenario that describes only the conversion of cloud droplets into rain in a dimensionless control volume. As superdroplet simulations are stochastic in nature, we use multiple realizations of simulations to train SuperdropNet. Hence, given a set of initial conditions, SuperdropNet is completely deterministic in nature and the bulk moments estimated by it are the equivalent of averaged superdroplet simulations (Sharma & Greenberg, 2024). The microphysical processes accounted for are accretion, autoconversion and self-collection of rain and cloud droplets. In ICON, the droplet collisions corresponding to warm rain processes are treated in a separate module where the process rates for accretion, autoconversion, and self-collection of rain and cloud droplets are calculated. The parameterization scheme is localized, i.e the process rates calculated for a grid cell depend only on the rain and cloud moments corresponding to that grid cell. Other microphysical processes and the vertical transport are accounted for in separate modules, which implies that the parameterization in ICON is structured such that all individual grid points can be considered zero-dimensional boxes. Thus, the parameterization setup for droplet collisions in ICON mimics the training data for SuperdropNet. This justifies the choice of using a test scenario in ICON for online coupling and testing of SuperdropNet.

Note that only the warm rain processes are replaced with SuperdropNet. In a cold atmosphere, SuperdropNet can still be coupled to ICON, but since warm rain processes are not relevant there, including SuperdropNet is expected not to change the experiment results.

B.2.4 *ICON program flow*

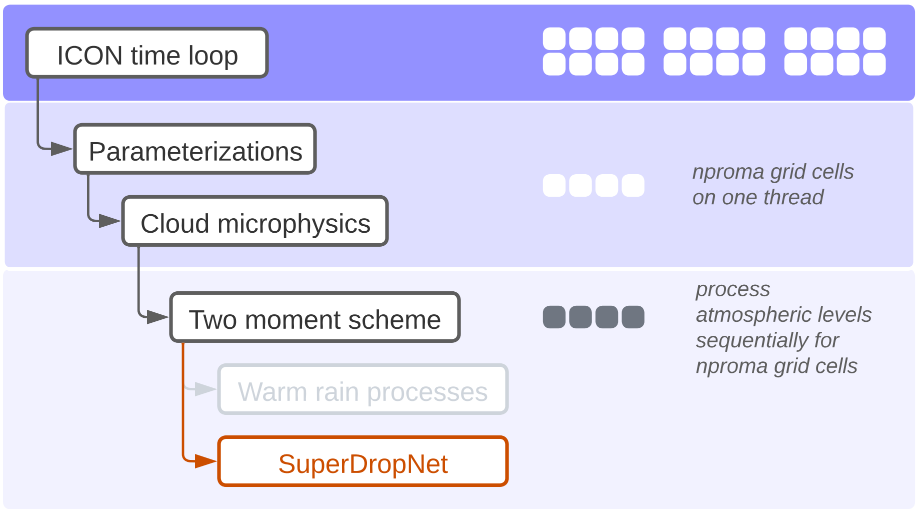


Figure B.1: We replace the warm rain processes (gray) with a call to SuperdropNet (orange). At this point, each thread has access to an *ik-slice*, a specific representation in the cloud microphysics parameterization that corresponds to one atmospheric level for one block of grid cells.

To illustrate at which point of program execution ML-ESM coupling becomes necessary, we show the flowchart for a single ICON time step in Fig. B.1, focusing only on the steps relevant to our application. Starting from the general ICON time loop, where the full grid information is available, we enter the cloud microphysics parameterization. At this point, a given thread has access to one block of grid cells with block length nproma, and all threads work in parallel. The two-moment scheme has its own grid representation, called *ik-slices*, where the block of grid cells is again divided by atmospheric levels. In our experiment, we simply replace the warm rain processes with a call to SuperdropNet, which provides updated moments for cloud and water droplets.

Since the call to the ML component is not at the grid level, but operates on *ik-slices* far down in the nested structure of the ICON program flow, we need to call SuperdropNet several times per time step – once for each block of grid cells and once for each atmospheric level. Note that saturation adjustments and evaporation are handled outside of the parts of the ICON code replaced by SuperdropNet.

B.3 INTEGRATING SUPERDROPNET IN ICON

There are several ways to integrate Python machine learning components into Fortran code (Partee et al., 2022). Based on a pre-selection of suitable methods, we have implemented three strategies, so-called Fortran-Python bridges. For convenience, we add a namelist to ICON that allows the selection of the coupling strategy. We perform the experiment with all three methods on the DKRZ Levante system. Levante is a BullSequana XH2000 supercomputer with 3042 compute nodes using the 3rd generation of AMD EPYC CPUs (Milan) with 128 cores per node, NVIDIA A100 GPUs, and a 130 Petabyte DDN filesystem. The nodes are connected to a Mellanox Infiniband HDR100 fabric.

B.3.1 *Embedding Python as a dynamic library*

Using the techniques in (Brenowitz, N., 2023), we develop a dynamic library based on Python code. The library is generated using the C Foreign Function Interface (CFFI) (Rigo & Fijalkowski, 2018) and is linked to ICON at compile time. At runtime, Python code is executed from the library. Employment of CFFI results in Python and Fortran sharing their address space, hence passing memory pointers is sufficient to access the same data. Jobs are run in a homogeneous setting, with Python code executed on the same CPU compute node as ICON.

B.3.2 *Using the coupling software YAC*

YetAnotherCoupler (YAC) (Hanke et al., 2023, 2016) is commonly used to couple different ICON components, e.g., atmosphere, ocean and I/O. YAC provides Python bindings so that external Python programs can be coupled with little effort to ICON.

YAC requires a definition of fields that are to be exchanged, and an exchange schedule that cannot be below the time step of ICON. For the warm-bubble scenario, we set the block length to the number of grid cells (880) and define two exchange fields per atmospheric level, one for the ICON-to-Python exchange, and one for the reverse exchange. This yields a total of 140 fields, that are exchanged at each time step. A smaller block length would require the developer to define more exchange fields, such that bulk moments in each grid cell can be exchanged at every time step.

Data transfer is building on Message Passing Interface (MPI) routines that are integrated in YAC. This offers the flexibility to use heterogeneous jobs, i.e., running ICON on CPU nodes and ML inference on GPU nodes. Due to current limitations of the scheduling software employed in the DKRZ Levante system, it was not possible to schedule simulations that span the CPU and the GPU partition of the system. Thus, we were not able to test the performance in a heterogeneous setting. With ICON shifting to GPUs, we foresee that in the future homogeneous jobs will be run on GPU nodes.

B.3.3 *Pipes*

We implemented a coupling between n ICON processes and one Python process running on the same node using FIFO (first-in-first-out) pipes. The first ICON MPI rank on the node will spawn a separate Python process that runs a worker script. Each rank also creates two pipes, one for each direction of communication (input and output to the Python worker). The worker iterates over all input pipes, performs the warm rain calculation on data being available and writes results back to the corresponding ICON process via its output pipe.

While this solution does not incur the potential overhead of using MPI to communicate locally, it is not a full shared memory solution relying on pointers exclusively. The corresponding extensions to ICON and the Python worker script are optimized to do as few memory copies as possible, though naturally some copying cannot be avoided when interacting with the pipes. As FIFO pipes only work on a local node, no cross-node setups are possible, such as running ICON and Python on different types of nodes (CPU, GPU). As the Python worker runs as a separate process on a dedicated core, the number of cores available to ICON is also marginally reduced by one.

B.3.4 *Other methods*

We note that the selection of methods in Sects. B.3.1–B.3.3 is by no means encompassing all the available tools and summarize here the alternatives to the best of our knowledge:

Four software libraries developed at ECMWF (Bonanni et al., 2022), the Cambridge Institute for Computing in Climate Science (Elafrou et al., 2023b), NVIDIA (Alexeev, D., 2023), and Tongji University (Mu et al., 2023) address ML inference directly by exposing the Tensorflow and Pytorch APIs for Fortran, respectively. This adds the benefit of not requiring a Python runtime environment at the time of execution. Since we require flexibility to use Python code beyond ML inference, and data exchange is done here via RAM comparable to the approach described in Sect. B.3.1, we did not investigate these libraries further.

During development, we noted that integrating SmartSim (Partee et al., 2022) would require a rewrite of the ICON startup routine that is beyond the scope of this project. On a similar note, the coupling routines developed in WRF-ML for the open source Weather Research and Forecasting (WRF) model cannot easily be adjusted to work with ICON (Zhong et al., 2023).

The Fortran-Keras bridge (Ott et al., 2020) allows for ML inference in Fortran based on ML algorithms developed in the Keras framework. This limits flexibility, since only those network layers and functionalities supported by the library can be used. On a similar note, the implementation of the ML algorithm in Neural Fortran (Curcic, 2019) is contingent on the library, and the Fortran InferenceEngine (Rouson et al., 2023) is restricted to feed-forward neural networks. We chose to forego these methods since we desire the flexibility to use any novel Pytorch developments without depending on their integration into an external library.

B.4 RESULTS

B.4.1 *Experiment description*

Using the three coupling techniques described in Sects. B.3.1–B.3.3, we integrate SuperdropNet in ICON. The experiment results are the same since the same network is called, but the impact on computational performance is different. We run the warm bubble scenario and the cold bubble scenario, both with a representation of warm rain processes using SuperdropNet, as well as using the existing bulk moment scheme in the two-moment cloud microphysics module.

We compare the effect of replacing warm rain processes with SuperdropNet on the experiment outcome in Sect. B.4.2. In Sect. B.4.3, we compare the impact on computational performance that is incurred by integrating SuperdropNet for all three coupling techniques.

B.4.2 *Comparison of the bulk moment scheme and SuperdropNet*

B.4.2.1 *Rain rates*

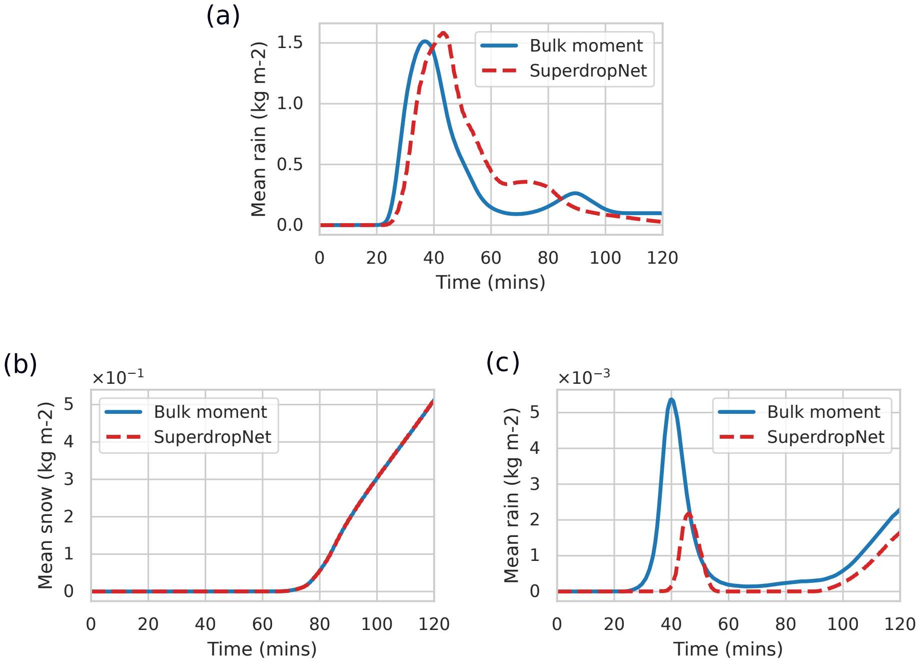


Figure B.2: Grid-averaged quantities for the bulk moment scheme and SuperdropNet under (a) warm bubble scenario, (b) cold bubble scenario and (c) mixed phases scenario.

Figure B.2a shows the grid-averaged rain rate in the warm bubble scenario deriving from warm rain processes using ICON's two-moment bulk cloud microphysics, with a comparison to SuperdropNet microphysics. Since SuperdropNet was trained on particle-based simulations that avoid certain statistical approximations of bulk moment schemes, we do not expect the rain rates in both scenarios to match. Due to the experimental setup, it is not possible to identify with certainty which model produces the more accurate rain rates. We do note, however, that SuperdropNet yields physically plausible rain rates. The rain rate obtained using SuperdropNet evolves in a predictable way, i.e, there is no rain at the beginning of the simulation, which eventually builds up to a peak and then slowly rescinds. At the end of the simulation, the rain rate is zero for both the simulations. No negative values are observed, and the coupling with SuperdropNet does not result in significant divergence of the simulation. This emphasizes that SuperdropNet is stable over longer simulation runs and overall behaves as a realistic ML based emulator for droplet collisions. One of the key differences in the evolution of the rain rate with the two different parameterizations is that the onset of rain is slightly delayed with SuperdropNet coupling which indicates a slower conversion of cloud droplets to rain droplets.

As a sanity check, we perform the cold bubble experiment using both the bulk moment scheme and SuperdropNet for the warm rain processes. In this scenario, warm rain processes are not relevant for the cloud microphysics, and we expect that including SuperdropNet does not affect processes with frozen particles. Figure B.2b shows the grid-averaged snow rate.

Both schemes show identical snow rates, which confirms that there are no undesired side-effects from coupling SuperdropNet when the conditions in the atmosphere do not allow warm-rain processes.

We also perform a mixed-phase experiment with the same setup. In this scenario, both frozen and non-frozen particles occur in the atmosphere. Figure B.2c shows the grid-averaged rain rate. The grid average values for all hydrometeors are included in the appendix. In this case, coupling to SuperdropNet significantly drops the total rain rate. Since the total water mass remains conserved in ICON, the suppression of rain formation leads to increased ice, cloud and snow formation (Figure B.7). In ICON, the warm rain processes are simulated before other processes such as ice nucleation, ice self-collection, snow melting etc. Hence, SuperdropNet's effect on decreasing rain formation is subsequently reflected in the excess of other hydrometeors.

B.4.2.2 *Heat Transport Fluxes*

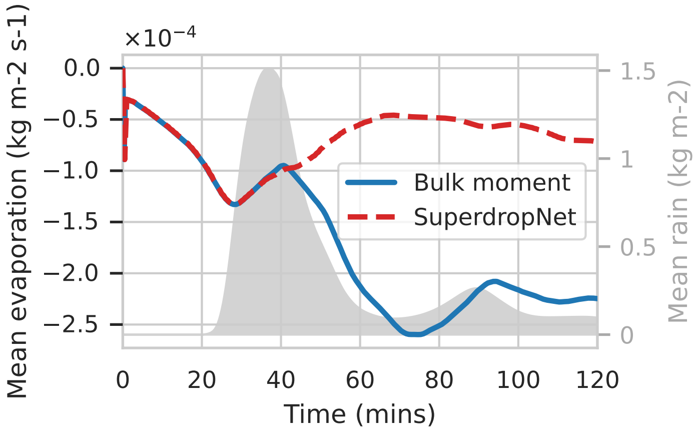


Figure B.3: Grid-averaged evaporative heat fluxes for the bulk moment scheme used in ICON two-moment cloud microphysics, and for SuperdropNet. The gray area shows the grid-averaged rain obtained using the bulk-moment scheme (see Figure B.2a). High negative values indicate a larger amount of heat transfer.

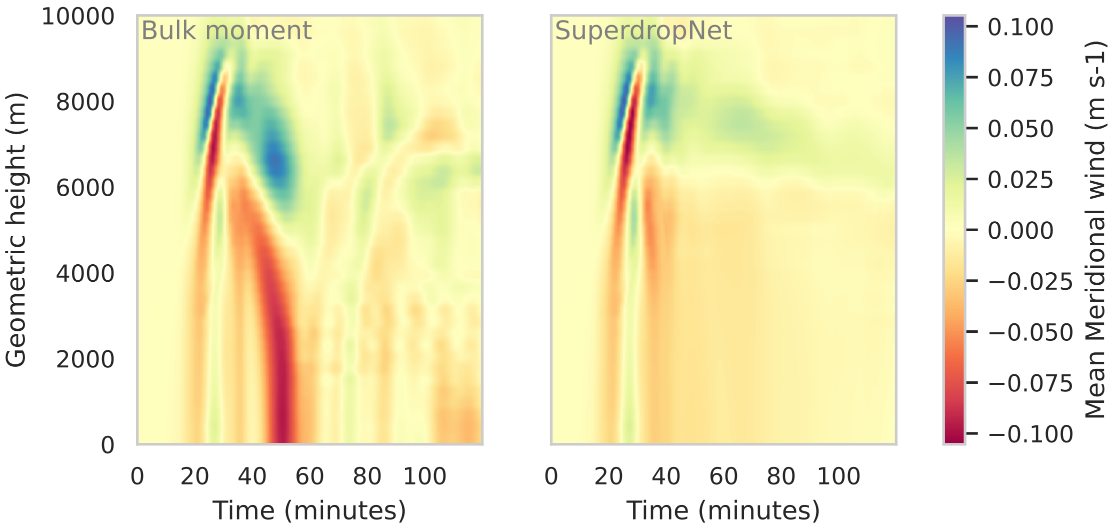


Figure B.4: Averaged meridional winds for the bulk moment scheme used in ICON two-moment cloud microphysics *(left)* and for SuperdropNet *(right)*.

Figure B.3 shows the grid-averaged evaporative fluxes as it evolves with time during the coupled warm-bubble simulation in ICON. While in the beginning both the bulk moment scheme and SuperdropNet produce similar fluxes, the values diverge approximately after about 30 minutes, corresponding to the onset of rain. This difference between the magnitude of fluxes is also reflected in the evolution of winds during the simulation. Winds are the primary source of energy transport and Fig. B.4 shows the evolution of meridional winds in the simulation. After approximately 40 minutes, which roughly corresponds to the end of the first rainfall with both parameterizations, the wind patterns are markedly different for the bulk moment and the SuperdropNet parameterizations. The winds appear much stronger in case of the bulk moment parameterization across the vertical column. The reduced magnitude of winds in SuperdropNet coupling corresponds to reduced heat fluxes in Fig. B.3.

Figure B.5 shows the vertical profile of specific humidity at different timesteps during the simulation. For the first 40 minutes of the experiment, both parameterization schemes produce similar specific humidity profiles but this changes during the later part of the simulation. Close to the surface, it can be observed that the bulk moment parameterization produces a stronger humidity gradient in comparison to SuperdropNet. This difference in the specific humidity gradient possibly results in a higher evaporative flux for the bulk moment coupling than the SuperdropNet coupled simulation.

Similarly, in Fig. B.6 the evolution of mean rain droplet mass ($\bar{X}_r$) is shown. The differences in $\bar{X}_r$ close to the surface as calculated using the bulk moment scheme vs. SuperdropNet become more visible

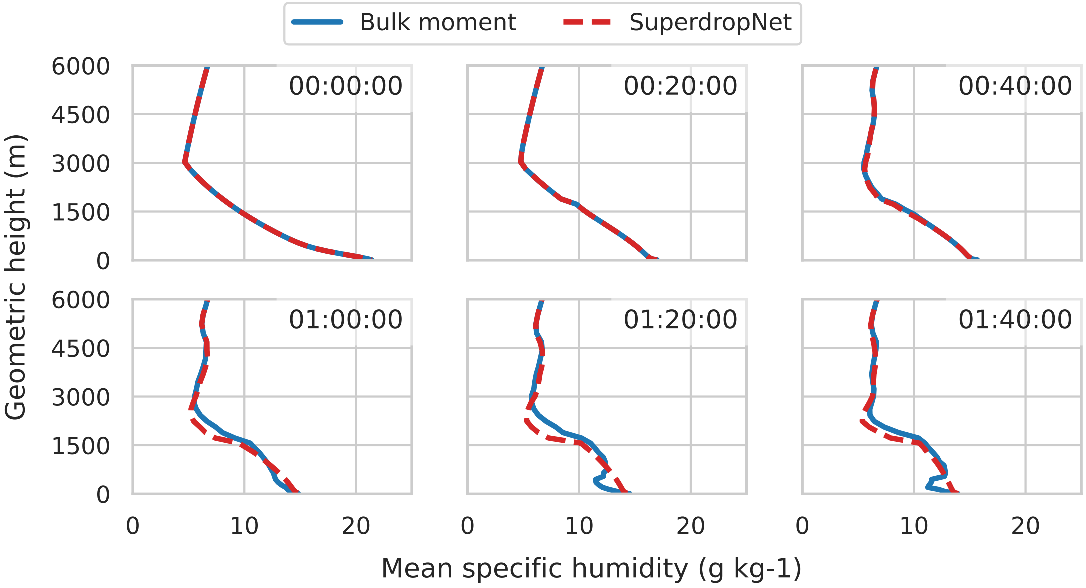


Figure B.5: Vertical profile of the specific humidity at different times for the bulk moment scheme and for SuperdropNet.

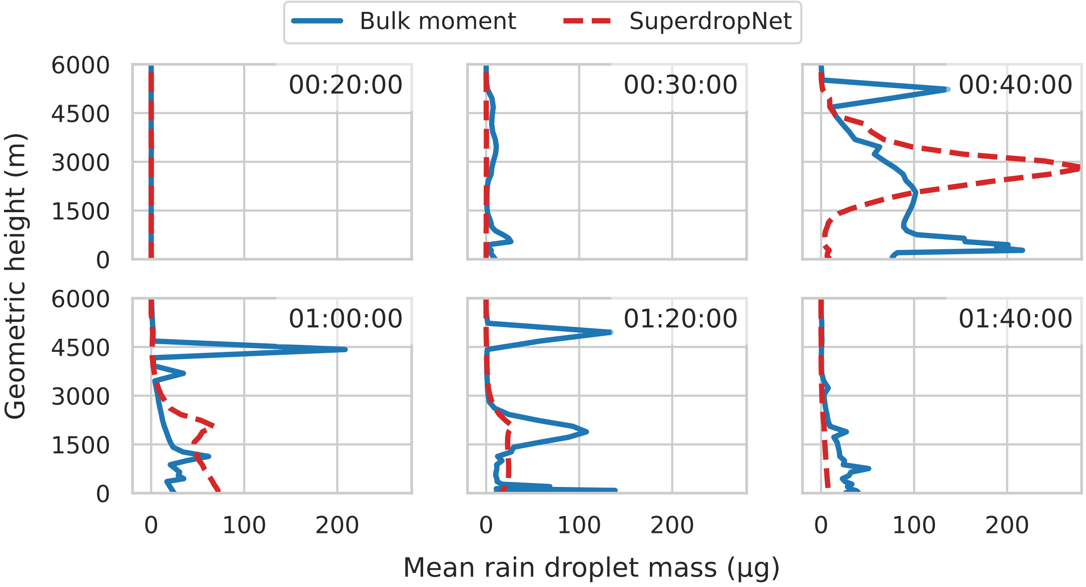


Figure B.6: Vertical profile of the rain droplet mass, calculated as the ratio of the specific rain content and the number concentration of rain droplets at different times for the bulk moment scheme and for SuperdropNet.

after 40 minutes. In general with the bulk moment parameterization $\bar{X}_r$ values are higher than those with the superdroplet parameterization close to the surface. Since the evaporative flux is propotional to the mean rain mass, higher $\bar{X}_r$ in bulk moment coupling results in higher heat fluxes. Throughout the vertical column, the SuperdropNet parameterization usually corresponds to lower $\bar{X}_r$, except at the 40 minutes time step where the high $\bar{X}_r$ value near the 3000 m height also corresponds to a higher amount of the vertically integrated rain rate as seen in figure B.2a.

Note that the warm-bubble scenario in ICON is highly sensitive to tiniest fluctuations within the assumptions made for cloud microphysics parameterization. Since

many other complex phenomena are simplified and the focus is only on the formation and dissipation of a single cloud, small deviations in the approximation of the cloud and rain moments lead to changes in other diagnostic variables that can accumulate over time.

B.4.3 *Computational performance upon including SuperdropNet*

B.4.3.1 *Benchmark*

Experiment		t_{2mom} (s)	Nodes
Bulk moment scheme (Fortran)		1.25	1
SuperdropNet (Pytorch)	CFFI	24.1	1
	Pipes	62.6	1
	YAC	49.5	2

Table 3: Time spent in the two-moment scheme in the ICON warm-bubble scenario, using the bulk-moment scheme (Fortran), and SuperdropNet (Pytorch) coupled to ICON. Note that by coupling SuperdropNet to ICON we introduce a scheme that would be computationally intractable for cloud microphysics in standard numerical simulations. A direct comparison of runtimes is therefore not possible.

We run the experiments on the Levante compute system at the German Climate Computing Center on compute nodes equipped with 2 AMD 7763 CPUs with a total of 128 cores and 256 GB main memory. The nodes are connected with a Mellanox Infiniband HDR100 fabric.

SuperdropNet provides a significant speedup by emulating processes that would otherwise be computationally infeasible to include in ICON, but when adding a Python component to the existing highly optimized Fortran code we expect an impact on computational performance. Table 3 summarizes the total time spent in the calculation of the two-moment scheme in the ICON warm bubble scenario, using the bulk moment scheme and SuperdropNet coupled to ICON through three different coupling strategies. The fastest time to solution is provided by including SuperdropNet via embedded Python, i.e. the C Foreign Function Interface (CFFI) (Sect. B.3.1). Coupling SuperdropNet via YAC (Sect. B.3.2) increases the relative runtime by a factor of two compared to embedded Python. Note that when coupling with YAC, the ICON and the Python main program run on two different computational nodes, which doubles the amount of computational resources required for the experiment. In the current configuration, YAC can only be used when the block length is equal to the grid size, which limits us to small experiments like the bubble scenarios. Coupling SuperdropNet and ICON using pipes is almost three times slower than embedded Python. On a qualitative note, implementing the coupling via pipes requires changes to core components of ICON beyond the cloud microphysics parameterization and may be an additional challenge for ML developers.

We note that coupling a superdroplet model directly to our test case in ICON is extremely challenging. ICON represents the warm rain processes as bulk moments, while McSnow represents them as droplet distributions. For an ideal benchmark simulation, we would need to completely overhaul the current representation of cloud microphysics processes in ICON and represent them as superdroplets for a two-way coupling. At the time of conducting this research, ICON did not allow for the representation of cloud microphysical processes as superdroplets, mainly because doing so would be computationally expensive. This is an active area of research but as of now, remains a work in progress, which makes SuperdropNet a cheaper, data-driven alternative to the superdroplet simulations.

B.4.3.2 *Detailed evaluation for coupling with embedded Python*

Process	Time (μs)	Fraction
Time reported by ICON	5.0×10^2	100%
Time reported by Python	4.8×10^2	96%
↪ out of which time reported for inference	4.4×10^2	87%
↪ out of which time reported for data transfer	4.2×10^1	8.5%

Table 4: Processes when coupling SuperdropNet to ICON via embedded Python and their associated duration. Machine learning inference is executed on a CPU node of the Levante compute system at the German Climate Computing Center.

We now turn to the fastest coupling scheme, embedded Python, and investigate the contribution of the individual steps to the total runtime. By including SuperdropNet, we incur computational cost for data exchange and for machine learning inference. Table 4 summarizes the contribution of the individual parts, measured with a block length of nproma $= 44$ grid cells using the ICON timer module. ICON averages the execution time across a total of $496,800$ calls to SuperdropNet. Most of the time can be attributed to model inference, while the actual data transfer is less significant. This could be attributed to the fact that ML inference has to be done on CPU. On a node equipped with an NVIDIA A100 GPU, we measure an inference time of 267 µs. This corresponds to 33% of the inference time reported on a CPU (see Table 4).

Note however that a heterogeneous setup, where moments are transfered to and from the GPU nodes via the Mellanox Infiniband network, would likely lead to a larger overall wall time. Given the successful efforts of porting ICON to GPU, a future experiment could be run exclusively on GPUs. By only applying SuperdropNet when at least one input moment is nonzero, we are already reducing the number of calls to ML inference to improve performance.

B.5 CONCLUSIONS

We have coupled SuperdropNet, a machine learning algorithm emulating warm rain processes in a two-moment cloud microphysics scheme, to ICON. In the warm bubble experiment, the ML emulator is stable, and the results are physically sound.

The strategies to bridge ICON and Python provide flexibility for the development of the ML component and account for the fact that ML development is done iteratively. Both embedded Python and YAC can be integrated with little programming overhead into ICON. For a later ML emulator, that replaces a full parameterization at the grid level, YAC can be used regardless of the block length. Coupling via pipes is comparatively slow and does not scale well. Since it requires an extensive rewrite of core components of ICON, we would not recommend it for implementation. Out of the three coupling strategies we tested, embedded Python provided the fastest performance. It can be used independent of the ICON grid to execute any Python code at any level of the ICON time loop.

We note that by coupling SuperdropNet to ICON we introduce a scheme that would otherwise be computationally intractable for cloud microphysics in a standard numerical simulations. A direct comparison of runtimes is therefore not possible. Note however that integrating a Python component will slow down the overall time to solution due to the incurred cost in network inference and data transfer. For applications that are more demanding than our warm bubble scenario test case, and if the ML component is thoroughly tested, a reimplementation in Fortran would likely increase performance, at the expense of losing the flexibility of development.

A natural extension of this work are more complex modelling scenarios. This would involve training machine learning based emulators for other cloud microphysical processes and/or introduction of other hydrometeors apart from clouds and rain. Apart from droplet collisions, processes such as sedimentation of droplets and deep convection can be challenging to represent with bulk moment parameterization schemes. Hence, in the future we want to explore the possibility of creating ML based proxies for these processes while continuing to use hybrid ML-ESMs for continuous online testing.

APPENDIX

B.5.1 *Mixed-phase bubble*

We include the grid-averaged cloud ice, cloud water, graupel, snow, and ice for the mixed-phase experiment described in Section B.4.2.1. The results are shown in Figure B.7.

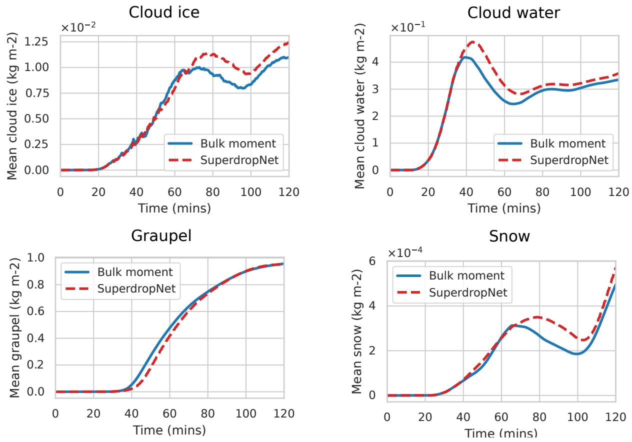


Figure B.7: Grid-averaged quantities for the bulk moment scheme and SuperdropNet under mixed phase scenario.

CODE AVAILABILITY

The SuperdropNet (version 0.1.0) inference code, trained model weights, and modules describing the coupling between SuperdropNet inference and generic Fortran code, analysis scripts and Jupyter notebooks, as well as the experiment description files are available under the MIT license here: https://doi.org/10.5281/zenodo.10069121. The license file is included in the repository. The ICON model code used for the simulations in this paper is available under https://doi.org/10.5281/zenodo.8348256. It is based on the ICON release 2.6.5 and includes additional code for coupling SuperdropNet. ICON is now publicly available under BSD-3-C license at https://www.icon-model.org. The experiment results obtained with SuperdropNet (version 0.1.0) coupled to ICON (version 2.6.5) are available under https://doi.org/10.5281/zenodo.8348266. We used McSnow (version 1.1.0) for generating the training data in a warm rain scenario. McSnow is not publicly available. Access to McSnow can be granted upon agreeing to the ICON licensing terms by the developers of McSnow (Brdar & Seifert, 2018).

ACKNOWLEDGEMENTS

We thank A.-K. Naumann and S. Rast for support with the ICON warm bubble scenario and helpful discussions, and we thank N.-A. Dreier and M. Hanke for support with integrating YAC. This work was supported by Helmholtz Association's Initiative and Networking Fund through Helmholtz AI [grant number: ZT-I-PF-5-01]. This work used resources of the Deutsches Klimarechenzentrum (DKRZ) granted by its Scientific Steering Committee (WLA) under project ID AIM.

BIBLIOGRAPHY

Abade, G. C., Grabowski, W. W., & Pawłowska, H. (2018). Broadening of cloud droplet spectra through eddy hopping: Turbulent entraining parcel simulations. *Journal of the Atmospheric Sciences*.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In: In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. 2016, 265–283.

Alexeev, D. (2023, August 1). PyTorch bindings for Fortran (v0.4).

Andrejczuk, M., Grabowski, W. W., Reisner, J. M., & Gadian, A. M. (2010). Cloud-aerosol interactions for boundary layer stratocumulus in the lagrangian cloud model. *Journal of Geophysical Research*, *115*.

Arabas, S., & Shima, S. (2012). Large-eddy simulations of trade wind cumuli using particle-based microphysics with monte carlo coalescence. *Journal of the Atmospheric Sciences*, *70*, 2768–2777.

Bartman, P., Bulenok, O., Górski, K., Jaruga, A., Łazarski, G., Olesik, M. A., Piasecki, B., Singer, C. E., Talar, A., & Arabas, S. (2022). PySDM v1: Particle-based Cloud Modeling Package for Warm-rain Microphysics and Aqueous Chemistry. *Journal of Open Source Software*, *7*, 3219.

Bauer, P., Thorpe, A., & Brunet, G. (2015). The quiet revolution of numerical weather prediction. *Nature*, *525*(7567), 47–55. https://doi.org/10.1038/nature14956

Beheng, K. D., & Doms, G. A general formulation of collection rates of cloud and raindrops using the kinetic equation and comparison with parameterizations. In: 1986.

Belochitski, A., & Krasnopolsky, V. (2021a). Robustness of neural network emulations of radiative transfer parameterizations in a state-of-the-art general circulation model. *Geoscientific Model Development*, *14*(12), 7425–7437. https://doi.org/10.5194/gmd-14-7425-2021

Belochitski, A., & Krasnopolsky, V. (2021b). Robustness of neural network emulations of radiative transfer parameterizations in a state-of-the-art general circulation model. *Geosci. Model Dev.*, *14*(12), 7425–7437. https://doi.org/10.5194/gmd-14-7425-2021

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. Curriculum learning. In: In *Proceedings of the 26th annual international conference on machine learning*. 2009, 41–48. ISBN: 978-1-60558-516-1. https://doi.org/10.1145/1553374.1553380

Beucler, T., Pritchard, M., Rasp, S., Ott, J., Baldi, P., & Gentine, P. (2021). Enforcing analytic constraints in neural networks emulating physical systems. *Physical Review Letters*, *126*(9), 098302. https://doi.org/10.1103/PhysRevLett.126.098302

Bodnar, C., Bruinsma, W. P., Lucic, A., Stanley, M., Brandstetter, J., Garvan, P., Riechert, M., Weyn, J., Dong, H., Vaughan, A., Gupta, J. K., Tambiratnam, K., Archibald, A., Heider, E., Welling, M., Turner, R. E., & Perdikaris, P. (2024). Aurora: A foundation model of the atmosphere.

Bonanni, A., Hawkes, J., & Quintino, T. (2022). Infero: A lower-level api for machine learning inference in operations (version 0.2.0).

Bonavita, M., & Laloyaux, P. (2020). Machine learning for model error inference and correction. *Journal of Advances in Modeling Earth Systems*, *12*, e2020MS002232. https://doi.org/10.1029/2020MS002232

Bony, S., Bellon, G., Klocke, D., et al. (2013). Robust direct effect of carbon dioxide on tropical circulation and regional precipitation. *Nature Geoscience*, *6*, 447–451. https://doi.org/10.1038/ngeo1799

Boucher, O., Randall, D., Artaxo, P., Bretherton, C., Feingold, G., Forster, P., Kerminen, V.-M., Kondo, Y., Liao, H., Lohmann, U., Rasch, P., Satheesh, S. K., Sherwood, S., Stevens, B., & Zhang, X. Y. Clouds and aerosols (T. F. Stocker, D. Qin, G.-K. Plattner, M. Tignor, S. K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex, & P. M. Midgley, Eds.). In: *Climate change 2013: The physical science basis. contribution of working group i to the fifth assessment report of the intergovernmental panel on climate change* (T. F. Stocker, D. Qin, G.-K. Plattner, M. Tignor, S. K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex, & P. M. Midgley, Eds.). Ed. by Stocker, T. F., Qin, D., Plattner, G.-K., Tignor, M., Allen, S. K., Boschung, J., Nauels, A., Xia, Y., Bex, V., & Midgley, P. M. Cambridge, United Kingdom, New York, NY, USA: Cambridge University Press, 2013.

Brandstetter, J., Worrall, D. E., & Welling, M. Message passing neural pde solvers. In: In *International conference on learning representations*. 2022.

Brdar, S., & Seifert, A. (2018). McSnow: A Monte-Carlo Particle Model for Riming and Aggregation of Ice Particles in a Multidimensional Microphysical Phase Space. *J. Adv. Model. Earth Sy.*, *10*, 187–206. https://doi.org/10.1002/2017MS001167

Brenowitz, N. (2023, January 13). Call Python from Fortran (v0.2.1). https://doi.org/10.5281/zenodo.7779572

Brenowitz, N. D., Beucler, T., Pritchard, M., & Bretherton, C. S. (2020a). Interpreting and stabilizing machine-learning parametrizations of convection. *Journal of the Atmospheric Sciences*, *77*, 4357–4375. https://doi.org/10.1175/JAS-D-20-0082.1

Brenowitz, N. D., & Bretherton, C. S. (2018). Prognostic Validation of a Neural Network Unified Physics Parameterization. *Geophysical Research Letters*, *45*(12), 6289–6298. https://doi.org/10.1029/2018GL078510

Brenowitz, N. D., Beucler, T., Pritchard, M., & Bretherton, C. S. (2020b). Interpreting and Stabilizing Machine-Learning Parametrizations of Convection. *J. Atmos. Sci.*, *77*(12), 4357–4375. https://doi.org/10.1175/JAS-D-20-0082.1

Brenowitz, N. D., & Bretherton, C. S. (2019). Spatially Extended Tests of a Neural Network Parametrization Trained by Coarse-Graining. *J. Adv. Model. Earth Sy.*, *11*(8), 2728–2744. https://doi.org/10.1029/2019MS001711

Brenowitz, N. D., Henn, B., McGibbon, J., Clark, S. K., Kwa, A., Perkins, W. A., Watt-Meyer, O., & Bretherton, C. S. (2020c). Machine learning climate model dynamics: Offline versus online performance. https://doi.org/10.48550/arXiv.2011.03081

Brenowitz, N. D., Perkins, W. A., Nugent, J. M., Watt-Meyer, O., Clark, S. K., Kwa, A., Henn, B., McGibbon, J., & Bretherton, C. S. Emulating fast processes in climate models. In: In *Machine Learning for the Physical Sciences, NEURIPS Workshop*. 2022. https://doi.org/10.48550/arXiv.2211.10774

Chantry, M., Hatfield, S., Dueben, P., Polichtchouk, I., & Palmer, T. (2021). Machine learning emulation of gravity wave drag in numerical weather forecasting. *Journal of Advances in Modeling Earth Systems*, *13*(7), e2021MS002477. https://doi.org/10.1029/2021MS002477

Chevallier, F, Morcrette, J.-J., Chéruy, F, & Scott, N. (2000). Use of a neural-network-based long-wave radiative-transfer scheme in the ecmwf atmospheric model. *Quarterly Journal of the Royal Meteorological Society*, *126*(563), 761–776. https://doi.org/10.1002/qj.49712656318

Chollet, F., et al. (2023). *Keras (v2.14.0)*.

Christensen, H., & Zanna, L. Parametrization in Weather and Climate Models. In: In *Oxford Research Encyclopedia of Climate Science*. 2022. ISBN: 978-0-19-022862-0. https://doi.org/10.1093/acrefore/9780190228620.013.826

Clark, T. L. (1974). Numerical modeling of the dynamics and microphysics of warm cumulus convection. *Journal of the Atmospheric Sciences*, *31*(7), 1712–1732. https://doi.org/10.1175/1520-0469(1974)031<1712:NMOTDA>2.0.CO;2

Curcic, M. (2019). A parallel Fortran framework for neural networks and deep learning. https://doi.org/10.48550/arXiv.1902.06714

Dong, W., Fritts, D. C., Liu, A. Z., Lund, T. S., Liu, H.-L., & Snively, J. (2023). Accelerating atmospheric gravity wave simulations using machine learning: Kelvin-helmholtz instability and mountain wave sources driving gravity wave breaking and secondary gravity wave generation. *Geophysical Research Letters*, *50*(15), e2023GL104668. https://doi.org/10.1029/2023GL104668

Dueben, P. D., Schultz, M. G., Chantry, M., Gagne, D. J., Hall, D. M., & McGovern, A. (2022). Challenges and Benchmark Datasets for Machine Learning in the Atmospheric Sciences: Definition, Status, and Outlook. *Artificial Intelligence for the Earth Systems*, *1*(3). https://doi.org/10.1175/AIES-D-21-0002.1

Dueben, P., Modigliani, U., Geer, A., Siemen, S., Pappenberger, F., Bauer, P., Brown, A., Palkovic, M., Raoult, B., Wedi, N., & Baousis, V. (2021). Machine learning at ecmwf: A roadmap for the next 10 years. *ECMWF Technical Memoranda*, (878). https://doi.org/10.21957/ge7ckgm

Elafrou, A., Orchard, D., & Cliffard, S. (2023a). Fortran-pytorch-lib (commit: Ffe833b66a6e1ce1c6cf023708d1f351a3a11f8b).

Elafrou, A., Orchard, D., & Cliffard, S. (2023b). Fortran-pytorch-lib (commit: Ffe833b66a6e1ce1c6cf023708d1f351a3a11f8b).

Farchi, A., Laloyaux, P., Bonavita, M., & Bocquet, M. (2021). Using machine learning to correct model error in data assimilation and forecast applications. *Quarterly Journal of the Royal Meteorological Society*, *147*(739), 3067–3084. https://doi.org/10.1002/qj.4116

Ferrier, B. S. (1994). A double-moment multiple-phase four-class bulk ice scheme. part i: Description.

Flossmann, A. I. (1998). Interactions of aerosol particles and clouds: A box model numerical study. *Journal of Geophysical Research*, *103*(D4), 3825–3835. https://doi.org/10.1029/97JD02712

Flossmann, A. I., Hall, W. D., & Pruppacher, H. R. (1985). A theoretical study of the wet removal of atmospheric pollutants. part i: The redistribution of aerosol particles captured through nucleation and impaction scavenging by growing cloud drops. *Journal of the Atmospheric Sciences*, *42*(6), 583–606. https://doi.org/10.1175/1520-0469(1985)042<0583:ATSOTW>2.0.CO;2

Geneva, N., & Zabaras, N. (2020). Modeling the dynamics of pde systems with physics-constrained deep auto-regressive networks. *Journal of Computational Physics*, *403*, 109056. https://doi.org/10.1016/j.jcp.2019.109056

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., & Yacalis, G. (2018). Could Machine Learning Break the Convection Parameterization Deadlock? *Geophysical Research Letters*, *45*(11), 5742–5751. https://doi.org/10.1029/2018GL078202
A very interesting use of machine learning for the purpose of cloud parameterization.

Gettelman, A., Gagne, D. J., Chen, C.-C., Christensen, M. W., Lebo, Z. J., Morrison, H., & Gantos, G. (2021). Machine Learning the Warm Rain Process. *Journal of Advances in Modeling Earth*

Systems, *13*(2), e2020MS002268. https://doi.org/https://doi.org/10.1029/2020MS002268
e2020MS002268 2020MS002268.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Grabowski, W. W., Dziekan, P., & Pawłowska, H. (2017). Lagrangian condensation microphysics with twomey ccn activation. *Geoscientific Model Development*, *11*, 103–120.

Gross, M., Wan, H., Rasch, P. J., Caldwell, P. M., Williamson, D. L., Klocke, D., Jablonowski, C., Thatcher, D. R., Wood, N., Cullen, M., Beare, B., Willett, M., Lemarié, F., Blayo, E., Malardel, S., Termonia, P., Gassmann, A., Lauritzen, P. H., Johansen, H., . . . Leung, R. (2018). Physics–dynamics coupling in weather, climate, and earth system models: Challenges and recent progress. *Monthly Weather Review*, *146*(11), 3505–3544. https://doi.org/10.1175/MWR-D-17-0345.1

Grundner, A., Beucler, T., Gentine, P., Iglesias-Suarez, F., Giorgetta, M. A., & Eyring, V. (2022). Deep Learning Based Cloud Cover Parameterization for ICON. *Journal of Advances in Modeling Earth Systems*, *14*(12), e2021MS002959. https://doi.org/10.1029/2021MS002959
e2021MS002959 2021MS002959.

Grzeszczuk, R., Terzopoulos, D., & Hinton, G. NeuroAnimator: Fast neural network emulation and control of physics-based models. In: In *Proceedings of the 25th annual conference on computer graphics and interactive techniques*. SIGGRAPH '98. New York, NY, USA: Association for Computing Machinery, 1998, 9–20. ISBN: 978-0-89791-999-9. https://doi.org/10.1145/280814.280816

Hanke, M., Dreier, N.-A., & Redler, R. (2023). Yetanothercoupler (yac) (version 2.6.1) [code].

Hanke, M., Redler, R., Holfeld, T., & Yastremsky, M. (2016). Yac 1.2.0: New aspects for coupling software in earth system modelling. *Geoscientific Model Development*, *9*, 2755–2769. https://doi.org/10.5194/gmd-9-2755-2016

Igel, A. L., Morrison, H., Santos, S. P., & van Lier-Walqui, M. (2022). Limitations of separate cloud and rain categories in parameterizing collision-coalescence for bulk microphysics schemes. *Journal of Advances in Modeling Earth Systems*, *14*(e2022MS003039). https://doi.org/10.1029/2022MS003039

Ionescu, R. T., Alexe, B., Leordeanu, M., Popescu, M., Papadopoulos, D. P., & Ferrari, V. How hard can it be? estimating the difficulty of visual search in an image. In: In *2016 ieee conference on computer vision and pattern recognition (cvpr)*. 2016.

Irrgang, C., Boers, N., Sonnewald, M., Barnes, E. A., Kadow, C., Staneva, J., & Saynisch-Wagner, J. (2021). Towards neural Earth system modelling by integrating artificial intelligence in Earth system science. *Nature Machine Intelligence*, *3*(8), 667–674. https://doi.org/10.1038/s42256-021-00374-3

Kelp, M. M., Jacob, D. J., Kutz, J. N., Marshall, J. D., & Tessum, C. W. (2020). Toward Stable, General Machine-Learned Models of the Atmospheric Chemical System. *Journal of Geophysical Research: Atmospheres*, *125*(23), e2020JD032759. https://doi.org/10.1029/2020JD032759
e2020JD032759 2020JD032759.

Kelp, M. M., Jacob, D. J., Lin, H., & Sulprizio, M. P. (2022). An Online-Learned Neural Network Chemical Solver for Stable Long-Term Global Simulations of Atmospheric Chemistry. *Journal of Advances in Modeling Earth Systems*, *14*(6), e2021MS002926. https://doi.org/10.1029/2021MS002926
e2021MS002926 2021MS002926.

Kessler, E. (1969). *On the distribution and continuity of water substance in atmospheric circulations* (Vol. 10).

Khain, A. P., Beheng, K. D., Heymsfield, A., Korolev, A., Krichak, S. O., Levin, Z., Pinsky, M., Phillips, V., Prabhakaran, T., Teller, A., Heever, S. C. v. d., & Yano, J.-I. (2015). Representation of microphysical processes in cloud-resolving models: Spectral (bin) microphysics versus bulk parameterization. *Reviews of Geophysics*, *53*(2), 247–322. https://doi.org/10.1002/2014RG000468

Kingma, D. P., & Ba, J. (2017). Adam: A method for stochastic optimization. https://doi.org/10.48550/arXiv.1412.6980

Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., & Hoyer, S. (2021). Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, *118*(21), e2101784118. https://doi.org/10.1073/pnas.2101784118

Kohl, G., Chen, L., & Thuerey, N. Benchmarking autoregressive conditional diffusion models for turbulent flow simulation. In: In *Icml 2024 ai for science workshop*. 2024.

Krasnopolsky, V. M., Fox-Rabinovitz, M. S., & Chalikov, D. V. (2005). New approach to calculation of atmospheric model physics: Accurate and fast neural network emulation of longwave radiation in a climate model. *Monthly Weather Review*, *133*(5), 1370–1383. https://doi.org/10.1175/MWR2923.1

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., Ravuri, S., Ewalds, T., Eaton-Rosen, Z., Hu, W., Merose, A., Hoyer, S., Holland, G., Vinyals, O., Stott, J., Pritzel, A., Mohamed, S., & Battaglia, P. (2023). Learning skillful medium-range global weather forecasting. *Science*, *382*(6677), 1416–1421. https://doi.org/10.1126/science.adi2336

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. Gradient-based learning applied to document recognition. In: In *Proceedings of the ieee. 86*. (11). 1998, 2278–2324.

Lessig, C., Luise, I., Gong, B., Langguth, M., Stadtler, S., & Schultz, M. (2023). AtmoRep: A stochastic model of atmosphere dynamics using large scale representation learning. https://doi.org/10.48550/arXiv.2308.13280

Leufen, L. H., & Schädler, G. (2019). Calculating the turbulent fluxes in the atmospheric surface layer with neural networks. *Geoscientific Model Development*, *12*(5), 2033–2047. https://doi.org/https://doi.org/10.5194/gmd-12-2033-2019
Publisher: Copernicus GmbH.

Lippe, P., Veeling, B. S., Perdikaris, P., Turner, R. E., & Brandstetter, J. (2023). Pde-refiner: Achieving accurate long rollouts with neural pde solvers. *Advances in Neural Information Processing Systems*, *36*.

List, B., Chen, L.-W., Bali, K., & Thuerey, N. (2024). How temporal unrolling supports neural physics simulators. *arXiv preprint arXiv:2402.12971*.

Lynn, B., & Khain, A. (2007). Utilization of spectral bin microphysics and bulk parameterization schemes to simulate the cloud structure and precipitation in a mesoscale rain event. *Journal of Geophysical Research: Atmospheres*, *112*(D22). https://doi.org/10.1029/2007JD008475

Manabe, S., & Bryan, K. (1969). Climate Calculations with a Combined Ocean-Atmosphere Model. *Journal of the Atmospheric Sciences*, *26*(4), 786–789. https://doi.org/10.1175/1520-0469(1969)026<0786:CCWACO>2.0.CO;2

Marshall, J. S., & Palmer, W. M. K. (1948). THE DISTRIBUTION OF RAINDROPS WITH SIZE. *Journal of the Atmospheric Sciences*, *5*(4), 165–166. https://doi.org/10.1175/1520-0469(1948)005<0165:TDORWS>2.0.CO;2

McGovern, A., Lagerquist, R., Gagne, D. J., Jergensen, G. E., Elmore, K. L., Homeyer, C. R., & Smith, T. (2019). Making the Black Box More Transparent: Understanding the Physical Implications of Machine Learning. *B. Am. Meteorol. Soc.*, *100*(11), 2175–2199. https://doi.org/10.1175/BAMS-D-18-0195.1

Meyer, D., Hogan, R. J., Dueben, P. D., & Mason, S. L. (2022). Machine Learning Emulation of 3D Cloud Radiative Effects. *J. Adv. Model. Earth Sy.*, *14*(3), e2021MS002550. https://doi.org/10.1029/2021MS002550

Morrison, H., Lier-Walqui, M. v., Fridlind, A. M., Grabowski, W. W., Harrington, J. Y., Hoose, C., Korolev, A., Kumjian, M. R., Milbrandt, J. A., Pawlowska, H., Posselt, D. J., Prat, O. P., Reimel, K. J., Shima, S.-I., Diedenhoven, B. v., & Xue, L. (2020). Confronting the Challenge of Modeling Cloud and Precipitation Microphysics. *J. Adv. Model. Earth Sy.*, *12*(8), e2019MS001689. https://doi.org/10.1029/2019MS001689

Mu, B., Chen, L., Yuan, S., & Qin, B. (2023). A radiative transfer deep learning model coupled into wrf with a generic fortran torch adaptor. *Frontiers in Earth Science*, *11*. https://doi.org/10.3389/feart.2023.1149566

Naumann, A. K., & Seifert, A. (2015). A Lagrangian drop model to study warm rain microphysical processes in shallow cumulus. *Journal of Advances in Modeling Earth Systems*, *7*(3), 1136–1154. https://doi.org/10.1002/2015MS000456

Nowack, P., Braesicke, P., Haigh, J., Abraham, N. L., Pyle, J., & Voulgarakis, A. (2018). Using machine learning to build temperature-based ozone parameterizations for climate sensitivity simulations. *Environ. Res. Lett.*, *13*(10), 104016. https://doi.org/10.1088/1748-9326/aae2be

O'Gorman, P. A., & Dwyer, J. G. (2018). Using Machine Learning to Parameterize Moist Convection: Potential for Modeling of Climate, Climate Change, and Extreme Events. *Journal of Advances in Modeling Earth Systems*, *10*(10), 2548–2563. https://doi.org/https://doi.org/10.1029/2018MS001351

Ott, J., Pritchard, M., Best, N., Linstead, E., Curcic, M., & Baldi, P. (2020). A Fortran-Keras Deep Learning Bridge for Scientific Computing. *Scientific Programming*, *2020*, Article ID 8888811. https://doi.org/10.1155/2020/8888811

Pal, A., Mahajan, S., & Norman, M. R. (2019). Using deep neural networks as cost-effective surrogate models for super-parameterized e3sm radiative transfer. *Geophysical Research Letters*, *46*(11), 6069–6079. https://doi.org/10.1029/2019GL082195

Palmer, T. N. (2001). A nonlinear dynamical perspective on model error: A proposal for non-local stochastic-dynamic parametrization in weather and climate prediction models. *Quarterly Journal of the Royal Meteorological Society*, *127*(572), 279–304. https://doi.org/https://doi.org/10.1002/qj.49712757202

Palmer, T. (2020). A Vision for Numerical Weather Prediction in 2030. *arXiv:2007.04830 [physics]*
Comment: For World Meteorological Organisation White Paper on Future of NWP.

Partee, S., Ellis, M., Rigazzi, A., Shao, A. E., Bachman, S., Marques, G., & Robbins, B. (2022). Using machine learning at scale in numerical simulations with smartsim: An application to ocean climate modeling. *J. Comput. Sci.-Neth.*, *62*, 101707. https://doi.org/10.1016/j.jocs.2022.101707

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In:

In *Advances in neural information processing systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

Qu, Y., & Shi, X. (2023). Can a Machine Learning–Enabled Numerical Model Help Extend Effective Forecast Range through Consistently Trained Subgrid-Scale Models? *Artificial Intelligence for the Earth Systems*, *2*(1). https://doi.org/10.1175/AIES-D-22-0050.1

Rasp, S. (2020). Coupled online learning as a way to tackle instabilities and biases in neural network parameterizations: General algorithms and Lorenz 96 case study (v1.0). *Geosci. Model Dev.*, *13*(5), 2185–2196. https://doi.org/10.5194/gmd-13-2185-2020

Rasp, S., Pritchard, M. S., & Gentine, P. (2018). Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, *115*(39), 9684–9689. https://doi.org/10.1073/pnas.1810286115

Rigo, A., & Fijalkowski, M. (2018). C foreign function interface for python, cffi [code].

Rouson, D., et al. (2023). *Inference engine (v0.10.0)*.

Sardina, G., Poulain, S., Brandt, L., & Caballero, R. (2017). Broadening of cloud droplet size spectra by stochastic condensation : Effects of mean updraft velocity and ccn activation. *Journal of the Atmospheric Sciences*, *75*, 451–467.

Seifert, A., & Beheng, K. D. (2006). A two-moment cloud microphysics parameterization for mixed-phase clouds. Part 1: Model description. *Meteorology and Atmospheric Physics*, *92*(1), 45–66. https://doi.org/10.1007/s00703-005-0112-4

Seifert, A., & Beheng, K. D. (2001). A double-moment parameterization for simulating autoconversion, accretion and selfcollection. *Atmospheric Research*, *59-60*, 265–281. https://doi.org/10.1016/S0169-8095(01)00126-0

Seifert, A., & Rasp, S. (2020). Potential and Limitations of Machine Learning for Modeling Warm-Rain Cloud Microphysical Processes. *Journal of Advances in Modeling Earth Systems*, *12*(12), e2020MS002301. https://doi.org/10.1029/2020MS002301 e2020MS002301 10.1029/2020MS002301.

Sharma, S., & Greenberg, D. (2024). SuperdropNet: A stable and accurate machine learning proxy for droplet-based cloud microphysics. https://doi.org/10.48550/arXiv.2402.18354

Shima, S., Kusano, K., Kawano, A., Sugiyama, T., & Kawahara, S. (2009). The super-droplet method for the numerical simulation of clouds and precipitation: A particle-based and probabilistic microphysics model coupled with a non-hydrostatic model. *Quarterly Journal of the Royal Meteorological Society*, *135*(642), 1307–1320. https://doi.org/10.1002/qj.441

Sölch, I., & Kärcher, B. (2010). A large-eddy model for cirrus clouds with explicit aerosol and ice microphysics and lagrangian ice particle tracking. *Quarterly Journal of the Royal Meteorological Society*, *136*.

Sonnewald, M., Lguensat, R., Jones, D. C., Dueben, P. D., Brajard, J., & Balaji, V. (2021). Bridging observations, theory and numerical simulation of the ocean using machine learning. *Environ. Res. Lett.*, *16*(7), 073008. https://doi.org/10.1088/1748-9326/ac0eb0

Stephens, G. L., Wild, M., Stackhouse, P. W., L'Ecuyer, T., Kato, S., & Henderson, D. S. (2012). The global character of the flux of downward longwave radiation. *Journal of Climate*, *25*, 2329–2340.

Stevens, B., & Bony, S. (2013). What are climate models missing? *Science*, *340*, 1053–1054.

Suzuki, K., Stephens, G. L., van den Heever, S. C., & Nakajima, T. Y. (2011). Diagnosis of the warm rain process in cloud-resolving models using joint cloudsat and modis observations. *Journal of the Atmospheric Sciences*, *68*, 2655–2670.

Ulbrich, C. W. (1983). Natural variations in the analytical form of the raindrop size distribution.

Um, K., Brand, R., Fei, Y. R., Holl, P., & Thuerey, N. Solver-in-the-loop: Learning from differentiable physics to interact with iterative PDE-solvers. In: In *Advances in neural information processing systems*. *33*. Curran Associates, Inc., 2020, 6111–6122.

Unterstrasser, S., Hoffmann, F., & Lerch, M. (2017). Collection/aggregation algorithms in Lagrangian cloud microphysical models: Rigorous evaluation in box model simulations. *Geoscientific Model Development*, *10*(4), 1521–1548. https://doi.org/10.5194/gmd-10-1521-2017

Veerman, M. A., Pincus, R., Stoffer, R., Van Leeuwen, C. M., Podareanu, D., & Van Heerwaarden, C. C. (2021). Predicting atmospheric optical properties for radiative transfer computations using neural networks. *Philosophical Transactions of the Royal Society A*, *379*(2194), 20200095.

Wacker, U. (2000). An analytical study on the evolution of the vertical profile of rain water concentration. *Geophysical Research Letters*, *27*, 1275–1278.

Wacker, U., & Seifert, A. (2001). Evolution of rain water profiles resulting from pure sedimentation: Spectral vs. parameterized description. *Atmospheric Research*, *58*(1), 19–39. https://doi.org/10.1016/S0169-8095(01)00081-3

Willis, P. T. (1984). Functional fit to some observed drop size distributions and parameterization of rain. *Journal of the Atmospheric Sciences*, *41*, 1648–1661. https://doi.org/10.1175/1520-0469(1984)041<1648:FFTSOD>2.0.CO;2

Wyant, M. C., Bretherton, C. S., Blossey, P. N., & Khairoutdinov, M. (2012). Fast cloud adjustment to increasing co2 in a super-parameterized climate model. *Journal of Advances in Modeling Earth Systems*, *4*.

Yuval, J., O'Gorman, P. A., & Hill, C. N. (2021). Use of neural networks for stable, accurate and physically consistent parameterization of subgrid atmospheric processes with good performance at reduced precision. *Geophysical Research Letters*, *48*(6), e2020GL091363. https://doi.org/10.1029/2020GL091363

Yuval, J., & O'Gorman, P. A. (2023). Neural-Network Parameterization of Subgrid Momentum Transport in the Atmosphere. *J. Adv. Model. Earth Sy.*, *15*(4), e2023MS003606. https://doi.org/10.1029/2023MS003606

Zhong, X., Ma, Z., Yao, Y., Xu, L., Wu, Y., & Wang, Z. (2023). WRF–ML v1.0: A bridge between WRF v4.3 and machine learning parameterizations and its application to atmospheric radiative transfer. *Geosci. Model Dev.*, *16*(1), 199–209. https://doi.org/10.5194/gmd-16-199-2023

Zängl, G., Reinert, D., Rípodas, P., & Baldauf, M. (2015). The ICON (ICOsahedral non-hydrostatic) modelling framework of DWD and MPI-m: Description of the non-hydrostatic dynamical core. *Quarterly Journal of the Royal Meteorological Society*, *141*(687), 563–579. https://doi.org/10.1002/qj.2378

EIDESSTATTLICHE VERSICHERUNG

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

I hereby declare upon oath that I have written the present dissertation independently and have not used further resources and aids than those stated.

Hamburg, Germany, August 2024

Shivani Sharma

Hinweis / Reference

Die gesamten Veröffentlichungen in der Publikationsreihe des MPI-M
„Berichte zur Erdsystemforschung / Reports on Earth System Science“,
ISSN 1614-1199

sind über die Internetseiten des Max-Planck-Instituts für Meteorologie erhältlich:
https://mpimet.mpg.de/forschung/publikationen

All the publications in the series of the MPI -M
„Berichte zur Erdsystemforschung / Reports on Earth System Science“,
ISSN 1614-1199

are available on the website of the Max Planck Institute for Meteorology:
https://mpimet.mpg.de/en/research/publications