Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Learning from Perception to Imagination: Towards General Multimodal Robot Manipulation

Doctoral thesis
at Research Group Technical Aspects of Multimodal Systems, TAMS
Prof. Dr. Jianwei. Zhang

Department of Informatics
MIN-Faculty
Universität Hamburg

submitted by
Wenkai Chen
Course of study: Informatik
Matrikelnr.: 7481393

Hamburg, 2024

Day of oral defense: 28.10.2024

The following evaluators recommend the admission of the dissertation:

Supervisor:
Prof. Dr. Jianwei Zhang
Department of Informatics
Universität Hamburg, Germany

Reviewer:
Prof. Dr. Stefan Wermter
Department of Informatics
Universität Hamburg, Germany

Chair:
Prof. Dr. Janick Edinger
Department of Informatics
Universität Hamburg, Germany

Abstract

Over the past decades, there has been a notable development in research of robot grasping and manipulation, seeking to leverage the growing accessibility of custom-built robot arms and grippers to elevate robot autonomy. However, it remains challenging for the new robot to operate in our everyday environments due to the variability and uncertainty of the real world. With the emergence of embodied artificial intelligence, the capability of robots to effectively perceive and act in their environments through their physical structures is becoming more important. Additionally, numerous approaches about robot learning are proposed to equip robots with the ability to learn from and adapt to new and unforeseen circumstances, as it is impractical to pre-program them with a detailed model of their environment, the objects within it, or the complex skills needed for manipulation. This dissertation explores the interplay between perception and action in robotics. It examines how robots' passive perception aids in gaining a contextual understanding of their environment and how interactive perception, integrated with action, enhances decision-making. Akin to human logic in manipulation, we further investigate the imagination potential of robotic manipulation outcomes after processing all perceptual information, hopefully boosting the efficiency of manipulation.

Especially for passive perception from visual understanding, several interesting ideas to improve grasping performance and enhance robot manipulability, *i.e.,* point cloud completion and 3D affordance, are introduced into the grasp pose detection algorithm. Moreover, a point cloud completion dataset based on the YCB Videos and a context-aware grasping dataset is constructed, separately. In parallel, we propose a transformer-based sparse shape completion framework and an affordance-based grasping detection framework to facilitate robots' interaction with objects or context-aware parts.

As the interactive perception, we explore more modality information from the hand-object and object-object interactions. To address the recognition challenges in dynamic hand-object actions, we employ event camera to gather data for an event-based hand-object dataset, named *EHoA*, and we have developed an attention-based spiking neural network that delivers benchmark results. The inference model is also effectively validated from real experiments of robot hand-object interaction. Furthermore, we study the strategy of multimodal learning in the challenged task setting of multiple peg-in-hole assembly, incorporating modalities like vision, proprioception, force, and torque. These are learned as compact representations within a sim2real transfer learning framework, with domain randomization and impedance control integrated into the policy training to better bridge the simulation-reality gap. Our real-world tests validate the effectiveness of our methods, demonstrating successful peg-in-hole assembly and adaptability to various object shapes.

Furthermore, this thesis explores the role of language in human-robot collaboration. We explore the robots' imagination ability on the tabletop rearrangement scenario after receiving perceptual information and human language instructions. A comprehensive 2D tabletop rearrangement dataset is first constructed, where a physical simulator is used to capture inter-object relationships and semantic configurations. Concurrently, we present DreamArrangement, a novel language-conditioned object rearrangement scheme, consisting of two primary processes: employing a transformer-based multi-modal denoising diffusion model to envisage the desired arrangement of objects, and leveraging a vision-language foundational model to derive actionable policies from text, alongside initial and target visual information. To reduce the average motion distance of objects, an efficiency-oriented rearrangement approach is also implemented. Our methods have been proven effective in real-world robotic trials, showing proficiency in handling complex, language-conditioned, and long-horizon tasks using a single model.

Overall, this dissertation proposes various robot learning methodologies to tackle challenges in robot perception and manipulation. Our findings cover different abstractions of the perception-action loop, from the low-level robot controller to the high-level contextual understanding and goal-oriented imagination. Although achieving fully generalizable embodied intelligence remains a distant goal, we hope our efforts represent significant steps forward in this challenging field.

Zusammenfassung

In den letzten Jahrzehnten gab es eine bemerkenswerte Entwicklung in der Forschung im Bereich des Robotergreifens und der Manipulation. Diese Forschung zielt darauf ab, die zunehmende Verfügbarkeit von maßgeschneiderten Roboterarmen und Greifern zu nutzen, um die Autonomie von Robotern auf ein bisher unerreichtes Niveau zu heben. Dennoch bleibt es eine Herausforderung, dass neue Roboter in unseren alltäglichen Umgebungen aufgrund der Variabilität und Unsicherheit der realen Welt operieren können. Mit dem Aufkommen der Embodied Intelligence ist die Fähigkeit von Robotern, ihre Umgebungen durch ihre physischen Strukturen effektiv wahrzunehmen und in ihnen zu agieren, viel wichtiger. Darüber hinaus werden zahlreiche Ansätze zum Robot Learning vorgeschlagen, um Robotern die Fähigkeit zu vermitteln, aus neuen und unvorhergesehenen Umständen zu lernen und sich daran anzupassen, da es unpraktisch ist, sie im Voraus mit einem detaillierten Modell ihrer Umgebung, der darin befindlichen Objekte oder der komplexen Fähigkeiten, die für die Manipulation erforderlich sind, zu programmieren. Diese Dissertation erforscht das Zusammenspiel von Wahrnehmung und Aktion in der Robotik. Sie untersucht, wie die passive Wahrnehmung von Robotern hilft, ein kontextuelles Verständnis ihrer Umgebung zu gewinnen und wie interaktive Wahrnehmung, integriert mit Aktion, die Entscheidungsfindung verbessert. Ähnlich der menschlichen Logik in der Manipulation untersuchen wir weiterhin das Imaginationspotenzial robotischer Manipulationsergebnisse nach der Verarbeitung aller Wahrnehmungen, in der Hoffnung, die Effizienz der Manipulation zu steigern.

Insbesondere werden aus der visuellen Wahrnehmung mehrere interessante Ideen zur Verbesserung der Greifleistung und zur Erhöhung der Manipulierbarkeit von Robotern, wie z.B. die Vervollständigung von Punktwolken und 3D-Affordanzen, in den Algorithmus zur Erkennung von Greifposen eingeführt. Darüber hinaus wird ein Datensatz zur Vervollständigung von Punktwolken basierend auf den YCB-Videos und ein kontextbewusster Greifdatensatz separat erstellt. Parallel dazu präsentieren wir einen transformer-basierten Rahmen für die Sparse Shape Completion und einen affordanzbasierten Rahmen für die Greiferkennung, um die Interaktion von Robotern mit Objekten oder kontextbewussten Teilen zu erleichtern.

Als interaktive Wahrnehmung erforschen wir mehr Modalitätsinformationen aus den Interaktionen zwischen Hand und Objekt sowie zwischen Objekten. Um die Erkennungsprobleme bei dynamischen Hand-Objekt-Aktionen zu lösen, verwenden wir die Event-Vision, um Daten für einen ereignisbasierten Hand-Objekt-Datensatz namens EH-oA zu sammeln, und wir haben ein Attention-Based Spiking Neural Network entwickelt, das Benchmark-Ergebnisse liefert. Das Inferenzmodell wird auch effektiv aus realen Experimenten der Interaktion zwischen Roboterhand und Objekt validiert. Wir untersuchen auch die Strategie des multimodalen Lernens in der herausfordernden Aufgabenumgebung der mehrfachen Peg-in-Hole-Montage, wobei Modalitäten wie Vision, Propriozeption, Kraft und Drehmoment eingebunden werden. Diese werden als kompakte Darstellungen innerhalb eines Sim2Real Transfer Learning Frameworks gelernt, wobei Randomisierung und Impedanzsteuerung in den Policy-Trainingsprozess integriert sind, um die Lücke zwischen Simulation und Realität besser zu überbrücken. Unsere Tests in der realen Welt bestätigen die Wirksamkeit unserer Methoden und zeigen erfolgreiche mehrfache Peg-in-Hole-Montagen und Anpassungsfähigkeit an verschiedene Objektformen.

Darüber hinaus erforscht diese Dissertation die Rolle der Sprache in der Zusammenarbeit zwischen Mensch und Roboter. Wir erforschen die Imaginationsfähigkeit der Roboter im Szenario der Objektneuanordnung nach dem Erhalt von Wahrnehmungsinformationen und menschlichen Sprachanweisungen. Wir erstellen zunächst einen umfassenden 2D-Datensatz für die Objektneuanordnung, bei dem ein physikalischer Simulator verwendet wird, um die Beziehungen zwischen den Objekten und die semantischen Konfigurationen zu erfassen. Parallel dazu präsentieren wir DreamArrangement, ein neuartiges sprachgesteuertes Objektneuanordnungsschema, das aus zwei Hauptprozessen besteht: dem Einsatz eines transformer-basierten multimodalen Denoising-Diffusionsmodells, um die gewünschte Anordnung der Objekte zu erahnen, und der Nutzung eines Vision-Language Foundational Models, um aus Texten neben den anfänglichen und zielspezifischen visuellen Informationen handlungsorientierte Policies abzuleiten. Wir implementieren auch einen effizienzorientierten Umordnungsansatz, um die durchschnittliche Bewegungsdistanz der Objekte zu reduzieren. Unsere Methoden haben sich in realen Roboterversuchen als wirksam erwiesen und zeigen Erfolge bei der Bewältigung komplexer, sprachgesteuerter und langfristiger Aufgaben mit einem einzigen Modell.

Insgesamt schlägt diese Dissertation verschiedene Robot Learning Methoden vor, um Herausforderungen in der Robotikwahrnehmung und -manipulation zu bewältigen. Unsere Ergebnisse decken verschiedene Abstraktionen der Wahrnehmungs-Aktions-Schl-eife ab, von der niedrigen Ebene der Robotersteuerung bis zum hohen Niveau des kontextuellen Verständnisses und der zielorientierten Imagination. Obwohl das Erreichen einer vollständig generalisierbaren verkörperten Intelligenz noch ein fernes Ziel ist, hoffen wir, dass unsere Bemühungen bedeutende Fortschritte in diesem anspruchsvollen Bereich darstellen.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Throughout history, humans have continuously developed and refined tools and machines, from simple hoes to complex looms, to minimize physical labor and streamline manual tasks. This tradition of innovation has propelled us into an era where affordable, lightweight, and flexible robots are built and adopted to increase productivity in our homes, offices and industry. As robots increasingly become part of everyday life, the need for them to operate autonomously and effectively in complex, unstructured environments is becoming paramount. Traditional robotic systems, which rely heavily on pre-programmed behaviors and responses, are limited in their ability to adapt to new or changing conditions. Recently, a concept of embodied intelligence is proposed where cognitive intelligence is not just brain-based but arises from the dynamic interactions between an agent's physical body and its environment. This idea is deeply inspired by the learning processes observed in human children, reflecting the integration of perception, cognition, and action, allowing a robot to adapt its behaviors based on sensory feedback and its physical capabilities. Just as a child learns to recognize objects by watching, touching, and manipulating them, a robot equipped with embodied intelligence uses its sensors and actuators in a coordinated manner to gain a deeper understanding of its operational context.

As a robotic researcher, how can we endow the physical robot with embodied intelligence? The famous psychology professor Lin Smith proposes six lessons on the embodied agent after observing babies' learning experiences [1]. Here, we summarize it as four points: (1) **Be Multimodal**. Just as humans use vision, sound, touch, and other senses to navigate and understand their environment, a multimodal approach in robotics allows machines to gather a broader range of data. This diversity in sensory information helps robots to create a more complete and accurate model of their surroundings, enhancing their ability to make informed decisions. Moreover, multiple sensory modalities provide redundancy, which is critical for reliability and robustness in dynamic or unpredictable environments. For instance, if visual information is compromised due to poor lighting conditions, a robot might rely on tactile feedback or auditory cues to continue functioning effectively. (2) **Be Incremental**. Children will progress to finish more complex tasks through incrementally mastering simple skills. For instance, a child learning to walk adjusts their balance and gait in response to the physical characteristics of the floor or the presence of obstacles. As the robots, grasping is the fundamental of all complex manipulation tasks. By implementing grasping and motion planning algorithms that allow robots to learn from sensor perception, robots can master different action skills to meet different requirements from human beings. (3) **Explore in a physical world.** In the early stage, babies usually move and act in highly variable and playful ways that are seemingly random. Physiologists think it is important for humans to find new problems and solutions by exploring an open-ended world intensively. For instance, a robot learns about object properties (like weight and texture) more effectively by manipulating the objects rather than by passive observation. Moreover, exploring the physical world also provides immediate and impactful feedback. This feedback is crucial for learning and adjusting behaviors in embodied agents. For example, a robot navigating a cluttered environment learns to modify its path-planning strategies based on obstacles encountered. (4) **Use language and be social**. Human babies acquire knowledge and skills through social interactions and communication with others. They learn language, social cues, and complex behaviors by observing and interacting with their parents and peers. For robots equipped with embodied intelligence, engaging in similar social interactions and language use enables them to learn in ways that are analogous to human learning, promoting more natural and intuitive behaviors. Furthermore, many tasks that robots might be expected to perform in human environments require understanding and acting upon complex instructions that involve multiple steps or conditions. Language comprehension and social interaction skills can greatly enhance a robot's ability to understand and execute such instructions accurately and efficiently. Especially, a robot with strong contextual understanding skills can function and generalize across a wider range of environments and tasks—from households and educational settings to more complex social situations.

These characteristics mentioned above of embodied intelligence are still hard for a physical robot to achieve simultaneously, and each characteristic can be an individual research topic in the robotic community. Figure 1.1 illustrates 36 kitchen actions in the English vocabulary. As human beings, a twelve-year-old child can do all these kitchen tasks, while today no robot can do that. To address these challenges and advance towards more capable embodied agents, several concrete strategies and research directions can be pursued from the perspective of robot learning:

- **Context-Aware Perception**: Robots must be able to perceive and understand their environment in a context-aware manner, recognizing not only objects but also their functionalities and task requirements. This involves sophisticated vision systems and the integration of deeper semantic understanding.

- **Multimodal Learning**: Integrating multiple types of sensory data (visual, auditory, tactile, etc.) can provide robots with a richer understanding of their environment. This approach mimics human sensory processing and can significantly improve the robot's ability to perform complex tasks that require the integration of different sensory inputs.

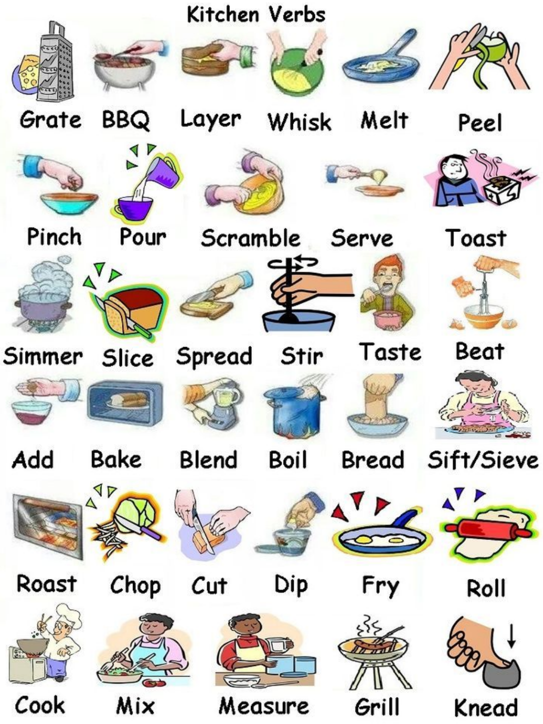
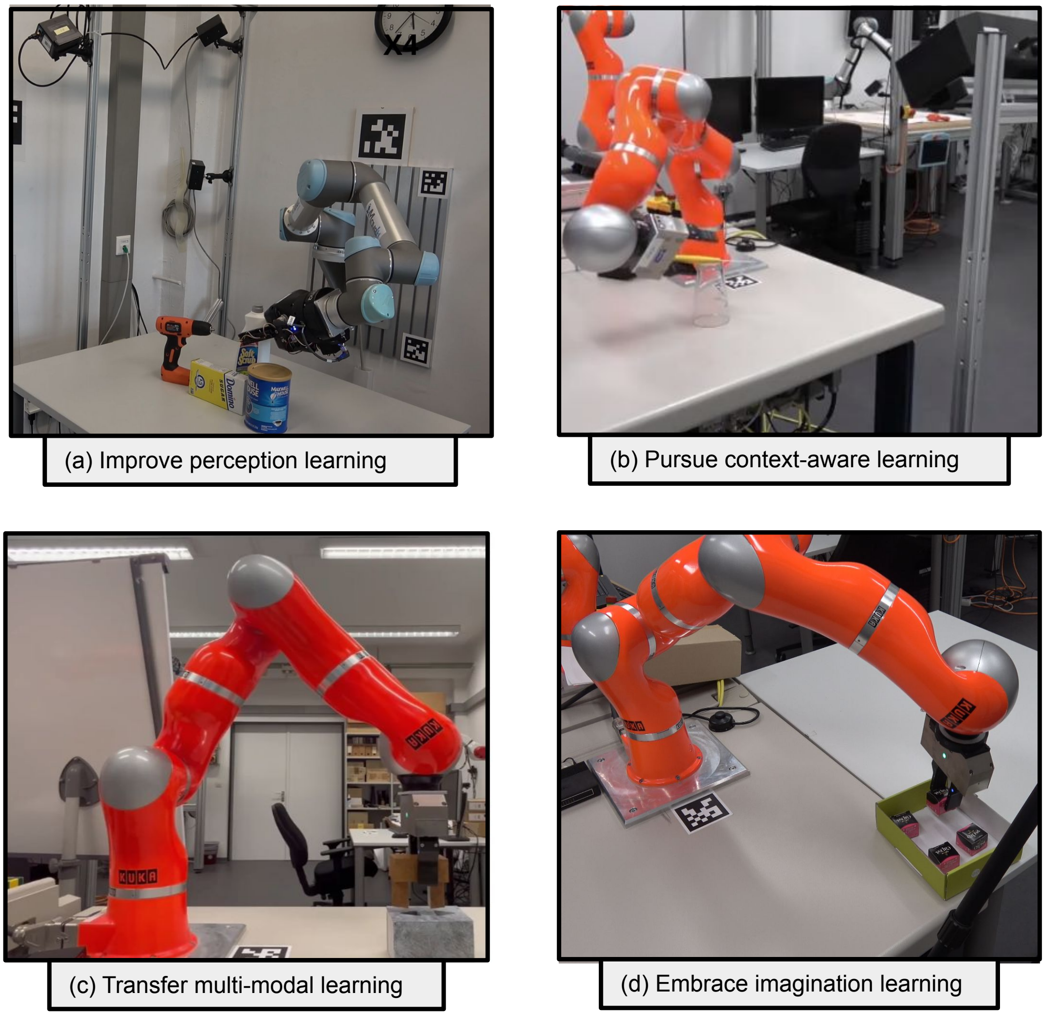


Figure 1.1: 36 kitchen verbs in the English vocabulary. Retrieved from `https://www.pinterest.com/pin/611082243195984573/`. Reprinted image: ©2024, Henrique Maldito.

- **Incremental and Lifelong Learning**: Robots need algorithms that enable them to learn incrementally and adaptively, much like a human child does. This includes developing capabilities for lifelong learning, where robots can continuously acquire new skills and refine existing ones over time without forgetting previously learned information.
- **Social Interaction and Communication**: Enhancing robots' abilities to understand and engage in human communication and social cues will improve their functionality in human environments. This includes understanding verbal instructions, non-verbal cues, and being able to execute tasks cooperatively with humans.
- **Task Planning and Execution**: Robots should be capable of complex planning and decision-making that involves sequencing tasks, timing, and adapting actions based on dynamic environmental changes. Cognitive architectures that support planning and reasoning will be crucial.

These proposed robot learning strategies can hopefully allow robots to adapt to the complexities of real-world environments. Unlike traditional programming, which requires explicit instructions for every possible scenario, robot learning enables robots to adjust their behaviors based on the data they gather through interactions. This is particularly important for tasks in unstructured environments like homes, offices, or outdoors, where predictability is low, and situations are highly variable. Moreover, it can also help robots



(a) Improve perception learning

(b) Pursue context-aware learning

(c) Transfer multi-modal learning

(d) Embrace imagination learning

Figure 1.2: Robotic task examples solved by four kinds of robot learning strategies in the thesis. (a) The UR5 robot is grasping the target bottle cleanser using improved geometric shape features from shape completion. (b) Based on the context-ware learning, the KUKA robot is grasping the *handle* part of a knife to execute the *cut* action potentially. (c) Using multiple modalities and an impedance controller trained from simulation, the KUKA robot is putting an *unseen triangle* pegs object into the holes object on the table. (d) After endowing the imaginative and planning ability from diffusion algorithms and Large Language Models (LLMs), the KUKA robot rearranges different small boxes into a big box container instructed by human languages.

to generalize learned skills across different tasks and environments. By extracting patterns and principles from past experiences, robots can apply this knowledge to new situations.

1.2 Aim of this Thesis

By using different robot learning algorithms, this thesis aims to address the following research questions.

Research Questions

- For a service robot, how to improve 6-DoF grasp performance to satisfy human requirements in our daily lives?
- How to leverage multiple modality information when the service robot interacts with objects?
- How to endow the imaginative ability to the service robot when collaborating with humans?

In summary, we hope to endow a physical robot with the ability to take human language and multisensory perception as input, then interact with the physical world as feedback, and finally achieve everyday tasks such as context-aware stable grasping and more complicated long-horizon manipulation tasks such as peg-in-hole assembly and tabletop rearrangement. As shown in Figure 1.2, a series of challenging robotic task examples are achieved using different robot learning strategies from raw perception to intuitive imagination. Specifically, the proposed research questions that this thesis tries to solve are:

- Currently, robotic grasping methods based on sparse partial point clouds often generate wrong grasping candidates due to the lack of geometric information on the object. Moreover, traditional voxel-based shape completion solutions in grasping mainly concentrate on recovering a single object, while they hardly consider the occlusion from other objects. To improve the grasping performance, a point cloud completion dataset in the robotic field and a corresponding advanced sparse point cloud completion method should be proposed to improve the perception ability of a robot.
- Recently, context-aware learning methods in robotic grasping are mostly based on pixel-level surfaces. These pixel-level approaches heavily rely on the accuracy of a 2D affordance mask, and the generated grasp candidates are restricted to a small workspace. To mitigate the limitation, it's necessary to construct a novel affordance-based grasp dataset and propose a 6-DoF task-oriented grasp detection framework based on the 3D point cloud.
- Now, there is little work exploring context-aware hand-object interaction from a dynamic perspective which actually exists in our daily lives. To fill the blank research field, a hand-object action dataset based on the dynamic interactions should be collected, which regular Complementary Metal–oxide–semiconductor (CMOS) cameras cannot handle. A supervised learning method should also be proposed to achieve such context-aware action recognition.
- Existing Reinforcement Learning (RL) methods are difficult to apply to multiple peg-in-hole issues due to more complicated geometric and physical constraints. In addition, previously limited solutions for multiple peg-in-hole assembly are hard to transfer into real industrial scenarios flexibly. To effectively address these

issues, a novel and more challenging multiple peg-in-hole assembly setup should be designed. An incremental learning scheme should be proposed to solve the challenged task, achieving a generalization across different object shapes in real-world scenarios.

- Prior solutions for robotic rearrangement have overlooked the significance of integrating human preferences and optimizing for rearrangement efficiency. Additionally, traditional prompt-based approaches struggle with complex, semantically meaningful rearrangement tasks without pre-defined target states for objects. Thus, it's necessary to construct an object rearrangement dataset, covering different inter-object relationships and semantic configurations. Moreover, a novel language-conditioned rearrangement scheme should be proposed to generate robot policies after receiving observation and human instructions.

1.3 Contribution of this Thesis

The main contributions of this thesis can be described as follows:

- **Improve Robotic Grasping Performance using Sparse Point Cloud Completion:** We construct a large-scale non-synthetic partial point cloud dataset based on YCB Video dataset [2]. As the dataset is captured by a real RGB-D camera, the natural noise will facilitate the generalization of our work, especially in real robot environments. Moreover, we propose a novel shape completion model (TransSC) [3]. This model has a transformer-based encoder to explore more point-wise features and a manifold-based decoder to exploit more object details using a segmented partial point cloud as input. Quantitative experiments verify the effectiveness of the proposed shape completion network and demonstrate that our network outperforms existing methods. Besides, TransSC is integrated into a grasp evaluation network [4] to generate a set of grasp candidates. The simulation experiment shows that TransSC improves the grasping generation result compared to the existing shape completion baselines. Furthermore, our robotic experiment shows that with TransSC, the robot is more successful in grasping objects of unknown numbers randomly placed on a support surface.

- **Learn 6-DoF Task-oriented Grasping Detection via Implicit Estimation and Visual Affordance:** We first construct a novel affordance-based grasp dataset and propose a 6-DoF task-oriented grasp detection framework [5], which takes the observed object point cloud as input and predicts diverse 6-DoF grasp poses for different tasks. Specifically, our implicit estimation network and visual affordance network in this framework could directly predict coarse grasp candidates, and corresponding 3D affordance heatmap for each potential task, respectively. Furthermore, the grasping scores from coarse grasps are combined with heatmap values to generate more accurate and finer candidates. Our proposed framework shows significant improvements compared to baselines for existing and novel objects on our simulation dataset. Although our framework is trained based on the simulated objects and environment, the final generated grasp candidates can be accurately and stably executed in real robot experiments when the object is randomly placed on a support surface.

- **Propose a Benchmark for Task-oriented Hand-Object Action Recognition using Event Vision:** We present a richly annotated task-oriented hand-object action dataset consisting of asynchronous event streams, captured by the event-based camera system on different application scenarios [6]. In addition, we design an Attention-based Residual Spiking Neural Network (ARSNN) by learning temporal-wise and spatial-wise attention simultaneously and introducing a particular residual connection structure to achieve dynamic hand-object action recognition. Extensive experiments are validated by comparing with existing baseline methods to form a vision benchmark. We also show that the learned recognition model can be transferred to classify real robot hand-object actions.

- **Transfer Robotic Multiple Peg-in-hole Assembly Skills from Simulation and Multi-sensory Signals:** We design a novel and more challenging multiple peg-in-hole assembly setup by using the advantage of transfer learning. We propose a detailed solution scheme to solve this task [7]. Specifically, multiple modalities including vision, proprioception, and force/torque are learned as compact representations to account for the complexity and uncertainties and improve the sample efficiency. Furthermore, RL is used in the simulation to train the policy, and the learned policy is transferred to the real world without extra exploration. Domain randomization and impedance control are embedded into the policy to narrow the gap between simulation and reality. Evaluation results demonstrate the effectiveness of the proposed solution, showcasing successful multiple peg-in-hole assembly and generalization across different object shapes in real-world scenarios.

- **Learn Language-conditioned Robotic Rearrangement of Objects via Denoising Diffusion and VLM Planner:** We first introduce a comprehensive 2D tabletop rearrangement dataset, utilizing a physical simulator to capture inter-object relationships and semantic configurations. Then we present DreamArrangement, a novel language-conditioned object rearrangement scheme, consisting of two primary processes: employing a transformer-based multi-modal denoising diffusion model to envisage the desired arrangement of objects, and leveraging a vision-language foundational model to derive actionable policies from text, alongside initial and target visual information. In particular, we introduce an efficiency-oriented learning strategy to minimize the average motion distance of objects. Given few-shot instruction examples, the learned policy from our synthetic dataset can be transferred to the real world without extra human intervention. Extensive simulations validate DreamArrangement's superior rearrangement quality and efficiency. Moreover, real-world robotic experiments confirm that our method can adeptly execute a range of challenging, language-conditioned, and long-horizon tasks with a singular model.

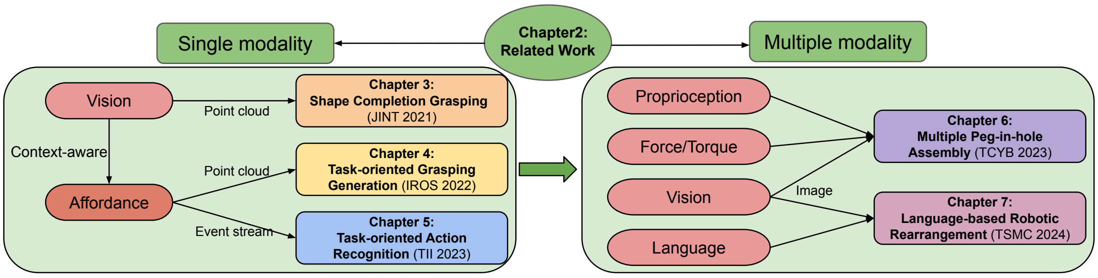


Figure 1.3: Overview of this thesis. It represents the research direction from single-modal learning to multi-modal learning. In single-modal learning diagram, it consists of three chapters that describe the main contributions of point cloud-based and event-based robotic tasks. In multi-modal learning diagram, it includes extra two chapters that describe the main contributions of robotic tasks combining proprioception, force/torque, image and language. Chapter 3 [3], Chapter 4 [5], Chapter 5 [6], Chapter 6 [7], and Chapter 7 [8] are organized from five separate papers of the author.

1.4 Structure of this Thesis

Inspired by the lessons about embodied intelligence, our thesis mainly conducts four kinds of research topics of robot learning, shown in Figure 1.2. It covers from single modality of vision to multiple modalities such as proprioception, force/torque and language. The structure of this thesis is shown in Figure 1.3. The rest of this section will introduce and give an abstract of each chapter.

Related Work

- Chapter 2: Related Work. This chapter describes some basic concepts and recent works related to our four kinds of robot learning strategies. To improve the perception learning on robotic grasping, we introduce basic work about robotic grasping based on the point cloud, dense point cloud completion and traditional shape completion methods in robotic grasping. To achieve context-aware learning in robotic tasks, we introduce the concept of affordance, traditional task-oriented grasping methods and hand-object action recognition methods. To transfer multi-modal learning, we introduce the robotic manipulation work about sim2real, reinforcement learning and similar work with our setting. Finally, we introduce recent works about foundation models based on vision and language to embrace imagination learning in our tabletop rearrangement task.

Single Modality

- Chapter 3: Shape Completion Grasping. This chapter describes a novel transformer-based shape completion framework. Input by a partial point cloud in an arbitrary camera view, the reconstructed object shape can help us improve the robotic grasping performance significantly, especially in the occluded conditions. This chapter is organized from the paper [3].

- Chapter 4: Task-oriented Grasping Generation. This chapter describes a novel 6-DOF context-aware grasping generation framework. A course-fine pose generation scheme is achieved using an implicit estimation network and a 3D affordance prediction network. This chapter is organized from the paper [5].

- Chapter 5: Task-oriented Action Recognition. This chapter describes a novel benchmark about hand-object action recognition captured by the event camera. An attention-based spiking neural network is also proposed to achieve state-of-the-art (SOTA) recognition accuracy. This chapter is organized from the paper [6].

Multiple Modality

- Chapter 6: Multiple Peg-in-hole Assembly. This chapter describes a challenging multiple peg-in-hole assembly setup. We propose a sim2real transfer learning architecture where the trained policy in the simulation can be transferred into real robot experiments directly. This chapter is organized from the paper [7].
- Chapter 7: Language-based Robotic Rearrangement. This chapter describes a challenged tabletop rearrangement task considering human preference and motion efficiency. Based on the setup, we design a novel language-conditioned object rearrangement scheme using a diffusion model and VLM planner. This chapter is organized from the paper [8].

Conclusion

- Chapter 8: Conclusion and Future Work. This chapter summarizes all research work and presents the findings and outcomes of my doctoral study. Ultimately, it also suggests potential future directions to address existing limitations and expand upon the current scope of my work.

Chapter 2

Related Work

When embodied agents interact with and manipulate their surroundings, it's crucial to equip them with the capability to initially comprehend the context of the environment. Similar to the function of all kinds of feeling organs from human beings, robotic perception is the cornerstone for robots to learn a series of complicated manipulation skills. Robotic perception can be divided into two kinds of types: passive perception and interactive perception. Passive perception involves observing and understanding the environment without any physical contact, such as human vision, smelling and listening. Take the vision sensor as an example, the robot can use camera images to recognize and localize the target objects intended to manipulate in a messy scene. Interactive perception, on the other hand, relies on the robot actively engaging with and altering its surroundings to gain a deeper or altered understanding such as haptic and tactile. For instance, a robot gripper can lift or push an object to estimate its weight. After using passive and interactive perception, each complicated environment that the robot interacts with can be regarded as a considerable structure. It consists of a collection of movable objects like apples, mugs, knives and plates, and stationary objects like walls and tables. By breaking down the high-dimensional 3D environment into individual objects and extracting their attributes, the robot can develop a feature representation of them. This approach facilitated the object feature and manipulation skill learning, enabling the robot to effectively apply its knowledge of similar objects across a variety of tasks.

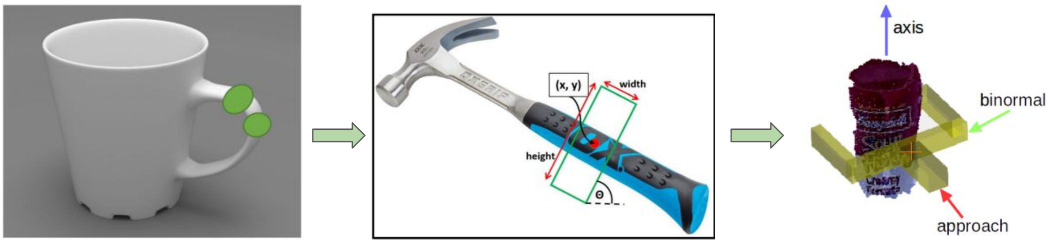
As the object feature learning, it can be represented in a hierarchical structure, encompassing levels that range from point-wise, part-wise and object-wise representations. This hierarchy transitions from finer detail to greater abstraction, reflecting the inherent geometric structure and components of the object. Geometric attributes capture the features of points, parts, and objects, while non-geometric attributes often define the nature or category of these elements. Beyond extracting individual object features, the robot is also capable of capturing the interactions among objects at various levels within this hierarchical framework, such as object-object, hand-object and hand-object-hand interaction. To accomplish a manipulation task effectively, the robot needs to further develop a policy based on the observation features, which usually involves a series of observation-action pairs with unknown numbers. For the simplest manipulation type *grasp*, the number of observation-action pairs is 1. It means that the gripper action is determined once the target object feature is well extracted and learned. However, for some challenged manipulation tasks like peg-in-hole assembly, many researchers will use reinforcement learning (RL) to learn a policy controller, where a fixed number of timesteps or a set of terminal states are defined to address the task. Recently, large foundation models based on language and vision have become a dominant paradigm in solving long-horizon robotic manipulation tasks, which indicates the potential that long-horizon manipulation tasks in our house life can also defined into discrete observation-action planning pairs in the prompt instructions to accomplish.

For reference, Section 2.1 introduces the drawbacks of robotic grasping of point-wise object feature learning from the raw partial point cloud, the recent progress of point cloud completion in the computer vision field and recent works in robot grasping using shape completion methods. Point-level representations provide the robot with the most flexible representations for capturing important details of objects and manipulation tasks. Section 2.2 further describes the concept of affordance based on the object's part-wise features and introduces state-of-the-art works in the fields of task-oriented grasping. Corresponding to sets of multiple contiguous points from the lower level of the hierarchy, part-level representations typically focus on parts associated with certain types of manipulations. For example, a *mug* can be seen as having an opening for *pouring*, a bowl for *containing*, a handle for *grasping*, and a bottom for *placing*. Section 2.3 emphasises the interactions or relations between the hand and the parts of different objects, where event vision is used to recognize the types of interactions. It mainly describes the traditional event-based dataset, the recent spiking neural network algorithms that are used to process event datasets, and some works about hand-object interaction recognition in traditional CMOS vision. Furthermore, for the manipulation skill learning, we first designed a novel but challenged multiple peg-in-hole assembly setup. Section 2.4 introduces the state-of-the-art work from RL-based robotic manipulation, sim2real transfer in robotic assembly, control strategies for RL-based manipulation, and previous robotic multiple peg-in-hole assembly. Finally, we leverage the diffusion models to endow the robot with an imaginative ability and utilize the vision-language model to assist the robot in accomplishing language-conditioned long-horizon rearrangement tasks. As a result, Section 2.5 discusses recent work from language-based manipulation, large foundation models, diffusion models and tabletop robotic rearrangement.

2.1 Shape Completion for Point Cloud Grasping

2.1.1 Grasping Basics from 2D to 3D

In most earlier works, grasps were represented as 2D points on images of actual scenes. Researchers used probabilistic models over possible grasping points to infer the grasp region, as shown in Figure 2.1(a). However, a significant drawback of such point-based grasps was that it only indicated where to grasp an object, without determining how wide the gripper should open or the required orientation. With the development of deep learning, many methods for deep visual grasping have been proposed. To overcome the drawback of 2D point representation, a new grasping configuration based on the rectangular shape is proposed, consisting of a Grasping center, Grasping orientation, Grip-



(a) 2D points (b) 2D rectangle (c) 3D point clouds

Figure 2.1: The evolution of different grasp representations for robotic grasping. (a) 2D points. (b) 2D rectangular shape [9]. Reprinted image: ©2015, IEEE. (c) 3D point clouds [14]. Reprinted image: ©2017, IEEE.

per width and height (See Figure 2.1(b)). Similar to 2D object recognition, rectangular representation from images was used to predict the grasp probability successfully [9]. In [10] and [11], a single RGB-D image of the target object was used to generate a 6D-pose grasp and effective end-effector trajectories by projecting the 2D rectangle to the 3D space. However, their works are not suitable for dealing with sparse 3D object information and spatial grasp. Compared with the 2D feature representations from images, 3D voxel or point cloud data could provide robotic grasping with more semantic and spatial information (See Figure 2.1(c)).

Given a synthetic grasp dataset, Breyer *et al.* [12] transformed scanning 3D object information into Truncated Signed Distance Function (TSDF) representations and passed them into a Volumetric Grasping Network (VGN) to directly output grasp quality, gripper orientation and gripper width at each voxel. [13] designed a special grasp proposal module that defines anchors of grasp centers and related 3D grid corners to predict a set of 6D grasps from a partial point cloud. Based on the scaled point cloud, Ten *et al.* [14] used hand-crafted outline features and a CNN-based method to build a grasp quality evaluation model and they also proposed a Grasp Pose Detection (GPD) algorithm to generate grasps using a sampling strategy. Based on the GPD algorithm, Liang *et al.* [4] used PointNet [15] to process point cloud for grasping evaluation, achieving to produce diverse grasp candidates of high quality. However, many generated grasps from GPD lack understanding of the surface and contours of objects on a physical interaction level, and it will cause lots of failures on larger and clearly outlined objects [16]. To address this problem, Mousavian *et al.* [17] introduced Variational Autoencoder (VAE) to generate grasp samples from point cloud, where generated successful grasps embody a certain understanding of the target object. However, due to the lack of complete geometric information on the object, we found that some grasp candidates are still infeasible and cause a collision with the object.

2.1.2 Dense Point Cloud Completion

The task of point cloud completion has been attracting more and more attention in the field of computer vision. Yuan *et al.* [18] firstly used Multi-Layer Perception (MLP) to extract the local geometric features of point clouds to accomplish the reconstruction. Groueix *et al.* [19] introduced a morphing learning strategy to generate different shapes of 3D surfaces, which shows great potential for point cloud and voxel reconstruction. Liu *et al.* [20] combined the above work and proposed a morphing and sampling network, which shows a higher fidelity and quality for the dense point cloud. Furthermore, Xie *et al.* [21] proposed a Gridding Residual Network to restore more structural details, especially for the dense point cloud. To summarise, most existing methods follow the encoder-decoder to reconstruct the complete point cloud from the learned contribution of shape prior, as shown in Figure 2.2. However, these methods cannot be applied to robotic research directly because all trained objects in their datasets are in a fixed pose and status. It's also much more challenging to restore the geometric details of the dense point clouds in real robotic tasks.

2.1.3 Shape Completion for Robotic Grasping

For robotic grasping, the critical challenge is recognizing objects in 3D space and avoiding potential perception uncertainty. When the RGB-D camera captures an object from a particular viewpoint, the 3D information on the object is incomplete, which means a lot of semantic and spatial information is missing. The missing complete 3D object information will lead to the grasp generation process generating wrong grasping poses.

Recently, researchers have proposed to use shape completion to enable robotic grasping. In [22], the observed object from 2.5D range sensors was firstly converted to occupancy voxel grid data. Then the voxelized data were input into a CNN and formed a high-resolution voxel output. Furthermore, the completion result was transformed into mesh and then loaded into Graspit! [23] to generate a grasp. Lundell *et al.* [24] used dropout layers to modify the network, which enabled the prediction of shape samples at run-time. Meanwhile, Monte Carlo Sampling and probabilistic grasp planning were used to generate grasp candidates. As traditional analytic grasping methods are computationally expensive, Lundell *et al.* [25] combined the shape completion of a voxel grid and a data-driven Grasping Quality Convolutional Neural Network (GQCNN) [26] to propose a structure called FC-GQCNN, where synthetic object shapes were obtained from a top-down physics simulator and grasps were generated from depth images. Traditional grasp-based shape completion solutions mainly concentrate on completing a single object from different camera views, while they hardly consider the lack of geometric information caused by occlusion from other objects.

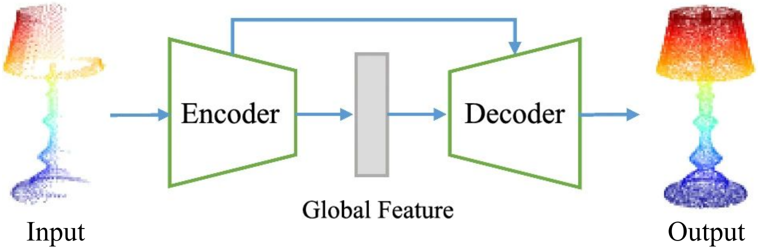


Figure 2.2: The encoder-decoder framework for dense point cloud completion.

2.2 Task-oriented 6-DoF Robotic Grasping

2.2.1 Affordance Basics

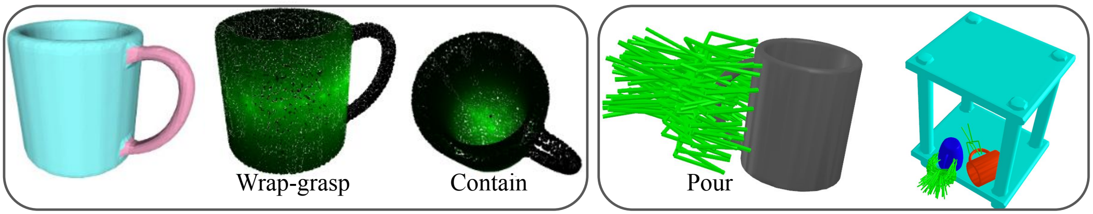
Gibson *et al.* [27] first introduced the concept of affordance, a term deeply ingrained in the interaction between agents and their environment, which signifies the potential actions an agent can execute within its surroundings. This concept, while abstract, plays a pivotal role in robotics, particularly in the identification and interaction with target objects. In essence, affordance in robotics involves perceiving an object, discerning the feasible actions associated with it, and understanding the consequences of these actions to determine if a task can be replicated. The application of affordances in robotics extends to visual and grasp affordances, each vital for the development of autonomous systems, shown in Figure 2.3.

In the realm of visual affordance, the concept has been integrated into semantic segmentation tasks as a type of labelling process [28]. Here, objects are identified not just by their physical characteristics but by the actions they enable—affordances. For instance, a cup may afford actions like ’pouring’, while a bed may be ’sittable’ and ’layable’. This perspective of affordance is crucial for autonomous systems to understand and interact with their environment effectively.

As the grasp affordance, it highlights the importance of context and previous experiences in establishing these affordance relations [29,30]. Humans excel at this, intuitively knowing that grasping a pair of scissors by the tip may be suitable for ’handing it over’ but not for ’cutting’. This principle has found new significance in robotics, where agents must manipulate novel objects for various tasks, each requiring a unique understanding of grasp affordances. In real-world applications, the multifaceted nature of objects means they afford multiple actions, with the success of a task hinging on the correct recognition and execution of the appropriate grasping region. By integrating affordance theory into robotics, we lay the foundation for creating more adaptable and intuitive autonomous systems. These systems, capable of complex decision-making and interaction with their environment, mark a significant step forward in the field of robotics and autonomous system development.

2.2.2 Task-oriented Grasping Detection

Widely accepted by the robotics community, the goal of affordance learning is to reason different physical and contextual meanings of objects [31]. The prediction of these con-



(a) Visual Affordance

(b) Grasp Affordance

Figure 2.3: The difference between visual affordance and grasp affordance.

textual affordances is thus a critical component in the complex problem of Task-oriented Grasping (TOG). It's very challenging to achieve affordance learning on the robotic scene, researchers proposed many approaches to learn different object parts through pixel-wise or point-wise features and corresponding semantic information [32–34].

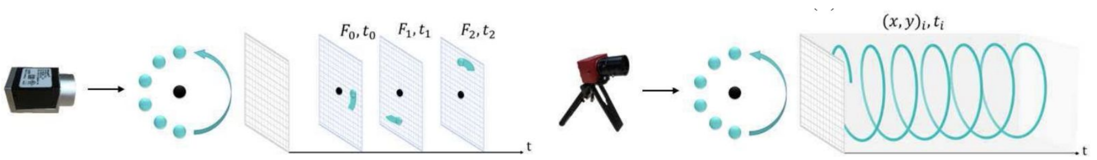
For the robotic grasping based on pixel-wise affordance, Vahrenkamp *et al.* [35] labelled different object parts as semantic information to guide the robot to grasp though it can only be applied to similar objects. To obtain a better generalization ability, Rezapour *et al.* [36] proposed to use data-driven approach to accomplish part-based affordance detection, which demonstrates robot could execute successfully after detecting the pixel-wise affordance. Furthermore, Liu *et al.* [37] proposed a context-aware grasping engine (database), which combines part affordance, part material, and tasks to train a semantic grasp network. That improves the relationship between grasping and objects though it cannot generate diverse grasp candidates automatically. On the basis of traditional pixel-wise part segmentation, Xu *et al.* [38] introduced an extra keypoint detection module, whose predictions consist of position, direction, and extent, guiding a more stable grasp pose. However, the problem with these pixel-based part affordance methods is that 6-DoF grasp detection is hard to embed in it, causing generated robotic grasp candidates in a very restricted workspace.

Only recently, some works started to study affordance-based learning on observed point clouds by extending semantic segmentation methods to the point-wise level. Hjelm *et al.* [39] used a demonstration method to learn tasks, specifically grasping based on visual point cloud. Ardon *et al.* [40] proposed a grasp affordance patch mapping method to generate optimal grasping region and then execute grasp while the whole execution process is cumbersome. Jiang *et al.* [41] proposed a GIGA framework to use implicit representation, jointly learning grasp affordance and 3D reconstruction. It achieves a great state-of-the-art grasping performance, while the weakness is that each object is related to a single affordance. Furthermore, Murali *et al.* [42] collected a TaskGrasp dataset by scanning real object point cloud and divided each object into diverse tasks, which also introduced graph knowledge to help task-oriented grasping generation. However, the grasps in this dataset are annotated purely through the geometric shape of the object, and assume that each affordance is true manually without considering the true context of object.

2.3 Event-based Action Recognition

2.3.1 Event Vision Basics and Related Datasets

Compared with a large amount of CMOS-based RGB datasets, few event camera-based datasets have been proposed to solve different challenging spatial-temporal recognition and estimation tasks [43–49], such as gesture classification, face recognition, pose estimation and driving recognition. Each pixel in an event-based camera independently and continuously monitors the intensity of light. When a change in light intensity surpasses a predefined threshold, the pixel generates an ”event”. The event camera is particularly advantageous in applications requiring high-speed motion detection, low latency, effi-



(a) Pixel changes from frame-based camera

(b) Event stream from event-based camera

Figure 2.4: The imaging difference between the frame-based camera and the event-based camera [51]. Reprinted image: ©2020, IEEE.

cient data handling, and operation in challenging lighting conditions. The visualization of the captured data format of the frame-based camera and the event-based camera is described in Figure 2.4.

DVS-CIFAR10 [43] is a classic neuromorphic dataset converted from the static CIFAR-10 format. Based on the transforming method of Repeated Closed-loop Smooth (RCLS) movement, it consists of 10000 event streams captured by an event camera with a resolution of 128×128 and contains 10 different spiking labels. Amir *et al.* [44] collected different moving hand gestures using a DVS128 event camera under different lighting conditions, where 29 volunteers were involved in the data collection, and 11 gesture classes were annotated. For human and hand pose estimation, Scarpellini *et al.* [45] and Rudnev *et al.* [50] developed a 3D human pose and hand pose dataset by utilizing the simulated events, respectively, which both enrich the human waking and hand moving scenarios. Event-based face recognition is another event-related research topic. Berlincioni *et al.* [46] and Lenz *et al.* [47] collected a rich annotation dataset based on different eye blinks and facial emotions, respectively. Moreover, Chen *et al.* [48] explored various datasets and algorithms of different autonomous driving tasks based on the neuromorphic vision sensor, which demonstrated that the event camera is becoming a valuable addition to conventional autonomous sensing modalities.

2.3.2 Learning from Spiking Neural Networks

Based on the gradient descent method, many Spiking Neural Network (SNN) models are developed to achieve high performance on event-based datasets. Among them, Hunsberger *et al.* [52] proposed the Leaky Integrate-and-fire (LIF) neuron model to make it easier to achieve ANN-based network architecture. Fang *et al.* [53] improved the LIF model by optimizing the membrane time constant and synaptic weight. Moreover, the attention mechanism was introduced to enrich the representation by focusing on the most informative elements of the input events. Cannici *et al.* [54] proposed two kinds of methods to utilize spatial-wise visual attention and demonstrate that they both can improve the effectiveness of Convolutional SNN to solve object recognition problems. Yao *et al.* [55] integrated the temporal-wise attention into SNN and found that the attention-score-based approach is beneficial to discard irrelevant features, yielding a notable reduction in computational cost. Yao *et al.* [56] further discussed the representation potential of different attention types and demonstrated that temporal-based, spatial-based, and channel-based attention can all facilitate the vanilla SNN backbone to achieve better performance.

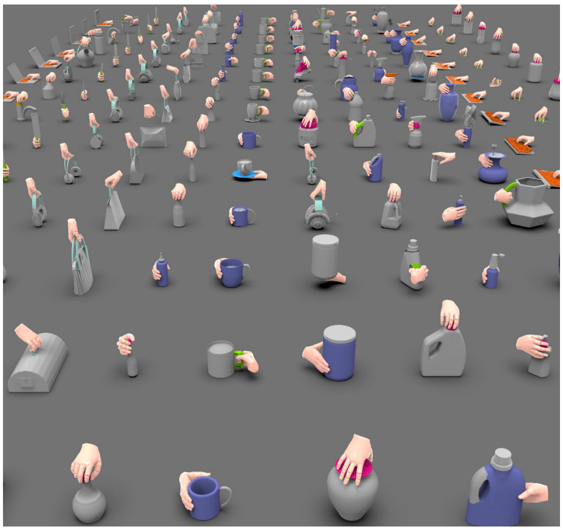


Figure 2.5: The illustration of different task-oriented hand-object relations [57]. Reprinted image: ©2023, IEEE.

2.3.3 Task-oriented Hand-object Action

Research on hand-object action tasks is parallel conducted in the robotics and computer vision fields. As robotics researchers, they pay attention to generating different object-grasping poses based on various tasks. Chen *et al.* [5] introduced a 3D affordance map to embed in the task-oriented 6-DOF grasp network, which greatly improves over a direct variational autoencoder (VAE) generation. Based on the event vision, [58,59] innovative collected a NeuroGrasp dataset and proposes a multimodal neural network to achieve grasp pose estimation successfully, showing a potential robotic application in the low latency environment. Researchers in the area of computer vision concentrate more on the 3D mesh and image relationship between the hand and the object. Hu *et al.* [60] adopted the MANO hand model and different objects to generate, reconstruct, and estimate hand-object action. Furthermore, [61–64] proposed different datasets to discuss hand-object action pose by capturing hand and object appearance. Taking AffordPose [57] as an example, the different hand-object relations are illustrated in Figure 2.5.

2.4 Transfer Learning in Robotic Assembly

2.4.1 RL Basics and Contact-rich Robotic Manipulation

Reinforcement Learning (RL) is a type of machine learning where an agent learns to make decisions by performing actions in an environment to maximize cumulative rewards. As shown in Figure 2.6, each control task can be assumed as the robot (agent) interacting with the environment. During the interaction process, the robot generates a series of actions according to its states. To make the robot accomplish the task, RL introduces a concept of reward. It represents a scalar feedback signal given to the robot after acting, guiding the learning process. When the robot finishes the learning process,

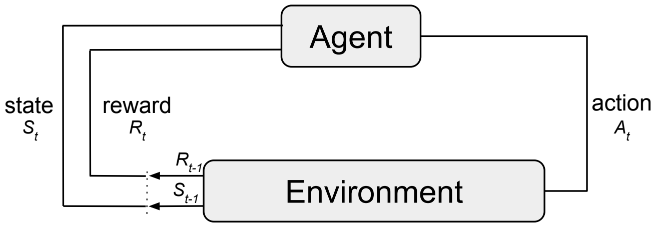


Figure 2.6: Basic concepts about reinforcement learning.

it will get a policy, which represents a strategy that the robot follows to determine its actions based on the current state.

Many RL-based methods have been developed and applied to solve different contact-rich robotic manipulation tasks, such as grasping, door opening, stacking, polishing, and assembly [65–72]. Liang *et al.* [65] used multi-modal perception to solve the complicated multi-finger grasping problem. Nemec *et al.* [66] used structure research combined with reinforcement learning to solve the physical constraint of the door opening problem. Furthermore, Englert *et al.* [67] combined constrained optimization, Bayesian optimization, and inverse optimal control to produce a high-dimensional policy using a few demonstration trials. To improve the stability and reliability of stacking problems, Lin *et al.* [73] introduced the extra force and torque perception to improve the policy performance. To avoid the polishing force signal suffering overshoot during the manufacturing process, Zhang *et al.* [74] proposed a press-release model to compensate for the robot deformation, and a model-based reinforcement learning algorithm was used to optimize the processing parameters. Automatic high-precision assembly is the most challenging case within RL for smart manufacturing, which is divided into two types: peg-in-hole insertion and electric connector bonding. Inoue *et al.* [70] firstly realized single peg-in-hole precision insertion skills through the recurrent neural network with reinforcement learning, showing great robustness against position and angle errors. Schoettler *et al.* [71] combined off-policy reinforcement learning with visual images to solve the industrial connector bonding task. These results have shown that RL is a feasible technique for handling contact-rich robotic manipulation tasks.

2.4.2 Sim2real Transfer in Robotic Assembly

For robotic assembly problems, obtaining real-world data for learning-based approaches is expensive. Especially, the RL algorithms need a large amount of data to sample and explore. Thus, learning from the simulation environment is a more efficient way. Chebotar *et al.* [75] changed the distribution of simulation using a few real-world roll-outs, demonstrating the trained policy can be reliably transferred to the real peg-in-hole assembly task. Beltran *et al.* [76] also used sim2real, domain randomization, and RL to bootstrap the training speed for position-controlled peg-in-hole tasks. Hebecker *et al.* [77] learned the compliant assembly representation in simulation and achieved a 90% success rate over the limited variations of goal position. Furthermore, Thomas *et al.* [78] leveraged the prior knowledge about geometric motion learning from the CAD model data with guiding RL, demonstrating a high precision, even without accurate state estimation. Multimodal learning was also proposed to combine with the Sim2real technique for the robotics assembly task. Wu *et al.* [79] proposed to learn dense rewards for multimodal observations to execute USB insertion, demonstrating better performance than sparse reward baselines. Lee *et al.* [80] combined the visual and haptic feedback to learn a multimodal representation in simulation. Real-world experiments showed that it could generalize over varying geometries, configurations, and clearances. However, sim2real transfer in multiple peg-in-hole assembly has not been fully considered.

2.4.3 Control Strategies for RL-based Contact-rich Tasks

Since simultaneous position and force control in the same axis for robot manipulation is quite challenging, an impedance strategy is developed to achieve a trade-off between compliance and precision for contact-rich manipulation tasks. It also has been sufficiently validated that impedance control is suitable for solving different robotic contact-rich tasks [81–87]. Luo *et al.* [88] incorporated force/torque information into RL and proposed a simple neural network to generalize the trained policy for the assembly gear task. Not randomly tuning the impedance parameters, Bogdanovic *et al.* [89] proposed to use RL to learn the optimal impedance parameters, which was validated successfully by assembling an HDMI connector in the simulation. For the single peg-in-hole assembly, Kulkarni *et al.* [90] proposed to use output action from the recurrent RL method as the reference of the operational impedance controller, where over 80% success rate was achieved under various object characteristics. Furthermore, Bogdanovic *et al.* [89] compared the policy performance by combining it with different position and force controllers, such as PD controller from fixed and variable gain, and torque controller from force signal directly, and finally facilitated them from real to reality. Based on that, Yang *et al.* [91] used a few real demonstration trajectories to train the RL framework with the variable impedance controller and showed that the learned variable impedance action could be deployed in real-world experiments. Similar to the impedance control, Spector *et al.* [92] proposed an admittance policy based on a residual strategy to correct the movements from baseline policy, which also achieved a good generalization performance over object side, shape, and space for the single peg-in-hole assembly. Inspired by these works, we implement impedance control for the multiple peg-in-hole assembly task in this work. Furthermore, we also aim to verify the performances of the RL policy under different control strategies.

2.4.4 Robotic Multiple Peg-in-hole Assembly

Since traditional single peg-in-hole assembly tasks have been extensively researched and validated based on RL, robotic multiple peg-in-hole assembly seems to be a more challenging problem. Because the increase in the number of pegs and holes will decrease the stability and reliability of the position and force from the output policy. Especially considering the dimension of rotation, when one peg is aligned exactly, other pegs may fail to enter the hole due to the change of angle. [93–95] described the geometric and force/torque characteristics for 3D Multi-peg-assembly (MPA) problem based on detailed theory and mechanical experiments, which demonstrates it has more complicated and dynamic contact states than single peg-in-hole assembly. Recently, some researchers have begun to use RL to solve this task. Hou *et al.* [96] firstly proposed a hybrid exploration strategy based on Deep Deterministic Policy Gradien (DDPG) with PD force controller, where simulation and real experiments both verified the methods. To improve learning efficiency, Xu *et al.* [97] utilized a feedback exploration and fuzzy reward to process the learning, demonstrating the improvement of performance. Based on the same setup, Li *et al.* [98] proposed a LADDPG approach to accelerate the learning process, where a safety assurance mechanism and adaptive impedance control were used to improve the performance. The latest work is that [99] modified their previous method [96] and introduced the time-scale prediction to reduce unnecessary policy exploration, and a fuzzy logic system (FLS) was introduced to map impedance parameter tuning.

Though these RL-based works on multiple peg-in-hole assembly work well in simulation and the real world, their experimental setup does not fit well with the accurate mechanical definition of multiple peg-in-hole assembly [93–95]. Lacking of visual feedback, their setup is also difficult to develop and apply to the real manufacturing industry.

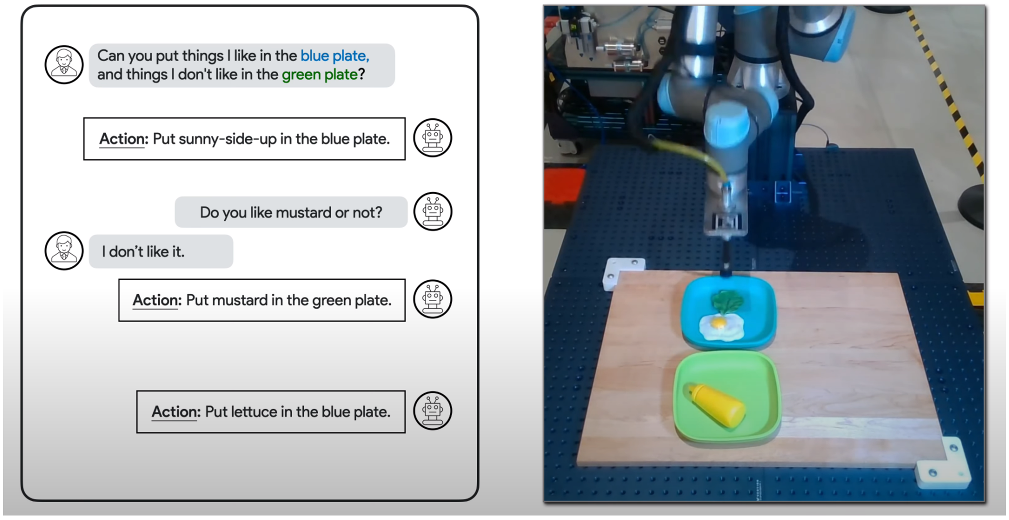
2.5 Language-conditioned Robotic Arrangement

2.5.1 Language-based Manipulation Basics

Language is a flexible and instinctive medium, enabling humans to specify tasks, communicate contextual details, and express their intentions. Much work about language-conditioned robot manipulations has been proposed to control a robot by generating low-level policies via RL or Imitation learning (IL) [100–103]. Jiang *et al.* [104] proposed to use language as abstract representations of hierarchical RL framework, demonstrating that the agent can learn compositional tasks like object sorting and multi-object arrangement in a simulation environment. Furthermore, Misra *et al.* [105] designed a novel RL agent that directly mapped language instructions and raw visual input to generate a sequence of actions without requiring intermediate representations and planning procedures. However, language-conditioned RL methods are difficult to deploy into real physical robots due to the challenge of learning the relationship between language and multimodal sensor data in the unstructured robot environment. To further improve learning efficiency, other researchers adopt the language-conditioned IL approaches, where agents are trained to perform tasks by mimicking the actions demonstrated by a human expert. Focusing on the *containing* task, Lynch *et al.* [106] first proposed a language-conditioned visuomotor policy utilizing unstructured and unlabeled data collected from a teleoperated robot in a physics simulator. Stepputtis *et al.* [107] further integrated the low-level motion controller into the language-conditioned learning framework. Both test results indicated that the robot can hopefully accomplish long-horizon tasks in the simulation environment. However, these IL-based methods require a large and diverse set of high-quality demonstration data. Acquiring such data on actual robots is a process that demands considerable time and resources. Contrary to prior efforts in language-conditioned research, our work emphasizes the utilization of language instructions to steer the denoising diffusion process, where the target states of objects will be estimated and used for the subsequent robot planning.

2.5.2 Large Foundation Models in Robotics

Recently, large foundation models based on language and vision have become a dominant paradigm in solving long-horizon robotic manipulation tasks [109–113]. They demonstrate strong few-shot or zero-shot reasoning ability to any text or vision input through just prompting by human instructions [7]. SayCan [110] used a large language model (LLM) to perform various tasks, where language objectives were destructed into a hierarchical sequence of instructions. These instructions were subsequently fed into skill-oriented value functions and search heuristics to obtain optimal action sequences. Informed by multimodal prompting, Socratic Models [111] exhibited a modular framework to capture multimodal information and leveraged LLMs to achieve zero-shot robotic perception and planning. Furthermore, Code-as-Policies [112] adopted the LLMs to generate a policy code of robot action, showing LLMs have a strong programming ability in controlling robots by recomposing perception and controller API functions. Utilizing the capability to generate codes, Huang *et al.* [113] used LLMs to integrate 3D value maps into the robotic observation space after inferring affordances and constraints from language instructions, which produced low-level control on the contact-rich manipulation tasks successfully. Nevertheless, the final goal states of each robot task from previous work on LLMs remain predominantly predefined, relying on human expertise or demonstrative guidance encapsulated within the prompt instructions. An example of multi-step robot arrangement using language language models is shown in Figure 2.7.



Can you put things I like in the blue plate, and things I don't like in the green plate?

Action: Put sunny-side-up in the blue plate.

Do you like mustard or not?

I don't like it.

Action: Put mustard in the green plate.

Action: Put lettuce in the blue plate.

Figure 2.7: Multi-step robot arrangement via language language models [108]. Reprinted image: ©2023, IEEE.

2.5.3 Diffusion Models

In the computer vision field, diffusion models have risen to prominence as leading generators of data, distinguished by their ability to accurately model complex distributions and generate a diverse array of high-quality samples [114,115]. The concept draws inspiration from the physical phenomenon of diffusion, where particles migrate from regions of higher concentration to lower concentration until a state of balance is achieved. Many applications from diverse domains, such as text, image, audio, and video, demonstrate that diffusion models can significantly improve the quality, realism, and creativity over previous generative models [116, 117]. Especially for text-to-image diffusion models, their groundbreaking synthesis capabilities using textual descriptions can significantly enhance creation efficiency [118,119]. However, these models offer limited control over the content they generate, primarily achieved through a single text-based input modality. Some techniques have been developed to enhance performance and gain more precise control using various input types, such as contextual layouts and class labels. These techniques strive to finely tune the creation of content by adjusting the generation process based on pre-trained models [120–122]. Taking an example of the inpainting task, Avrahami *et al.* [122] proposed a solution to achieve image inpainting successfully by leveraging a pre-trained vision-language model like CLIP [114], where the inpainting process was guided from a text description along with an ROI mask. Figure 2.8 describes a robotic example using diffusion models. Kapelyukh *et al.* [123] proposed to utilize DALL-E, a web-scale artificial intelligence-generated content (AIGC) model, to generate a target image that implicitly incorporates various objects the robot observes. Nevertheless, the exclusive reliance on textual input for image generation has proven to be notably unstable and inefficient in real-world robot manipulation, primarily due to the neglect of crucial observational cues.

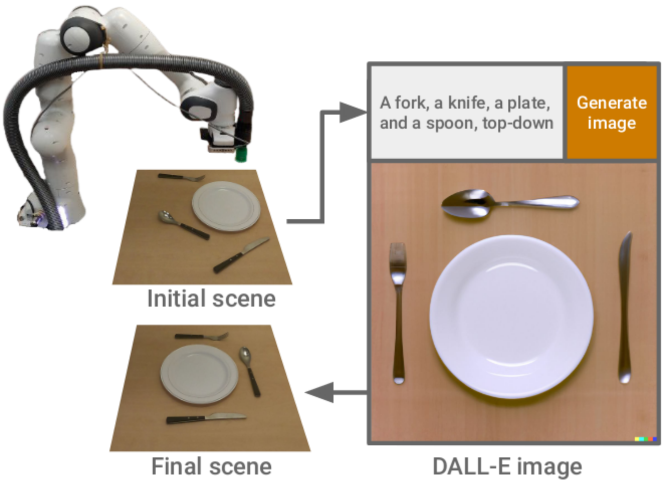


Figure 2.8: The overview of DALL-E Bot system. DALL-E is used to generate an image of a human-like arrangement of those objects [123]. Reprinted image: ©2023, IEEE.

2.5.4 Tabletop Robot Rearrangement

The objective of an intelligent robotic rearrangement system is to equip robots with the ability to understand their surroundings and interact with humans, thereby achieving precise and efficient object repositioning according to different structures or criteria that reflect human preferences [124]. Various approaches have been explored to tackle this challenge. Typically, Tang *et al.* [125] proposed to utilize an RL strategy based on the Proximal Policy Optimization (PPO) algorithm to push irregular objects on the table inside a crate, which was hard to generalize to other rearrangement tasks because of the fixed position of the crate on the table. To improve the generalizability, VIMA [126] introduced prompt-based learning to train a multimodal generalist agent, achieving a simple zero-shot robot arrangement setting in the simulation environment. Nevertheless, it was still difficult to deploy in real robot experiments due to the lack of human-designed visual prompts. Moreover, Liu *et al.* [124] first introduced the concept of semantic structure in the robot arrangement task, which necessitated a robot's ability to understand the relationships between scattered objects and subsequently rearrange them into a spatial structure instructed by human languages. However, the efficiency was compromised by its sequential processing, where the goal state of the current object was estimated only after finishing the arrangement of the previous object. To address this inefficiency, StructDiffusion [127] implemented a 3D diffusion-based approach based on the same dataset, achieving a better rearrangement performance. However, we found that the predicted object states for a given structure demonstrate negligible layout adaptability on the table when the initial messy observation and motion distance of objects between the messy scene and the rearranged scene are not taken into account. This issue largely results from all target object states being derived from predetermined Gaussian noise throughout the denoising diffusion process. Moreover, the inherent design of the dataset presents challenges in enabling scattered objects to form varied structures upon completion of the rearrangement process.

Chapter 3

Transformer-based Shape Completion for Robotic Grasping

Currently, robotic grasping methods based on sparse partial point clouds have attained excellent grasping performance on various objects. However, they often generate wrong-grasping candidates due to the lack of complete geometric information on the object. In this chapter, we propose a novel and robust sparse shape completion model (TransSC). This model has a transformer-based encoder to explore more point-wise features and a manifold-based decoder to exploit more object details using a segmented partial point cloud as input. Quantitative experiments verify the effectiveness of the proposed shape completion network and demonstrate that our network outperforms existing methods. Besides, TransSC is integrated into a grasp evaluation network to generate a set of grasp candidates. The simulation experiment shows that TransSC improves the grasping generation result compared to the existing shape completion baselines. Furthermore, our robotic experiment shows that with TransSC, the robot is more successful in grasping objects of unknown numbers randomly placed on a support surface.

3.1 Introduction

Robotic grasping evaluation is a challenging task due to incomplete geometric information from single-view visual sensor data [128]. Many probabilistic grasp planning models have been proposed to address this problem, such as Motel Carlo, Gaussian Process and uncertainty analysis [11,24,129]. However, these analytic methods are always computationally expensive. With the development of deep learning techniques, data-driven grasp detection methods have shown great potential [4, 12–14] to solve this problem. They generate lots of grasp candidates and estimate the corresponding grasp quality, resulting in better grasp performance and generalization. However, as most of these methods still rely on original sensor inputs like 2D (image) and 2.5D (depth map), there exists a physical grasping defect when the gripper interacts with real object surfaces or edges because of the incomplete pixel-wise and point-wise representations. Otherwise, traditional data-driven grasping algorithms [4, 14, 17] are mostly based on the partial point clouds. Due to the object's missing geometric and semantic information, these algorithms easily generate wrong grasp candidates and cause a research gap.

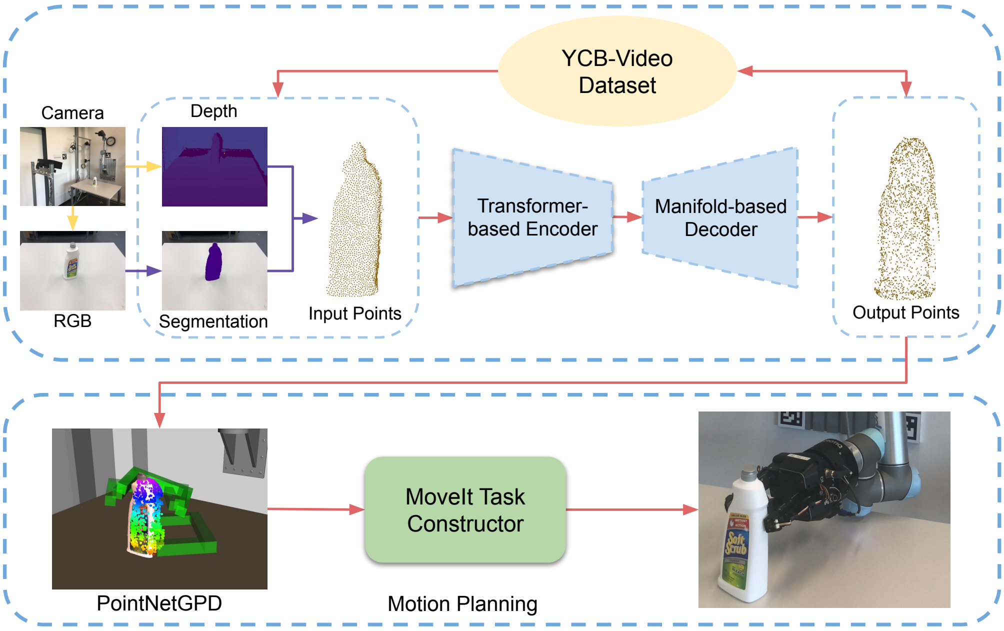


Figure 3.1: Overview of our shape completion-based grasp pipeline. The top row shows the shape completion module. In this module, a segmented partial point cloud ζ_p with n points is first input into a transformer-based encoder to extract point-wise and self-attention features, which outputs a latent vector with m dimensions. Then, the latent vector is concatenated with another latent feature from a flat/spatial point seed generator to predict multiple spatial surfaces in the manifold-based decoder. Finally, these surfaces are assembled into a complete point cloud ζ_c. The bottom row is the grasp evaluation module, and the complete point cloud ζ_c is the input of our grasp detection pipeline PointNetGPD to compute the grasp quality $\mathcal{Q}_i$. The grasp with the highest score $\mathcal{G}_{best}$ will be sent to the MoveIt task constructor to calculate a collision-free trajectory and will be executed in a real robot experiment.

To improve grasp performance, the sparse point cloud is necessary to be restored or repaired to generate a better grasping interaction. Additional sensor input such as a tactile sensor can be regarded as a supplement of original vision sensing [130]. However, object uncertainty still exists and extra sensor interference with the object will directly affect the final grasping result. Another strategy is to use shape completion to infer the original object shape while traditional grasping-based shape completion methods use a high-resolution voxelized grid as object representation [22, 24, 25], causing a high memory cost and information loss due to the sparsity of the sensory input. To avoid extra sensor costs and obtain complete object information, a novel transformer-based shape completion module is proposed in this chapter based on an original sparse point cloud. Compared with the traditional convolutional network layer, the transformer has achieved state-of-the-art results in visual recognition and segmentation [131, 132], which enables our shape completion module to achieve better performance.

As illustrated in Fig. 3.1, we present a novel grasping pipeline that uses a sparse point cloud to execute the grasp directly, without converting it into discrete voxel grids during the shape completion process and then transforming it into a mesh in the grasp planning process. The pipeline consists of two sub-modules: The transformer-based shape completion module and the grasp evaluation module. In the first module, a non-synthetic segmented partial point cloud dataset based on YCB objects was constructed. Not cropping the object randomly or viewing the object in a physical simulator, our dataset contains many real cameras and environmental noise, which guarantees an improved grasping interaction in a real robot environment. Based on this dataset, we propose a novel point cloud completion network (TransSC), where the segmented partial point cloud of an object is input, and the complete point cloud is output. In the second module, our previous work [4] is involved. We use PointNet [15] to obtain feature representations of the repaired point cloud, and build a grasp detection network to generate and evaluate a set of grasp candidates. The grasp with the highest score will be executed in a real robot experiment. The proposed pipeline is validated in a simulation experiment and robotic experiments, which both demonstrate that our shape completion pipeline can significantly improve grasping performance.

Our contributions in this chapter can be listed as:

- A large-scale non-synthetic partial point cloud dataset is constructed based on the YCB-Video dataset. As the dataset is based on 3D point cloud data captured by a real RGB-D camera, the noise that comes from it will facilitate the generalization of our work, especially in real robot environments.

- A novel point cloud completion network TransSC is proposed. The transformer-based encoder and manifold-based decoder are introduced into the shape completion task to improve its performance.

- Combining our previous work PointNetGPD for grasp evaluation and the MoveIt Task Constructor for motion planning, we demonstrate that a robust grasp planning pipeline using the shape completion result as input can achieve a higher grasp success rate compared to the single view work without shape completion.

3.2 Problem Formulation

We consider a setup consisting of a robotic arm with parallel-jaw grippers, an RGB-D camera, and objects of unknown number that are set on a flat support surface while we define a target object via user's input. Meanwhile, we assume that the RGB-D camera captures the depth map of objects, where a semantic segmentation network is used to extract the mask of the target object and convert it into a 2.5D partial point cloud $\mathcal{P} \in \mathcal{R}^{N\times3}$. For simplicity, all spatial quantities are in camera coordinates.

Given a gripper configuration $\mathcal{C}$ and camera observation $\mathcal{O}$, our goal is firstly to extract the target object point cloud $\mathcal{P}$ using semantic segmentation. Then a point cloud completion network is used to repair the segmented 2.5D partial point cloud $\mathcal{P} \in \mathcal{R}^{N\times3}$, turning it into a complete 3D point cloud $\mathcal{P}_c \in \mathcal{R}^{N\times3}$. After that, a grasp evaluation network based on $\mathcal{P}_c$ is used to predict a set of grasp candidates $\mathcal{G}_i$ and compute the relative grasp quality $\mathcal{Q}_i$. The grasp with the highest score $\mathcal{G}_{best}$ and highest kinematic possibility, *i.e.*, a collision-free grasp, will be executed in the real robot experiment.

3.3 Shape Completion Dataset

3.3.1 Dataset Construction

Traditional shape completion models use synthetic CAD models from the ShapeNet [133] or ModelNet [134] datasets to generate partial and corresponding complete point cloud data, while these synthetic data contain no real-world noise. As a result, synthetic data often do not work well in the real world. To tackle this problem, we summarize a shape completion dataset from the YCB-Video Dataset [2]. Non-synthetic RGB-D video images ($\sim$ 133,827 frames) in the YCB-Video Dataset are first chosen, while most of them vary insignificantly. Thus, a preprocessed image dataset is obtained by reducing every five frames. Meanwhile, to cover distinguishable shapes with different levels of detail, 18 objects are also chosen from the YCB-Video dataset. In this work, the ground-truth point cloud of 18 objects is created by the farthest point sampling (FPS) of 2048 points on each object model. Not randomly sampling or cropping complete point clouds on the unit sphere to get partial point clouds, RGB-D images and related object label images in the preprocessed dataset are loaded to compute the matching partial point clouds using related camera intrinsic parameters. To approximate the distribution of point cloud data of real objects and retain the semantic information, a large number of cameras and environmental noise data are kept on, though a small radius filter is used to remove partial outliers. For the convenience of network training, the partial point clouds are also unified into the size of 2048 points by FPS or replicating points. To enable an accurate comparison with existing baselines, the canonical center of the partial point cloud of each object is also transformed into the same center of the ground-truth point cloud using pose information. Finally, more than 70,000 partial point clouds are collected in our dataset. Compared to other synthetic point cloud datasets, our dataset does well at preserving the real point cloud distribution of occluded objects.

3.3.2 Semantic Segmentation

As shown in Fig. 3.1, the scene of our grasping task is that objects of unknown number are set on a flat support surface. To obtain the target object point cloud, we first build a semantic segmentation network branch, where different YCB objects are assigned a particular semantic label value. It can be seen that the performance of the segmentation network is good enough that it can also be deployed in a grasping task of multi-object occlusion.

Our segmentation network [135] takes an RGB image as input and outputs a binary mask of the expected object. The network has an encoder-decoder architecture based on CNN, where the encoder consists of 13 convolutional layers with ReLU activation followed by max-pooling layers. At the same time, the decoder utilizes upsampling operations whereby the pooling indices from the corresponding encoder layers are recalled. Moreover, several data augmentation strategies like adjusting brightness, contrast and saturation are used to make the network generalize well. After getting the expected object mask, the sparse 2.5D point cloud $\mathcal{P} \in \mathcal{R}^{N \times 3}$ of the target object could be extracted through the corresponding depth image. Meanwhile, we also remove the

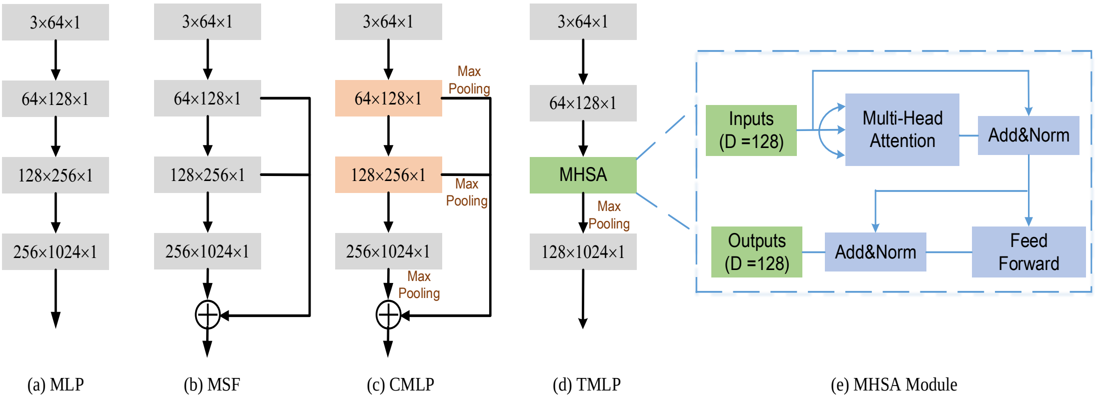


Figure 3.2: Illustration of various encoder structures for point cloud completion. (a) is a simple multiple-layer perception (MLP) structure. (b) is a multi-scale fusion (MSF) module, which can fuse features from different layers directly. (c) is concatenated multiple layer perception (CMLP), which can also concatenate multi-dimensional latent features while the max pooling operation is used to extract latent features further. (d) shows our Transformer-based multiple layer perception (TMLP) module, which integrates the Multi-head Self-attention (MHSA) module into the MLP structure. (e) depicts the architecture of the MHSA module.

redundant background (support surface) point cloud by setting a threshold value of the z-axis (support surface height).

3.4 TSC-Net Architecture

3.4.1 Transformer-based Encoder Module

As shown in Fig. 3.2, we compare our proposed encoder module with several common competitive methods. Multi-layer Perception (MLP) is a simple baseline architecture to extract point features. This method maps each point into different dimensions and extracts the maximum value from the final K dimensions to formulate a latent vector. A simple generalization for MLP is to combine semantic features from a low-level dimension with those of a high-level dimension. The MSF (Multi-scale Fusion) [136] module inflates the dimension of the latent vector from 1024 to 1408 to obtain semantic features from different dimensions.To enhance the performance of the feature extractor, L-GAN [137] introduced the use of a Maxpooling layer effectively. Similarly, Concatenated Multiple Layer Perception (CMLP) [138] applied multiple Maxpooling operations to the outputs of the last k layers, ensuring that multi-scale feature vectors are concatenated directly. An overview of our proposed Transformer-based multi-layer perception (TMLP) module is shown in Fig. 3.2(d). Without an extra skip connection structure and a Maxpooling operation from different layers, the Multi-head Self-attention (MHSA) [139] module is introduced to replace the traditional convolutional layer $[128 \times 256 \times 1]$.

MHSA aims to transform (encode) the input point feature into a new feature space, which contains point-wise and self-attention features. Fig. 3.2(e) shows a simple MHSA architecture used in TMLP, which includes two sub-layers. In our first layer, the multi-head number is set to 8 and the input feature dimension for each point is 128. Unlike natural language processing (NLP) problems, the 128-dimensional feature vector $\mathcal{A}_{in} \in \mathcal{R}^{2048 \times 128}$ will enter into the multi-head attention module directly without positional encoding. This is because each point in the point cloud has its unique $x - y - z$ coordinates. The output feature $\mathcal{Z}$ is formed by concatenating the attention of each attention head. A residual structure is also used to add and normalize the output feature $\mathcal{Z}$ with $\mathcal{A}_{in}$. This process can be formulated as follows:

$$\mathcal{A}_i = SA_i(\mathcal{A}_{in}) \quad i = 1, 2, ..., 8 \tag{3.1}$$

$$\mathcal{Z} = concat(\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_8) * W_0 \tag{3.2}$$

$$\mathcal{A}_{out} = Norm(\mathcal{A}_{in} + \mathcal{Z}) \tag{3.3}$$

where SA_i represents the i-th self-attention layer, each has the same output dimension size with input feature vector $\mathcal{A}_{in}$, and W_0 is the weight of the linear layer. $\mathcal{A}_{out}$ represents the output point-wise features of the first sub-layer.

The second sub-layer is called the Feed-forward module, which is a fully connected network. Point-wise features $\mathcal{A}_{out}$ are processed through two linear transformations and one ReLU activation. Furthermore, a residual network is also used to fuse and normalize the output features. Finally, we can get the MHSA module output $\mathcal{FF}_{out} \in \mathcal{R}^{2048 \times 128}$ as:

$$\mathcal{FF} = ReLU(\mathcal{A}_{out} * W_1 + b_1) * W_2 + b_2 \tag{3.4}$$

$$\mathcal{FF}_{out} = Norm(\mathcal{A}_{out} + \mathcal{FF}) \tag{3.5}$$

where W_1, W_2 and b_1, b_2 represent the weight and bias value of the corresponding linear transformation, respectively.

3.4.2 Manifold-based Decoder Module

Inspired by the AtlasNet [19], a manifold-based decoder module is designed to predict a complete point cloud from partial point cloud features. As shown in Fig. 3.3, a complete point cloud can be assumed to consist of multiple sub-surfaces. Therefore, we only concentrate on generating each sub-surface, then we gather them and make an appropriate montage to form the final complete point cloud. To obtain each sub-surface, a point seed generator is used to concatenate with global feature vector $\mathcal{P}_g \in \mathcal{R}^{2048 \times 1024}$ output from the encoder, where point initialization values are computed from a flat (f) or spatial (g) sampler. As the coordinate values of the ground-truth point cloud are limited to between [-1, 1], point initialization values are also limited in this range. After that, the concatenated feature vector $\mathcal{P}_{concat} \in \mathcal{R}^{2048 \times M}(M = 1026 \text{ or } 1027)$ is input into K convolutional layers, where all sampled 2D or 3D points will be mapped to 3D points on each sub-surface. In our decoder, the sub-surface number is set to 16. Unlike other voxel-based shape completion methods, our decoder module achieves an arbitrary resolution for the completion results.

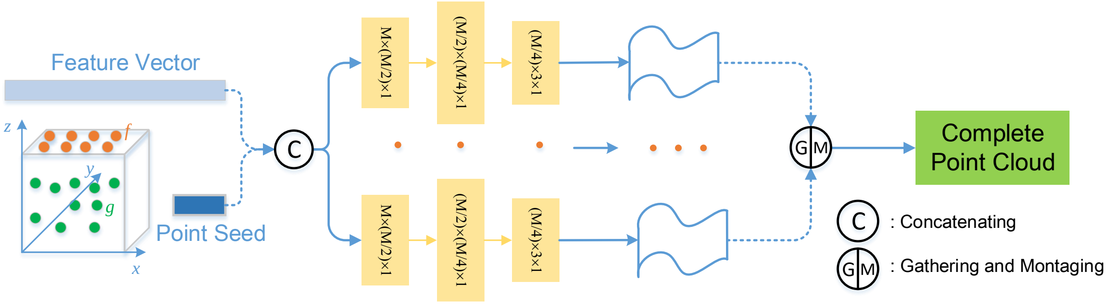


Figure 3.3: Illustration of the decoder structure for point cloud completion. The feature vector with m dimensions from the encoder is firstly concatenated with a latent feature from a special point seed generator f or g. Then three convolutional layers as the backbone are used to extract features and form different manifold-based surfaces, respectively. Finally, these surfaces are gathered and montaged into a complete point cloud.

Evaluation Metrics To evaluate our shape completion results, we used two permutation-invariant metrics called Chamfer Distance (CD) and Earth Mover's Distance (EMD) as our evaluation goal [140]. Given two arbitrary point clouds S_1 and S_2, CD measures the average distance from each point in one point cloud to its nearest point coordinates in the other point cloud.

$$d_{CD}\left(S_1, S_2\right) = \frac{1}{S_1}\sum_{x\in S_1}\min_{y\in S_2}\|x-y\|_2^2 + \frac{1}{S_2}\sum_{y\in S_2}\min_{x\in S_1}\|y-x\|_2^2 \tag{3.6}$$

While Earth Mover's Distance considers two equal point sets S_1 and S_2 and is defined as:

$$d_{EMD}\left(S_1, S_2\right) = \min_{\emptyset:S_1\to S_2}\frac{1}{S_1}\sum_{x\in S_1}\|x-\emptyset(x)\|_2 \tag{3.7}$$

The CD has been widely used in most shape completion tasks because it is efficient to compute. However, EMD is chosen as our completion loss because CD is blind to some visual inferiority and ignores details easily [137]. With $\emptyset : S_1 \to S_2$ being bijective, EMD could solve the assignment and transformation problem in which one point cloud is mapped into another.

3.4.3 Grasping Detection Module

Given the complete point cloud from the previous steps, we put the point cloud into a geometric-based grasp pose generation algorithm (GPG) [141], which outputs a set of grasp proposals $\mathcal{G}_i$. We then transform $\mathcal{G}_i$ into a gripper coordinate system and use points inside the gripper as the input of PointNetGPD [4], a data-driven grasp evaluation framework. The output grasp will then be sent to the MoveIt Task Constructor [142] to plan a feasible trajectory for a pick-and-place task.

PointNetGPD [4] is trained on a grasp dataset generated using a reconstructed YCB object mesh and evaluates the input grasp quality. The grasp candidates in the grasp dataset all proceeded collision-free to the target object. As a result, the grasp evaluation

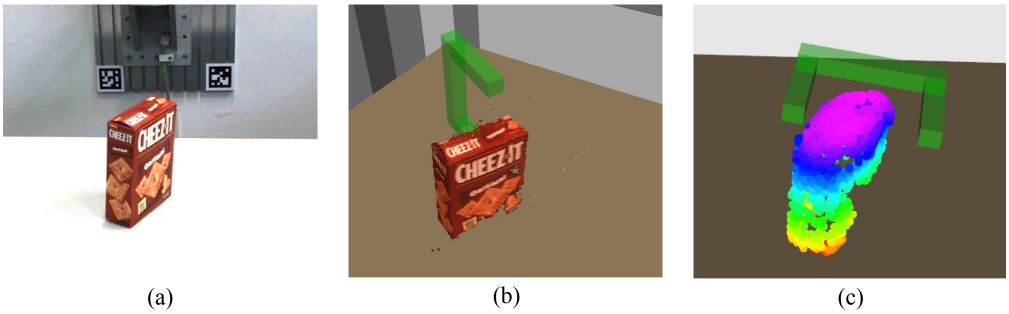


Figure 3.4: Comparison of grasp candidates generated using GPG. (a) RGB image to show the example environment, (b) grasp generated with the partial point cloud, (c) grasp generated with the complete point cloud.

network assumes that all the input grasp candidates are not colliding with the object. If the object has occlusion due to the camera viewpoint, the current geometric-based grasp proposal algorithm will generate grasp candidates that collide with the object. Thus, using a complete point cloud can ensure that the grasp candidate generation algorithm generates grasp sets that do not collide with the graspable objects. Fig. 3.4 shows the comparison of the grasp generation result using GPG [141] with and without point cloud completion, where Fig. 3.4(b) shows a candidate generated using a partial point cloud and Fig. 3.4(c) shows a grasp candidate generated using a complete point cloud. We can see that the grasp in Fig. 3.4(b) collides with the real object while Fig. 3.4(c) avoids generating that kind of grasp.

3.5 Simulation Experiments

Training and Implementation details To evaluate model performance and reduce training time, eight categories of different objects in our dataset are chosen to train the shape completion model. The training set and validation set are split into 0.8:0.2. We implement our network on PyTorch. All the building modules are trained using the Adam optimizer with an initial learning rate of 0.0001 and a batch size of 16. All the parameters of the network are initialized using a Gaussian sampler. Batch Normalization (BN) and ReLU activation units are all employed at the encoder and decoder module except the final tanh layer producing point coordinates, and Dropout operation is used in the MHSA module to suppress model overfitting.

3.5.1 Quantitative Evaluation

In this subsection, we compare our method against several representative baselines that are also used for point cloud completion, including AtlasNet [19], MSN [20] and GRNet [21]. The Oracle method means that we randomly resample 2048 points from the original surface of different YCB objects. Corresponding EMD and CD distances between the resampled point cloud and the ground-truth point cloud provide an upper bound for the performance. Relative comparison results are shown in Table 3.1 and Ta-

Table 3.1: Comparison of Earth Mover's Distance with different sparse point cloud completion models for 2048 points and multiplied by 10^3

Model	**cracker box**	**banana**	**pitcher base**	**bleach cleanser**	**bowl**	**mug**	**power drill**	**scissors**	**average**
Oracle	3.4	1.7	4.6	2.9	1.9	2.0	3.8	1.5	2.7
AtlasNet [19]	9.7	4.9	10.5	10.0	8.8	5.3	15.0	5.2	8.7
MSN (fusion) [20]	10.7	4.6	12.4	14.0	11.5	12.9	23.4	5.3	11.8
MSN (vanilla) [20]	11.0	**3.8**	9.3	8.3	10.2	3.9	5.9	**3.4**	7.0
GRNet (sparse) [21]	**8.4**	4.3	8.8	6.0	6.0	4.3	5.8	4.5	5.3
Our (flat)	8.5	3.9	9.4	6.7	6.0	**3.7**	**5.2**	4.1	**4.9**
Our (spatial)	10.1	4.4	**8.4**	**5.8**	**5.6**	**3.7**	7.0	3.9	6.1

Table 3.2: Comparison of Chamfer Distance in different sparse point cloud completion models for 2048 points and multiplied by 10^3

Model	**cracker box**	**banana**	**pitcher base**	**bleach cleanser**	**bowl**	**mug**	**power drill**	**scissors**	**average**
Oracle	0.24	0.52	0.28	0.12	0.10	0.09	0.13	0.38	0.23
AtlasNet [19]	4.51	0.87	4.97	5.61	4.21	1.37	6.18	0.92	3.58
MSN (fusion) [20]	5.59	1.25	5.71	2.77	10.81	1.77	8.34	1.58	4.73
MSN (vanilla) [20]	6.01	**0.71**	4.01	4.68	7.51	0.76	1.28	**0.38**	3.17
GRNet (sparse) [21]	**2.28**	0.97	3.78	1.67	2.85	0.76	1.48	0.88	1.90
Our (flat)	3.28	0.92	4.09	1.50	**2.55**	**0.66**	**1.25**	0.82	**1.88**
Our (spatial)	5.81	0.87	**3.19**	**1.20**	2.79	0.69	2.54	0.66	2.22

ble 3.2. Since the point clouds are normalized within this [-1, 1] range, the distance values indicate the relative differences in position within the normalized space. Our method is developed into two models based on the different point seed generators (f/g) in the decoder module. It can be seen that our method outperforms other methods in most objects on both EMD and CD distances. Though for some objects like banana and cracker box, the evaluation metrics of Earth Mover's Distance and Chamfer Distance from our both models are bigger than other baselines. However, for other objects in our datatset, our flat/spatial models both achieve a better performance than other baselines. More importantly, the final average evaluation metrics of EMD and CD of Our (flat) model are both the best evaluation results. For the same completion loss function, our (flat) model achieves an average of about 9% improvement in terms of the EMD distance to the latest GRNet model. Since our dataset contains much noise from the camera and the environment, we found that fusing the output completion result with the original point cloud makes the performance significantly worse, which can be seen from the comparison of MSN (fusion) and MSN (vanilla). It also implies that our model is robust enough, which is conducive to rapid deployment in real robot experiments. Furthermore, compared with ideal results from the Oracle method, it demonstrates that point cloud completion remains an arduous task to solve.

To understand the computational complexity of the proposed transformer-based model, we analyse the Floating-point Operations (FLOPs) and the number of network param-

Table 3.3: Number of FLOPs and network Parameters

Method	**AtlasNet**	**MSN**	**MSN(fusion)**	**GRNet**	**Ours**
# Params (M)	29.46	30.32	33.65	76.71	30.02
# FLOPs (GMac)	14.36	21.46	——	25.90	9.87

Table 3.4: Comparison of EMD and CD from different encoder structures

Earth Mover's Distance (EMD)	**MLP**	**CMLP**	**MSF**	**TMLP**	**Chamfer Distance (CD)**	**MLP**	**CMLP**	**MSF**	**TMLP**
Mug	6.01	3.69	9.45	**3.69**	Mug	2.15	**0.65**	13.80	0.66
Bleach cleanser	10.51	8.10	11.70	**6.70**	Bleach cleanser	6.88	2.63	13.89	**1.50**

Table 3.5: Comparison of average EMD and CD from different point generators

Similarity	**Uniform Distribution:**			**Gaussian Distribution:**			**ZERO**
Metrics	0:1	-0.5:0.5	-1:1	0.5,0.5/3	0,0.5	0,1	
Avg EMD	**5.94**	7.09	6.50	6.34	6.15	**6.14**	9.88
Avg CD	**1.89**	3.25	2.42	2.39	2.38	**2.12**	6.17

Table 3.6: Influence of different surface numbers in the decoder

Earth Mover's Distance (EMD)	**n=4**	**n=8**	**n=16**	**n=32**	**Chamfer Distance (CD)**	**n=4**	**n=8**	**n=16**	**n=32**
Mug	4.71	3.94	3.70	**3.61**	Mug	9.01	6.70	**6.61**	6.69
Bleach cleanser	10.10	7.82	6.69	**5.94**	Bleach cleanser	3.69	1.70	**1.51**	1.53

eters and summarize them in Table 3.3. It can be seen that the self-attention module introduced in our transformer-based encoder is lighter than the traditional convolution layer, reducing the computational complexity. Moreover, after removing a large number of redundant convolution layers existing in traditional dense shape completion, our FLOPs value is also decreased significantly.

3.5.2 Ablation Studies

This section provides a series of ablation studies on our YCB-based dataset to evaluate our proposed shape completion model comprehensively. Accordingly, the effectiveness of each particular module in our model is analyzed as follows: We first evaluate our transformer-based encoder module with other representative encoder modules under the same setting of convolutional/transformer layer number and object inputs. As shown in Table 3.4, our encoder has a better result overall, though CMLP gets a great result on Mug's completion. When the point seed in the decoder is flat, we further analyze the influence of different point seed distributions and surface numbers in Table 3.5 and Table 3.6. We can see that both Uniform and Gaussian sample methods can achieve a better result at $(0, 1)$. We choose $Uniform(0, 1)$ in our model to achieve the best results. Like the weight parameters in the neural network, the initialization value of points cannot be close to zero, which predicts the worst result. As illustrated in Table 3.6, when the sub-surface number increases, the overall model performance improves. However, the improvement of completion results is limited when the number is above 16.

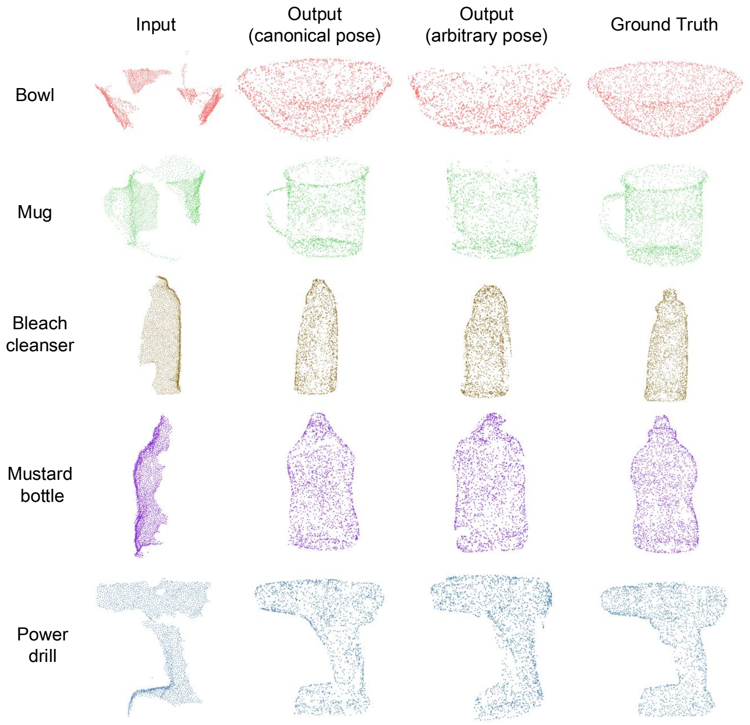


Figure 3.5: Shape completion result using TransSC. The canonical pose result is trained under a fixed point cloud coordinate system while the arbitrary pose result is trained under the camera perspective. In the robot experiment, the arbitrary pose training result is used to generate grasps.

3.5.3 Visualization Analysis

Fig. 3.5 shows the visualized shape completion results using our TransSC. In the visual analysis, each object's input partial point cloud is first preprocessed to remove noisy data from the camera and the environment. It can be seen that the geometric loss of the input point cloud in our dataset comes from the change of the camera viewpoint and the occlusion by other objects, which causes a big challenge for our model. The output results of the canonical pose show that our model works well on all simple and complex objects. Moreover, our model can generate realistic structures and details like the mug handle, bowl edge and bottle mouth. In robotic grasping, as the target object pose is randomly put on the support surface, another shape completion model based on the arbitrary ground-truth pose is retrained. This is done by transforming the ground truth pose to the original pose of the input partial point cloud. The completion results are also shown in Fig. 3.5. Arbitrary output is not as good as the canonical output but it still restores the overall shape of each object well. It also demonstrates that achieving object completion of arbitrary poses in a real environment is still a formidable task.

3.5.4 Simulation Grasp Experiments

Experimental Setup of Simulation Experiments We use Graspit! [23] to evaluate the quality of shape completion similar to [22]. First, the Alpha shapes algorithm [143] is

Table 3.7: Comparison of the average difference between grasp joints from different completion types

Error	**Partial**	**Mirror**	**Voxel-based**	**Ours (canonical)**	**Ours (arbitrary)**
Grasp Joint (degree)	10.07	4.42	2.17	**1.15**	2.02

used to implement surface reconstruction of the completion object. The output 3D mesh is then imported into GraspIt! Simulator to calculate grasps. To have a fair comparison, we also use a Barrett Hand to generate grasps. After finishing the grasp generation, we remove the completion object and import the ground-truth object into the same place. Meanwhile, the Barrett Hand is moved back 20 cm along the approach direction and then approaches the object until the gripper detects a collision or reaches the calculated grasp pose. Furthermore, we adjust the gripper to the calculated grasp joint angles and perform the auto-grasp function in GraspIt! to ensure the gripper makes contact with the object surface or reaches the joint limit. The different values of joint angles at different positions are then recorded. We use four objects (bleach cleanser, cracker box, pitcher base and power drill) from the YCB objects set and calculate 100 grasps for each object in our experiment.

Assuming the grasp pose is the same, we compare the average difference of the joint angle from our shape completion model to that of Laplacian smoothing in Meshlab (Partial), mirroring completion [144] (Mirror) and voxel-based completion [22]. Note that we use two different models, canonical and arbitrary. The **canonical** model means the training process is based on the same object coordinate system and the **arbitrary** model means all the training data are transformed into the camera's coordinate system. Although we can see from Fig. 3.5 that the canonical model has a better shape completion result, it requires an accurate 6D pose of the target object if we want to deploy the complete point cloud into the real world. To avoid this complication of adding a 6D pose estimation module and real robot experiments can be achieved, the arbitrary model is also trained. The simulation result is shown in Table 3.7. It can be seen that Ours (canonical) gets the best simulation grasping performance, which outperforms other completion types. Ours (arbitrary) also obtains a great simulation result though its average joint angle is slightly smaller than voxel-based methods. Moreover, the average difference between the two models also demonstrates that a perfect shape completion in an arbitrary pose is much harder than in a canonical pose.

3.6 Robot Experiments

3.6.1 Robotic Experiments on Single Objects

Experimental Setup of Single Objects To evaluate the performance improvement using a complete point cloud for robotic grasping, we choose six YCB objects to test the grasping success rate. The robot for evaluation is a UR5 robot arm equipped with a Robotiq 3-finger gripper. The vision sensor is an Industrial 3D camera from Mech-

Table 3.8: Robotic grasping performance on a single object

Method	cracker	mug	meat	pitcher	bleach	power	average
	box		can	base	cleanser	drill	
WOSC	70%	70%	80%	80%	90%	40%	71.67%
WSC	80%	100%	100%	80%	90%	50%	83.33%

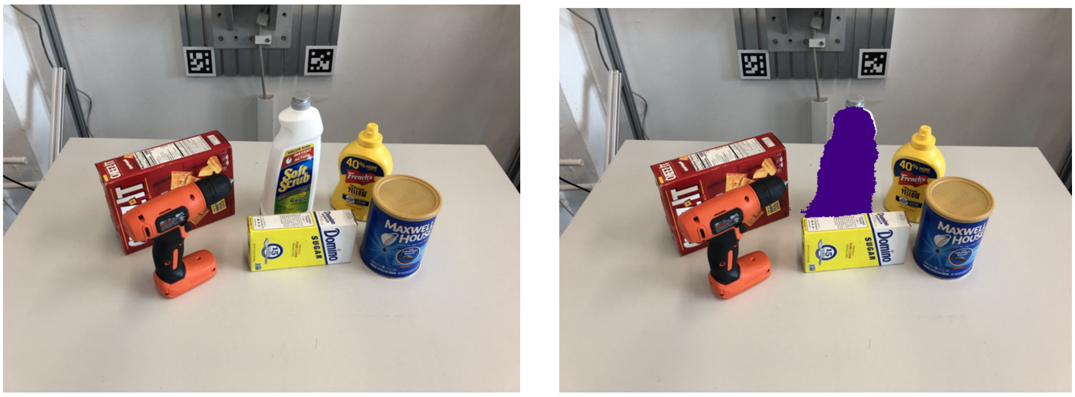


Figure 3.6: The target object and segmentation results with different occlusion settings.

mind [1] to acquire a high-quality partial point cloud. The selected six objects are listed in Table 3.8. We select these objects because they are typical objects that may fail to generate good grasp candidates without shape completion. Other objects such as a banana or a marker are quite simple and small, so that improvement of shape completion on the grasping result is minor. In our robotic experiments, each YCB object is firstly placed on the center of flat table and then moves randomly as long as it can appear in the field of the vision sensor and within the executable range of our UR5 robot arm.

For the selected six objects, we perform grasp evaluation based on PointNetGPD [4] on two different methods: Without our shape completion (WOSC) and with our shape completion (WSC). We run the robot experiment by randomly putting the object on the table and grasping it ten times, then calculating the success rate. The experiment result is shown in Table 3.8. We can see that all six objects' grasp success rates from our grasp pipeline outperform or are even with the original method. The low success rate of the power drill for both methods is due to the contact area of the power drill head being too slippery for the robot to grasp. The failures of WOSC with the observed point cloud input are mainly due to the limit of the camera viewpoint, and GPG generates grasp candidates that sink into the object. An explanation of this situation is illustrated in Fig. 3.4, which demonstrates that our shape completion model can improve the grasp success rate in some particular objects.

[1] https://en.mech-mind.net/

3.6.2 Robotic Experiments on Object Occlusion

Experimental Setup of Object Occlusion When there are different objects on the flat table, the occlusions from other objects will cause a lack of geometric information on the target object. To simulate this scene, we choose bleach cleanser as the target object and other YCB objects are picked as a potential occluder where the occluder as the foreground is placed directly in front of the target object. All objects are placed in a natural vertical position while the horizontal distance between the two types of objects is set to 8 cm. The experimental objects and segmentation result of the target object can be seen in Fig. 3.6. The robot arm and camera are the same as in the robotic experiment on the single object. Furthermore, in real experiments, the target object is placed near the center of the table to ensure that the vision camera can capture it accurately and then we randomly change the 6D pose of the target object to grasp ten times.

As shown in Fig. 3.7, we compare the grasping performance of WOSC and WSC when five different YCB objects occlude the target object (bleach cleanser). The average successful grasping rate of WSC is 88% while WOSC is 50%. It demonstrates that our shape completion method can significantly increase the successful grasping rate up to 32% comparing the original grasping strategy. However, we found that some irregularly shaped objects like the Mustard bottle and Power drill will divide the original partial point cloud of the target object into multiple surface parts. Because PointNetGPD [4] cannot understand that these separated point clouds are from the same object, WOSC generates more wrong grasp candidates without our shape completion.

Furthermore, we explored the effect of the occlusion ratio on the grasp performance by stacking different blocks in front of the target object as an occlusion. Because the target object and stacking blocks are all placed on the table vertically and the horizontal length of each block is bigger than the maximum horizontal width of the target object, the occlusion ratio is calculated by measuring the vertical height of stacking blocks ($\mathcal{H}_b$) and the target object ($\mathcal{H}_t$). As seen from Fig. 3.8, we conducted six additional experi-

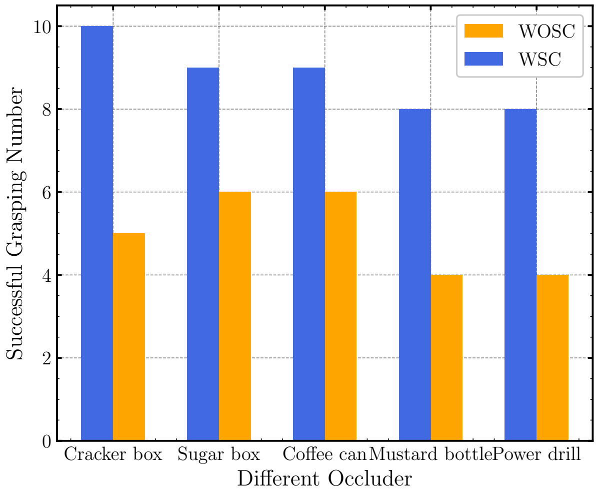


Figure 3.7: Grasping performance comparison when the target object is behind different occluders.

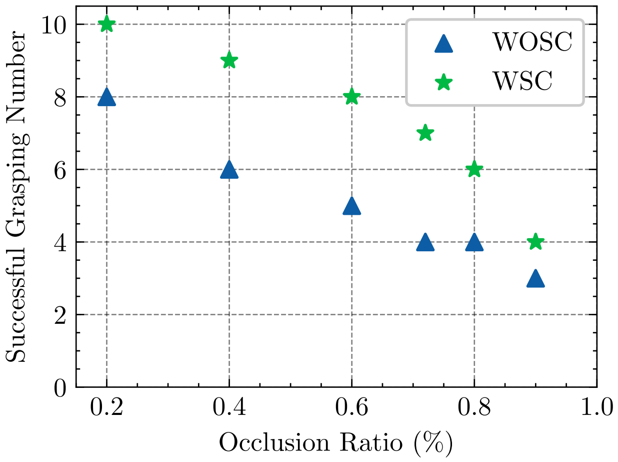


Figure 3.8: Grasping performance comparison when the target object is in a different occlusion ratio.

ments with an occlusion rate between 0.2 and 0.9 to compare the two methods. When the occlusion ratio is less than 0.6, the grasping success rate of WSC is significantly improved over that of WOSC. However, because there are few high occlusion scenes in the YCB video dataset, it is still difficult for TransSC to repair the partial point cloud, especially when the occlusion ratio is higher than 0.8. Furthermore, when the occlusion ratio is between 0.8 and 1.0, it means that the target object has been completely obscured. The vision information of the target object is too little, so it's also much more difficult to use shape completion to restore complete object information. According to our observation in daily life, we found 0.2-0.6 is the most common object occlusion ratio and our experiments showed that our shape completion method can effectively improve the successful rate within this range.

3.7 Discussion and Summary

In this chapter, we present a novel transformer-based sparse shape completion network (TransSC). This network includes a transformer-based encoder and manifold-based decoder that we designed, enabling our model to achieve a great completion result and outperform other representative methods. The experiments show that our network is robust to sparse and noisy point cloud input. Besides, simulation grasping experiments show our model could achieve a smaller grasp joint error than traditional robotic completion methods. Finally, when executing real robotic experiments of single objects and object occlusion, we demonstrate that our TransSC can be easily embedded into an existing grasp evaluation module and improve grasping performance significantly in both scenes.

The lack of object geometric information in our dataset is due to the change in the camera viewpoint and the occlusion by different objects. Thus, our grasp pipeline can solve both situations occurring in the grasping task successfully. However, similar to the research issue of 6 DoF pose estimation, it is still challenging to achieve shape completion of an arbitrary object at an arbitrary pose due to the limited object categories in our dataset. So the main limitation in this chapter is that the object categories in our constructed dataset are still small and they are only limited to the YCB objects, which causes our shape completion model not to be generalized into other novel objects. In future work, our goal is to collect more object categories to achieve a better generalization for unseen but similarly shaped objects. Furthermore, we will also consider more data augmentation strategies like adding more data representations, different object 6 DoF poses and different point cloud missing ratios as our experiments have shown, which can hopefully achieve a better grasp performance from our shape completion model in the real robotic experiments.

Chapter 4

Task-oriented Grasping Generation Based on 3D Visual Affordance

Currently, task-oriented grasp detection approaches are mostly based on pixel-level affordance detection and semantic segmentation. These pixel-level approaches heavily rely on the accuracy of a 2D affordance mask, and the generated grasp candidates are restricted to a small workspace. In this chapter, we first construct a novel affordance-based grasp dataset and propose a 6-DoF task-oriented grasp detection framework, which takes the observed object point cloud as input and predicts diverse 6-DoF grasp poses for different tasks. Specifically, our implicit estimation network and visual affordance network in this framework could directly predict coarse grasp candidates, and corresponding 3D affordance heatmap for each potential task, respectively. Furthermore, the grasping scores from coarse grasps are combined with heatmap values to generate more accurate and finer candidates. Our proposed framework shows significant improvements compared to baselines for existing and novel objects on our simulation dataset. Although our framework is trained based on the simulated objects and environment, the final generated grasp candidates can be accurately and stably executed in real robot experiments when the object is randomly placed on a support surface.

4.1 Introduction

Recently, task-oriented robotic grasping and manipulation have received more and more attention from the robotics community [34, 145, 146], which aims to generate different robotic actions and interactions for the same object representing of a potential scenario. The outcome of task-oriented robotic motion can benefit a robot's ability to understand the semantic context of objects better. For example, traditional robotic grasping detection methods (pixel-based and point cloud-based) all generate random grasp candidates around the target object. However, these methods lack the understanding of object context (such as global and local texture).

To cope with this limitation, some researchers introduce the concept of affordance into the robotic field, which plays a key role as a mediator, organising the diversity of possible perceptions into tractable presentations that can support reasoning processes to

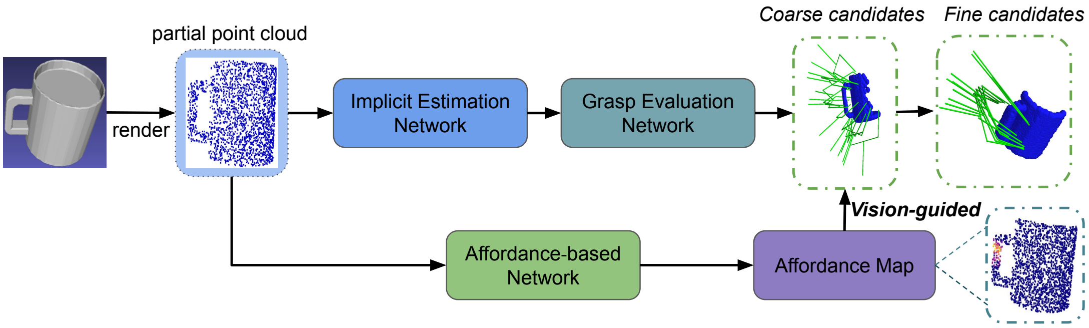


Figure 4.1: Overview of our 6-DoF task-oriented grasp detection framework for affordance-based robotic grasping. The observed point cloud from the RGB-D camera is sampled as input $\mathcal{P} \in \mathcal{R}^{N\times 3}$. It will pass into two parallel modules, the grasping affordance module and the visual affordance detection module. The first module finally outputs coarse grasp candidates $\mathcal{G}_C$ in the form of $SE(3)$ and corresponding confidence scores; the other module outputs a 3D heatmap $\mathcal{H}^N$ where the values of each affordance label are predicted. Finally, the visual affordance map values combine with confidence scores to guide the coarse grasp candidates $\mathcal{G}_C$ in becoming more accurate and fine. And fine grasp candidates $\mathcal{G}_\mathcal{F}$ will be executed in the real robotic experiments.

improve the generalization of tasks [147,148]. However, most of the current mainstream affordance-based robot grasping methods require prior pixel-level target detection and semantic segmentation [32,38,149–151], where grasping candidates are generated after obtaining the target object part. Nonetheless, this kind of grasp detection strategy cannot essentially couple with the contextual information of objects, and the pose of pixel-based grasp is at a limited dimension. To combine 6-DoF grasp detection with affordance knowledge, some works are also proposed to use the observed point cloud as the model input recently [37,41,42,152]. These methods either rely solely on a generative model to generate target region grasping, or use masks to assist grasping detection, which cannot achieve a great trade-off between grasp quality and generalization ability.

As illustrated in Fig. 4.1, we present a novel 6-DoF task-oriented grasp detection framework for affordance-based robotic grasping tasks. Specifically, the input of our framework is the partial object point cloud captured by the RGB-D camera. Our framework consists of two modules: the grasping affordance detection module and the visual affordance prediction module. Motivated by the grasp generation approach based on the generative model (VAEs) [17], it achieves great grasp generation results for some clearly outlined objects like a mug, a scissor and a bottle. Due to the complex data distribution and geometry structures of points, we postulate that the uni-modal distribution assumption can be violated in multi-affordance grasping generation tasks. Especially the generated grasp poses from point-wise features can vary vastly within different affordances. Thus, we design each task with an implicit representation to better represent the complex distribution. Otherwise, a grasp evaluation network is designed to evaluate the generated coarse grasp candidates. Both networks are trained in our self-constructed affordance grasp dataset. For the visual affordance prediction module, inspired by the work of 3D affordanceNet benchmark [28], we designed an attention-aware bilinear feature learning network to capture the geometric dependencies and semantic correlations by learning point features and edge features, simultaneously. Sequentially, the predicted affordance map guides coarse grasp candidates to centrally distribute around the spatial region of the maximum map value, which could significantly improve the accuracy and stability of the generated grasps.

The main contributions of this chapter can be summarized as:

- Based on the work of ACRONYM [153], we introduce an affordance-based grasp dataset where each successful grasp of each object is annotated as a task-oriented label. Each grasp is evaluated by the simulated engine.

- An implicit multi-stream network is proposed to generate diverse affordance-based grasp candidates directly, showing a better performance than the VAEs model.

- To use the spatial context of objects, we design an attention-aware bilinear feature learning network and first introduce 3D visual affordance to real robotic grasping, which effectively guides coarse grasp candidates to become more affordance-centric and finer. We also demonstrate that these grasps generated from simulated objects can be transferred to the real world.

4.2 Problem Formulation

We consider a setup consisting of a robotic arm with parallel-jaw grippers, an RGB-D camera and an object on a planar tabletop to be grasped. A single-view depth map is captured by the RGB-D camera to convert into a 2.5D partial point cloud $\mathcal{P} \in \mathcal{R}^{N \times 3}$ and then passes into the pipeline. For simplicity, all spatial quantities are in camera coordinate frames.

Our pipeline consists of two models: the Grasping Affordance Detection Model and the Visual Affordance Prediction Model. The first model aims to learn a posterior distribution $\mathcal{D}((\mathcal{G}(\mathcal{T})^*)|\mathcal{P})$, where $\mathcal{P}$ is the input partial point cloud and $\mathcal{G}(\mathcal{T})^*$ represents successful grasps of different mini-task actions $\mathcal{T} \in (0, M)$, such as wrap, grasp, pour, and cut, where M is the total number of mini-task categories. This model outputs coarse 6-DoF grasp detection candidates $\mathcal{G}_{\mathcal{C}}$ and associated confidence scores $\mathcal{S}_{\mathcal{O}}$. Furthermore, the function of the second model is to predict a 3D visual affordance map $\mathcal{S}_{\mathcal{M}} \in [0, 1]^N$ for different mini-task actions $\mathcal{T}$, which are combined with original grasp confidence scores $\mathcal{S}_{\mathcal{O}}$ to obtain fine grasp candidates $\mathcal{G}_{\mathcal{F}}$. Each generated grasp $G_i \in (\mathcal{G}_{\mathcal{C}}, \mathcal{G}_{\mathcal{F}})$ is denoted as $(R, T) \in SE(3)$. We trained our grasp framework by randomly rotating the objects with different affordance in a simulated rendering environment, where final generated grasps $\mathcal{G}_{\mathcal{F}}$ are defined according to the object reference frame whose axes are parallel to the camera. Finally, the $\mathcal{G}_{\mathcal{F}}$, representing fine successful grasps of a certain task affordance, will be transformed into the camera coordinate frame to be executed in real robot experiments.

Table 4.1: Statistic of object number and corresponding affordance categories in the affordance grasp dataset

Object	Mug	Bottle	Knife	Hat	Bowl	Scissor
Affordance	grasp, wrap, pour, contain	grasp, wrap, contain	grasp, cut/stab	grasp, wear	grasp, wrap	grasp, cut
Number	60	33	42	8	52	8

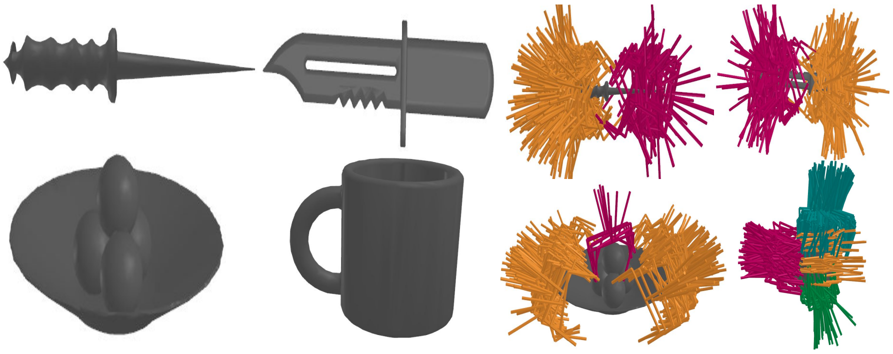


Figure 4.2: Visualization of our partial affordance grasp dataset. (Left) 3D model of ShapeNet objects. (Right) All markers represent successful grasps and different colors of markers indicate different grasp tasks.

4.3 Affordance-based Grasping Dataset

Inspired by the 3D AffordanceNet [28] and ACRONYM dataset [153], we focus on constructing a new dataset for task-oriented grasping based on simulated ShapeNet [154] objects. We chose the ACRONYM dataset as our grasp prototype because it is a large and well-established dataset for robot grasp planning, which in total contains more than 17M parallel grasps and diverse objects from different categories. Moreover, each grasp in this dataset is evaluated and then judged as a successful or failed one through a physics simulator. As shown in Table 4.1, we exclude many object categories because they lack affordance meaning and could not be applied to household robotic scenarios.

Rather than using different object parts as different affordance representations [28, 37, 38], we annotate all successful grasps of selected objects with different affordance labels. For example, *grasp* affordance in the mug instance means all successful grasps around the mug handle while *pour* affordance means all successful grasps around the upper mug rim. During our annotating process, the affordance label number for different objects of the same categories are assumed to be the same though we find the distribution of successful grasps of a few objects is not similar. Taking the mug as an example, all successful grasps of some mugs with special shape can only be divided into two kinds of affordance types. As a result, we use a constant grasp pose value to indicate successful grasp for the other two kinds of affordance types. All failed grasp candidates of selected objects existing in [153] are also reserved in our dataset as negative grasp samples. Otherwise, we transform all objects and corresponding grasps into a uniform coordinate frame, which is conducive to the training of two subsequent modules. Fig. 4.2 visualizes different affordance results of selected objects in our dataset. Finally, our affordance

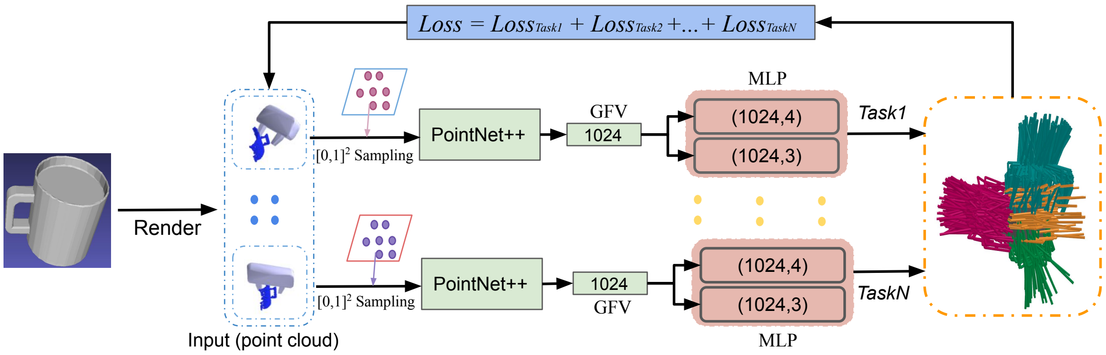


Figure 4.3: The architecture of the proposed implicit multi-stream estimation model.

grasps dataset consists of 203 household objects from 6 categories, and more than 100K successful grasps are selected and annotated into 6 common tasks in our daily lives.

4.4 Grasp-Affordance Framework

In Fig. 4.1, our grasping affordance detection framework consists of two sub-networks: an implicit estimation network and a grasping evaluation network. Firstly, based on the proposed dataset, the object point cloud is captured by the camera by rotating the object at a random pose in the rendering environment, and each point cloud is sampled to 2048 points through Farthest Point Sampling (FPS). Then the implicit estimation network takes the partial point cloud as input and outputs diverse grasp candidates corresponding to its affordance label. On the other side, the grasping evaluation network takes different grasps as input and learns a classifier to recognize success and failure. Finally, coarse grasp candidates of each affordance label will be obtained when the generated grasps from the implicit estimation network are input into the trained grasping evaluation network. Below, we present details of these two sub-networks.

4.4.1 Implicit Estimation Network

Generative modelling is a cornerstone for machine learning, which has been widely used in the 2D vision field, like image tampering and image compositing. In previous 3D point cloud-based robotic research [17, 42], variational autoencoders (VAEs) are commonly chosen for numerous grasp candidates. However, an accurate VAEs model usually needs a prior partition function to predict the distribution of ground truth, like a mixture of Gaussian, hidden Markov or Boltzmann machines. Especially when we need to predict $SE(3)$ grasps of different affordance tasks simultaneously, it is challenging to sample from these models. On the other hand, using an implicit model is a more natural way in terms of the sampling strategy [155–157], which can be simply expressed by the following sampling procedure:

1. Sample $\zeta \sim \mathcal{M}(0, I)$
2. Return $\chi := \mathcal{N}(\zeta)$

Where $\mathcal{M}$ is a latent distribution and $\mathcal{N}$ is a highly expressive function approximator, usually replaced by a neural network.

To encode both affordance and geometry from the partial point cloud, we use the implicit maximum likelihood estimation method [156] to predict grasp poses. As shown in Fig. 4.3, we propose a multi-stream neural network architecture for jointly predicting $SE(3)$ grasps of different affordance tasks. Each stream represents a different affordance label of certain object. At each network stream, the sampled point cloud $\mathcal{P}^{2048\times3}$ is concatenated with a latent indicator $\mathcal{L}$ and then is input into the PointNet++ [158] architecture to extract spatial information between the point cloud and the potential grasp pose. After that, a 1024-d generalized feature vector (GFV) can be obtained. We parametrize each GFV with separated small full-connected layers. The output rotation and translation values of the target grasp pose are expressed as:

$$R_{\mathcal{P}}^{\mathcal{T}} \longrightarrow [quat_1, quat_2, quat_3, quat_4] \tag{4.1}$$

$$T_{\mathcal{P}}^{\mathcal{T}} \longrightarrow [X, Y, Z] \tag{4.2}$$

where $\mathcal{T}$ and $\mathcal{P}$ separately represent the affordance label and the object point cloud, and rotation values are predicted as a form of quaternion.

4.4.2 Grasping Evaluation Network

Similar to the evaluation method of [17], we choose to combine all grasp poses $(\mathcal{G}_s, \mathcal{G}_f)$ with object point cloud $\mathcal{P}$ as the input of the network, where a gripper point cloud corresponding to each grasp pose is used to approximate the real gripper. An extra binary value is also used to judge whether the point from the combined point cloud belongs to the object or gripper. Furthermore, like the implicit estimation network, we still use PointNet++ [158] to explore the spatial relationship between object point cloud and gripper point cloud. The output module consists of three full-connected layers [1024, 512, 256] and a final sigmoid layer. Finally, according to the binary ground truth label (success or failure), it is easy to train a classifier to predict the successful probability of each input grasp. After finishing the evaluator training, this model is used to deal with output results from the implicit estimation network, which can guarantee final grasp candidates are all successful.

4.4.3 Visual Affordance Prediction Module

Fig. 4.4 illustrates our attention-aware visual affordance network architecture, which consists of two main components: embedding network and metric decoder. The embedding network is the most important part of our network since the performance of the metric decoder relies on learned embedding space. We expect this embedding work to realize two critical functions: 1) to encode the geometric relationship of the local region, especially for different affordance parts. 2) to encode global semantic information based on global context.

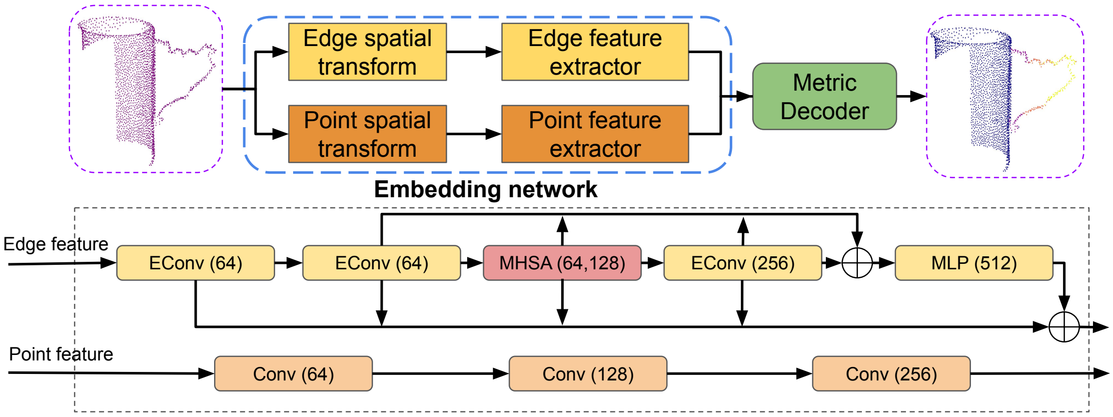


Figure 4.4: The architecture of our proposed visual affordance network, where EConv is the EdgeConv layer and MHSA is the multi-head self-attention module.

Based on this idea, we design an attention-aware bilinear learning framework that incorporates point features and edge features to extract local and global semantic information. In particular, we adopt the PointNet [15] and DGCNN [159] as our backbone to extract different semantic features respectively. Based on the PointNet, three sequential convolutional layers (Conv(64, 128, 256)) are used to produce global semantic features. [159] proposed a dynamic graph network architecture, which can effectively output local geometry features from the first EdgeConv layer (EConv(64)) and global spatial features from the final multi-layer perceptrons layer (MLP(512)). To further obtain local correlations for each affordance part, the multi-head self-attention (MHSA(64, 128)) module is applied to generate more semantic features, encouraging point-wise features to aggregate with global context for the affordance-meaning point cloud. After obtaining the point feature and edge feature, they are concatenated together to input into the metric decoder, which consists of a stack of multi-layer perceptron layers. It finally predicts a probability distribution based on the partial point cloud corresponding to different affordance tasks:

$$\mathcal{H}^N = \left\{H_1^N, ..., H_M^N | H_i^N = map_i(\mathcal{P})\right\} \tag{4.3}$$

$$H_i^N = \left\{h_i^0, ..., h_i^{N-1}\right\}, h_i^j \in [0, 1] \tag{4.4}$$

where H_i^N is the predicted point cloud map values of affordance label i. In our network model, the parameters of N and M represent the sampled point cloud number and the affordance task number, respectively.

4.4.4 Fine Grasp Candidate Generation

For each affordance label i, all coarse grasp candidates $\mathcal{G_C}^i$ output by the grasping detection module are sorted in descending order according to their confidence scores $\mathcal{S_C}^i$. Then, the predicted values of the affordance heatmap from the visual affordance module can be also obtained and sorted descending as H_i^N. To reduce the computational cost for the sampling sparse point cloud, we can define $\mathcal{U}_i \subset \mathcal{P}_i$ as the subset of points corresponding to the top 100 maximum values in H_i after filtering noisy outliers. Then the reduced point cloud and its associated values can be represented as our affordance label $(\mathcal{P}_i^{(100)}, H_i^{(100)})$, where:

$$\mathcal{P}_i^{(100)} = \{\mathbf{p} \in \mathcal{P}_i \mid H_i(\mathbf{p}) \in \mathcal{U}_i\} \tag{4.5}$$

$$H_i^{(100)} = \{H_i(\mathbf{p}) \mid \mathbf{p} \in \mathcal{P}_i^{(100)}\} \tag{4.6}$$

Note that most of the values of H_i^N are approximately 0 because they do not belong to label i. After that, each point from $\mathcal{P}_i^{100}$ is selected to compute the L2 distance with the middle control point P_i^{cm} of each coarse candidate, the minimal distance is regarded as the vision-guided score for each grasp:

$$\mathcal{S_V}^i = min(||\mathcal{P}_i^j - P_i^{cm}||_2), \text{for } j \in [0, 100) \tag{4.7}$$

The final fine grasp candidates can be obtained by combining the value of $\mathcal{S_C}^i$ and $\mathcal{S_V}^i$:

$$\mathcal{S_F}^i = \alpha_1 * \mathcal{S_C}^i + \alpha_2 * \mathcal{S_V}^i \tag{4.8}$$

where $\mathcal{S_F}^i$ represents the score of fine grasp candidates for affordance label i and the hyperparameters of α_1, α_2 are both set to 0.5.

4.5 Simulation Experiments

4.5.1 Implementation Details

Data Augmentation: The object point cloud was captured through rotating at a random Euler angle $(x, y, 0)$, where the range of rotation is $x \in [0, 2\pi], y \in [-\pi/2, \pi/2]$. The rotation number of each object is 900 times, where jitter and dropout operations are added in each rotation. After that, the observed partial point cloud is sampled to 2048 points through the farthest point sampling (FPS) algorithm during training. And they are also processed through a mean-centred and unit-scaled trick. For the visual affordance prediction, where we followed [28], the point cloud needs to be further normalized during the training.

Training Details: The grasp affordance module is trained based on our proposed grasp affordance dataset, and the visual affordance prediction module is trained on the 3D AffordanceNet dataset [28], which causes two kinds of training loss. For the grasp affordance loss, we firstly compute minimum L1 loss between any predicted grasp pose $\mathcal{G_C} = [quat_1, quat_2, quat_3, quat_4, X, Y, Z]$ with the ground-truth grasp poses $\mathcal{G(T)}^*$ for each affordance task $\mathcal{T}$. To simplify the computational complexity, each grasp pose is transformed to 7 control points representing a gripper to make training simpler. Thus, the loss function of the implicit estimation network is denoted as $\mathcal{L}_i$:

$$\mathcal{L}_i = \frac{1}{M}\sum_i^M \left(min(\frac{1}{k}||\mathcal{G_C} - \mathcal{G(T)}^*||_1) \right) \tag{4.9}$$

Moreover, for the loss function of the grasping evaluation network, we adopt the standard binary cross-entropy loss between the predicted grasp status $\overset{*}{o} \in [0, 1]$ and the ground-truth grasp label $o \in \{0, 1\}$ (0 for failure, 1 for success). Thus, the loss of the grasp evaluator is denoted as $\mathcal{L}_e$:

$$\mathcal{L}_e = -(olog(\overset{*}{o}) + (1 - o)log(1 - \overset{*}{o}) \tag{4.10}$$

As visual affordance loss, the same training target in benchmark [28] is adopted:

$$\mathcal{L}_v(\overset{*}{m}, m) = W_1 * L_{ce}(\overset{*}{m}, m) + W_2 * L_{dice}(\overset{*}{m}, m) \tag{4.11}$$

Where $\overset{*}{m} \in [0, 1]$ denotes predicted affordance map values and $m \in [0, 1]$ denotes the ground-truth propagation score (0 is the minimum value of correlation for each affordance label, 1 is the maximum value of correlation). L_{ce} is the cross-entropy loss and dice loss L_{dice} is also introduced to mitigate the imbalance issue caused by the dataset. In our training process, hyperparameters W_1 and W_2 are set as 0.4 and 0.6, separately.

4.5.2 Ablation Study

We design two baselines for comparison with our method. 1) Baseline1: this is from a similar work [42], which takes the scanned point cloud as input to train a framework to generate different task-oriented grasps. The grasp detection benchmark from this work is adapted from [17] and a knowledge graph is introduced to connect diverse tasks and objects. However, we find that the effect of the knowledge graph is limited in our dataset due to fewer tasks and object categories. 2) Baseline2: this can be considered as a degraded version of our method, where the multi-stream implicit estimation method is

Table 4.2: Comparison of different generators: IEN and VAEs

ESM	Mug	Bottle	Bowl	Hat	Scissors	Knife	Average
IEN	0.062	0.058	0.081	0.122	0.157	0.071	0.092
VAEs	0.106	0.112	0.142	0.186	0.211	0.147	0.151

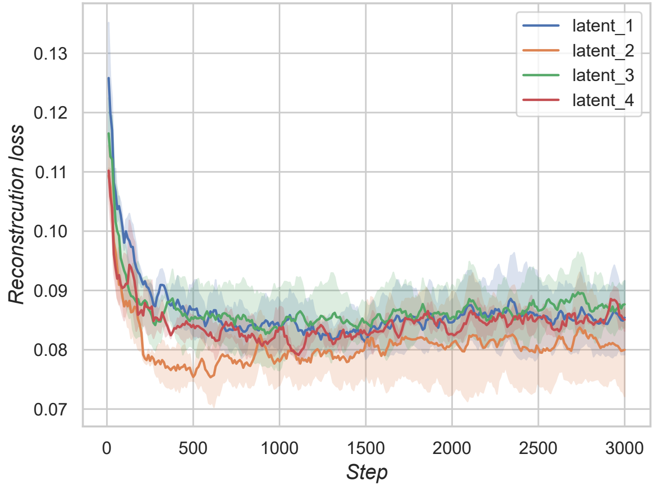


Figure 4.5: The effect of length of a latent vector in our implicit estimation network.

Table 4.3: Ablation study for the visual affordance module

Point feature	Edge feature	MHSA module	average AP	average AUC	average IOU
✗	✓	✓	0.4201	0.8325	0.1098
✓	✗	✓	0.3737	0.7982	0.1473
✓	✓	✗	0.4241	0.8249	0.1618
✓	✓	✓	**0.4281**	**0.8360**	**0.1628**

Table 4.4: Evaluated similarity metric from different existing objects for different-oriented tasks

Task	**Grasp**						**Wrap/Cut**					
	Mug1	Mug2	Mug3	Bottle1	Knife1	Scissor1	Mug1	Mug2	Mug3	Bow1	Knife1	Scissor1
Baseline1	0.167	0.147	0.107	0.149	0.151	0.321	0.141	0.088	0.096	0.203	0.185	0.185
Baseline2	0.061	0.051	0.048	0.092	0.138	0.248	0.097	0.094	0.084	0.133	0.133	0.139
Our	0.046	0.032	0.033	0.110	0.083	0.179	—			0.121	0.117	0.144

employed to detect grasp without the guiding of the 3D visual affordance. To demonstrate the effectiveness of our method, we also design an evaluation metric to compare the grasp similarity between predicted grasp and ground truths in our dataset. Similar to the loss function of the implicit estimation network, we sample 100 predicted grasps and ground truth grasps randomly and then we will compare the L1 distance of each predicted grasp $\mathcal{G}$ with $\mathcal{G}(\mathcal{T})^*$, the minimum value is assumed as its similarity value. The mean value of the sum of 100 minimum values can be computed as the evaluated similarity metric (ESM). A smaller ESM value means a better similarity between predicted grasps and ground truths.

We study the effects of the grasp detection model from our implicit estimation network (IEN) and widely used VAEs model [17], and the comparison results of ESM are listed in Table 4.2. It shows that our network can achieve a better prediction result than the VAEs model. Moreover, we illustrate the effect of the length of the latent vector in our implicit estimation network. As can be seen from Fig. 4.5, when the length of the latent vector equals 2, the network achieves the best performance in the test set because a slightly bigger latent vector can cause an over-fitting problem.

As the vision-guiding module, the visual affordance network is also the most important component of our framework. Thus, we continue to study the effects of various designs existing in visual affordance network. Though point-level (PointNet [15] and PointNet++ [158]) and edge-level (DGCCN [159]) methods are used in the benchmark [28], we denote the levels of new features, i.e. point feature, edge feature, and MHSA module, respectively. The results of three variants are listed in Table 4.3. Eventually, the integration features of the three levels give us the best performance on the visual affordance prediction.

4.5.3 Comparison with Baselines

Table 4.4 and 4.5 summarize the ESM results of comparing our method to the baselines on existing objects and novel objects, respectively. It is not surprising that the use of implicit representation leads to improvements in task-oriented grasp prediction. Moreover,

Table 4.5: Evaluated similarity metric from different novel objects for different-oriented tasks

Task	Grasp					Wrap/Cut				
	Mug4	Mug5	Scissor2	Knife2	Knife3	Mug4	Mug5	Scissor2	Knife2	Knife3
Baseline1	0.130	0.136	0.192	0.156	0.178	0.082	0.109	0.234	0.185	0.195
Baseline2	0.117	0.056	0.161	0.091	0.122	0.076	0.101	0.187	0.131	0.162
Our	0.059	0.037	0.161	0.052	0.104	—		0.162	0.104	0.134

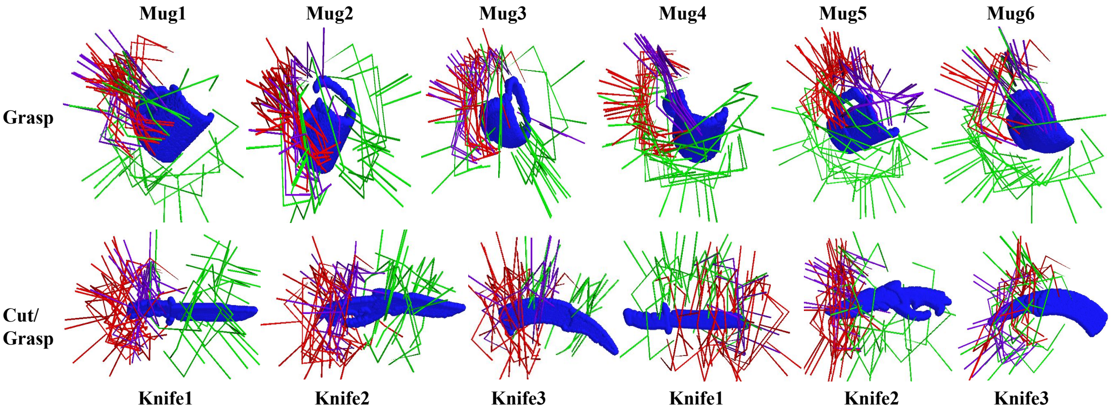


Figure 4.6: Visualization results of our proposed method for task-oriented grasp prediction. For each affordance label, green means the generated grasp without visual guidance, purple means grasp candidates from our method, and red means the ground truth from our dataset. (Tasks: *grasp* of the mug, *cut* of the knife and *grasp* of the knife.)

for most object point clouds, the 3D affordance map in our method could effectively improve the final prediction result. We also observe a phenomenon that the coarse version of our approach (Baseline2) sometimes is better than our method. This is probably because the input partial point cloud sometimes lacks the affordance context. For example, *grasp* of *mug* corresponds to the mug handle. If the captured mug point cloud misses the handle completely, that will cause a bad affordance heatmap. Our method also shows the best performance for novel objects, demonstrating a great ability for generalization.

4.5.4 Visualization Analysis

Fig. 4.6 and 4.7 show the visualization results of our proposed method for task-oriented grasp prediction, respectively. The results of affordance-based grasp candidates from our method are compared with the predictions from the degraded version and ground truths. As seen from Fig. 4.6, it is very challenging to predict the grasps restricted in a small affordance region. The coarse grasp candidates from a degraded version of our method are not ideal because many predicted grasps are not centric-around the affordance context (like *grasp* around the handle of a mug, and *cut* around the handle of a knife). However, the predictions from our method could be more concentrated on the affordance area, and the fine grasp candidates are more accurate. Fig. 4.7 also shows the visualization results of the 3D affordance map and corresponding grasp candidates. We find an interesting result when the affordance region is evenly distributed over the con-

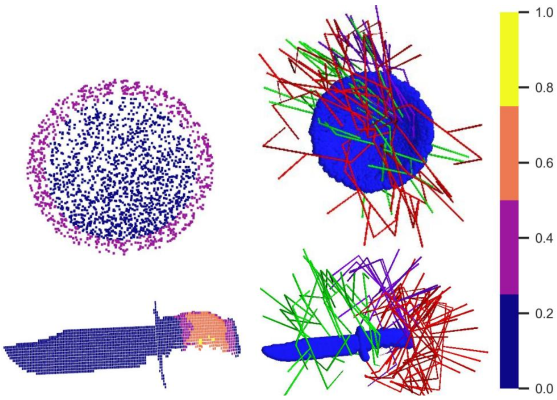


Figure 4.7: Visualization results of the 3D affordance map and corresponding grasp candidates (*wrap* of the bowl and *cut* of the knife). The different colours of the right bar are just used for the affordance map.

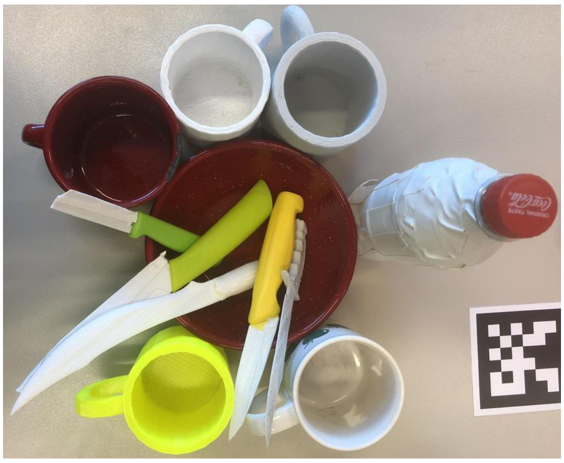


Figure 4.8: All novel objects that are tested in our real robot experiment.

tour of the object (like *wrap* of the bowl and mug), where the predictions of our method will be similar to the coarse candidates. That is because the groundtruths are uniformly distributed around the whole point cloud, causing the vision-guided score ${\mathcal{S}_{\mathcal{V}}}^i$ cannot effectively improve the original grasping score ${\mathcal{S}_{\mathcal{C}}}^i$.

4.6 Robot Experiments

To evaluate the performance using our task-oriented grasping detection framework for robotic grasping, we run real robot experiments to demonstrate that our model trained from simulation transfers well to the real robot environment. In the experiment setup, we put a single object on a flat table at an arbitrary pose without any clutter. As shown in Fig. 4.8, all the objects we test are unknown to the system. To avoid causing damage to real objects and gripper, some objects like mugs and knives are obtained through 3D-print technology. Near the target object, a KUKA LWR robot is fixed on the table with a 2-fingered WSG-50 gripper. And about 1.2 m in front of the robot and target object, a

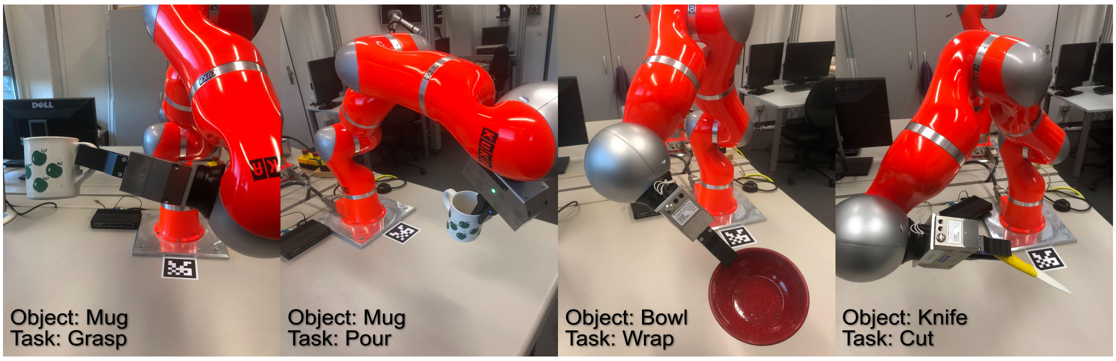


Figure 4.9: Evaluation results based on different tasks and objects from real robot grasping.

Mechmind RGB-D camera is suspended on a bracket to capture the object point cloud. The obtained point cloud is input into the trained framework. For each task, over 3000 successful coarse grasp candidates are output first. Then a 3D visual affordance map for this affordance label is predicted to couple with the original grasp score. Finally, 20 fine-sampled grasp candidates are sent to the robot for execution according to their final evaluated grasping scores in descending order, where the first grasp (highest score) will be executed if there is no problem with its planning in MoveIt. The whole framework is trained and inferenced on the desktop PC with NVIDIA GTX 2080Ti GPU. Fig. 4.9 shows the evaluation examples based on different tasks and objects from real robot grasping. It demonstrates that our approach trained in simulation can be validated successfully in a real environment. During the process of experiments, we found that if more affordance features like the handle of the mug and knife can be captured by the camera, the final execution grasp is more stable and accurate. The gap between the simulation and the real environment also exists when the texture of the real object is very smooth and irregular though the generated grasp looks good. Nonetheless, for almost every experiment, our fine grasp poses mainly focus on the corresponding affordance areas, showing our framework can well reason the relationship between grasp detection and task-oriented affordance.

4.7 Discussion and Summary

This chapter investigates the challenging problem of task-oriented robotic grasping. Focusing on 6-DoF grasp detection, we proposed a novel solution by designing three modules: an implicit estimation network, a grasp evaluation network, and an attention-aware visual affordance network, which achieves consistent and clear improvements over baselines for existing and novel objects in our self-constructed affordance grasp dataset. This work provides several key insights into task-oriented grasping: 1) the learning of implicit representations from objects and grasp poses from each affordance label is the core of 6-DoF grasping affordance detection. 2) the exploitation of point-based, edge-based features and the attention mechanism are necessary to achieve a better affordance map prediction. 3) the generated 3D affordance map could effectively guide coarse grasp candidates to become more accurate and finer for a specific affordance task. For future work, we hope to use the advantage of the simulation environment to rapidly extend the number of object categories and affordance labels, improving the generalization ability of our framework. Moreover, we also want to explore the possibility of combining the trained framework with hand-over tasks, which is beneficial to increase the use of affordance learning in the robotic field.

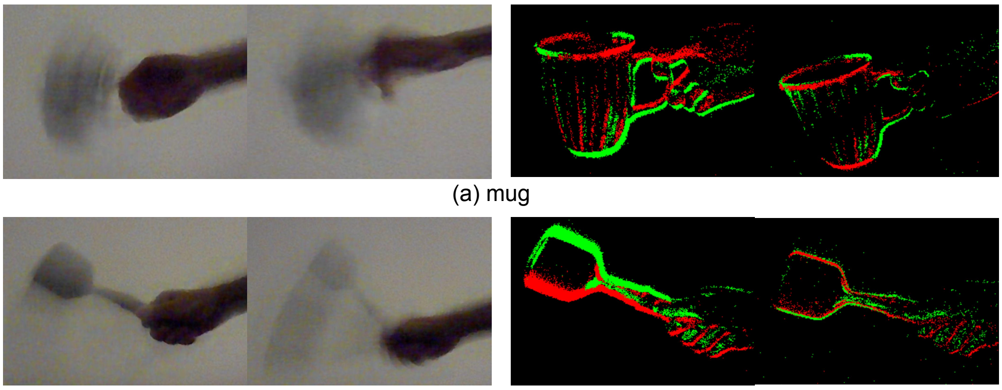
Chapter 5

Task-oriented Hand-Object Grasping Recognition Using Event Vision

In the last chapter, we discussed the grasping generation based on the context-aware object parts. Continually, we will discuss the different hand-object grasping types when they are in dynamic motion conditions, especially involving another vision modality. The event-based camera is a novel neuromorphic vision sensor that can perceive different dynamic behaviors due to its low latency, asynchronous data stream, and high dynamic range characteristics. There has been much work based on event cameras to solve problems such as object tracking, visual odometry, and gesture recognition. However, the adoption of event vision to analyze hand-object action in a dynamic environment, a problem that regular CMOS cameras cannot handle, is still lacking in relevant research. This work presents a richly annotated task-oriented hand-object action dataset consisting of asynchronous event streams, captured by the event-based camera system on different application scenarios. In addition, we design an attention-based residual spiking neural network (ARSNN) by learning temporal-wise and spatial-wise attention simultaneously and introducing a particular residual connection structure to achieve dynamic hand-object action recognition. Extensive experiments are validated by comparing with existing baseline methods to form a vision benchmark. We also show that the learned recognition model can be transferred to classify a real robot hand-object action.

5.1 Introduction

In many situations, including augmented reality (AR), the Metaverse, and human-computer interaction (HCI), accurate motion recognition for the scene of a hand interacting dynamically with an object is very basic but essential. The computer vision and robotics communities have made a great effort to study different hand-object interaction behaviors from visual perception, pose estimation, and grasp prediction. However, there is little work exploring hand-object interaction from a dynamic perspective which is actually existing in our daily life. The interaction behavior between hand and object is not time-static grasping but is accompanied by moving manipulation. For example, if you want to drink a cola, *you will first pick up a cola bottle, then unscrew the bottle cap,*



(a) mug

(b) spatula

Figure 5.1: The visual difference captured by the CMOS camera and event camera when a hand interacts with (a) *mug* and (b) *spatula* with a dynamic motion.

and finally put it near your mouth to drink it. This sequence of dynamic motions including $handover$, $unscrew$, and $drink$ changes drastically over time. Currently, most hand-object grasping datasets [62, 63] are based on a simulated environment, making it challenging to obtain dynamic action data. A few naturally situated datasets [61, 64] capture the real visual information of the hand-object action to analyze while the dynamic characteristics of actions are barely considered. Therefore, it is challenging but necessary to collect a dynamic hand-object action dataset containing spatial-temporal information and then visually analyze different grasping behaviors within the dataset.

In this chapter, we define our task as recognizing the action that is being performed as a person manipulates an object, by observing solely the hand and object. Based on that, we propose a rich-annotated event-based hand-object action dataset, namely EHoA, to solve the task-oriented action recognition problem. Unlike traditional CMOS cameras, the event-based vision sensor can capture extensive visual information both in temporal and spatial channels. Primarily, it can well perceive motion change under different lighting conditions due to a high dynamic range and lighting sensitivity. In most high-level vision tasks like object tracking, recognition, and detection, insufficient illumination, and limited exposure time poses a great challenge for model evaluation. To construct the EHoA dataset from different light conditions, we devise a set of task commands and invite participants to interact with various objects in our daily scenarios to achieve these tasks. Finally, we hope not only that this dataset can apply to human beings' action recognition but also that the learning experience can be transferred to the robot.

Recently, lots of works [55, 160, 161] from the spiking neural networks (SNNs) field demonstrate that this kind of spike-based model can achieve a better performance than ANNs and CNNs, especially for the event-based recognition tasks, such as gesture recognition, image classification, and audio recognition. It has unique event-triggered computation characteristics that endow them a better inherent temporal dynamics and efficiency in processing spatiotemporal data [162]. To demonstrate that our proposed EHoA dataset can be applied to hand-object action recognition, we also propose an attention-based residual spiking neural network (ARSNN) to predict recognition results. Inspired by the development of residual architecture and attention mechanism,

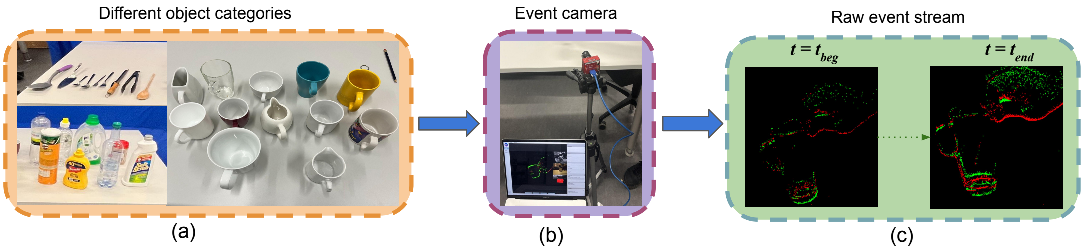


Figure 5.2: The data collection system: (a) objects from different application scenarios (b) DAVIS346 Red Event Camera. The computer is hidden for recording data and better illustration of the system. (c) The output event stream data when the participant interacts with a mug, where t_{beg} and t_{end} are determined as 0s and 5s, separately.

ARSNN will embed and stack multiple mixed attention modules into the deep residual structure where the main residual blocks are used to generate trunk features and mixed attention modules are adopted to fuse temporal-wise and spatial-wise attention to generate mask features. The features of both branches will be further concatenated to form an attention-based residual block. After successively stacking multiple attention-based residual blocks, the encoded high-dimensional vector with rich semantic information is propagated forward to predict the label of the interaction task. The primary contributions and novelties of this chapter are as follows:

1) Considering the dynamic motion, we propose a novel dataset of hand-object action represented by the event streams. For vision researchers, it complements previous work [44] well and provides a new neuromorphic dataset from real scenes for the researchers to develop their recognition algorithm. For robotic researchers, it also provides an inference benchmark to promote human-robot interaction in dynamic scenes.

2) We propose a novel attention-based residual spiking neural network (ARSNN) to address the challenging dynamic hand-object action recognition problem. The ARSNN can fuse spatial-wise and temporal-wise attention into the residual spiking blocks and achieve an excellent recognition performance.

3) We make a benchmark based on this dataset by evaluating it with the state-of-art baseline methods. Moreover, we also demonstrate that the learned model can be transferred to recognize the robot manipulation task in a dynamic environment.

5.2 Event-based Hand-object Grasping Dataset

5.2.1 Data Acquisition from Dynamic Interaction

Object Selection Inspired by the work [5], we select three categories of objects that achieve different hand-object action behaviors. The categories are defined as office, kitchen, and grocery store. These categories come from real interactive scenes of daily life, such as *pouring water from a mug, unscrewing a bottle cap to drink, using a spoon to enjoy dessert*, etc. Displayed in Fig. 5.2(a), each participant will interact with 30 objects during the collection process, where objects can be transparent, translucent, and opaque.

Capture Device and Recording Setup The data capturing system is shown in Fig. 5.2. We invite able-bodied volunteers to our laboratory to collect data in the following process: First, an object from the repository is randomly put on a flat table. Next, the participants stand in the data collection area and hold out their right hand to wait for commands to interact with the objects. Since we designed 8 different action types between hand and object, we tell participants the difference between different action commands and demonstrate each action type before they begin the data collection experiments. After they get familiar with different action commands while adhering to their daily habits in interacting with objects like grasping and moving, they are further instructed to move the object for around 5 seconds for each command. In order to ensure that the recording device fully captures the motion information both of the hand and the object, we suggest that participants move their hands in translation or rotation along one direction in space instead of moving randomly. It can be found that each participant will exhibit variations in the action direction and grasping posture for the same action type.

To capture the dynamic interaction process, we adopt DAVIS346 Red as the event camera module to record it. Unlike the conventional CMOS camera, which records the entire image at a fixed frame rate (e.g., 20fps), the event camera captures motion changes at a microsecond level. As shown in Fig. 5.1, when the interaction motion between hand and object is fast under natural light, the object appears deformed and blurred in the RGB image. At this time, it is not easy to distinguish the object type and analyze motion action through the traditional deep convolutional neural network, especially when the existing transparent objects will increase the challenge. In contrast, by observing the frame-based representation converted from the event stream captured by the event camera, the dynamic features of the hand and the object are well preserved.

Since the event camera also has a good dynamic range even though the lighting conditions change drastically [51] , similar to the work of [44], we designed three different illumination conditions for each experiment to simulate the lighting background in our daily lives. They are divided into *Low*(nightfall), *Normal*(morning), and *High*(afternoon) intensity. *Normal* means the experiment is conducted in the laboratory with all windows open, *High* means all LED lights on the ceiling are further turned on, and *Low* means all windows and curtains are closed and the lights are turned off.

5.2.2 Data Process and Analysis

Task-oriented Annotation To ensure adequate signal-to-noise ratios (SNR) conducive to task performance, aggregating events over a certain temporal window is the most efficient approach, which provides a more comprehensive view of the scene and is easier to annotate, label, and analyze for supervised machine learning [163]. Therefore, we convert the original event stream data E_t consisting of a sequence of e_j into a frame-based representation $F^{\tau,0} \in \mathbb{R}^{H \times W \times 2}$ in an accumulated aggregation method [53], which can be simply expressed as:

$$E_t = \{e_j | e_j = [x_j, y_j, t_j, p_j]\} \tag{5.1}$$

$$F^{\tau,0} = \mathcal{G}(E_t) \tag{5.2}$$

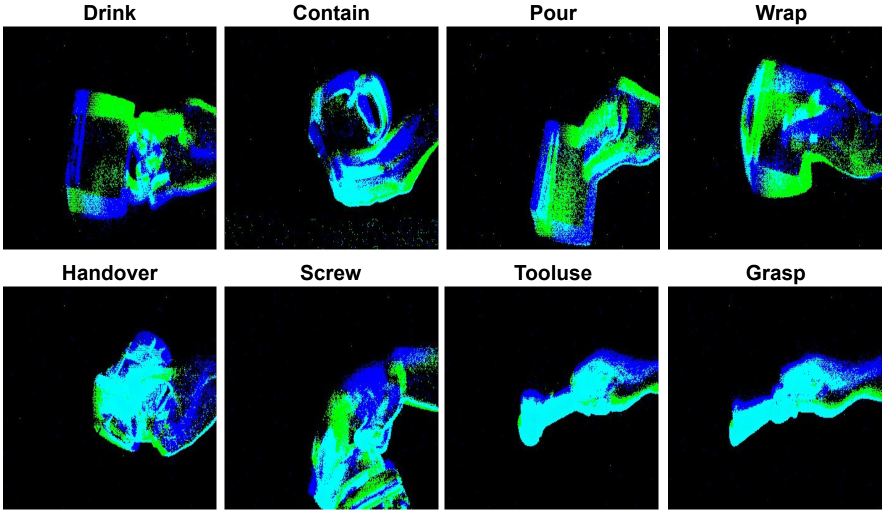


Figure 5.3: Eight hand-object action tasks are presented. Each sample is visualized by capturing high-rate event frames, which are transformed from continuous event streams. Many black areas indicate that the event intensity value of each pixel is zero.

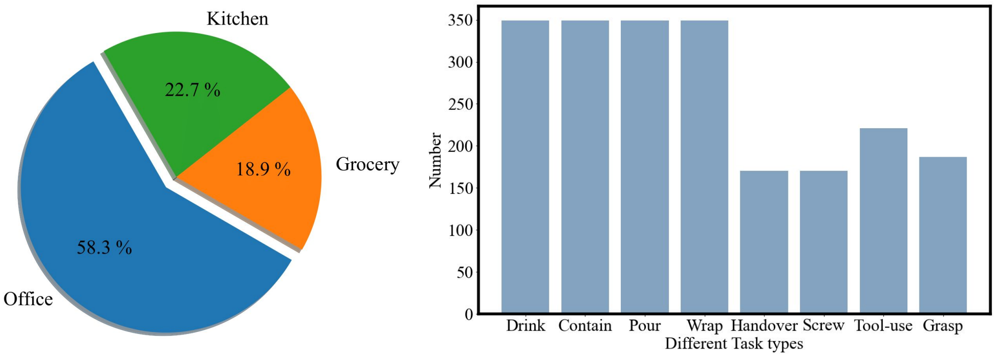


Figure 5.4: Distribution of different application scenarios (left) and corresponding objects from different tasks (right).

Where x_j, y_j, p_j means the spatial coordinates and polarity of event stream-based data at timestamp t_j, respectively. $\tau \in \{1, 2, 3, ..., T\}$ is the time-step index of the frame-based data and $\mathcal{G}$ is the aggregation function that splits the event's number into T slices with nearly the same number of events in each slice and integrates events into frames. Specifically, if assuming a two-channel frame as $F(j)$ and a pixel at (p, x, y) as $F(j, p, x, y)$, the pixel value p can be integrated from the events data as follows:

$$F(j) = \sum_{i=j_{index1}}^{j_{index2}-1} \mathcal{I}_{p,x,y}\left(p_i, x_i, y_i\right) \tag{5.3}$$

Where $index1$ and $index2$ represent the split indices from event data, respectively. The function $\mathcal{I}_{p,x,y}\left(p_i, x_i, y_i\right)$ is an indicator function that equals 1 only if (p, x, y) matches (p_i, x_i, y_i). Without the need to iterate over the entire sample explicitly, we further use the ”bincount” operation to efficiently accumulate the information on event polarity to

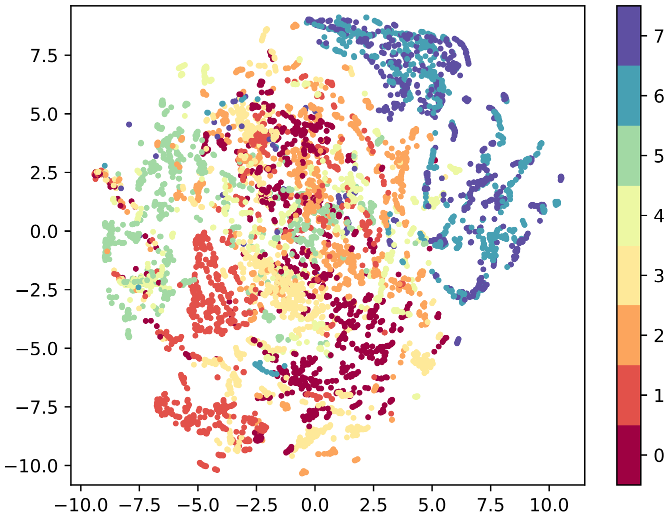


Figure 5.5: Visualization of the t-SNE embedding distribution of our 8 hand-object action tasks.

obtain the final pixel value p. Fig. 5.3 shows samples for each action task in a frame-based representation. The mug object at the office can be interacted with four kinds of task-oriented hand poses. While the bottle in the grocery store and the spoon in the kitchen can be manipulated with two kinds of hand poses. Thus, each data sample is annotated as a label from a task subset (Drink, Contain, Pour, Wrap, Handover, Screw, Tool-use, and Grasp). To ensure that the time length of each raw event stream sample is approximate, we trim all data samples into a similar length. In addition, we removed some data because some participants moved the object considerably, causing the object or hand to move out of the field-of-view (FOV) of the event camera.

Statistics Analysis Our dataset consists of 2144 event stream samples. Fig. 5.4 shows the proportion of each application scenario in the dataset and displays the exact number of each task-oriented type. Moreover, we employ the t-SNE method to investigate the distribution of 8 hand-object action types after converting raw event streams into event frames. As the accumulated event frames encompass information from both temporal and spatial dimensions, their dimensionality exceeds a single RGB image. To reduce the computational complexity of embedding, we amalgamate the dimensions of the temporal and spatial channels as a feature vector $H \times W \times 2$ to do the TSNE operation. Based on the TSNE method, we transform the data into lower dimensions

Table 5.1: Comparison with different datasets based on different features

Feature	DVSGesture [44]	FPHA [61]	ContactDB [62]	ContactPose [63]	H2O [64]	Ours
Natural hand shape	✔	✗	✗	✔	✔	✔
Natural object shape	✗	✗	✗	✗	✔	✔
Natural action	✔	✔	✗	✗	✔	✔
Task-oriented	✔	✗	✔	✔	✔	✔
Event streams	✔	✗	✗	✗	✗	✔
#Subjects	29	6	50	50	15	10
#Objects	0	4	50	25	15	30

using a perplexity value of 40 to get the lowest KL Divergence. The final segmented 2D visualization results are described in Fig. 5.5. Seeing the distribution of blue points (class 6) and purple points (class 7) from the kitchen scene, they cluster closely together due to their similar task and the distribution exhibits a discernible separation when the scenes of application differ. Furthermore, we compare our EHoA dataset with related datasets captured using an event or CMOS camera based on the number of subjects and objects, whether it is built from real scenes, and whether it involves hands, objects, and task-oriented intentions. Table. 5.1 shows a comparison of our EHoA dataset with other hand-object related datasets, where natural hand, object, and action represent that all data are collected from real human-object motion processes rather than through virtual environments or simulated action engines. We first compare the DVSGesture [44] that captures data based on the event camera while it does not consider the interaction with objects. Next, we compare recent hand-object datasets like FPHA [61], ContactDB [62], ContactPose [63], and H2O [64] that do not consider the dynamic interaction scenario.

5.3 ARSNN Architecture

5.3.1 Spiking Neuron Model

The spiking neuron is the fundamental computing unit of SNNs, where the LIF model and the Parametric Leaky Integrate-and-Fire (PLIF) model are commonly used to study the behavior of individual neurons. In the field of neurocomputing, both models capture the basic behavior of individual neurons and generate insights into how neural activity contributes to complex behaviors and functions.

The LIF model initially achieves a trade-off between capturing the intricate spatiotemporal dynamics exhibited by biological neurons and maintaining a simplified mathematical representation. In this model, the membrane potential of a neuron is modelled as a capacitor with a leaky current. When a neuron receives an input signal, its membrane potential increases until it reaches a threshold, at which point it *fires* and sends an output signal. The basic mechanism can be expressed as a differential function [164]:

$$\frac{dV}{dt} = \frac{-gL(V - E_L) + I(t)}{C_m} \tag{5.4}$$

Where V represents the neuron's membrane potential, gL represents the leak conductance, E_L represents the leak reversal potential, $I(t)$ represents the input current at time t, and Cm represents the membrane capacitance. To facilitate neural network training and inference, a simple iterative representation of LIF neuron can be expressed as:

$$\begin{cases} X^{t,n} = g\left(W^n, F^{t,n-1}\right) \\ U^{t,n} = H^{t-1,n} + X^{t,n} \\ S^{t,n} = Hea\left(U^{t,n} - u_{th}\right) \\ H^{t,n} = V_{reset} S^{t,n} + \left(\beta U^{t,n}\right) \odot \left(1 - S^{t,n}\right) \end{cases} \tag{5.5}$$

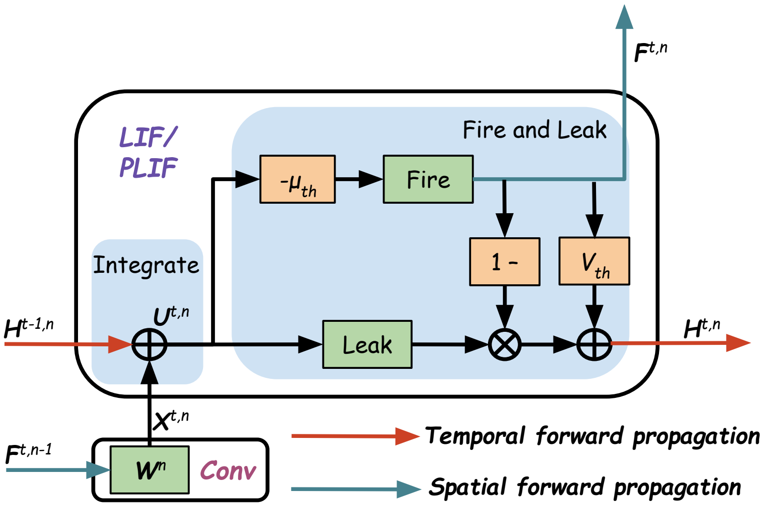


Figure 5.6: The fundamental spiking neuron unit (LIF or PLIF) in the Conv-based SNN layer.

where t and n denote the time step and layer. As stated in Section 5.2.2, we first transform the raw event streams from our EHoA dataset into frame-based representations. Frame-based representation $F^{t,n-1}$ is naturally compatible with traditional image and signal processing techniques, making it easier for researchers to develop different SNN algorithms, especially leveraging well-established convolutional neural network (CNN) frameworks. Therefore, g is a convolutional operation that converts the frame-based input $F^{t,n-1}$ into spatial features $X^{t,n}$:

$$X^{t,n} = AvgPool(BN(Conv(W^n, F^{t,n-1}))) \tag{5.6}$$

Within the integration phase, $U^{t,n}$ represents the membrane potential formed through the integration of the temporal input $H^{t-1,n}$ and the spatial feature $X^{t,n}$. In the spiking generation (*fire*) phase, u_{th} serves as the threshold for determining whether to activate the output spiking tensor $S^{t,n}$ or maintain it at zero. Hea(.) is a Heaviside step function that outputs 1 when the input is 0, otherwise, it outputs 0. Subsequent to the release of output spiking, V_{reset} denotes the reset potential, $\beta = e^{-\frac{dt}{\tau}} < 1$ represents the decay factor, and $\odot$ signifies element-wise multiplication. And the decayed value of membrane potential $H^{t,n}$ will serve as the temporal input for the subsequent timestep. The PLIF model is an extension of the LIF model that introduces a learnable membrane time constant to better simulate human neurons' behavior, which changes over time and affects the neuron's sensitivity for firing, which is proposed in [53] and can be described as follows:

$$U^{t,n} = H^{t-1,n} + \frac{1}{\tau}(X^{t,n} - (H^{t-1,n} - V_{reset})) \tag{5.7}$$

where τ denotes the membrane time constant. Fig. 5.6 shows the basic mechanism of spiking neuron unit from the LIF and PLIF models and visualizes the process of temporal-spatial-based forward propagation. The LIF-based and PLIF-based neuron models are both evaluated in our baseline methods to compare the performance based on our EHoA dataset.

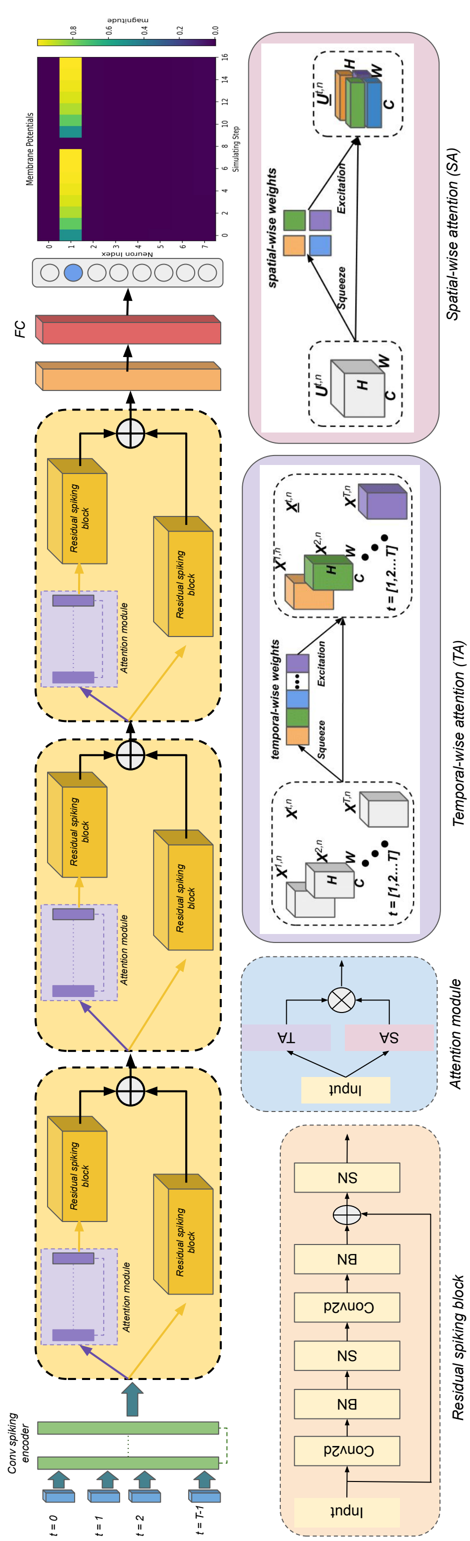


Figure 5.7: The upper row describes the overall architecture of the attention-based residual spiking neural network (ARSNN). The bottom row illustrates network details about the residual spiking block and temporal-spatial attention modules, Where conv2d, BN, and SN denote convolution operation, batch normalization, and spiking neurons, respectively. The final spiking output of each neuron is limited in $[0, 1]$ and the heatmap also visualizes the difference in their membrane potentials.

5.3.2 Temporal-wise and Spatial-wise Attention

As seen in Fig. 5.7, we first use a convolutional spiking encoder consisting of the Conv and spiking neuron to process the input event frames. The kind of conv-based spiking encoder is capable of feature extraction from input data and subsequent transformation into firing spikes occurring at various time steps. Furthermore, we introduce a novel attention-residual module designed for integration within deep spiking neural networks (SNNs). Our approach departs from the conventional practice of consecutively stacking attention modules. Instead, we introduce a trunk branch and a mask branch, facilitating the concatenation of the residual unit.

Temporal-wise and Spatial-wise Attention Instead of substituting the convolution layer with self-attention within the contemporary transformer structure, our aim is not to alter the fundamental meta-operator of established SNNs. Instead, we seek to introduce attention as a saliency component, making it seamlessly compatible with existing SNN architectures to enhance their representation capabilities. Temporal-wise attention (TA) in SNNs is a mechanism that allows the network to focus selectively on important temporal features of the input frames. It consists of two operations, the squeeze operation and the excitation operation [165]. The squeeze operation is a process where temporal information from all channels is consolidated into a singular vector through techniques like global average pooling or global max pooling. The subsequent excitation operation involves selectively amplifying or suppressing the informative temporal features obtained from the squeeze operation. This is achieved by applying a collection of learnable parameters or weights to the squeezed tensor and translating the compressed features into a set of input scores. Notably, the spatial-wise attention (SA) module follows a similar squeeze-and-excitation procedure while concentrating on the frame pixel information, much akin to the extraction of saliency from RGB images. If the average-pooling operation is chosen to aggregate the channel information, the output TA and SA scores vector can be expressed as:

$$\mathcal{S}_t = \sigma(W_{i1}(ReLU(W_{i0}(AvgPool(X^n))))) \tag{5.8}$$

$$\mathcal{S}_s = \sigma(W_{i2}(Conv(AvgPool(X^t)))) \tag{5.9}$$

where $AvgPool(X^n) \in \mathbb{R}^{T\times 1\times 1\times 1}$ means the output temporal attention feature and $AvgPool(X^t) \in \mathbb{R}^{1\times C\times H\times W}$ means the output spatial attention feature. W_{i0}, W_{i1}, W_{i2} are the weights of shared MLP layers and convolutional layers, and σ means the sigmoid function. After obtaining the temporal-wise and spatial-wise scores, we use a maximum function to merge the two attention score vectors based on input feature X to obtain the output mask feature $Max(\mathcal{S}_t, \mathcal{S}_s|X)$ of the attention module.

5.3.3 Attention-based Residual Learning

Similar to the idea of attention-based residual learning solving the image classification problem, it demonstrates that mask attention can well assist its counterpart without attention to improve performance [166]. In this work, we adopt the residual spiking block as the basic feature learning module, which is modified from the ResNet18 backbone

Algorithm 1 Feedforward of attention-based residual module

```
Input: $X$ - Input tensor with dimensions $[T, N, C, H, W]$
Output: Output tensor of the single module
procedure SkipConnection($X$)
    $identity \leftarrow$ ResidualSpikingBlock($X$)
    $\mathcal{S}_t = \sigma(W_{i1}(\text{ReLU}(W_{i0}(\text{AvgPool}(X^n)))))$
    $\mathcal{S}_s = \sigma(W_{i2}(\text{Conv}(\text{AvgPool}(X^t))))$
    $X^* \leftarrow \text{Max}(\mathcal{S}_t, \mathcal{S}_s|X)$
    $X \leftarrow$ ResidualSpikingBlock($X^*$) $+ identity$
    return $X$
end procedure
```

model. The output of the stacked residual attention module is defined as:

$$H(X) = Res(X) + Res(Max(\mathcal{S}_t, \mathcal{S}_s|X)) \tag{5.10}$$

Where X is the input feature of the stacking module. The mask branch feature in spiking neural networks is designed to selectively amplify informative features while suppressing noise and irrelevant features from the trunk features. As shown in Fig. 5.7, we stack three attention-based residual modules sequentially to refine the feature maps, hopefully improving the recognition performance consistently. Furthermore, the architecture integrates an additional residual spiking block, culminating in a fully connected (FC) layer tasked with the generation of a predictive action class. The details of the forward process for a single attention-based residual module are summarized in Algorithm 1.

Throughout the training procedure, we incorporate both the neuron models of LIF and PLIF, along with the newly introduced ARSNN network (as illustrated in Fig. 5.7), to facilitate both forward learning and backpropagation. Given an input data, our objective is to activate the neuron corresponding to the class exhibiting the highest response intensity while ensuring the rest of the neurons remain in an inactive state. To quantify the training progress, we employ the Mean Squared Error (MSE) as the chosen loss function:

$$Loss = \frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{\mathcal{M}}\sum_{i=0}^{\mathcal{M}-1}\left(y_{t,i}^{'} - y_{t,i}\right)^2 \tag{5.11}$$

where T is the time steps of frame-based data and $\mathcal{M}$ is the class number. $y_{t,i}^{'}$ and $y_{t,i}$ are the predicted output and target output.

5.4 Network Evaluation

5.4.1 Implemention Details

The EHoA dataset was partitioned into training and testing subsets, maintaining a distribution ratio of approximately 8:2. To account for the diversity in subjects and objects originating from our raw dataset, we split two distinct types of training and testing

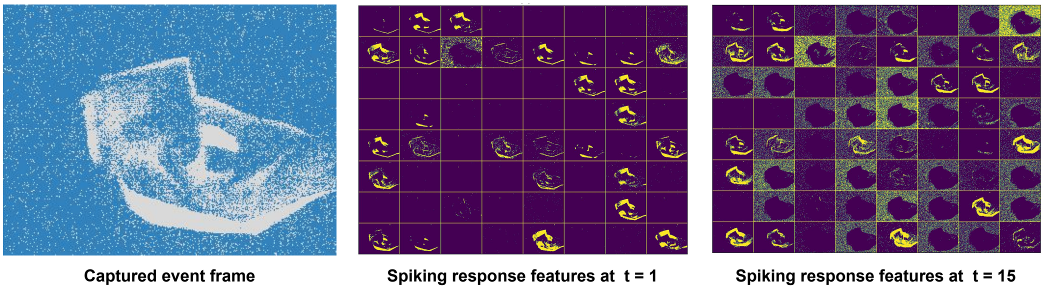


Figure 5.8: Visualization of spiking response of action ”*Contain*”. From left to right: a captured event frame of the ”*Contain*” action sample, and visual representations of the spike response features derived from 64 channels from the first spiking neuron layer at t = 1 and 15.

sets to evaluate model generalization. These two types are categorized as subject-based and object-based, implying that the test set comprises previously unseen hand shapes and unseen objects, respectively. Detailed specifications are available in our publicly released dataset. The details of the proposed network parameters are clarified as follows: The input data is represented using tunable frames. In our work, its tensor shape is described as $[N, T, C, H, W]$, where N is the number of batch size, channel C is 2, H is 260, and W is 346. To effectively evaluate the effect of other parameters, the frame number T of each data sample is set as 16. As seen in Fig. 5.7, we first use a convolutional spiking encoder to extract the basic temporal-spatial features, which could be expressed as {c64k7s2p3-BN-LIP/PLIF-MPk3s2}. $c64k7s2p3$ represents the output $channels = 64, kernel\_size = 7, stride = 1$, and $padding = 3$ from the 2D convolutional layer. And $MPk3s2$ means $kernel\_size = 3, stride = 2$ from the 2D max-pooling layer. For more hyper-parameter details, we use the Adam optimizer with a learning rate of 0.001 to optimize the gradient-descent process. The NVIDIA 2080Ti is used to train the proposed network, and the batch size N is set as 2 due to limited GPU memory. The method of automatic mixed precision is also adopted to improve the training speed and reduce memory consumption.

5.4.2 Visualization of Spiking Encoder

To more intuitively explain the temporal-spatial features learned by the spiking encoder on our EHoA dataset, we input an event frames sample of action ”*Contain*” in the test set to evaluate the trained ARSNN model. And the output spike responses $S_c^{t,n}$ from different c-th channels, n-th spiking layers and timestep $t \in [1, 2, .., 16]$ can be inferred as a visualized feature map. As shown in Fig. 5.8, we choose the first spiking neuron layer ($n = 1$) to visualize spikes from all output channels ($c = 1, 2, 3, ..., 64$) because a deeper neuron layer may generate richer semantic information while visualized feature maps are harder to understand. If we further select the number of output channels as 1, then the obtained dimension of single feature map $S_1^{t,1}$ is computed as (130, 173), retaining the same dimension from the output of the convolutional layer ($c64k7s2p3$).

Each pixel in the feature map $S_1^{t,1}$ corresponds to the spiking activity rate of an individual neuron, where the yellow pixel indicates a high spiking activity rate, while the

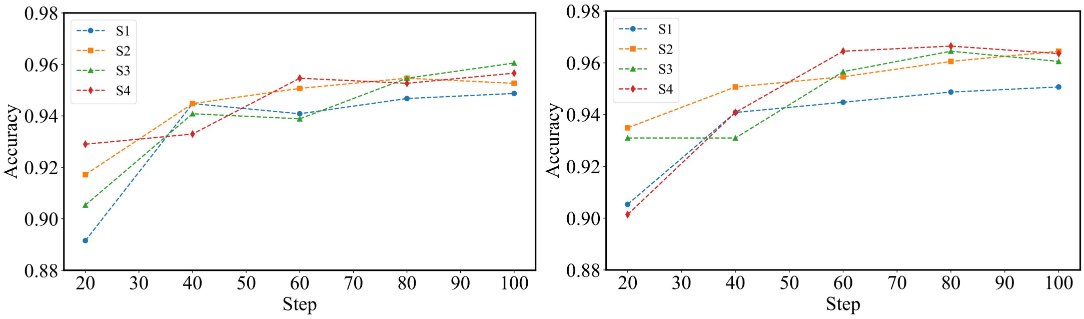


Figure 5.9: Accuracy comparison from different RTSA positions based on the EHoA dataset, split by the objects. Left: LIF neuron model. Right: PLIF neuron model.

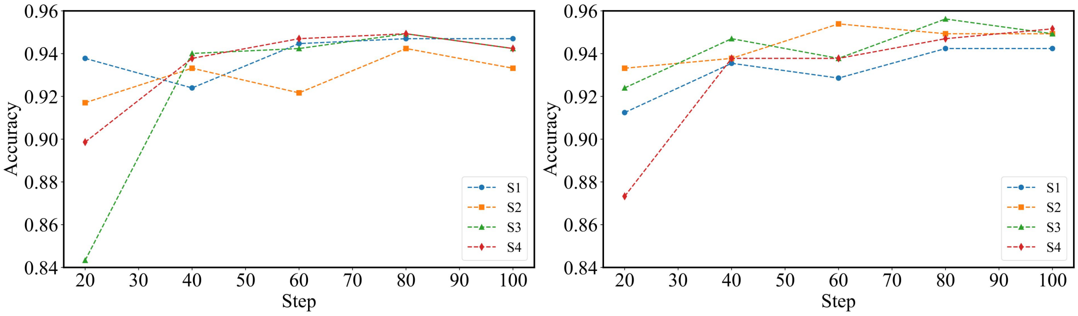


Figure 5.10: Accuracy comparison from different RTSA positions based on the EHoA dataset, split by the subjects. Left: LIF neuron model. Right: PLIF neuron model.

purple pixel indicates a spiking activity rate approaching zero. It can be seen that spiking response features vary across different time steps. Specifically, at time step $t = 1$, the network appears to prioritize the separated details from hand and object, while at $t = 15$, it seems to focus on detecting a distinct edge from hand-object motion. The overall spike responses at different timesteps show that the spiking neuron encoder can effectively perceive both spatial-variant and temporal-variant input data from dynamic hand-object motion.

5.4.3 Ablation Study of Attention Position

When training our proposed ARSNN, it is necessary to evaluate the effect of different positions of the residual temporal-spatial attention (RTSA) module. we define four different RTSA positions: S1: pure residual spiking block without residual attention; S2: insert RTSA in the first residual block; S3: insert RTSA in the last residual block; S4: insert RTSA in the whole residual block illustrated in Fig. 5.7. Through the four settings, we evaluate the accuracy of LIF vs. PLIF neuron units based on the EHoA dataset split by the variety of objects and subjects. When the max training epoch is set as 100, Fig. 5.9 and Fig. 5.10 demonstrate that the introduction of an attention-based residual module can improve performance in most cases. In addition, the S4 setting can achieve the best average accuracy. We can also observe that the recognition performance from the PLIF neuron model is better than the LIF neuron model overall after adding the RTSA module.

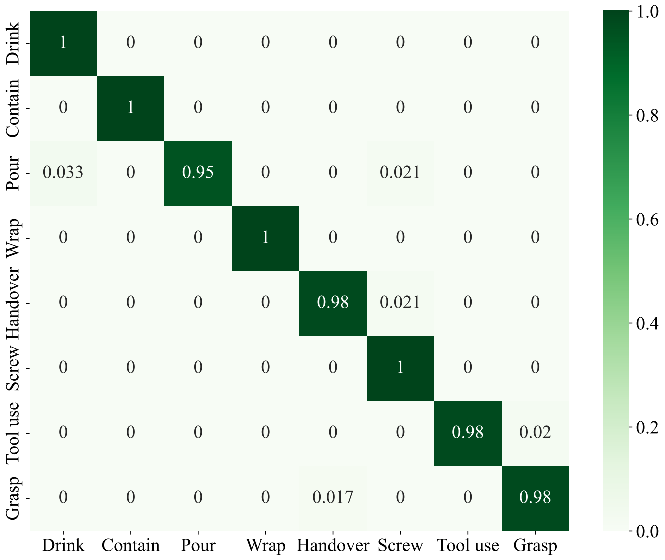


Figure 5.11: Confusion matrix for the EHoA dataset, split by objects using the proposed AR-SNN framework.

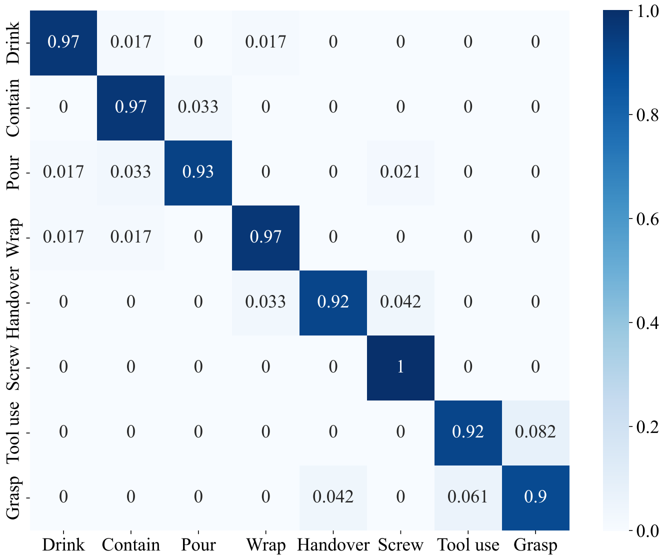


Figure 5.12: Confusion matrix for the EHoA dataset, split by subjects using the proposed AR-SNN framework.

5.4.4 Evaluation of Action Recognition

In our evaluation experiments, we first use the PLIF neuron model to embed in our AR-SNN network framework and different train sets splitting by subjects and objects are adopted to train the inference models. After that, the inference models are used in different test sets to obtain recognition accuracy. Fig. 5.11 and Fig. 5.12 represent the confusion matrix maps split from objects and subjects, respectively. Overall, the accuracy of the objects-based dataset is higher than that of the subjects-based dataset. For example, the performances of tasks for *Handover*, *Tool-use*, and *Grasp* actions are 98%, while the performances of subjects for these action labels are about 92%. It demonstrates that the

Table 5.2: Accuracy and computational cost of models for the EHoA dataset using the LIF and PLIF neuron

Method	Features	LIF		PLIF		Param	Flops
		Object Acc. (%)	**Subject Acc.** (%)	**Object Acc.** (%)	**Subject Acc.** (%)	(M)	(G)
RSNN-FB [167]	Recurrent Feedback	88.36	87.56	89.94	86.41	1.24	1.59
RSNN-SF [167]	Stateful Synapse	91.12	89.84	92.11	89.86	1.24	1.59
CSNN [53]	ANN-SNN	93.29	89.86	94.28	91.71	0.71	1.58
Resnet18-SNN [161]	ANN-SNN	94.87	**94.70**	95.01	94.93	11.17	53.76
Resnet34-SNN [168]	Hybrid training	94.67	89.86	92.31	90.53	21.28	110
TASNN [55]	Temporal attention	94.01	91.01	94.93	93.12	11.17	62.01
ARSNN	Residual attention	**95.66**	94.24	**95.86**	**95.16**	11.18	78.53

Table 5.3: Comparison results for other frame-wise representations and pure CNN frameworks based on our EHoA dataset

EHoA	Frame-wise representaion			3D CNN		ARSNN
	TBR [169]	Polarity [170]	SAE [171]	ResNet [172]	X3D [173]	Our
Object-based	95.38	94.21	93.49	87.56	93.08	**95.66**
Subject-based	**94.65**	93.10	92.52	85.33	92.11	94.24

variation of the human hand and hand pose during the interaction process makes it more difficult to generalize compared to the variation of objects. Otherwise, we also report the recognition accuracy in Table. 5.2 using LIF and PLIF neuron models by comparing with the state-of-art SNNs-based baseline methods. The results demonstrate that the PLIF neuron model has a better out-spiking accuracy than the LIF model. And introducing a residual temporal-spatial attention module can also improve the final performance, especially compared with temporal attention-based SNNs [55]. In the context of evaluating model complexities, it can be seen that the comprehensive number of parameters and the flops of our model are bigger than the simple RSNN and CSNN models because our proposed ARSNN framework employs the ResNet18 architecture as its foundational backbone. However, compared with a more sophisticated SNN model, such as Resnet34-SNN, our proposed model demonstrates that the introduction of spatial-temporal attention can effectively mitigate complexity without necessitating an increase in the network's depth, while better performance can also be obtained. Furthermore, compared with the TASNN model, the number of parameters changes little while our model for all four kinds of evaluation methods can achieve a better performance. We recognize that it's still challenging to achieve a trade-off between advanced model performance and simplified model complexity

In this work, we refer to the work [53] to accumulate events to frames by setting the frame number. we also compare three baselines from other aggregation approaches by accumulating time: TBR [169], Polarity [170] and SAE [171]. Table. 5.3 shows that TBR representations can achieve similar performance with our method (LIF-ARSNN) because they both preserve polarity features and temporal information. Furthermore, we

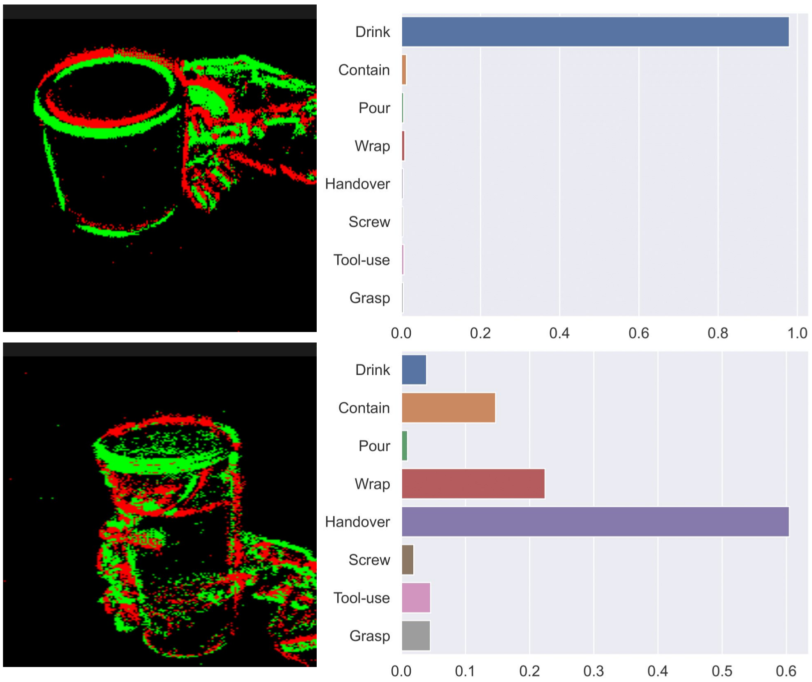


Figure 5.13: Visualization of the raw event streams and corresponding average response intensities output by our learned model. Two action tasks: *Drink* (top) and *Handover*(down).

also try to use recent 3D CNN methods modified from ResNet [172] and X3D [173] to process our event frames by regarding them as two-channel high-rate video and compare their performance with our LIF-ARSNN framework. It can be seen that the evaluation results of pure CNN methods are worse than spike-based SNN methods, indicating that the SNNs have a better performance to naturally handle sparse data by generating spikes in response to events, making them more efficient for processing such information compared to dense CNNs.

5.5 Robot Hand-object Grasping Recognition

Since our ARSNN framework demonstrates proficiency in recognizing various hand-object actions from actual human hand interactions, we are also keen to extend this capability to mechanical dexterous hands and virtual hands. This expansion holds significant promise for a wide range of applications, particularly in the context of humanoid robots and the burgeoning Metaverse. In this work, we design a robotic experiment to apply the learned model for the recognition of dexterous hand-object actions performed by the PR2 humanoid robot, which has been retrofitted with ShadowHands within our laboratory. We collected 8 extra event stream data by controlling the robot hand to manipulate a variety of objects across different action types. Fig. 5.13 illustrates two samples of raw event stream data alongside their corresponding average spiking response intensities output by our model. Specifically, the *Drink* label means the hand will only grasp the ”handle” part of the ”mug” category, and the *Handover* label means the hand will only grasp the ”body” of the ”bottle or can” category. It’s noted that the original spiking neuron's output is binary, and its reliability can be affected by inherent noise stemming from the encoding process. Consequently, to derive our model's output, we consider the average response intensity across the final output over a specified time period T. Validated by the results from *Drink* and *Handover* actions, our ARSNN model can also recognize actions for robot-object manipulation successfully. It can also observed that the *Handover* action is pretty challenging to recognize because it is similar to the *Wrap* and *Contain* actions. Otherwise, we also found in real deployment experiments, it will take a response time of 250-350ms to achieve smooth recognition for all eight robot hand-object action samples. However, the entire inference time from our original ARSNN model is about 5ms, we speculate that this discrepancy can be attributed to hardware and motion limitations that affect the robots' ability to perform certain actions smoothly compared to human hands.

5.6 Discussion and Summary

In this chapter, we propose a novel event-based dataset, EHoA, for visual analysis of dynamic human-object action recognition. Eight types of annotations for action behaviors with 10 subjects and 30 objects under three light conditions are involved in providing comprehensive labelled tasks. We also propose a novel ARSNN and benchmark the state-of-the-art baseline methods to complete the visual analysis and achieve challenging dynamic action recognition tasks. To our knowledge, EHoA is the first dataset using event vision for human-object action tasks, which has been extensively studied in the computer vision field and robotics community. Ultimately, we also show the potential for a trained recognition model to be transferred to real robot manipulation tasks.

In the future, we want to combine the IMU and other motion sensors to collect real-time pose information when hands interact dynamically with objects. Furthermore, we will fuse multi-modal data and design better visual analysis algorithms to explore potential hand-object pose estimation tasks. We view this as an initial step toward building a more extensive dataset in the domain of event vision, specifically for human-object interaction tasks.

Chapter 6

Multimodal Transfer Learning for Robotic Multiple Peg-in-Hole Assembly

Apart from investigating hand-object interactions, we further examine object-object manipulations from the perspective of interactive perception. Robotic rigid contact-rich manipulation in an unstructured dynamic environment requires an effective resolution for smart manufacturing. As the most common use case for the intelligence industry, a lot of studies based on reinforcement learning (RL) algorithms have been conducted to improve the performances of single peg-in-hole assembly. However, existing RL methods are difficult to apply to multiple peg-in-hole issues due to more complicated geometric and physical constraints. In addition, previously limited solutions for multiple peg-in-hole assembly are hard to transfer into real industrial scenarios flexibly. To effectively address these issues, this chapter designs a novel and more challenging multiple peg-in-hole assembly setup by using the advantage of the industrial Metaverse. We propose a detailed solution scheme to solve this task. Specifically, multiple modalities including vision, proprioception, and force/torque are learned as compact representations to account for the complexity and uncertainties and improve the sample efficiency. Furthermore, RL is used in the simulation to train the policy, and the learned policy is transferred to the real world without extra exploration. Domain randomization and impedance control are embedded into the policy to narrow the gap between simulation and reality. Evaluation results demonstrate the effectiveness of the proposed solution, showcasing successful multiple peg-in-hole assembly and generalization across different object shapes in real-world scenarios.

6.1 Introduction

To prompt the high-quality development of the industry, intelligent robots have become indispensable in realizing many manufacturing processes [174, 175]. Taking the assembly task as an example, the global intelligent assembly market is expected to grow by 30% over the next four years [176]. The most obvious characteristic of an assembly task is that it involves mechanical interaction and fits between two or more objects, such as clearance fits, transition fits, and interference fits. Therefore, in order to achieve a high-

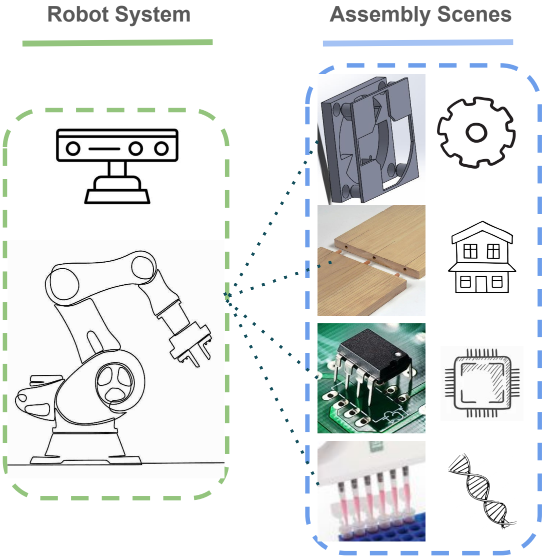


Figure 6.1: Examples of multiple peg-in-hole assembly scenes from smart manufacturing processes. From top to bottom: mechanical fits, furniture assembly, semi-conductor insertion, and multiple-channel pipette.

precision assembly, research in multiple dimensions should be considered, such as the redundancy and clearance of the robot's own mechanical precision, pose uncertainties between peg and hole objects, and the complex physical models involved in each assembly scene, consisting of geometry, contact force, and kinematics [70, 177–179]. To improve assembly performance, reinforcement learning (RL) has been recently introduced to learn assembly skills through interaction with environments. The most significant benefit of using RL algorithms to learn assembly skills is that they enable the continual solution of the search and insertion processes to generalize previously unknown assembly challenges, such as various clearance and shape requirements [180]. Especially for the single peg-in-hole assembly, many studies are conducted to achieve promising results [70,75,77,181]. However, there exist few studies on multiple peg-in-hole manipulation because of a more complicated geometric and physical interaction model [93]. Fig. 6.1 shows different applications related to multiple peg-in-hole assembly tasks in the intelligent industry, e.g., mechanical fits, furniture assembly, semi-conductor insertion, and multi-channel pipette. Despite this, the experimental setup of previous multiple peg-in-hole has many flaws, like the peg is fixed on the end-effector, the 6-DOF pose of the holes object stays constant, the shape of the holes object and pegs object is immutable, and lacking the visual feedback. Practically, their setup with these limitations is not in line with the actual multiple peg-in-hole assembly scenes [96–98].

On the other hand, the industrial Metaverse as an important branch of Metaverse has attracted more and more attention from robotics researchers, which hopefully improves the safety and efficiency of applications in each phase of manufacturing. The concept of the industrial Metaverse can be defined as a new digital twin system of the real industry, consisting of large-scale industrial data processing, industrial process simulation, and natural human-machine or human-robot interaction [182]. Previous work also demonstrates that industrial Metaverse is highly related to Digital Twin and Simulation, and transferability is the key technology for the deployment of the industrial Metaverse [183, 184].

In this chapter, we design a new multiple peg-in-hole assembly setup to solve the flaws mentioned above from previous work and maximize the transferability which means a successful policy learned from our setup can be easier to deploy in real manufacturing scenarios. Based on this more challenging task, we also propose an end-to-end multimodal learning architecture using reinforcement learning, where features of multiple modalities are compacted into latent representations at a high level via a tokenization-based model. It enables robotic agents to leverage the complementary nature of these sensing modalities for policy learning. By exploring the previously mentioned concept of the Industrial Metaverse, a simulation environment with the setup is constructed to train the policy with a Soft Actor-Critic (SAC) algorithm [185]. The learned policy can be transferred to the real world directly. In addition, domain randomization is used in simulation to narrow the gap between the simulation and the real experimental setup. Furthermore, impedance control is designed and embedded into the proposed architecture, which helps the policy deal with our physical contact-rich task. More specifically, we aim to enable the RL agent to learn impedance control strategies, as it has been widely demonstrated that impedance control can improve system compliance, which is quite essential for contact-rich tasks [186, 187]. Finally, the proposed assembly task is evaluated both in the simulation and real robot experiments, demonstrating that the proposed multiple modality-driven impedance-based policy trained with domain randomization achieves successful dynamic assembly. To the best of our knowledge, this is the first work to learn such a challenging assembly skill for multiple peg-in-hole on a real robot. The primary contributions and novelties of this chapter are:

1) We define a novel and more challenging experimental setup for multiple peg-in-hole assembly task, which is easier to apply to the real application scenario than previous work [96–98].

2) To achieve the trade-off between excellent visual representation and fast visual training, we sample a visual dataset with different object shapes based on our setup and then construct a special module to learn a visual feature representation.

3) A tokenization method based on the transformer architecture is proposed to extract features from robot proprioception and force/torque signals and the extracted features are further fused into a compact multimodal representation.

4) With domain randomization and impedance control, the policy for dynamic assembly can be learned successfully in simulation and then transferred to reality without extra exploration.

5) Experimental results show the trained policy could achieve generalization to tasks with different peg shapes under object uncertainties.

6.2 Problem Formulation

We consider a scenario, where a robot manipulator consisting of a robot arm and a parallel-jaw gripper interacts in finite episodes over discrete steps with a multi-peg ob-

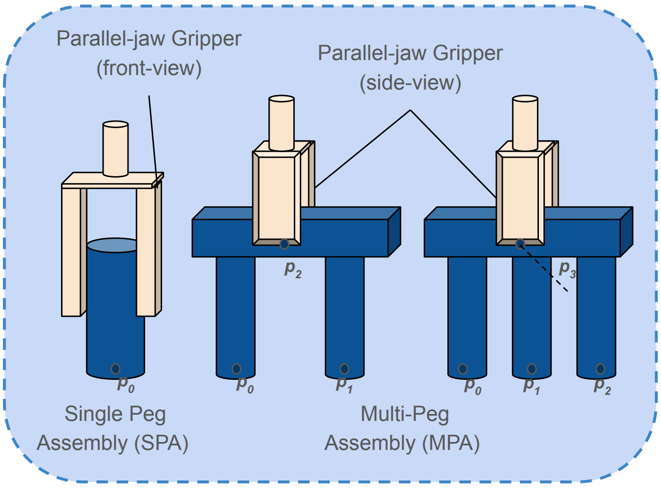


Figure 6.2: The geometric and contact difference between single peg assembly (left) and multi-peg assembly (right) with parallel-jaw gripper, p_i means the center of each object element.

ject that needs to be inserted in a multi-hole object on a flat table. As seen from Fig. 6.2, the parallel-jaw gripper is movable and used to grasp the top connector from multiple pegs, instead of directly grasping the backbone part of the peg as the single-peg-assembly (SPA) problem. In addition, the contact points of the gripper and the connector are not fixed. So unlike the previously defined multiple-peg-assembly (MPA) problem [98, 99], the state information of the peg cannot be implicitly represented through the pose of the end-effector. Furthermore, the position and orientation of the multi-hole object vary in a limited range.

Since full states of the peg and object can not be obtained directly from the real world, an extra RGB camera is set up to provide their state information. To address this challenging problem, multiple modalities like visual image, proprioception, and force/-torque, observed from the manipulator and camera, are used to learn an impedance-based control policy. Finally, we formulate the dynamic multiple peg-in-hole assembly task as a Partially Observable Markov Decision Process (POMDP). The optimal policy π^* is finally obtained by maximizing the expected cumulative reward value:

$$\pi^* = \underset{\pi}{\operatorname{argmax}} \mathbb{E} \left[\sum_{n=1}^{N} \gamma R\left(s_t, a_t, g\right) \right] \tag{6.1}$$

where $S = \{s_1, s_2, ..., s_t\}$ is a set of states, $A = \{a_1, a_2, ..., a_t\}$ is a set of actions, $S : S \times A \rightarrow \mathbb{R}$ and $\gamma \in [0, 1]$ is the discount factor. The goal g is used to determine whether the problem is solved successfully or the current training step is at the end of the episode. Although the policy π^* is trained in simulation, it can be directly transferred to the real world. Furthermore, both simulation and real setup embed an impedance controller to generate torque commands to drive the manipulator executing assembly.

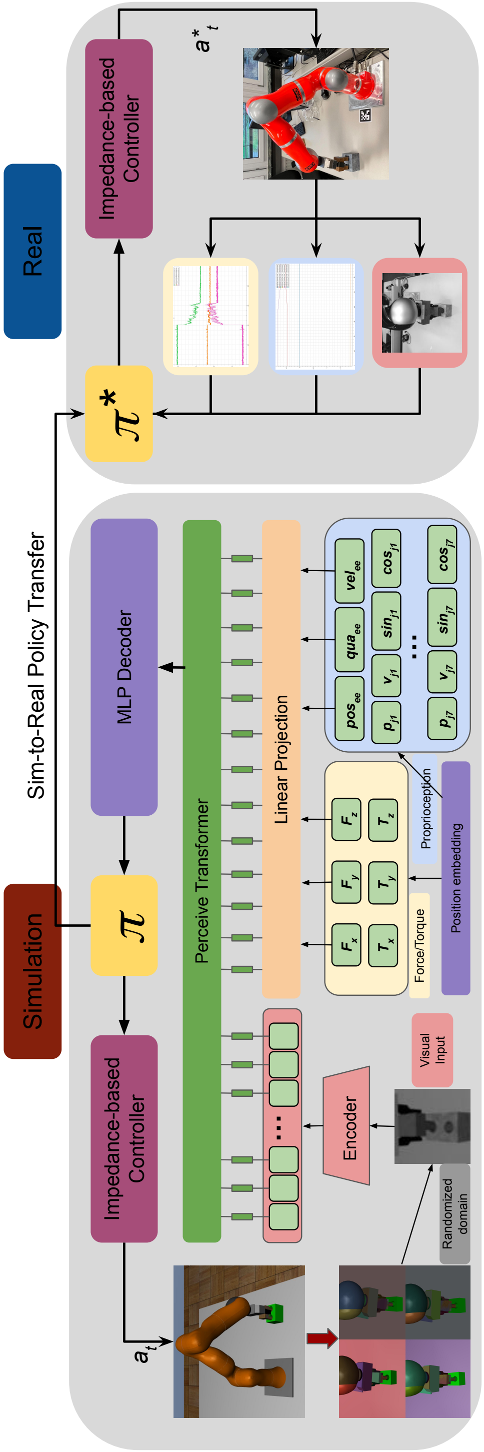


Figure 6.3: Overview of the proposed architecture. The left part represents policy training in the simulation environment, where domain randomization is used to sample the interaction of the assembly task like color, lighting, camera, and robotic dynamics (left). During the training process, multiple modalities including visual image, proprioception, and force/torque signals are all tokenized and fused into a perceived transformer module. Each predicted policy π is embedded into an impedance controller to execute torque commands to control the manipulator. Finally, the trained policy π^* is transferred to the real world without additional exploration (right).

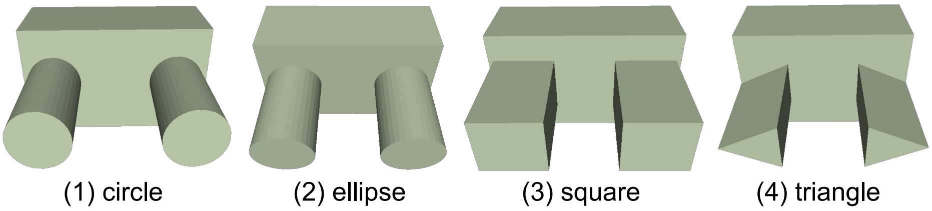


Figure 6.4: Visualization of different multi-peg object shapes (1) circle (2) ellipse (3) square (4) triangle during the collecting process of the visual assembly dataset.

6.2.1 Latent Representation Learning Framework

In this section, we focus on utilizing the multiple modalities in the simulated robotic environment to learn a robust policy and then transfer it to the challenging assembly task. An overview of our proposed architecture for multiple peg-in-hole assembly is depicted in Fig. 6.3. First, we propose a pretraining-training approach to learn the latent representation from the visual frames where domain randomization is used to collect a robust assembly dataset. Seeing the simulation part in Fig. 6.3, the image observation is input into the pre-trained encoder during the policy training process to obtain a latent feature embedding. Second, a self-attention-based transformer module is applied to learn the dependencies between robot proprioception states, force-torque signals in the Cartesian space, and the extracted visual embedding, where position embedding and linear projection are used to get the token features of proprioception and force/torque values. Next, the robot actions are predicted and mapped to the impedance controller after the compact multimodal features pass through a Multilayer Perceptron (MLP) decoder, and the SAC-based RL algorithm is adopted to obtain each predicted policy π during the whole training process. After obtaining the optimal policy π^* in simulation, the policy is transferred to control the manipulator and finish real assembly tasks directly by capturing real visual images, robot proprioception, and force/torque signals.

6.2.2 Pretraining-training from Visual Input

We first collect an image dataset with different peg and hole shapes. As shown in Fig. 6.4, different object shapes like circle, ellipse, square, and triangle are used for the due-hole assembly. To collect the dataset, we use the ShuffleNet-v2 [188] as the visual model to extract visual features and execute policy learning using our proposed architecture. When the robot moves 5 steps, the camera in the simulation environment captures an image and saves it into the dataset. In order to prevent insufficient memory during RL learning, the original image is further cropped. Finally, the height and width of each image are set to 64, the amount of each object shape in the dataset is set to 4000, and the interaction details about the peg and hole should be retained.

After finishing the dataset construction, our visual representation module is firstly used to pretrain all data, shown in Fig. 6.5. With the pretraining of the encoder-decoder architecture, a prior of the visual image for each standard peg and hole shape can be learned. The latent space vector $\mathcal{Z}$ is learned from the encoder FCN layers $E_\phi(m)$. At this stage, we aim to learn a function that maps the image pixel $m \in \mathbb{R}^{H \times W \times 1}$ to its

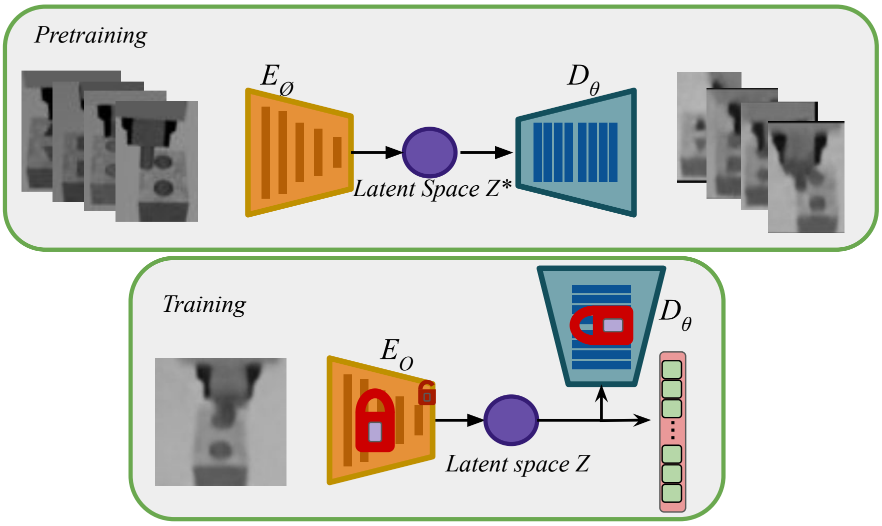


Figure 6.5: Overview of the visual representation module. 1) pretraining: We first pre-train the feature encoder and decoder. After that, the latent space Z^* to recognize different object shapes is obtained. where ϕ and θ means different weights, E_ϕ and D_θ means the encoder network loaded with weights ϕ and the decoder network loaded with weights θ, respectively. 2) training: The decoder and most encoder layers are frozen, and we train the encoder during the policy learning process.

probability distribution function (PDF) $p \in \mathbb{R}$:

$$D_\theta(\mathcal{Z}, m) = PDF(m) \tag{6.2}$$

In this work, the probability distribution of PDF follows a Gaussian distribution. In addition, we use the reparameterization trick to sample the PDF and pass it through the decoder D to get a distribution $p(\mathcal{Z})$ for the predicted image $f(\mathcal{Z})$. As sampling details, the mean and covariance functions g and h are separately used to minimize the Kullback-Leibler (KL) divergence between the approximation $p(\mathcal{Z})$ and the target $q_m(\mathcal{Z})$. Finally, the loss function L_r that regresses the distribution value for each image pixel m is defined as:

$$q_m(\mathcal{Z}) = \mathcal{N}(g(m), h(m)) \tag{6.3}$$

$$L_r = \underset{(f,g,h)}{\arg\max} \left(\mathbb{E}_{\mathcal{Z}\sim q_m}\left(\log D_\theta(\mathcal{Z}, m)\right) - KL\left(q_m(\mathcal{Z}), p(\mathcal{Z})\right)\right) \tag{6.4}$$

where $\mathcal{N}$ is a gaussian-based sampling distribution with mean $g(m)$ and covariance $h(m)$. A potential dilemma for our visual representation module is to align the latent vectors produced by the encoder to significant latent vectors for the decoder with slightly different correspondence due to the restricted amount of our image data. As a result, only the weights ϕ of the encoder $E_\phi(m)$ are optimized during the training phase, whereas the weights θ of the decoder D_θ remain frozen. To speed up the whole policy learning, we further freeze all encoder feature layers except the output layer. Although in the pretraining process, we need multiple images of the different hole and peg shapes. Our final pipeline only needs a single image captured by the camera at inference time during RL training.

Our encoder architecture consists of five hidden blocks where each block includes a Conv2D layer, batch norm layer, and Leaky ReLU activation function. The hidden dimension of each block is expressed as [32, 64, 128, 256, 512]. After that, the output feature map $f \in \mathbb{R}^{512\times512\times1}$ is flattened into a feature vector $\mathcal{Z}$. The latent mean and covariance features are split by concatenating the feature vector with the bi-stream FC layer. In our experiments, we only choose the latent mean feature as the potential visual embedding. As the decoder, it first uses a fully connected layer to reduce the feature $\mathcal{Z}$ to a dimension of 2048. Then, it is reshaped into a feature map $F \in \mathbb{R}^{512\times2\times2}$ and put into five reverse blocks in the encoder. After that, two Conv2d layers are concatenated to restore the feature map into the same shape with visual input $v \in \mathbb{R}^{H\times W\times1}$.

6.2.3 Tokenization Model Based on Multiple Modalities

The proposed architecture for multiple peg-in-hole assembly tasks is presented in Fig. 6.3. The modalities will be tokenized consisting of vision, proprioception, and force/torque values.

First, the latent mean feature vector (from Section 6.2.2) is regarded as the visual embedding of our transformer encoder architecture. The force/torque signals are obtained from the end-effector, represented by states in three directions, which could be expressed as $[F_x, F_y, F_z]$ and $[T_x, T_y, T_z]$. The force and torque values will change sharply when the peg grasped in the gripper moves down to the target hole especially if there exists touch and friction force between pegs and holes. That demonstrates that the force and torque values are important modality features to learn in our assembly task. The proprioception consists of 38-dimensional states. Concretely, $[p_{ji}, sin_{ji}, cos_{ji}, v_{ji}]$ indicates 7-DOF robot joint states where i means the joint index number, v means the joint velocity value, and sin, cos means the different mathematical operation to the joint position state. $[pos_{ee}, qua_{ee}, vel_{ee}]$ indicates the robot end-effector states, representing the position, quaternion, and velocity values, separately.

In the traditional NLP and computer vision field, the transformer-based module has been widely used to learn common dependencies between word embeddings and image batch embeddings [139, 189]. In this chapter, we try to learn the dependencies between different states from multiple modalities because all modalities are necessary for the challenging multiple peg-in-hole assembly task. For example, the visual modality could help the robot detect the peg's relative position and orientation to the hole. Proprioception and force/torque can well reflect the motion states of the robot when performing tasks. However, some recent work stitches them together uniformly and then uses a large transformer framework to process them [190, 191]. This big framework often includes three submodules, like encoder, process, and decoder, each of which introduces a complex attention mechanism. This will undoubtedly cause the system to require huge computing and memory resources.

As seen from Fig. 6.3, our method attempts to reduce the complexity of the big model. To achieve successful dependency learning between each modality, the proprioception and force/torque states are first separated into scalar values. Then, we use a one-hot vector of 44 dimensions to distinguish each scalar value. This vector is added to the concatenation of proprioception and force/torque states as the positional embed-

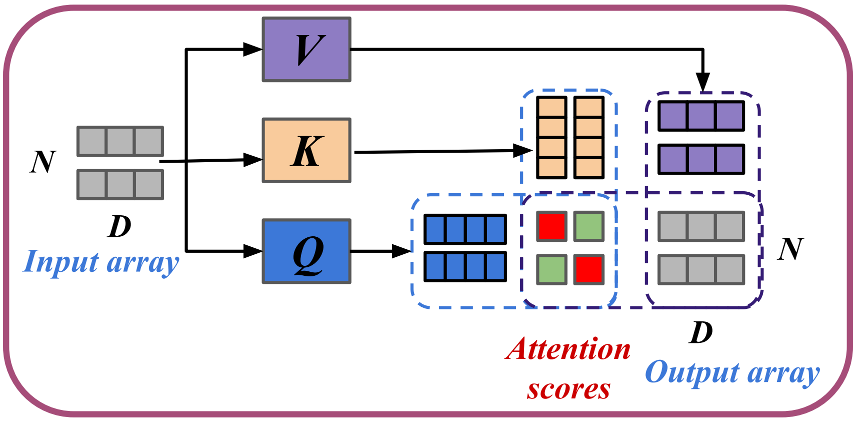


Figure 6.6: The attention module for perceived transformer block. X is the input array $\in \mathbb{R}^{N \times D}$, $X_{\mathcal{QK}}$ is the attention scores, visualized as red and green colors, and $X_{\mathcal{QKV}}$ is the output array $\in \mathbb{R}^{N \times D}$.

ding. Furthermore, these encoded robot states are fed into a linear projection block, expanding the representation features of the concatenated robot states into 64 dimensions. After that, the expanded representations are concatenated with the latent mean embedding from visual input into a 128-dimensional feature embedding. Finally, it is fed into three consecutive blocks from the perceived transformer architecture.

Following the structure of Perceiver [190] and our previous work [3,5], the attention module of our transformer block is shown in Fig. 6.6. This block processes the input latent array using a global query-key-value (QKV) attention operation (N=1, D=128). Following this, the multi-layer perceptrons (MLP) are used to independently process each element of the index dimension. The linear projection layers alone with MLP and QKV operation ensure the output latent array of the block has the same index dimension as the input. The QKV operation in this block could be formulated as:

$$\mathcal{Q} = f_{\mathcal{Q}}(X); \mathcal{K} = f_{\mathcal{K}}(X); \mathcal{V} = f_{\mathcal{V}}(X) \tag{6.5}$$

$$X_{\mathcal{QK}} = \text{softmax}\left(\mathcal{QK}^T / \sqrt{SF}\right) \tag{6.6}$$

$$X_{\mathcal{QKV}} = f_O\left(X_{\mathcal{QK}} \mathcal{V}\right) \tag{6.7}$$

As shown in Fig. 6.6, X is the input array $\in \mathbb{R}^{N \times D}$, $X_{\mathcal{QK}}$ is the attention scores, visualized as red and green colors, and $X_{\mathcal{QKV}}$ is the output array $\in \mathbb{R}^{N \times D}$. The functions $f_{\mathcal{Q}}, f_{\mathcal{K}},$ and $f_{\mathcal{V}}$ are linear layers that map each input to a common feature dimension SF, representing as $\mathcal{Q}$ (blue feature blocks), $\mathcal{K}$ (Tan features blocks), $\mathcal{V}$ (violet feature blocks). The output attention dimension is further projected from another linear layer f_O. After the QKV operation, a two-layer MLP with a GELUs activation function is concatenated. Finally, the output feature vector stays the same size with 128 dimensions.

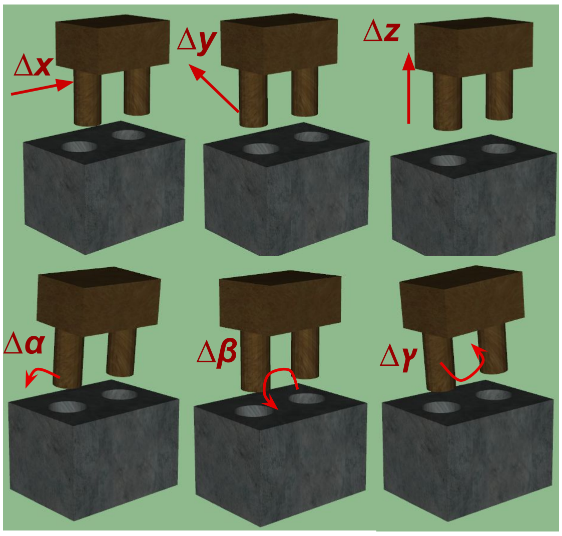


Figure 6.7: Elementary movement in 6 directions from our trained policy, consisting of moving of position, and orientation from the end-effector.

6.3 Assembly Policy Learning

6.3.1 Impedance-based Action Control

The encoded feature vector from perceived transformer is input into an MLP decoder with three hidden layers to predict the gripper and end-effector actions. Seen from Fig. 6.7, the action space in our task is 7-dimensional, consisting of moving of position, orientation from the end-effector, and the open/close state of the gripper. The action a_t is defined as the difference between the current kinematic state and the desired kinematic state:

$$a_t^{pos} = s_{t+1}^{pos} - s_t^{pos}, a^{pos} \in [\Delta x, \Delta y, \Delta z] \tag{6.8}$$

$$a_t^{ori} = s_{t+1}^{ori} * inv(s_t^{ori}), a^{ori} \in [\Delta\alpha, \Delta\beta, \Delta\gamma] \tag{6.9}$$

$$a_t^{gri} = s_{t+1}^{gri} - s_t^{gri}, a^{ori} \in [0, 1] \tag{6.10}$$

where t is the timestamp of the current action, and s^{pos}, s^{ori}, and s^{grpi} represent the scalar value for position, orientation, and gripper. inv means the inverse operation for the orientation matrix. The orientation of the action is represented by the differences in a three-dimensional axis angle. The gripper scalar value is a binary signal. When it equals 1, the gripper will open and it will close when there exists a reverse signal. This will work at the beginning and end of the task.

For the robot action control part, let us first consider a rigid robot manipulator with the following dynamics in the joint space,

$$H(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau + \tau_{ext} \tag{6.11}$$

where $q \in \mathbb{R}^{M\times 1}$, $\dot{q} \in \mathbb{R}^{M\times 1}$, $\ddot{q} \in \mathbb{R}^{M\times 1}$ are the joint angle, velocity, and acceleration vectors, respectively, and M is the number of Degree of Freedom (DoF). $H(q) \in \mathbb{R}^{M\times M}$ is the mass matrix; $C(q, \dot{q})) \in \mathbb{R}^{M\times M}$ are the Coriolis and centrifugal forces; $G(q) \in$

$\mathbb{R}^{M\times 1}$ represents the gravity term. $\tau_{ext} \in \mathbb{R}^{M\times 1}$ is the torque vector due to the external force during the interaction, and $\tau \in \mathbb{R}^{M\times 1}$ represents the torque control input vector. Here, we aim to validate the performances of the RL policy embedded with different control strategies. To do so, three commonly used torque control strategies are selected below.:

Controller 1: velocity-based joint impedance control. In this case, the output of the RL agent at each step is the desired joint velocity $\dot{q}_d$, the command control torque input τ is calculated by,

$$\tau = K_d(\dot{q}_d^q - \dot{q}) + G(q) \tag{6.12}$$

where $K_d^q \in \mathbb{R}^{M\times M}$ is a gain matrix.

Controller 2: the widely used PD-like impedance controller. The control force in the Cartesian space is computed by,

$$\begin{aligned} F_c^{pos} &= K_p^{pos}\Delta p - K_d^{pos}\dot{p} \\ F_c^{ori} &= K_p^{ori}\Delta R - K_d^{ori}\omega \end{aligned} \tag{6.13}$$

where $F_c^{pos} \in \mathbb{R}^{3\times 1}$ and $F_c^{ori} \in \mathbb{R}^{3\times 1}$ are the control force vectors for the 3-D position and 3-D orientation, respectively. Δp is the position error between the desired robot endpoint position $p_d \in \mathbb{R}^{3\times 1}$ and the current one $p \in \mathbb{R}^{3\times 1}$, i.e., $\Delta p = p_d - p$. ΔR is the orientation error between the desired orientation matrix $R_d \in \mathbb{SO}(3)$ and the current one $R \in \mathbb{SO}(3)$, i.e., $\Delta R = R_d \ominus R$, where $\ominus$ means the subtraction operation in $\mathbb{SO}(3)$. Note that the desired control variables are obtained according to the output of the RL agent at each time step, i.e., a^{pos} and a^{ori}. $\dot{p} \in \mathbb{R}^{3\times 1}$ and $\omega \in \mathbb{R}^{3\times 1}$ represent the translation velocity and orientation velocity, respectively. $K_p^* \in \mathbb{R}^{3\times 3}$ and $K_d^* \in \mathbb{R}^{3\times 3}$ are the gain matrices, and $K_d^* = \zeta\sqrt{K_p^*}$, where ζ is a positive constant.

Then, the joint torque control input τ is calculated by,

$$\tau = J^T \begin{bmatrix} F_c^{pos} \\ F_c^{ori} \end{bmatrix} + N\xi + G(q) \tag{6.14}$$

where $J \in \mathbb{R}^{6\times M}$ represents the Jacobian matrix. $N \in \mathbb{R}^{M\times M}$ is the null space operation matrix, and $\xi \in \mathbb{R}^{M\times 1}$ is a joint force vector.

Controller 3: the dynamical decoupling impedance controller (see, e.g., [192]). In this case, the joint torque control input τ is given by,

$$\tau = H\bar{J}(\ddot{x}_r - \dot{J}\dot{q}) + N\xi + G(q) \tag{6.15}$$

where $\bar{J}$ is the inertia-weighted pseudo-inverse Jacobian matrix, computed by

$$\bar{J} = H^{-1}J^T(JH^{-1}J^T)^{-1} \tag{6.16}$$

and $\dot{J}$ is the change rate of the Jacobian matrix with respect to time. $\ddot{x}_r$ is the auxiliary control variable, given by

$$\ddot{x}_r = \begin{bmatrix} \ddot{p}_r \\ \dot{\omega}_r \end{bmatrix} \tag{6.17}$$

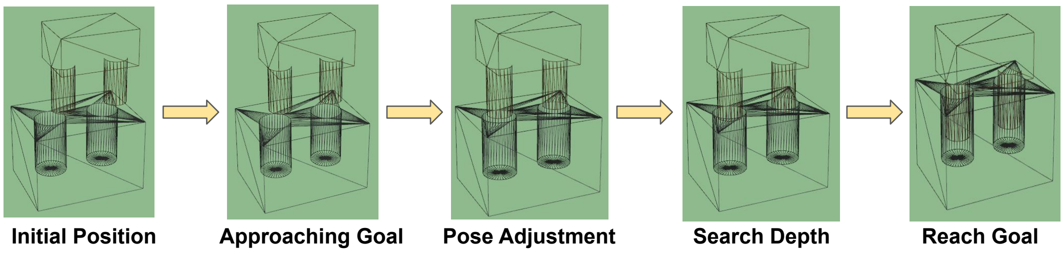


Figure 6.8: A successful policy exploration process for our multiple peg-in-hole assembly task, which consists of four transition states: from the initial position to approaching goal, from approaching goal to pose adjustment, from pose adjustment to search depth, and from search depth to reach the goal.

with

$$\begin{aligned} \ddot{p}_r &= \ddot{p}_d + K_p^{pos}(p_d - p) + K_d^{pos}(\dot{p}_d - \dot{p}) \\ \dot{w}_r &= \dot{w}_d + K_p^{ori}(R_d \ominus R) + K_d^{pos}(\omega_d - \omega) \end{aligned} \tag{6.18}$$

where $(*)_d$ represent the desired control variables, similar to that in *Controller 2*.

After obtaining action a_t during each simulation step, the impedance controller framework computes the necessary joint torques to minimize the error between the desired and the current pose according to specified impedance parameters and torque limitations. Furthermore, SAC is used to train the multi-modality impedance-based assembly policy. The hindsight experience replay (HER) is also used to improve sample efficiency and speed up the training process. To facilitate transferring the policy into the real robot experiments, the policy and Q-functions of SAC are both fed by the observations of the visual images and scalar proprioception and force/torque values, which could be directly obtained through the RGB camera and robot control system. The specific details on policy learning in the physical simulator are shown in the Experiments Section.

6.3.2 Sim2Real Dynamic Assembly

Seeing from Fig. 6.8, our task involves a set of policy exploration processes and a challenging configuration space that cannot be reached with quasistatic manipulation to execute assembly, which performs a dynamic primitive. To achieve Sim2Real transfer learning for the robotic multiple peg-in-hole assembly task, two methods are used to prompt the dynamic learning process: 1) an impedance-based controller is embedded for the policy learning 2) domain randomization is facilitated to simulate the noise of the real environment, including background color, camera parameters, lighting condition, and robot dynamic parameters.

Sampling from the policy, the impedance-based controller will map the desired end-effector positions and orientations into real joint torque values for the real robotic manipulator, aiding the RL agent's implicit learning of the dynamic impedance parameters. Based on the cascaded structure, we couple the low-frequency policy during simulation training with an impedance-based controller working at the high frequencies necessary

Algorithm 2 Multimodal RL for dynamic assembly control

```
Initialize domain randomization parameters $M = [\mu^b, \mu^c, \mu^l, \mu^d]$
Select impedance controller parameters $K_p, T_{limit}$
Initialize policy and $Q$-function network; replay buffer $D$
repeat
    Initialize hole object randomly in a limited range
    Observe state $s_t = \mathbf{Env}(M)$, select action $a_t$
    Calculate joint torque $T_t = \mathbb{I}(s|K_p, T_{limit})$
    Execute $T_t$, observe next state and reward $s_{t+1}, r_t$
    Store $(s_t, a_t, r_t, s_{t+1}, succ)$ in $D$
    If $s_{t+1}$ is terminal, reset simulator and resample $M$
    for $j$ in range(training steps) do
        Randomly sample a batch of transitions from $D$
        Compute targets for $Q$-functions
        Update all network parameters using gradient
        Update target network
    end for
until convergence
```

for actual robot experiments. The learned multimodal policy is executed at a lower frequency because there exist real-world time constraints for camera and force/torque signal observing and extracting. For the coupling process, we use interpolation operation to achieve high-frequency reference between different policy steps. For example, the current and desired end-effector state is s_t, s_{t+1}. The low-frequency action can be expressed as $a_t = s_{t+1} - s_t$. If we interpolate K times between this time interval, then the sub-step action could be expressed as :

$$a(t, i) : \begin{cases} a_{t+1/k} = s_{t+1/k} - s_t, \\ a_{t+2/k} = s_{t+2/k} - s_{t+1/k}, \\ \qquad \ldots\ldots, \\ a_{t+i/k} = s_{t+i/k} - s_{t+(i-1)/k} \end{cases} \tag{6.19}$$

The second important element for dynamic assembly is domain randomization. Firstly for the image background, there only exist a robot and table rendered in the physics simulator while the actual robot working environment is full of various background objects and colors. This is pretty difficult to match in the simulation. Thus, we randomize the color and texture of the manipulator and the table with the holes object, which is interpolated from the adjacent colors and sampled from texture collections. As for the camera parameters, we sample some noise from the uniform distribution to the position, orientation, and field-of-view (FOV) after obtaining them through real camera calibration. The lighting condition is very important because we use a grey image as the visual modality input. Different lighting conditions between simulation and the real world can have a significant impact on visual characteristics. We incorporate Gaussian noise into the lighting position, direction, and diffusion, ensuring the range of illumination changes in simulation can cover as much as possible of the natural light and artificial lights in the real assembly scene. Finally, small random errors from robot dynamic parameters like friction, inertia, and stiffness are considered for potential changes in robot mechanics. The details of training multimodal RL for dynamic assembly control are summarized in Algorithm 2.

6.4 Simulation Experiments

In this section, we first construct the multiple peg-in-hole assembly environment in the simulation. Secondly, to evaluate the proposed architecture for learning multi-modality-driven policies via impedance-based manipulation, we conduct a set of comparison experiments. Finally, to evaluate the performance of the policies trained in the simulation matching the real world, the real experimental system is set up and a set of experiments are performed with a real robot.

6.4.1 Simulation Setup and Training

The simulation environment of the multiple peg-in-hole assembly task is implemented in MuJoCo where a model of the KUKA LWR arm and Schank WSG Gripper is included along with a white table similar to the real world. To satisfy the needs of robot assembly in different application scenarios as much as possible, the peg objects and hole objects we test are not fixed. They both can move on the table and slip along the gripper surface. Otherwise, to evaluate the generalization of the assembly task, we use four different test shapes $S_{real} = \{S_{circle}, S_{ellipse}, S_{square}, S_{triangle}\}$ (See Fig. 6.4).

Each episode consists of 60 training steps, where the controller runs in the simulator at 60 Hz while the policy runs at 10 Hz. The interpolation operation following equation (6.19) is applied to mimic the real system. An episode is terminated when the episode steps end or the peg grasped in the gripper is within the distance threshold $\delta = 2.0cm$ from its goal. Fig. 6.8 shows an ideal dynamic assembly process for the trained RL robot to finish the challenging task, which consists of position initialization, approaching goal, pose adjustment, searching depth, and reaching goal. During the approaching process, a handy-shaped reward is designed by computing the L2 distance between the key point p_i $(i = 0, 1, 2)$ in the peg to the target location p_i^* in the hole object (See Fig. 6.2). The reward is expressed as:

$$R_t = \begin{cases} 1 - \tanh((\sum_{i=0}^{n} ||p_i - p_i^*||)/3) \\ 1 + (1 - \tanh(||p_0 - p_0^*||)), \ \sum_{i=0}^{n} ||p_i - p_i^*|| < \delta_1 \end{cases} \tag{6.20}$$

where p_0 and p_1 represent the center points of two peg bases from the peg object. And p_2 represents the top center point of the top connector from the peg object. The height difference between p_2 and p_0, p_1 is $0.09m$. And $\delta_1 = 0.6$ indicates the transition between pose adjustment and searching depth in the dynamic assembly process while the pegs enter the holes. This rewards the agent for searching the depth until reaching the goal and improving the learning process.

The number of training epochs is set as 250, where 6000 simulated time steps are contained in each epoch. The object shapes are trained during the policy learning process consisting of *circle*, *ellipse*, and *square*. The replay buffer collects the states, actions, and rewards at the end of each cycle. As the state-of-the-art off-policy RL algorithm, SAC has achieved the best performance based on its high sample efficiency on many simulation-based RL tasks [185]. Inspired by the previous work [193, 194], SAC also shows great potential to solve challenging robotic manipulation tasks, especially considering a combination of proprioceptive and object-specific observations, robots, and controllers. Thus, we choose the SAC algorithm for the whole policy optimization in our robotic multiple peg-in-hole assembly task. The evaluation episodes are conducted parallel to measure the success rate during the training process. Except for the gripper action, the position and orientation actions of the end-effector output from each policy step are uniformly scaled to the range of [-0.005, 0.005]. For the object initialization, the pegs object is initialized at a fixed position, then the gripper will be closed to grasp it immediately. However, the position and orientation of the holes object on the table will change at a limited variation range $x \in [-1.5, 1.5]cm$, $\theta \in [-5°, 5°]$. Furthermore, specific hyperparameters of the MPA-SAC network and domain randomization are shown in Table. 6.1 and Table. 6.2, respectively. We also employ the method of controlling variables to fine-tune and determine each hyperparameter. Take the position parameter of the camera in domain randomization as an example, we first set the camera position perturbation as 0 after calibrating and obtaining the camera parameters in the real robot environment. Then, we capture and observe the related visualized images consisting of robot gripper and pegs/holes objects when doing our assembly task in the simulation.

Table 6.1: Selected MPA-SAC hyperparameter

Parameter	Value
original captured image	$540 \times 960 \times 3$
visual image	$64 \times 64 \times 1$
proprioception and force/toque	44
batch size	256
replay buffer size	1e6
policy network learning rate	1e-3
Q-functions learning rate	5e-4
reward scaling	0.1

Table 6.2: Selected domain randomization hyperparameter from our assembly task in the simulation

Perturbation Parameter	Value
color and texture interpolation	0.3
camera position and orientation	0.02
lighting position and diffusion	0.2
robot stiffness and friction	0.1

Table 6.3: Ablation study of assembly success rate from proposed architecture in the simulation

baselines	unfrozen encoder	perceive transformer	success rate
1	✗	✗	0.72
2	✓	✗	0.80
3	✗	✓	0.75
Ours	✓	✓	0.85

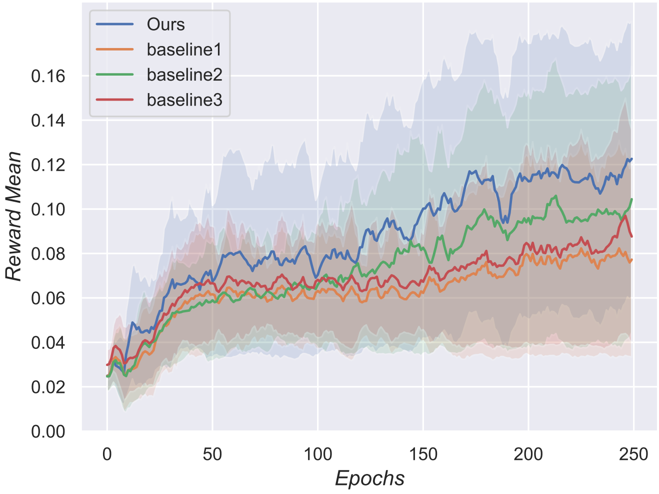


Figure 6.9: Ablation study for the proposed architecture considering the effect of the unfrozen encoder and transformer module.

After that, we will increase the value of position perturbation to 0.01 gradually and related images will be captured. We must ensure that robot gripper and pegs/holes objects are seen in the images when executing assembly. If the perturbation value is set to 0.03 and the information of robot gripper and pegs/holes objects during some simulation steps is lost, then we set the final value of position perturbation as 0.02.

6.4.2 Ablation Study

First, to evaluate the proposed architecture, ablation studies about the unfrozen encoder module from visual representation and the perceived transformer module from multimodal tokenization are analyzed. The mean rewards value and success rate during the policy training are used as different indicators, shown in Table. 6.3 and Fig. 6.9. Note that without an unfrozen encoder module means, the weights of both the encoder and decoder are all fixed during the policy training process and without perceived transformer means the concatenation of latent visual feature, proprioception, and force/torque feature is processed by full-connected layers. The ablation studies show that the proposed architecture could achieve better performance. However, the success rate of the proposed approach is not so high because we find the peg grasped by the gripper cannot be reliably fixed at the same position in the MuJoCo simulator especially when the pegs collide with the holes object, causing some failing cases during pose adjustment

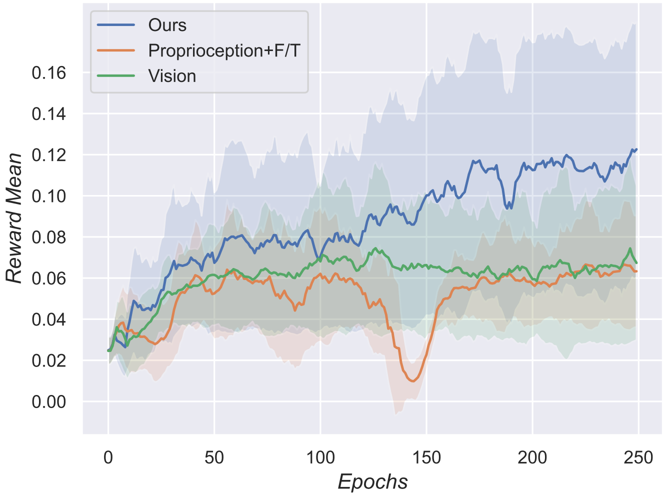


Figure 6.10: Comparison of multimodal training (Ours) and various single-modal training (Pure vision and Proprioception+F/T) runs for multiple peg-in-hole assembly experiments.

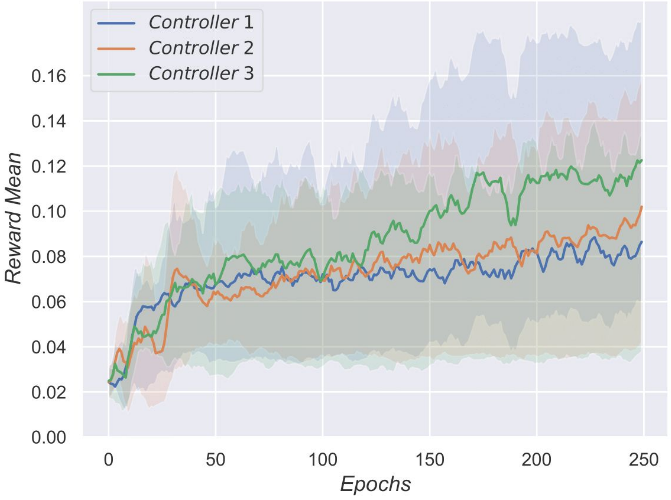


Figure 6.11: The learning curves of our RL agent for representation are trained from three different controllers described in Section 6.3.1.

and search depth periods. Moreover, compared with baseline2 without perceived transformer, the improvement of success rate from our method is only 5%. It also implies that the modality features existing in our assembly task are not so complicated. Except for the visual features, the force/torque features and proprioception features are relatively simple, causing the improvement from perceived transformer is not so obvious compared to some language-based robotic tasks [126, 190].

Furthermore, we discuss the effect of different modalities on our challenged assembly task. Unlike the previous multiple peg-in-hole assembly work [99], the position and orientation values of the holes object on the table are constant, and the pose of the peg can be implicitly represented by the pose of the end-effector. Our task introduces flexible grippers and a movable pose of the holes object, so vision is necessary to obtain the relative position between pegs and holes. Based on this, we use pure visual features and pure state information inside the robot to compare the performance of multi-modal fusion. As shown in Fig. 6.10, the training curves demonstrate that the fusion of multiple modalities can significantly improve the reward performance of this task. And the absence of vision, or force/torque modality negatively affects our assembly task.

6.4.3 Comparison Analysis of The Robot Controllers

First, we compare the performances of these three robot action controllers presented in Sec. 6.3.1. The RL reward performances under these controllers are given in Fig. 6.11. It clearly shows that *Controller 3*, *Controller 2*, and *Controller 1* obtain the best, middle, and worst performances, respectively. Specifically, during the first 100 epochs, they do not show large differences. But after 100 epochs, the mean reward under *Controller 3* increases obviously faster than the other two. And at the 250th epoch, the mean reward under *Controller 3* finally reaches 1.5 times that under *Controller 1*. This comparison result indicates that the choice of the low-level robot action controller largely affects the RL performance in a physical contact-rich task like assembly in this work. entropy regularization existing in SAC and TD3 provide a better exploration than DDPG.

Then, we discuss the effect of the different settings of the gain matrix K_p^{pos} after choosing the *Controller 3* as our impedance controller. To do so, we set $K_p^{pos} = 800I$, $K_p^{pos} = 1600I$, $K_p^{pos} = 2500I$, $K_p^{pos} = 3600I$, respectively, where $I \in \mathbb{R}^{3\times3}$ is an identity matrix. As seen in Fig. 6.12, in the first 150 episodes, as the value of K_p^{pos} increases, the upward trend of the training curve is significantly accelerated, which indicates that the increase of K_p^{pos} ensures the accuracy of performing actions to a certain extent. However, comparing the changing trend of $K_p^{pos} = 2500I$ and $K_p^{pos} = 3600I$ after 150 episodes, it can be observed that the upward trend with $K_p^{pos} = 3600I$ gradually slows down. This may imply that an excessive K_p^{pos} will increase the instability

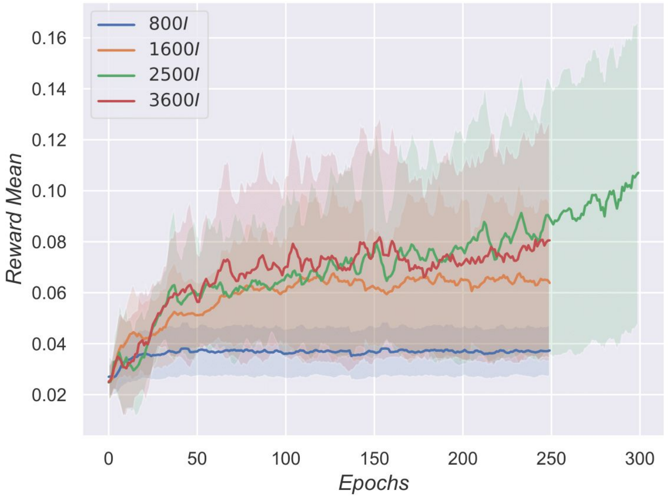


Figure 6.12: The learning curves of our RL agent for representation are trained from different gain matrices in the impedance controller.

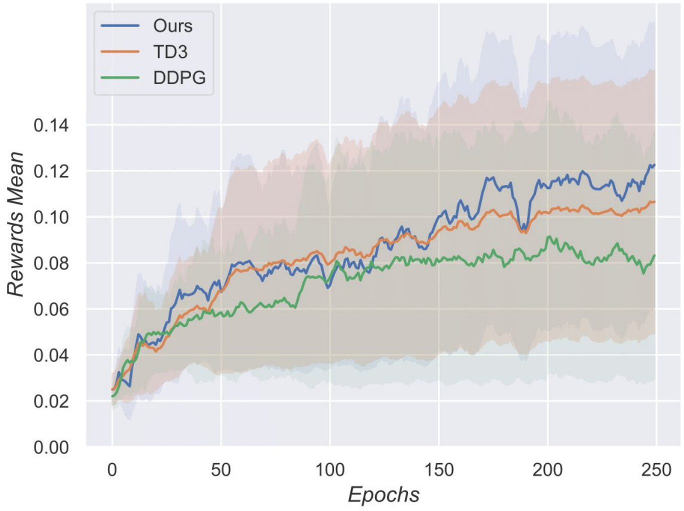


Figure 6.13: The learning curves of our RL agent for representation are trained from different off-policy RL algorithms.

of action execution, which leads to a decrease in the performance of the final learned policy. This result demonstrates that a careful choice of the parameters in the impedance controller is also crucial for the RL performance.

Furthermore, we discuss the effect of different off-policy RL algorithms for our simulated task. Since on-policy methods like PPO and TRPO are more suited to domains with real data, we decided to compare the SAC algorithm in our proposed methods with two off-policy methods (TD3 and DDPG) that are more efficient for sim2real learning. Seeing the learning curves from Fig. 6.13, it shows that the SAC can achieve the best reward performance than other algorithms in our assembly task though TD3 can also achieve a high reward. That indicates the stochastic nature and

6.5 Robot Experiments

To verify the trained MPA-SAC policy in the real world, we also set up an experimental system shown in Fig. 6.14. The 7-DOF KUKA LWR arm and Schank WSG50 gripper are connected as the robot manipulator, fixed on the table. The center position of the fixed point between the robot arm and the table can be formulated as $[x, y, z] = [0.0, 0.0, 0.8]$ in the real-world coordinate system. The F/T signals and robot states can be accessed through the KUKA robot control interface. The pegs object and holes object of different shapes are obtained through 3D printing technology, and 1.8mm clearance in all directions is measured for the printed model. The vision system we use is the Kinetic V2 and qhd mode is set to get the same height and width of the RGB image with the simulation environment. The AprilTag on the table is used to calibrate the camera and get the camera position, orientation, and FOV values. To obtain the image observation at a valid frequency and avoid the effect of latency, we use the maximum frequency setting for camera frames. It guarantees that the latest camera observation can be input

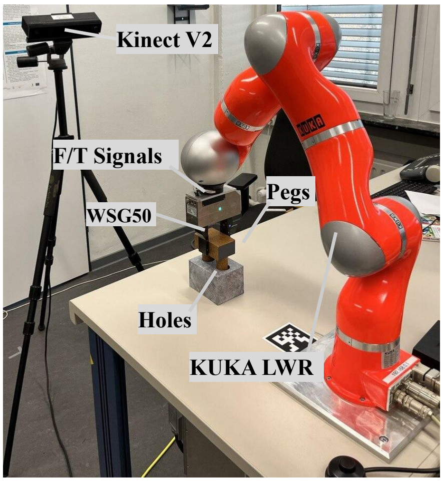


Figure 6.14: Experimental setup for robotic multiple peg-in-hole assembly in the real world, consisting of the robotic arm, gripper, vision system, and assembly objects.

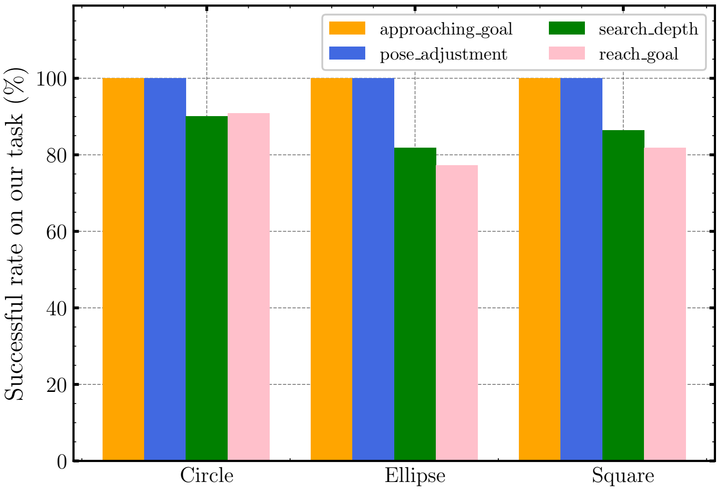
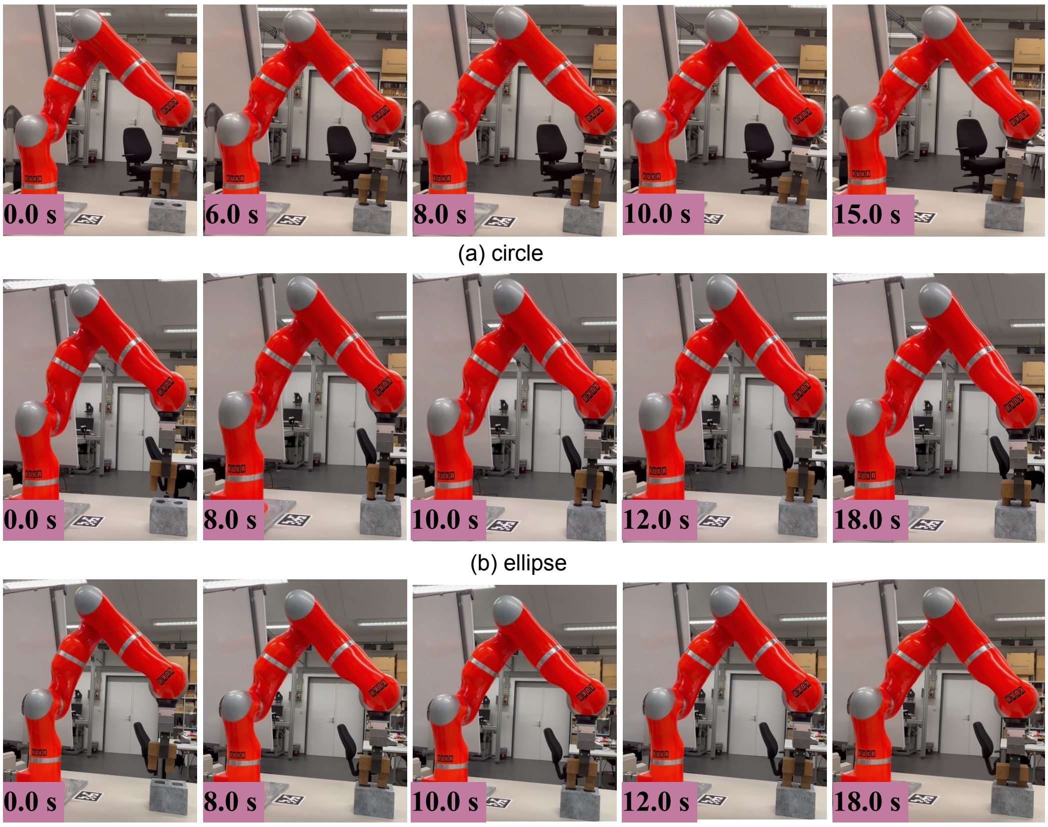


Figure 6.15: Comparison of the success rate of our task for different transition states described in Fig. 6.8 from Circle, Ellipse, and Square object shapes in the real robot experiments.

into the policy at inference time though the output policy acts at 10 Hz. The whole experimental system is operated by a ROS interface and the policy is implemented using PyTorch. Prior to running one testing experiment, the pegs object is put in the center of the gripper, and the arm with the gripper is initialized to the same joint angle $qpos$ = [0.0, 0.445, 0.0, -1.867, 0.0, 0.830, 0.0, -0.001, 0.001] for each episode, where the last two values represent the gripper being closed to grasp the pegs object. In addition, the lighting conditions and camera configurations are all kept fixed when doing the testing experiments.

By randomly adjusting the position and orientation of the holes object in a limited range, we run 22 evaluations for each object. Firstly, in order to compare against the transition states described in Fig. 6.8, we evaluate the success rate from different object shapes in the real robot experiments. Looking at Fig. 6.15, our solution can achieve this



(a) circle

(b) ellipse

(c) square

Figure 6.16: Example of the proposed solution on multiple peg-in-hole assembly for different trained object shapes (a) circle (b) ellipse (c) square. Take the circle shape as an example, the robot gripper first grasps the peg and approaches the goal hole (∼6.0s), adjusts its 6-DOF pose to prepare to put it into the hole (∼8.0s), searches the depth when moving along the hole (∼10.0s), and finally reaches the goal (∼15.0s).

challenging task at a high success rate. Especially, for the circle object, the final trained policy could achieve a success rate of over 90%. Otherwise, the transition period from pose adjustment to search depth is the main period causing failure cases. Referring to the single peg-in-hole work from [80], we guess that there exists an inevitable gap between reality and simulation, especially on the visual level, which poses significant challenges to high-precision and high-stability assembly. Fig. 6.16 shows the real-world example of the proposed solution on multiple peg-in-hole assembly for different object shapes. It can be seen that it takes quite a short time to adjust the 6-DOF posture of the pegs and then insert them into the holes, which indicates the policy needs to perceive little changes in the high-dimensional representation features of the entire system. In addition, we also observe that the assembly performance will be affected by the initial position and pose of the object on the table though we have considered a limited variation range of them in the policy training process. We found that a large orientation change will affect the assembly performance significantly while the effect of position is pretty small, demonstrating the learning of 6-DOF pose during the dynamic assembly process is still pretty challenging.

Table 6.4: The comparison of average success rate for different environmental perturbations in the real robot experiments

	Lighting condition (%)			Camera position (%)		
	Low	Normal	High	X(±0.01)	Y(±0.01)	Z(±0.01)
Circle	68.2	90.9	90.9	95.4	90.9	90.9
Ellipse	63.4	77.3	81.8	86.4	77.3	81.8
Square	68.2	81.8	90.9	86.4	81.8	81.8

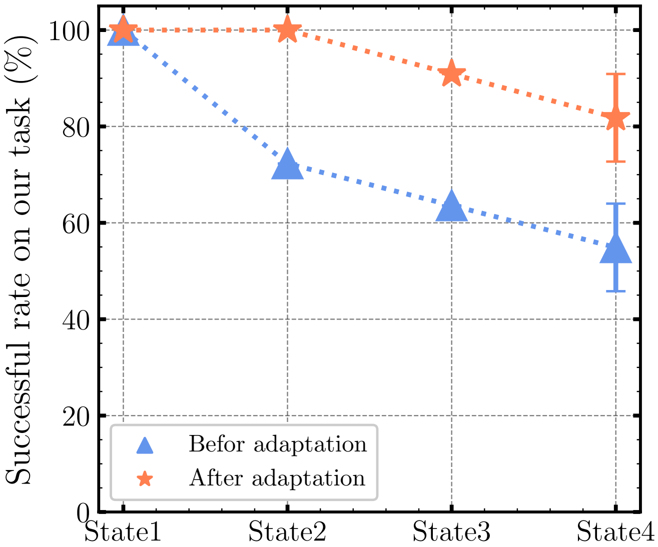


Figure 6.17: Comparison of the success rate of different transition states from triangle object in the real assembly experiments. After/Before adaption refers to whether the triangle object is added to the policy training or not.

When we train the policy in the simulation, the strategy of domain randomization is adopted to increase the robustness of the learned policy for different perturbations. To discuss the effect of perturbations on real robot experiments, we select two important environmental factors: lighting condition and camera position. As the lighting condition, it consists of low, normal, and high modes, where low lighting means natural light in our laboratory room, normal lighting means we will turn on the fluorescent lights above the robotic manipulator, and high lighting means all fluorescent lights in our laboratory room are turned on. As for the camera position, we add a noise value to the position value from different axes as we set the perturbation parameter as 0.01 in the simulation. The experimental results are shown in Table. 6.4. For the camera position, it can be seen that the perturbations have little effect on the average success rate because of the same perturbation setting between the simulation and the real world. For the lighting conditions, the assembly performances are both great for the normal and high lighting conditions because we also set two lights above the robot manipulator in the simulation. However, for low lighting conditions, the performance suffers a significant decline. We guess it is

Table 6.5: The comparison of final success rate and assembly time for different object shapes in the real robot experiments

Object shape	Circle	Ellipse	Square	Triangle*	Triangle
Time (s)	15.0	18.0	18.0	24.2	17.6
Success rate % avg (std)	90.9 (9.09)	77.3 (13.6)	81.8 (9.09)	54.5 (9.09)	81.8 (9.09)

Note: the untrained object shape is represented with *.

Table 6.6: The comparison of final success rate considering the effect of domain randomization and impedance controller in the real robotic experiments

Object shape	Circle	Ellipse	Square	Triangle*	Triangle
w/o domain randomization % avg (std)	13.6 (9.09)	9.09 (9.09)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
w/o impedance controller % avg (std)	63.6 (13.6)	54.5 (9.09)	50.0 (9.09)	22.7 (9.09)	45.5 (13.6)

Note: the untrained object shape is represented with *.

caused by the fewer low-light images existing on our collected image dataset and the trained visual representation module can not well perceive the low-lighting condition.

To validate the generalization ability for the trained policy, we chose the triangle shape as the test object since it's not used in the policy training. As seen from Fig. 6.17 and Table. 6.5, the trained policy from the circle, ellipse, and square could be generalized to the new triangle shape. The mean assembly time of each object shape is also computed to justify the relationship between task difficulty and assembly performance. We find that the assembly time of a visual circle shape from pegs/holes is shorter than the ellipse and square shapes while the triangle shape takes the longest time duration. It can be explained that a smaller visual perception field from a triangle shape makes it more difficult to learn a good visual representation. Finally, after adapting to the simulation environment again, the test on the triangle object shape achieves a significant improvement of over 25% in the success rate and a reduction of over 6.5s in the assembly time, which demonstrates that our model has a good generalization ability over uncertainties and object shapes.

As seen in Table. 6.6, we further retrain the policy without adding domain randomization for all kinds of object shapes. We found that the success rate for circle, ellipse, square, and triangle shapes are 13.6%, 9.09%, 0, and 0, respectively. And the final trained model cannot be generalized to a new object shape. When we randomize our simulation domain, we consider all kinds of permutations from camera setting, environment setting, lighting setting, etc., which exist indeed in real robotic experiments. Especially for the lighting conditions, domain randomization can satisfy all kinds of lighting conditions in our laboratory. Without it, the visual representation learned from the simulation is significantly different from the real environment, which causes the visual perception to fail in our assembly task. Furthermore, we retrain the policy without adding impedance control and introduce basic position control to execute the output action. It can be seen that the success rates for all object shapes are significantly lower than our method, demonstrating the embedding of impedance skills is necessary to improve the assembly performance.

6.6 Discussion and Summary

We present a solution for the multiple peg-in-hole assembly (MPA) task using a multimodal representation by transferring the trained policy in simulation to the real world without extra exploration. A special visual representation module and tokenization-based transformer module are separately proposed to compact the feature as the backbone of reinforcement learning. Furthermore, our policy learning process incorporates domain randomization and an impedance controller, which expedites the transfer process and narrows the gap between simulation and reality. We extensively evaluate the performance of our solution in a simulation environment, demonstrating a high success rate in a more challenging multiple peg-in-hole assembly setup. Moreover, our solution's generalization ability is validated across different object shapes.

This chapter sets the number of pegs and holes as two, with the experimental objects comprising four different shapes. Moving forward, it is important to extend our research to include more complex object interactions, accounting for factors such as the number, size, and shape of the objects. Additionally, the current input modality features in our assembly task are relatively simple, leading to less noticeable improvements in performance from our perceived transformer fusion module. To address this, we aim to incorporate additional modalities such as point cloud, tactile feedback, and language to enhance and diversify our work. Furthermore, certain limitations in our proposed method need to be addressed. Currently, some parameters related to domain randomization and impedance control are manually fine-tuned, and it would be beneficial to introduce optimization algorithms or learning-based approaches to optimize these parameters. We firmly believe that further research on robotic multiple peg-in-hole assembly, specifically through the application of multimodal reinforcement learning, can significantly enhance the efficiency of related manufacturing processes.

Chapter 7

Language-conditioned Diffusion Learning for Robotic Rearrangement

In the preceding chapters, we addressed robotic manipulation challenges using both passive and interactive perception. However, how can we enable robots to think and act more like humans, particularly when faced with novel task conditions? In this chapter, we explore potential solutions through diffusion learning and vision-language models to achieve intensive imagination [1]. The capability for robotic systems to rearrange objects based on human instructions represents a critical step towards realizing embodied intelligence. Recently, diffusion-based learning has shown significant advancements in the field of data generation while prompt-based learning has proven effective in formulating robot manipulation strategies. However, prior solutions for robotic rearrangement have overlooked the significance of integrating human preferences and optimizing for rearrangement efficiency. Additionally, traditional prompt-based approaches struggle with complex, semantically meaningful rearrangement tasks without pre-defined target states for objects. To address these challenges, this chapter first introduces a comprehensive 2D tabletop rearrangement dataset, utilizing a physical simulator to capture inter-object relationships and semantic configurations. Then we present DreamArrangement, a novel language-conditioned object rearrangement scheme, consisting of two primary processes: employing a transformer-based multi-modal denoising diffusion model to envisage the desired arrangement of objects, and leveraging a vision-language foundational model to derive actionable policies from text, alongside initial and target visual information. In particular, we introduce an efficiency-oriented learning strategy to minimize the average motion distance of objects. Given few-shot instruction examples, the learned policy from our synthetic dataset can be transferred to the real world without extra human intervention. Extensive simulations validate DreamArrangement's superior rearrangement quality and efficiency. Moreover, real-world robotic experiments confirm that our method can adeptly execute a range of challenging, language-conditioned, and long-horizon tasks with a singular model. The demonstration video can be found at https://youtu.be/fq25-DjrbQE.

[1] This chapter has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

7.1 Introduction

From the perspective of embodied intelligence, how can we empower the household robots with the capability to discern *how and where they should rearrange messy tabletop objects* especially involving ambiguous human instructions? Comprehensive reasoning and planning across diverse constraints from object geometry, language-conditioned tasks, collision physics, and human preference, pose a significant challenge for autonomous robots operating within varied and unstructured household scenarios, such as automated packaging and sorting in warehouses, kitchen cleaning, and complex assembly tasks in manufacturing. In this work, we study this challenge by introducing human-like imagination and planning ability to the robots in the context of human instructions and prior observations.

Robotic rearrangement can be defined as a canonical task: given a previously unseen environment, the robot needs to rearrange each object into an appropriate pose to form a specified structure following human preference. This paradigm can also encompass a diverse array of activities, such as making a bed, ironing clothes, and cleaning a room. However, we specifically concentrate on investigating tabletop object arrangements, considering this challenging but tractable [195]. Recently, some approaches that leverage large language models (LLMs) have demonstrated a strong generalization for robots to understand complex semantic contexts and generate long-horizon planning for tabletop arrangement task [112, 196, 197]. However, the goal states of different objects still need to be manually specified in the prompt instructions. Furthermore, to estimate the target states of objects intelligently, some generative work based on VAE [198] and diffusion models [123, 127] has been proposed to endow the robot with human-like imagination, hopefully generating and refining the distribution of object poses. For instance, [123] proposes to utilize DALL-E, a web-scale artificial intelligence-generated content (AIGC) model, to generate a target image that implicitly incorporates various objects the robot observes. Nevertheless, the exclusive reliance on textual input for image generation has proven to be notably unstable and inefficient in real-world robot manipulation, primarily due to the neglect of crucial observational cues. Inspired by this prior work, building a model that conducts observation reasoning first and then imagines goal states intuitively via language is a crucial step towards autonomous robotic rearrangement.

On the other hand, considering functional and stylistic inter-object relationships emerges as a critical dimension for real-world robotic rearrangement [199]. For a given “messy” scenario, a “clean” arrangement should not be deterministic because there exists a plurality of desirable layouts from different human preferences. Thus, beyond the initial phase of estimating the goal poses of objects, the subsequent phase involves reorganizing the global layout of rearranged objects to align with human preferences, often communicated through language instructions. Moreover, to improve the real-world rearrangement efficiency, we also need to reduce the duration cost of the long-horizon manipulation by considering the motion distance of each object as much as possible. Despite notable advancements in learning-based scene synthesis and robotic rearrangement methods [124, 127, 199, 200], there remains a challenge to meet diverse desiderata in real human-robot cooperation environments.

In this chapter, we design a novel robotic arrangement scheme to solve the aforementioned flaws and maximize versatility and adaptivity, where the robot can rearrange objects in different goal poses and structures via language instructions without extra manual intervention. Specifically, we first construct a kitchen-based tabletop arrangement dataset consisting of four different global structures - *horizontal, vertical, circle, and containing*, and two local regularities - *symmetry* and *uniform*, where 22-class objects with different shape scales and texture materials are selected. Given that the input is a messy scene with human language instructions, we propose a transformer-based multi-modal denoising diffusion framework to estimate the goal states of objects by implicitly reasoning multi-object semantic relations.

Furthermore, we treat the planning problem of robotic rearrangement as a long-horizon estimation task by utilizing a frozen vision-language model (VLM) like GPT-4 to bridge connections between language text, visual perception, and robotic action. When prompted with several examples followed by the corresponding rearrangement policy, VLM planners can take in new language instructions and semantic contexts from initial "messy" scenes and predicted "clean" scenes, autonomously generating a new robotic arrangement policy. An example of the whole human-robot arrangement process of the *containing* structure can be visualized in Fig. 7.1. It describes a task scene in which a household robot needs to place all objects on the table into a container like a plate or box without causing objects collision and penetration. Finally, the proposed scheme is evaluated in both simulation and real robot experiments and compared with several state-of-the-art baselines, demonstrating that it can achieve better rearrangement quality and efficiency for different structure-based rearrangement tasks. The primary contributions of this chapter are described as follows:

1) Considering the differentiated requirements of inter-object relationships and human preference in the robotic rearrangement task, we construct a 2D kitchen rearrangement dataset consisting of a variety of household object scenes with different global structures and local regularities.

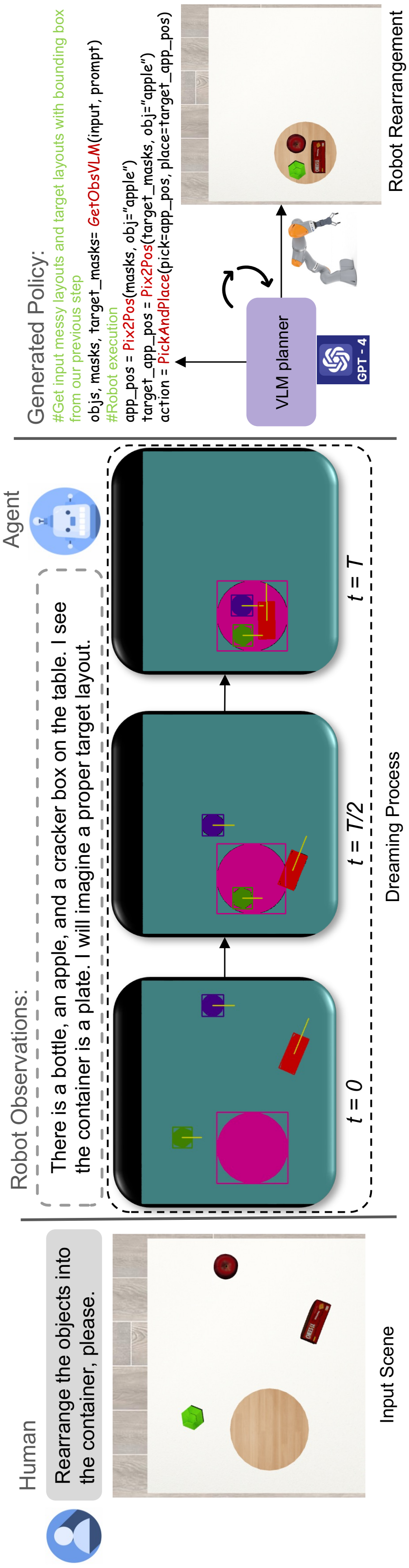
2) To generate a high-quality rearranged scene, we propose a transformer-based multi-modal denoising diffusion model, which can effectively reason semantic and geometric relations from diverse objects, and explicitly predict the goal states of objects instructed by contextualized language representation.

3) To obtain the optimal layout in the real world, we propose an efficiency-oriented rearrangement learning strategy, which pursues a minimal average motion distance of objects.

4) Inspired by prompt-based learning, we integrate the generative model with VLMs to formulate a VLM planner, which outputs robot action policies in different arrangement tasks and can be directly deployed into a physical robot.

7.2 Problem Formulation

We introduce DreamArrangemnt, a novel robotic arrangement scheme designed to comprehend diverse human language instructions and the distribution of 2D object scenes


Human
Rearrange the objects into the container, please.
Input Scene
Robot Observations:
There is a bottle, an apple, and a cracker box on the table. I see the container is a plate. I will imagine a proper target layout.
Agent
$t = 0$
$t = T/2$
$t = T$
Dreaming Process
Generated Policy:
VLM planner
GPT - 4
Robot Rearrangement

```
#Get input messy layouts and target layouts with bounding box
from our previous step
objs, masks, target_masks= GetObsVLM(input, prompt)
#Robot execution
app_pos = Pix2Pos(masks, obj="apple")
target_app_pos = Pix2Pos(target_masks, obj="apple")
action = PickAndPlace(pick=app_pos, place=target_app_pos)
```

Figure 7.1: Overview of the proposed scheme on the robotic rearrangement task when the multi-object structure setting is *containing*.

including variations in attributes like semantic classes, geometric shapes, and placements of multiple objects, which shows the ability to perform a long-horizon manipulation task autonomously.

We consider the initial tabletop scenes where all objects are scattered in an image coordinate system, starting from the top left corner as the origin. In each messy scene S, we depict a combination of of a table T and objects $\{o_1, ..., o_N\}$. To achieve semantic rearrangement based on human preference, a structure-based language instruction $\mathcal{L}$ (e.g., "rearrange all objects into a circle shape") is also given. To enhance contextual understanding, we further employ approaches from text summarizing (e.g., prompt-based LLM parsing or search-based word dictionary) to decompose the abstract language into specified word tokens $\mathcal{L} \to (l_1, l_2, ..., l_n)$. This study primarily explores the challenge of generating a language-conditioned clean object scene S^* for a robot **r**. S^* can be directly used in the planning phase as a visual prompt module in the VLM planner, finally generating a manipulation policy $\mathcal{P}$. We formulate this as an optimization problem to use the robot **r** to rearrange a "messy" scene S under a language instruction $\mathcal{L}$ via learning a bijection f of paired objects and minimizing their motion distance, referring to the ground truth "clean" scene $\tilde{S}$:

$$\begin{aligned} f^* = \underset{f}{\operatorname{argmin}} \quad & \mathcal{F}_{arrangement}(S, \mathcal{L}) + \lambda \mathcal{F}_{motion}(S, \mathcal{L}), \\ s.t. \quad & \mathcal{F}_{arrangement}(S, \mathcal{L}) = f(S, \mathcal{L}) - \tilde{S}, \\ & \mathcal{F}_{motion}(S, \mathcal{L}) = f(S, \mathcal{L}) - S, \end{aligned} \tag{7.1}$$

where λ is the weight hyperparameter of the $\mathcal{F}_{motion}(S, \mathcal{L})$ term. Then the policy can be expressed as:

$$\mathcal{P} = VLM(S, f^*(S, \mathcal{L}), \mathcal{L}), \tag{7.2}$$

More specifically, each object o in the input scene S is defined by its semantic class $c \in \mathbb{R}^C$, 2D oriented bounding box size $s \in \mathbb{R}^2$, object translation $t \in \mathbb{R}^2$, and object rotation $r \in SO(2)$[2], respectively. Since the *containing* structure is a special semantic scene, we define an additional 'mask' object class m to represent *containers* like plates and tables. Besides, we use type $tp \in \mathbb{R}^T$ instead of c to differentiate different containers. In summary, we denote each scene S as follows:

$$S = \{m_i, ..., o_i, ...\}, m_i = (t_i, r_i, s_i, tp_i), o_i = (t_i, r_i, s_i, c_i). \tag{7.3}$$

The object semantic class label c_i and container type label tp_i are represented as one-hot vectors of C and T classes, respectively, and the 2D bounding box size s_i is obtained by performing the principal component analysis (PCA) and then computing the positional relation of 4 corners. The values of translation t_i and rotation r_i are characterized by calculating the center position and the orientation angle of the bounding box. To facilitate a stable training process, we further use the normalization operation to transform t_i and s_i into the same range of $[-1, 1]$ as r_i.

[2] The first column of the rotation matrix is used.

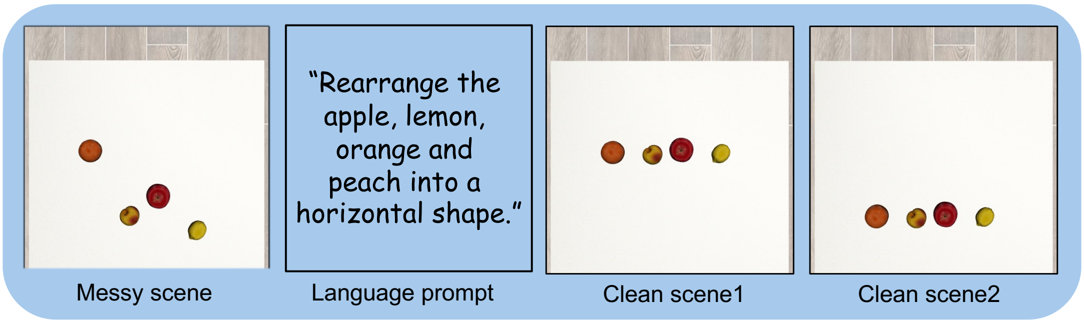

"Rearrange the apple, lemon, orange and peach into a horizontal shape."
Messy scene
Language prompt
Clean scene1
Clean scene2

Figure 7.2: Comparison of different generation results for clean scenes based on the same messy scene and language prompt.

7.3 Physics-based Object Rearrangement Dataset

To facilitate tabletop robotic rearrangement, it's necessary to collect a large object rearrangement dataset which includes different object categories and spatial structures. However, collecting such a dataset involving complex physical interactions in the real world can be time-consuming, labor-intensive, and costly. In this work, we collect a 2D synthetic dataset based on the Mujoco physics simulator, consisting of 2223 clean object scenes. A physics simulator can help us precisely control the position, orientation, scale, and texture of each object and keep each object in the rearrangement scene collision-free and penetration-free. Additionally, it is convenient for us to describe each clean rearrangement structure with high-level language instruction. Specifically, we adopt 22 household object models from the YCB objects and ShapeNet objects as our object database. For each valid clean scene in the dataset, we preprocess it using instance segmentation and extract the oriented bounding box of each object as its explicit representation. The obtained scene $\tilde{S}$ will be regarded as our target output of the generation model. To simulate different messy scenes in our daily lives, we further perturb the target scene on the fly to generate clean-messy pairs and re-associate objects within each category, where the generated messy scenes will serve as the input data.

More importantly, high-level language instruction corresponding to a structure usually conveys different object layouts in the real world. As seen in Fig. 7.2, when we tell the household robot: "Based on the current messy scene, please rearrange the apple, lemon, orange, and peach into a horizontal shape", the mainstream solutions [123, 127, 198] will make the robot arrange objects into a centred layout as in clean scene1, which is a common pattern in their training data. However, according to our human experience, we prefer to arrange the unordered objects into clean scene2, because it can save a large amount of time and effort. Therefore, to avoid the drawback in previous works that diverse initial scenes are arranged into the identical layout given the same instruction, we adopt the technique of data augmentation to enrich layout variations of target scenes with the same structure in our dataset.

Another rearrangement setting is that we want to arrange the same configuration on the table into different structures given different language prompts. We further design four kinds of physically meaningful spatial structures to pair with text descriptions.

Table 7.1: Object attributes and spatial structures in our dataset

Entity	Type	Name
Object attributes	class (22)	apple, bear, banana, bowl, box, can, cracker_box, cup, fork, knife, lemon, milk...
	material (3)	YCB texture, metal, wood
	scale ratio (3)	0.8, 1.0, 1.2
Spatial structures	global structure	horizontal, vertical, circle, containing
	local regularity	symmetry, uniformity

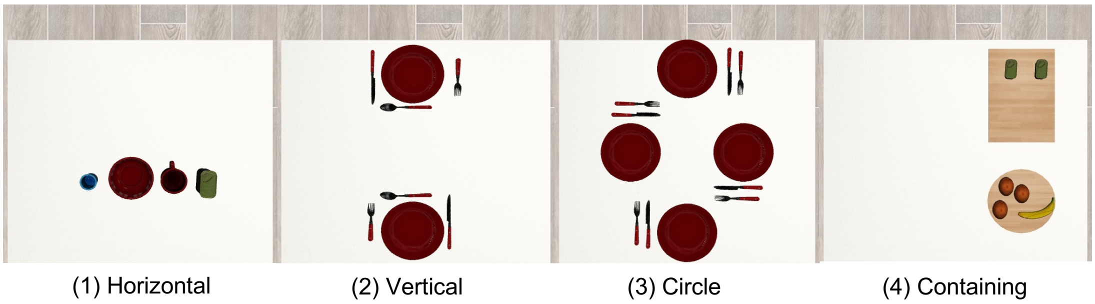

(1) Horizontal (2) Vertical (3) Circle (4) Containing

Figure 7.3: The four kinds of global structures in our dataset: horizontal, vertical, circle, and containing.

As shown in Fig. 7.3, the structures of *horizontal*, *vertical*, and *circle* represent all objects forming a horizontal, vertical, and circle shape globally, respectively. Since the *containing* structure involves the additional 'mask' object class m_i, we describe it as placing different objects in different containers, including plate-like containers and box-like containers. The semantic and geometric parameters of these containers will also be employed in the *containing* rearrangement task.

Moreover, to distinguish the difference of local distribution in real-world table settings, we introduce the concept of *symmetry* and *uniformity* in language instruction. Taking forks, knives, and plates as an example, *symmetry* represents that a pair of knives and forks are placed on varied sides of the plate while *uniformity* denotes that knives and forks are positioned on the same side of the plate. Finally, all object attributes and spatial structures in our dataset are shown in Tab. 7.1.

7.4 Rearrangement Diffusion Framework

7.4.1 Score-based Denoising Diffusion Model

Denoising Diffusion Models [114, 201] are a class of generative models that learn data distribution by progressively denoising from a tractable noise distribution. Below, we provide a brief preliminary introduction from a score-based perspective. For more details, please refer to [201]. Given various samples from an unknown data distribution $q_0(x)$, our goal is to train a model capable of generating new samples that mimic the original distribution $q_0\left(x\right)$. A critical mechanism employed in this endeavor is Langevin dynamics, a concept borrowed from the domain of physics. This approach can produce samples from a distribution $p_{data}(x)$ when its *score*, defined as its gradient $\nabla_x \log p_{data}(x)$, is known. Starting from x_T of any prior distribution, the Langevin method recursively denoises the data as follows:

$$x_{t-1} = x_t + \alpha_t \nabla_{x_t} \log q_0\left(x_t\right) + \beta_t \epsilon, \tag{7.4}$$

where α_t and β_t are pre-defined step sizes associated with the time step t and $\epsilon \sim \mathcal{N}(0, I)$ is a stochastic term. As T becomes sufficiently large, the final obtained x_0 will converge to a sample drawn from $q_0(x)$.

We aim to train a neural network s_θ to approximate the *score* of the target distribution. The denoising score matching technique [202] is adopted to make the estimation of *score* tractable, with the key insight being to utilize conditional distribution settings. This involves perturbing $x_0 \sim q_0(x)$ with various noise kernels $q_t\left(x_t|x_0\right)$ across a spectrum of step parameters $t \sim \mathcal{U}[1, T]$. The original score matching objective of the perturbed distribution $q_t(x)$ can be expressed as $\mathbb{E}_{q_t(x_t|x_0)q_0(x_0)} \left\|s_\theta(x_t) - \nabla_{x_t} \log q_t\left(x_t|x_0\right)\right\|^2$. As demonstrated in [201], the final optimal network parameter θ^* for this objective should ensure $s_{\theta^*}(x) \approx \nabla_x \log q_t\left(x\right)$. Moreover, when employing Gaussian kernels $q_t\left(x_t|x_0\right) = \mathcal{N}(x_0, \sigma_t^2)$ with pre-defined noise levels σ_t, the *score* of the conditional probability density can be analytically derived as $\nabla_{x_t} \log q_t\left(x_t|x_0\right) = \frac{x_0 - x_t}{\sigma_t^2}$. Consequently, the unified objective amalgamating all procedural steps is formulated as:

$$L_{score}(\theta) = \mathbb{E}_{t \sim \mathcal{U}[1,T], q_t(x_t|x_0)q_0(x_0)} \lambda_t \left\| s_\theta(x_t) - \frac{x_0 - x_t}{\sigma_t^2} \right\|^2, \tag{7.5}$$

where λ_t denotes the objective weight, pragmatically set to σ_t^2.

In summary, we need to first optimize the *score* network s_θ to minimize objective Eq. 7.5. After that, we use the trained model $s_{\theta^*}(x_t)$ to incrementally refine the approximation of $\nabla_{x_t} \log q_0\left(x_t\right)$ as per the Langevin dynamics, facilitating an update along the Markov chain with Eq. 7.4 to generate new samples ultimately.

7.4.2 Vision and Language Parsing

Based on our collected synthetic dataset, we further propose a scheme for solving the tabletop rearrangement task as shown in Fig. 7.4. A messy scene S and a high-level text description $\mathcal{L}$ from human language are taken as the input.

Object Detection and Parameterization: In order to obtain the geometric and semantic attributes of all objects, including o_i and m_i, in the messy scene S, we first adopt the latest Grounded Segment Anything Model (GSAM) [203], which has shown a strong zero-shot ability for object recognition and instance segmentation. Then the segmented results are subjected to Principle Component Analysis (PCA) to get the oriented bounding box of each object. Specifically, the values of object translation t_i and rotation r_i are derived by averaging all pixel points p_m of the object and establishing the covariance matrix:

$$t_i = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} p_m, \tag{7.6}$$

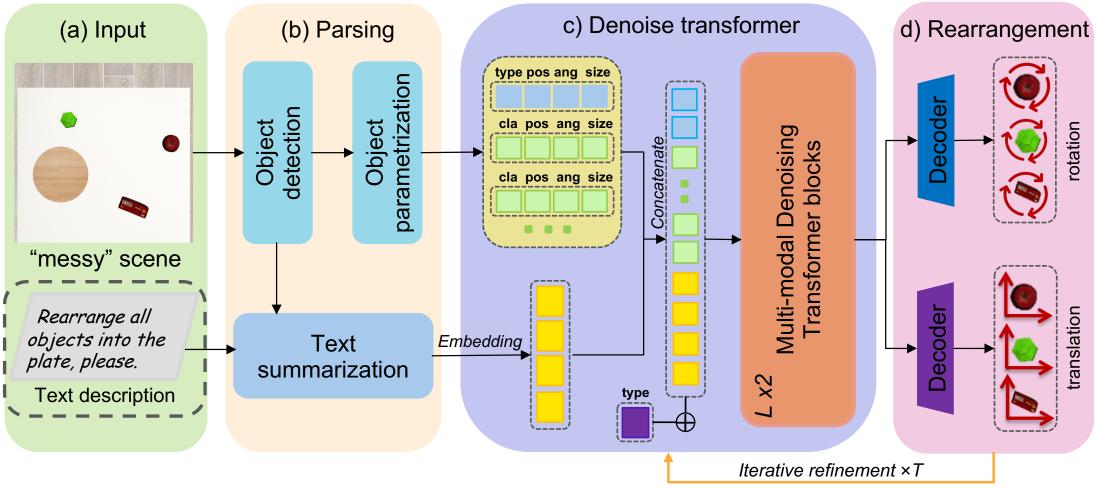
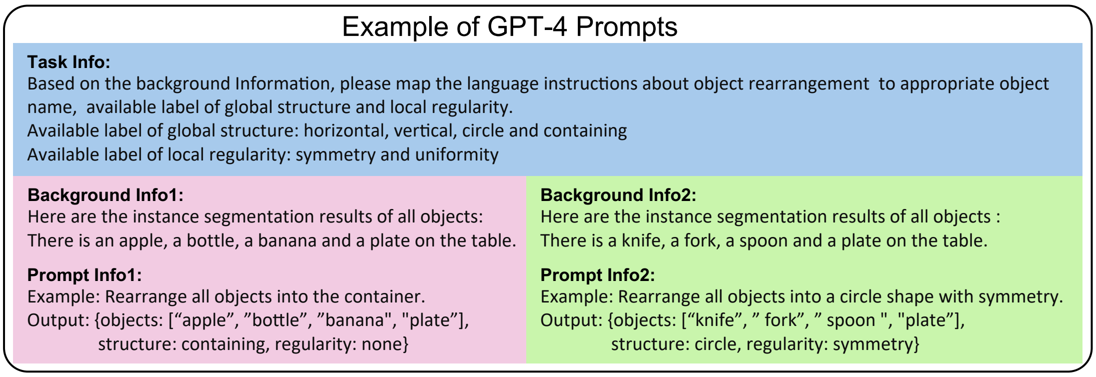


Figure 7.4: An overview of the proposed conditional rearrangement diffusion network. (a) We sample the combinations of text descriptions from the humans with the messy observation as input. (b) The parsing process is to obtain explicit object attributes and word tokens from input. (c) We build a denoising diffusion framework with transformer architecture that separately encodes object attributes and word tokens into latent space. (d) To achieve the rearrangement task, the direction of translation and rotation of each instance are iteratively refined during the limited denoising steps T.



Example of GPT-4 Prompts

Task Info:
Based on the background Information, please map the language instructions about object rearrangement to appropriate object name, available label of global structure and local regularity.
Available label of global structure: horizontal, vertical, circle and containing
Available label of local regularity: symmetry and uniformity

Background Info1:
Here are the instance segmentation results of all objects:
There is an apple, a bottle, a banana and a plate on the table.

Prompt Info1:
Example: Rearrange all objects into the container.
Output: {objects: [“apple”, ”bottle”, ”banana", "plate”],
structure: containing, regularity: none}

Background Info2:
Here are the instance segmentation results of all objects :
There is a knife, a fork, a spoon and a plate on the table.

Prompt Info2:
Example: Rearrange all objects into a circle shape with symmetry.
Output: {objects: [“knife”, ” fork”, ” spoon ", "plate”],
structure: circle, regularity: symmetry}

Figure 7.5: Example of GPT-4 prompts that map human language instruction and segmentation results from vision parsing to concise word tokens.

$$r_i = \underset{v,\ \|v\|_2=1}{\operatorname{argmax}} \quad v^T \left[\frac{1}{\mathcal{M}-1} \sum_{m=1}^{\mathcal{M}} (p_m - t_i)(p_m - t_i)^T \right] v, \tag{7.7}$$

where $\mathcal{M}$ is the number of pixel points in the segmented object and the vector v indicates the projection direction to be searched for. Next, we employ separate neural network layers to encode the geometric and categorical features to obtain the instance embedding for the regular object o_i and the mask embedding for the container object m_i.

Text Summarization: To encode the natural language instruction into implicit representation, we need to distill the most important information and convert it into a condensed form. In this work, we adopt the concept of text summarization to capture the key essence from the text description and visual clues and then stitch them together. For most language-conditioned robotic works [103, 124], they generally need to retrain an extra language model based on a pre-trained CLIP or MiniLM model on their self-deigned task-oriented sentence dataset to achieve text summarization.

To enhance the efficiency and multimodal adaptability of the summarization process, we use prompt-based learning via GPT-4 to achieve contextual understanding and generate word tokens. As the most advanced language model, GPT-4 has a vast knowledge base and linguistic proficiency, allowing it to produce the concise summaries that humans want. An example of prompts in Fig. 7.5 shows that GPT-4 can learn to produce outputs tailored to our specific mapping tasks by providing prompts that are representative of the summarization task. To enable word embedding, we further use the strategy of label encoding to assign a unique integer to each class of labels in the generated word tokens.

7.4.3 Conditional Rearrangement Diffusion Network

The architecture of the proposed conditional rearrangement diffusion network is illustrated in Fig. 7.4. The transformer structure is employed as it is adept at fusing the information from different modalities. We first encode various scene object attributes and parsed word tokens into latent tokens, which are then processed by the multi-modal transformer. The network outputs translation and rotation predictions for each instance, and a diffusion scheme is adopted to successively refine the pose of each object. Below, we elaborate on each component of our network.

Token Encoder: The input tokens of the transformer include word embedding, mask embedding, and instance embedding. The word embedding represents the language instruction used to specify the target configuration. We map the parsed global structure and local regularity types to learned embeddings, which is conducive to identifying the commonalities of instructions faster during training compared to encoding the whole sentence with language models. Next, as defined in Section. 7.2, the attributes of container objects and regular objects contain continuous variables such as translation, rotation, and size, as well as discrete variables like type and class. Similar to [199], we employ positional encodings of certain frequencies and subsequent linear layers to convert t_i, r_i, and s_i into vectors. As for discrete properties, a multilayer perceptron (MLP) is adopted to map one-hot vectors to high-dimensional latent. The above features are concatenated and then processed through an MLP to form the mask embedding and the instance embedding. This object-centric representation encodes each object separately, and 2 sets of specific MLPs are applied for mask and instance, respectively. Furthermore, a learned type embedding $T_{\mathcal{T}}$, which is utilized to distinguish different types of tokens ($\mathcal{W}$ord, $\mathcal{M}$ask, and $\mathcal{I}$nstance), is concatenated to the aforementioned embeddings as follows:

$$\hat{T}_{\mathcal{W},\mathcal{M},\mathcal{I}} = T_{\mathcal{W},\mathcal{M},\mathcal{I}} \oplus T_{\mathcal{T}}^{\mathcal{W},\mathcal{M},\mathcal{I}}, \tag{7.8}$$

where T and $\hat{T}$ represent the embedding of each modality and the final input token for the transformer, respectively.

Multi-Modal Transformer: We adopt the conventional encoder-only transformer architecture as the backbone. Our multi-modal transformer is a stack of several standard transformer blocks [139], consisting of the multi-head self-attention module and the position-wise feed-forward module. The self-attention mechanism helps the model enact the interactions between multiple objects, which allows it to be regarded as a fully connected graph structure. Besides, the language token and the mask token serve as conditional constraints and affect posture prediction through attention calculation. In the end, we build our network decoder as a two-layer MLP to output the denoised direction of t_i and r_i for each instance.

Efficiency-oriented Rearrangement Learning: To improve the interpretability of the network in our rearrangement task, we reparameterize the original score-based model s_θ to $\epsilon_\theta = \sigma_t^2 s_\theta$ as our multi-modal denoising diffusion model, indicating that the optimization objective evolves into a noise prediction problem. For the forward process during model training, we add sampled Gaussian noise with a specific standard deviation to the translation and rotation parameters of each object in the clean scene to formulate noisy S_t, which allows for the generation of various perturbed scenes with different levels of noise for one clean scene. After that, a reversed denoising process is learned by projecting S_t to the clean scene manifold via noise prediction.

As formulated in Eq. 7.1, we also want to minimize the motion distance between the initial messy scene and the rearranged clean scene. For most denoising diffusion works based on high-dimensional image space, it is typically presumed that the original image constitutes the nearest projection to its version perturbed by noise. However, each object instance's pose information in our task is low-dimensional data. This discrepancy suggests that with the introduction of different noise levels, the optimal projection target for a messy scene might not necessarily be consistent with the initially intended clean scene. Especially when applied to practical applications, such as food preparation or tabletop arrangement, the efficiency cost is enormous if persistently converting diverse messy scenes into the same specific layout.

Thus, we propose several techniques to ensure that the rearranged scene shares more similarities with the initial messy scene. First, we re-associate instances within the same class. Taking a language instruction as an example: *"Please rearrange all small boxes into a circle shape"*, we re-establish the pairing relationship p among all boxes between the current messy scene S and the target clean scene $\tilde{S}$ by computing their Earth Mover's Distance [204]:

$$\mathrm{EMD} = \min_{p} \frac{1}{n} \sum_{i=1}^{n} \left\| t_i - \tilde{t}_{p(i)} \right\|_2^2, \tag{7.9}$$

where n is the number of instances in the scene, t and $\tilde{t}$ represent translation parameters of S and $\tilde{S}$, respectively. During training, we choose $\tilde{t}_{p^*(:)}$ with the optimal pairing relationship p^* instead of $\tilde{t}_{:}$ to construct $\tilde{S}$, which hopefully encourages a more efficient movement during rearrangement.

Second, as shown in Fig. 7.2, horizontal and vertical structures possess a certain degree of ambiguity. Drawing on the principles of least squares approximation from statistical analysis, we further propose to pan the clean scene along the relevant axis. This is to ensure that the average position of all instances in the optimal target scene $\tilde{S}^*$ aligns with the average position of all instances in the messy scene S. Through this procedure,

```
import numpy as np
from vision_utils import get_obj_masks, PCA
from language_utils import LLM_parsing
from diffusion_utils import diffusion_pipeline
from robot_utils import pixel2pos, pick_and_place
```

Figure 7.6: Statements of python APIs in our rearrangement task.

we analytically guarantee minimal movement during the arrangement process, which can be formalized as:

$$\tilde{t_i^v}^* = \tilde{t_i^v} + \frac{1}{n}\sum_{i=1}^{n}\left(t_i^v - \tilde{t_i^v}\right) \ or\ \tilde{t_i^h}^* = \tilde{t_i^h} + \frac{1}{n}\sum_{i=1}^{n}\left(t_i^h - \tilde{t_i^h}\right), \tag{7.10}$$

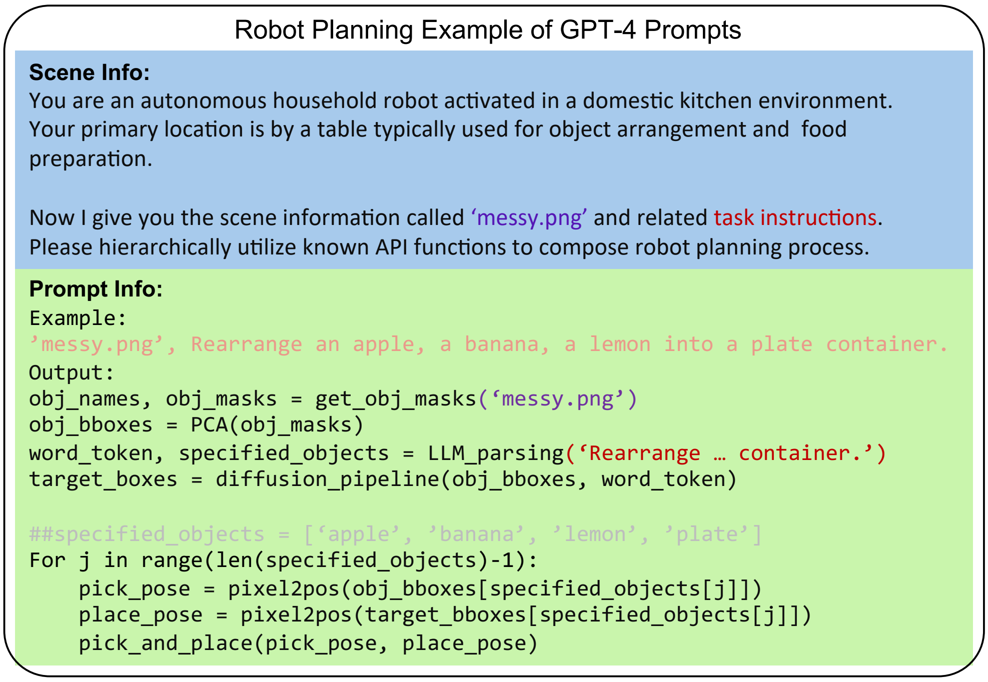
where t_i^v and t_i^h represent the coordinates of the vertical and the horizontal axis, respectively. We operate on the vertical axis for the horizontal structure and on the horizontal axis for the vertical structure.

Inference: During inference, we pursue the typical diffusion scheme shown in Eq. 7.4, where the learned $\epsilon_{\theta^*}(S_t)$ is asymptotically proportional to $\nabla_{S_t} \log q_0 (S_t)$ as t declines. Given a messy scene S with attributes extracted, we treat it as S_T with a specific time step T and recursively predict the layout of the "cleaner" scene. We update the translation and rotation parameters of each instance and iterate continuously. α_t and β_t in Eq. 7.4 are designed to decrease as denoising progresses. The final attained S_0 is our rearrangement of the messy scene.

7.5 VLM Programming as Planner

Recently, much work [112, 205] from LLM-based robotic manipulation has demonstrated that language models have the potential to directly generate code snippets by parametrizing object states and the robot controller API. However, they all have to define the goal object states manually to finish different manipulation tasks. In this work, we use the conditional diffusion model to imagine different goal object states and integrate them as prompts into a multimodal vision-language model (VLM) to generate programming policy. In practice, we use the OpenAI GPT-4 model as our VLM cornerstone. Moreover, the generated output from the GPT model is expected to be valid Python code that covers programs from visual perception, language parsing, and denoising diffusion to robotic control.

Specifically, we first need to define our own Python function libraries that can inform the GPT model of which APIs are available and provide type hints on how to use these APIs. Fig. 7.6 shows all statements that can be imported into our rearrangement task. Furthermore, a few demonstration examples are used as prompts to instruct the GPT model to present contextual understanding and few-shot learning ability. Fig. 7.7 gives an example that directly outputs executable planning code comprising the capability to perform arithmetic, call API functions, and implement other Python language



Robot Planning Example of GPT-4 Prompts

Scene Info:
You are an autonomous household robot activated in a domestic kitchen environment. Your primary location is by a table typically used for object arrangement and food preparation.

Now I give you the scene information called ‘messy.png’ and related task instructions. Please hierarchically utilize known API functions to compose robot planning process.

Prompt Info:

```
Example:
’messy.png’, Rearrange an apple, a banana, a lemon into a plate container.
Output:
obj_names, obj_masks = get_obj_masks(‘messy.png’)
obj_bboxes = PCA(obj_masks)
word_token, specified_objects = LLM_parsing(‘Rearrange … container.’)
target_boxes = diffusion_pipeline(obj_bboxes, word_token)

##specified_objects = [‘apple’, ’banana’, ’lemon’, ’plate’]
For j in range(len(specified_objects)-1):
    pick_pose = pixel2pos(obj_bboxes[specified_objects[j]])
    place_pose = pixel2pos(target_bboxes[specified_objects[j]])
    pick_and_place(pick_pose, place_pose)
```

Figure 7.7: An example of GPT-4 prompts that generate a planning policy based on human language instruction and visual information of the messy scene when the robot executes the object rearrangement task.

features. It can be seen that the GPT-4 model can well process multimodal inputs as instructions, then convert them into high-level perception features programmatically via vision, language, and denoising diffusion APIs, and finally call the low-level robot controller APIs to generate rearrangement actions. Owing to our denoising diffusion model being trained on a self-constructed synthetic dataset, it is usually challenging for traditional work to overcome the sim2real gap to implement our robotic arrangement task. However, by combining the open-vocabulary Grounded SAM model as a vision module, our GPT-based prompt-learning method can generalize to new objects and environments in real experimental scenarios well.

7.6 Simulation Experiments

7.6.1 Implementation Details

The token encoder produces 512-dimensional features as the input for the transformer, which has 2 layers with 8 heads of attention. The hidden layers of the transformer have 512 dimensions. We optimize our model on the proposed object rearrangement dataset, which contains 1640 clean scenes for training and 583 clean scenes for testing. We adopt the Adam optimizer with a base learning rate of 10^{-4}. The batch size is selected as 64 and the denoising model is trained on an A800 GPU for $30,000$ iterations, which takes about 3 hours. During inference, we choose to iterate 35 steps after considering both rearrangement efficiency and generative effectiveness, which takes seconds to rearrange a messy scene.

7.6.2 Evaluation Metrics & Baselines

To thoroughly estimate the performance of our proposed model in the simulated object rearrangement task, we utilize the following quantitative metrics:

Discrepancy between Results and the Ground Truth ($Dist2GT$): To measure the rearrangement quality, we compare the difference between the rearrangement result and the pre-perturbed scene. We compute the Earth Mover's Distance (EMD) between our rearranged configuration and the ground truth. We further calculate the cosine distance between the orientation of instances in the scene before and after rearranging consulting the new assignment from EMD. We report the average difference in position and orientation of scenes in the test dataset separately.

Distance Moved ($Movement$): To measure the rearrangement efficiency, we compute the average movement distance required for the scene to clean up. More specifically, we calculate the average Euclidean distance between the paired instances in the messy and rearranged layout for each scene. Then we report the mean value of all scenes in the test dataset. It is essential to consider the initial messy configuration and provide a solution that moves instances as little as possible to save time and energy for the robot.

Intersection over Union Threshold ($IOU_{threshold}$): In our work, we take the 2D oriented bounding box to represent the geometric attribute of the object. To quantitatively evaluate the arrangement accuracy, we compute the Intersection over Union (IoU) values between the predicted-target bounding box pairs for each instance. If the IoU value of arbitrary bounding box pairs is larger than a threshold δ (e.g. $\delta = 0.25, 0.5$), it is regarded as a success. Then we report the mean success rate of all rearranged objects in the test dataset.

In summary, $Dist2GT$ represents quality and alignment, $Movement$ shows the efficiency, and $IOU_{threshold}$ indicates success. For $Dist2GT$ and $Movement$, a lower value denotes a better performance of the generated results while a higher $IOU_{threshold}$ indicates a higher success rate. The metrics for real-world experiments will be introduced later.

Baselines: We reproduce 2 state-of-the-art baselines about object rearrangement on our dataset for comparison: 1) StructDiffusion [127], an object-centric and language-based iterative method that utilizes point clouds and instructions to learn global structures of object rearrangement. Unlike our approach, it introduces an extra time embedding in its diffusion framework to iterate from pure Gaussian noise without considering the initial messy configuration that the robot encounters. 2) LEGO-Net [199], a transformer-based data-driven method that learns to rearrange objects in messy rooms, where the concept of the moving distance of each object is first introduced. However, it lacks specialized designs for rich language conditions and semantic structures, and it cannot be directly applied to robot manipulation tasks. We reproduce these methods on our 2D rearrangement dataset, where training and evaluation splits remain the same as our method.

Table 7.2: Quantitative comparisons on the task of rearranging into three kinds of global structures in the simulation experiments.

Method	Horizonal					Vertical					Circle				
	$Dist2GT_T\downarrow$	$Dist2GT_R\downarrow$	$Movement\downarrow$	$IoU_{0.25}\uparrow$	$IoU_{0.5}\uparrow$	$Dist2GT_T\downarrow$	$Dist2GT_R\downarrow$	$Movement\downarrow$	$IoU_{0.25}\uparrow$	$IoU_{0.5}\uparrow$	$Dist2GT_T\downarrow$	$Dist2GT_R\downarrow$	$Movement\downarrow$	$IoU_{0.25}\uparrow$	$IoU_{0.5}\uparrow$
StructDiffusion [127]	0.308±0.004	0.071±0.010	0.481±0.004	13.5±0.7	3.3±0.5	0.249±0.004	0.016±0.003	0.448±0.006	21.0±1.0	6.2±0.6	0.248±0.003	0.011±0.002	0.429±0.013	23.8±0.8	10.3±0.4
LEGO-Net [199]	0.192±0.014	0.077±0.014	0.404±0.016	42.2±2.0	23.0±0.9	0.190±0.016	0.087±0.020	**0.394±0.009**	39.9±2.4	23.4±1.5	0.205±0.003	0.079±0.006	**0.375±0.008**	33.1±1.2	17.9±0.9
Ours	**0.103±0.002**	**0.005±0.002**	**0.397±0.012**	**51.8±0.7**	**28.5±1.2**	**0.109±0.002**	**0.001±0.000**	0.415±0.006	**53.0±0.9**	**27.6±1.8**	**0.153±0.002**	**0.001±0.000**	0.383±0.007	**43.4±0.7**	**21.6±0.5**

Table 7.3: Quantitative comparisons on the task of placing into containers in the simulation experiments.

Method	$Dist2GT_T \downarrow$	$Dist2GT_R \downarrow$	$Movement \downarrow$	$IoU_{0.25} \uparrow$	$IoU_{0.5} \uparrow$
StructDiffusion [127]	0.106±0.001	0.002±0.000	0.479±0.011	19.7±1.0	7.4±0.4
LEGO-Net [199]	0.095±0.001	0.001±0.000	0.482±0.005	26.2±1.0	11.4±1.0
Ours	**0.085±0.001**	**0.001±0.000**	**0.458±0.024**	**37.1±2.4**	**17.8±1.4**

7.6.3 Quantitative & Qualitative Analysis

Quantitative Results:

We compare our method against the baselines mentioned above. We perturb clean scenes in the test set and rearrange these messy scenes with various methods. We conduct 5 replication experiments for each algorithm and report the average and confidence interval values on several metrics. The main results are presented in Tab. 7.2 and Tab. 7.3, with the best results shown in **bold** and the inferior results within the confidence interval underlined. Our method is shown to outperform previous methods in most aspects. Due to StructDiffusion [127] starting denoising from pure noise, it ignores the initial configuration. Therefore, the rearrangement results obtained require a longer movement distance. As for LEGO-Net [199], it does not consider language conditions, thus causing uncertainty about achieving which kinds of structure. Our multi-modal transformer network increases the controllability of the rearrangement process, allowing for more precise implementation of various regularities. Moreover, the “container” object serves a distinct function compared to the regular objects being arranged. By introducing an extra “mask” object class and applying dedicated models to handle it, we achieve a better performance in the *containing* task, as evidenced in Table 7.3.

Qualitative Results: Considering different human preferences from the same messy scene, we visualize some rearranged results of our model in Fig. 7.8. We use oriented bounding boxes to represent instances on the table, with different colors conveying different classes, whereas containers are depicted by directionless bounding boxes. For the same messy scene of each task, our method can rearrange it into different layouts according to different conditions. For instance, the final placements of forks (red) and knives (blue) for food preparation conform to the local regularities of *symmetry* and *uniformity* while the horizontal structure is also achieved. In addition, for the object containing task, our model can rearrange the objects from different categories into corresponding containers regardless of their location variations. One noteworthy point is that the misalignment of containers in column 4 has not appeared in the training split. To sum up, our model can learn how to leverage multi-modal conditional constraints for rearrangement, which makes our method applicable and generalizable to practical scenarios.

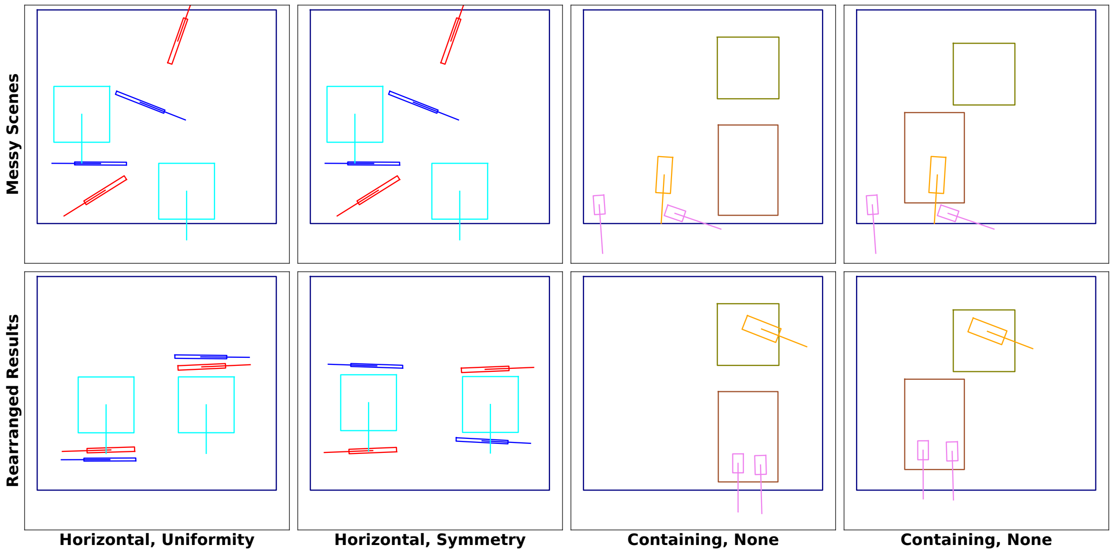


Figure 7.8: Qualitative results from different rearrangement tasks: Food preparation (column 1 and column 2): knives, forks, and plates; Object containing (column 3 and column 4): cans and a banana. Our model can rearrange the same messy scene of both tasks following different instructions.

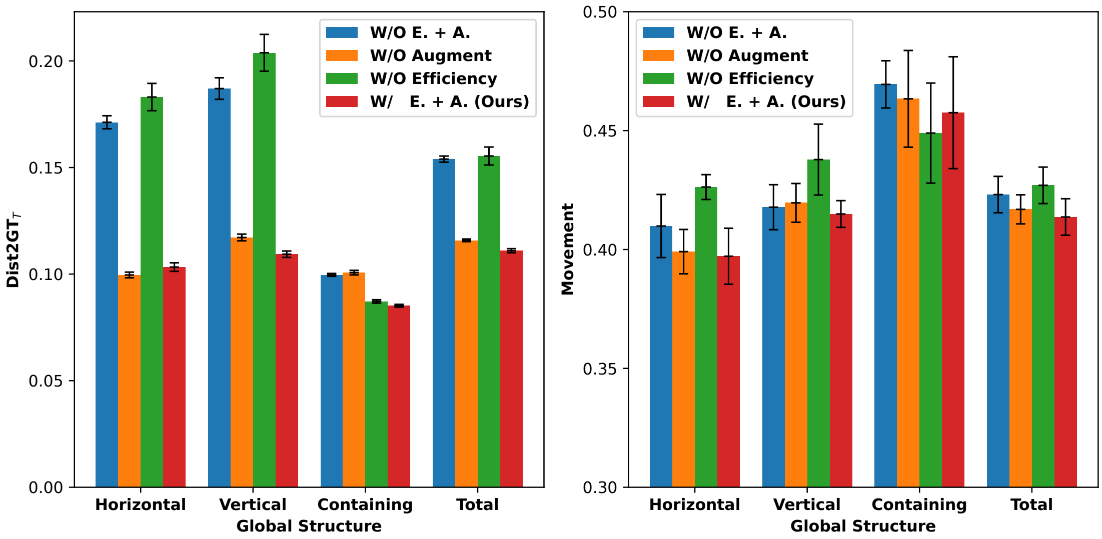


Figure 7.9: Rearrangement results on different global structures. We compare our method with several variants under 2 metrics.

7.6.4 Ablation Study

As one of our contributions is to propose an efficiency-oriented rearrangement method, we further compare our method with “W/O Efficiency” that does not employ the operations in Eq. 7.10, “W/O Augment” that does not adopt data augmentation, and “W/O E. + A.” that utilizes neither. These variants are evaluated on the test split across various global structures. As shown in the results of “Total” in Fig. 7.9, our method achieves

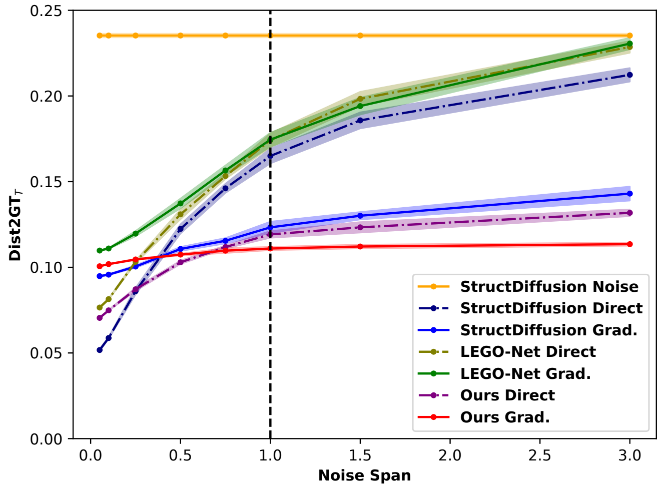


Figure 7.10: Rearrangement results under different noise spans. Various baselines and inference strategies are compared. The black dashed line represents the noise span we adopted for training.

a better overall performance than other variants in terms of both quality and efficiency of rearrangement. Especially for the *horizontal* and *vertical* structures, a significant improvement in the $Dist2GT_T$ metric can be seen owing to mitigating the ambiguity of the projection target. Besides, due to obtaining the optimal target on the clean scene manifold, we achieve a smaller motion distance in the $Movement$ metric. Moreover, the data augmentation operation can slightly improve the performance on various structures in both metrics, especially for the *containing* structure.

Since we can perturb the clean scene with different noise levels, we further investigate the model's denoising ability in dealing with different perturbing noises. Following [199], we use a noise span hyper-parameter σ to characterize the spectrum of σ_t, which originates from the positive half of $\mathcal{N}(0, \sigma^2)$. When we disturb scenes with σ_t derived from σ, the larger the σ value, the more likely the mess becomes severe. Meanwhile, as the trained denoising network ϵ_{θ^*} approximates the added noise $S_0 - S_t$, we can set $\alpha = 1$ and $\beta = 0$ in Eq. 7.4 to directly obtain S_0 and denote it as the *Direct* denoising strategy, distinguished from the standard *Gradual* denoising strategy.

Based on the $Dist2GT_T$ metric, we evaluate StructDiffusion [127], LEGO-Net [199], and our method combined with these inference strategies on the test split. For StructDiffusion, we further adopt its original inference process starting from pure noises and name it *Noise*. As shown in Fig. 7.10, our *Gradual* recipe demonstrates the best denoising performance as the increment of noise spans, indicating that the proposed multi-modal transformer architecture can stably reconstruct a regular scene, even though the perturbation added to the scene is quite significant. Moreover, it can be seen that the *Direct* strategy exhibits a worse denoising ability than the *Gradual* strategy in handling high-noise scenes among all methods, possibly due to inaccurately estimated scores in low-density regions. The comparison results prove that iterative denoising is crucial for rearrangement, as it can gradually update data to high-density regions that possess more accurate estimates.

7.6.5 Visualization Comparison

In Fig. 7.11, we further visualize several comparison results of rearranged scenes in the physical simulator, where the dynamics of object collisions are accounted for. Given the special language command and the messy scene, it can be seen that among all scenes, our method can achieve the most precise arrangement of objects that conforms to human intentions while meeting the demand for efficiency. For example, in the second scene, the rearranged result from StrcutDiffusion [127] appears satisfactory, yet a more substantial movement for each object is needed compared to our rearranged result. In the case of LEGO-Net [199], a physical collision occurred between the mug and the meat can, leading to their dispersion across the table's surface.

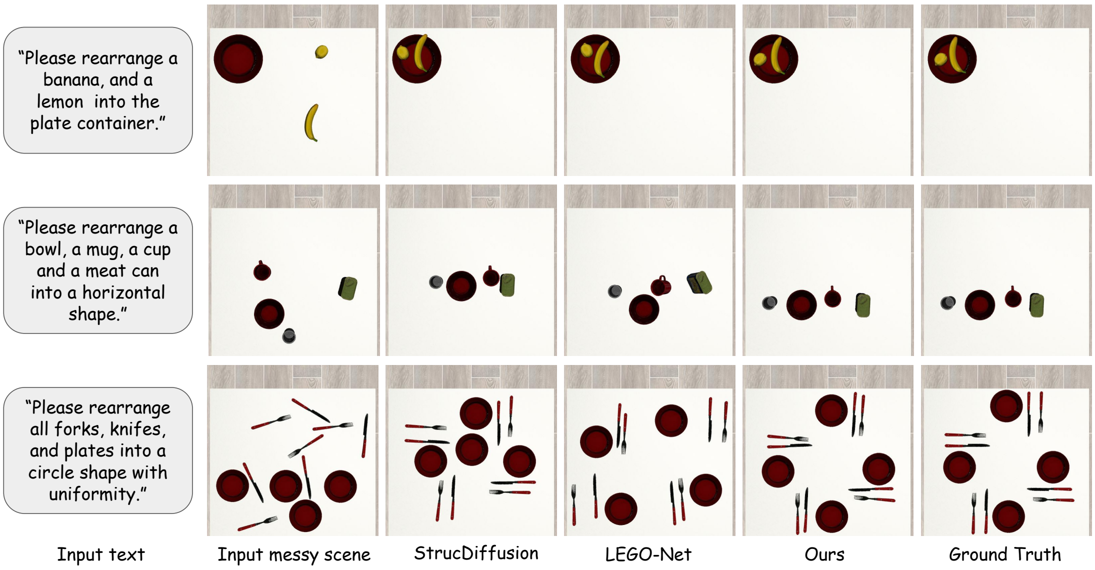

"Please rearrange a banana, and a lemon into the plate container."
"Please rearrange a bowl, a mug, a cup and a meat can into a horizontal shape."
"Please rearrange all forks, knifes, and plates into a circle shape with uniformity."
Input text
Input messy scene
StrucDiffusion
LEGO-Net
Ours
Ground Truth

Figure 7.11: Visualization results in simulation. We compare our method with state-of-the-art methods StructDiffusion [127] and LEGO-Net [199].

7.7 Robot Experiments

7.7.1 Experimental Setup

To verify the proposed scheme in the real world, we also deploy a robotic experimental system shown in Fig. 7.12. The robotic manipulator selected for our setup consists of a 7-DOF KUKA LWR arm paired with a Schunk WSG50 gripper, which is mounted on the side of a table. The fixed point where the robot arm connects to the table is considered the base, with its centre position in the real-world coordinate system formulated as $[x, y, z] = [0.0, 0.0, 0.8]$. Our vision system incorporates the Kinect V2 in *qhd* mode to capture raw images. The *qhd* mode, while offering a wide field of view (FOV), also introduces the challenge of potentially detecting extraneous objects, such as camera fixtures and robotic equipment, as noise for the open-vocabulary Grounded SAM model. To mitigate this, we trim the raw image data to a uniform size of 448 pix-

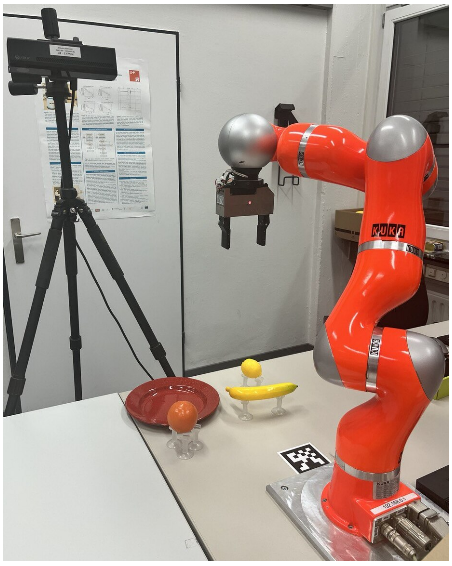
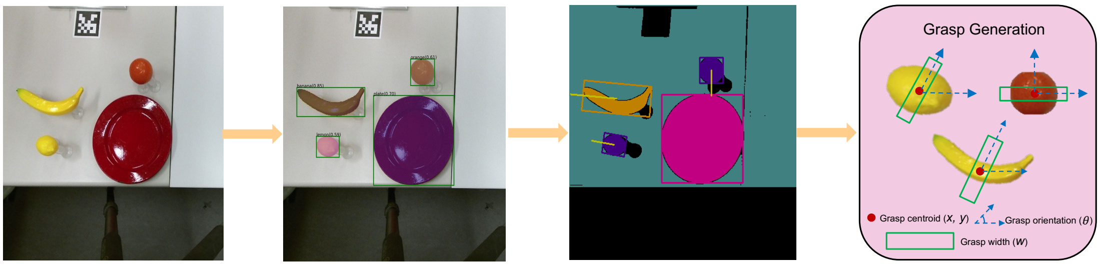


Figure 7.12: Experimental setup for tabletop robotic rearrangement in the real world, consisting of the robotic arm, gripper, vision system, and messy scene.



Input messy scene Object recognition Oriented bounding box Output grasp

Figure 7.13: The process of grasp generation for each object in the real robot experiments.

els for both height and width. An AprilTag located on the table is used to calibrate the vision system, which will further facilitate the transformation of object pose from pixel coordinates to world coordinates. Finally, the process of generating grasps for the real robot experiment is illustrated in Fig. 7.13. To calculate the grasp on each object, we employ the antipodal method on its oriented bounding box. This involves determining the grasp point (x, y) and orientation θ based on the bounding box's average position and rotation value. The whole experimental system is operated by a ROS interface and the *pick_and_place* API of the robot in our VLM planner is achieved based on MoveIt!.

7.7.2 Robotic Rearrangement results

We conduct 12 evaluations for each task by altering the position and orientation of objects within a messy scene and arranging them according to a language-conditioned structure. For the structures defined as *horizontal*, *vertical*, and *circle*, the categories of objects in the messy scene include small boxes, toothpaste boxes, knives, forks, and spoons. As for the *containing* structure, the scene's objects consist of small boxes, box

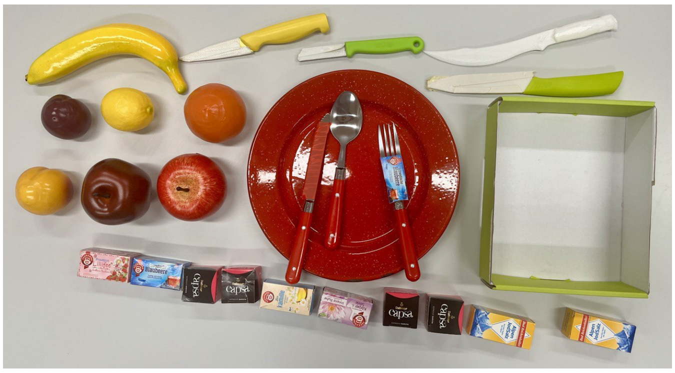


Figure 7.14: All testing objects for robotic table rearrangement in the real world.

Table 7.4: Quantitative comparisons on the task of rearranging into different structures in the real robot experiments.

Object structure	Horizontal	Vertical	Circle	Containing
Duration (s)	75.5	87.0	69.0	79.5
Collision-free rate (%) avg (std)	75.0 (16.7)	83.3 (8.33)	75.0 (8.3)	91.7 (8.33)
Success rate (%) avg (std)	66.7 (16.7)	75.0 (8.3)	58.3 (8.3)	83.3 (8.33)

containers, plate containers, and various fruits. All objects utilized in our robotic experiments are displayed in Fig. 7.14.

In Tab. 7.4, we first compare the average rearrangement duration, collision-free rate, and final successful rate for different structures in the real robot experiment. The collision-free rate is calculated by observing whether all objects in the rearrangement scene predicted by our denoising diffusion model are collision-free. Additionally, the success rate is assessed after the robot finishes each rearrangement task. Unlike the abstract estimation metrics in simulation, a real-world rearrangement is considered successful only if the objects are positioned without any collisions and the overall structure adheres to the semantic constraints set forth by the provided language instructions. It can be seen that the *horizontal*, *vertical*, and *circle* configurations present significant challenges due to the entirely novel nature of various boxes in our dataset and the limited tolerant positions for sequential placement. The requirement to rearrange the same categories of objects into these three distinct structures simultaneously further complicates the task for our inference model. Additionally, we encounter failures when the vision system struggles to accurately perceive the depth of particular objects, such as spoons, knives, forks, and bananas, due to their reflective surfaces. This incorrect depth data prevents the robot's gripper from achieving a stable grasp.

Furthermore, we compare the average success rate with baselines on four kinds of language-conditioned structures, shown in Fig. 7.15. Our method outperforms other baselines for all global structures, with an average improvement of 15% on the suc-

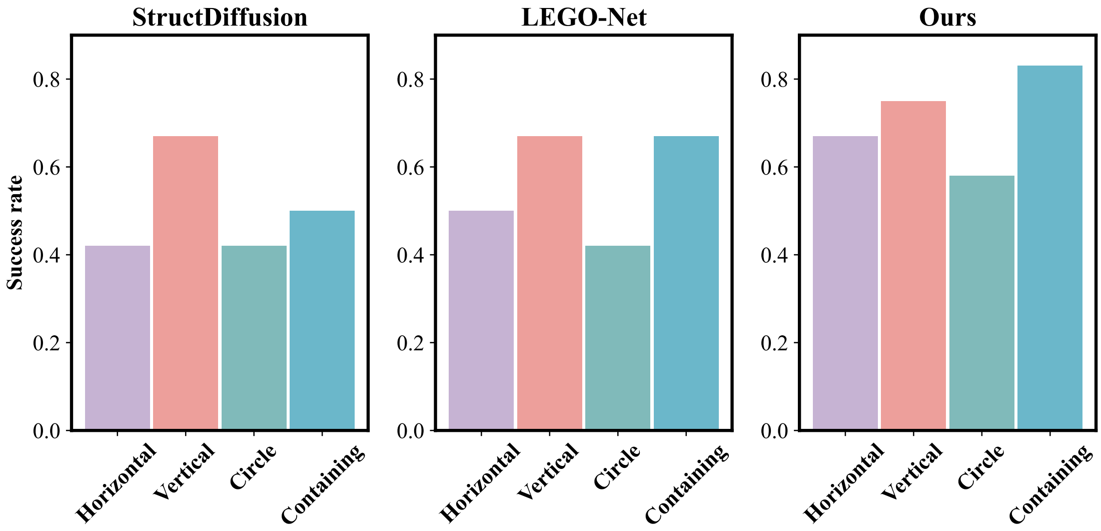


Figure 7.15: Comparisons of the average success rate on various rearrangement structures for different methods in the real robot experiments.

cess rate compared to LEGO-Net [199]. We also find that the real-world rearrangement becomes more challenging when more objects are added to the initial messy scene. This may be attributed to the struggle of diffusion models to learn a more complicated inter-object relationship, along with the increase in robot planning and execution horizons.

7.8 Discussion and Summary

In this chapter, we present a solution for the language-conditioned robotic rearrangement task with different global structures and local regularities. Firstly, we collect and process a 2D synthetic arrangement dataset based on the physical simulator. To capture long-range dependencies between visual and textural inputs, we build our conditional diffusion model based on the multi-modal transformer architecture, which endows the robot with the ability to imagine the target pose information of different objects from the observation scene. In particular, we introduce an efficient-oriented rearrangement learning strategy to reduce object motion distance and create a more appropriate layout. Inspired by the recent prompt-based learning, we further integrate the generative model into the most advanced VLM module (GPT-4) to generate robot planning and action policy. Finally, we carefully design three kinds of quantitative metrics to evaluate our model in the simulation experiments, showing that our generative model outperforms related state-of-the-art methods. Extensive experiments on the real robot further demonstrate that our proposed scheme can satisfy the human language-based requirements and finish different rearrangement tasks successfully on diverse unseen objects.

Concerning limitations, this work simplifies the inter-object relationship in a clean rearrangement scene based on human preference, with the global structure and the local regularity limited to 4 and 2, respectively. In addition, when dealing with more complex object interactions, our approach tends to rearrange objects into a simpler layout composed of fewer objects by overlapping some instances. Therefore, extending our research to include and comprehend the inter-object relationship on a larger scale is important. Moreover, since we use prompt-based learning to achieve language instruction parsing and robot action policy generation, we still need to pre-define a series of examples to instruct the VLM module to interpret the prompt. As for future work, the introduction of explicit collision avoidance mechanisms in the denoising process can be explored, which may make the generated layout more plausible. Besides, we currently use the same number of inference steps, and we wonder whether it is possible to determine a more accurate number of denoising steps by assessing the level of mess in the present scene, which may enhance efficiency. Finally, designing an end-to-end VLM model to estimate robot actions directly rather than separating the dreaming and planning process may further improve the effectiveness of robot manipulation.

Chapter 8

Conclusions and Future Work

To achieve embodied intelligence, we have discussed four kinds of robot learning strategies that enable physical robots to interact with unstructured environments through different perception modalities. We hope to endow the robots with the ability to flexibly perform a wide range of tasks and to adaptively generalize new tasks. In detail, the thesis aim outlined in Section 1.2 is broken down into five sub-achievements, which are illustrated in Section 8.1. Section 8.2 details the limitations of this thesis, while Section 8.3 outlines the future efforts the author intends to undertake to achieve a general robot learning toward embodied intelligence.

8.1 Achievements

As stated in Section 1.2, we introduced our work that analyzes perception-action robot learning at four levels of abstraction:

- **Improve perception learning:** A pioneering transformer-based sparse shape completion network (TransSC) is proposed to reconstruct the raw partial point cloud. It comprises a transformer-based encoder and a manifold-based decoder. This design enables our model to achieve superior completion results, outperforming other baseline methods. Our experiments demonstrate that our network is resilient to sparse and noisy point cloud inputs. Additionally, simulation grasping experiments indicate that our model can achieve lower grasp joint errors compared to traditional robotic completion methods. Furthermore, in real robotic experiments involving single objects and object occlusion, we show that our TransSC can be seamlessly integrated into an existing grasp evaluation module, significantly enhancing grasping performance in both scenarios.

- **Pursue Context-aware Learning:** Inspired by the concept of affordance, we explore the issue of task-oriented 6-DoF robotic grasping and hand-object action recognition. As the grasping detection, we introduce an implicit estimation network, a grasp evaluation network, and an attention-aware visual affordance network. Our solution demonstrates significant improvements over existing baselines for both familiar and novel objects within our specially constructed affordance grasp dataset. For action recognition, we benchmark the first dataset using event vision for hand-object action tasks and demonstrate the potential for transferring to real robot manipulation tasks.

- **Transfer multi-modal learning:** We further study a solution for the multiple peg-in-hole assembly (MPA) problem using a multimodal representation, allowing the transfer of a trained policy from simulation to the real world without the need for additional exploration. We propose a specialized visual representation module and a tokenization-based transformer module to effectively compact features as the core of the reinforcement learning framework. We conduct extensive evaluations of our solution in a simulation environment, where it achieves a high success rate in the more demanding multiple peg-in-hole assembly scenario. Finally, the generalization capability of our solution is also confirmed across various object shapes.

- **Embrace imaginative learning:** Beside the ability of perception learning, we also want to the robot have the ability of imagination to generate an appropriate target for his next action similar to human kinds. Therefore, we present a solution for the language-conditioned robotic rearrangement task, where a conditional diffusion model based on the multi-modal transformer architecture and cutting-edge VLM module (GPT-4) to formulate robotic planning and action policies. Further extensive experiments with a real robot show that our proposed approach can effectively meet language-based requirements and successfully complete various rearrangement tasks involving diverse, unseen objects.

8.2 Limitations

While this thesis has advanced our understanding of robot manipulation learning, it is important to recognize the inherent limitations associated with our research methods and scope. Addressing these limitations is crucial for interpreting the findings and guiding future research for the robotic community.

Our work focuses on different robotic manipulation tasks using a parallel gripper, extending these capabilities to a service robot equipped with a multi-fingered hand introduces complexities in learning dexterous manipulation skills. Moreover, the task settings we have used are relatively simple compared to the multifaceted daily tasks performed by humans, such as folding clothes or washing dishes.

Our current robot learning methods achieve limited generalization across different task settings. For example, in the multiple peg-in-hole assembly task, real robot experiments show that the assembly policy has limited generalization to different object shapes and positions. This limitation highlights the difficulty in transferring policies trained on one robot task to different tasks in the real world. Achieving broad generalizability across various tasks and robots is essential for realizing embodied intelligence.

Our research also underscores the critical role of datasets in different learning strategies. Although we have developed specific datasets for certain manipulation tasks, these are not without limitations. For instance, in the work on affordance grasping, the variety of object categories and the corresponding affordance annotations are restricted, which limits the applicability of the inference model to unseen object categories. With the rapid advancements in deep learning, particularly with the emergence of technologies like transformers and diffusion models, robotic researchers can more effectively process multisensory information to learn diverse latent manipulation skills. However, the creation of a comprehensive vision-language-action (VLA) dataset is crucial for the robotics community. Such a dataset would be akin to the ImageNet dataset in the computer vision field and could significantly advance the development of embodied intelligence.

8.3 Future Work

There is a lot that can be done to guide the future work of the thesis topic:

- **Introducing language into context-aware manipulation:** Language serves as a bridge between human instructions and robotic actions, facilitating a more intuitive interaction. By understanding context-aware commands such as *"cut the carrot" and "open the door"*, robots can interpret the context and intentions behind tasks, allowing for more accurate and appropriate interaction with objects.

- **Exploring better algorithms for transfer across substantially different families of tasks:** The goal of an intelligent system is to not only perform well on tasks similar to those they have encountered during training but also adapt seamlessly to entirely new, related or unrelated tasks. We have explored the use of deep learning architectures that can abstract higher-level features across tasks, and domain randomization techniques that minimize the gap between simulation and reality domains. The discrepancies in the complexity and type of data available across

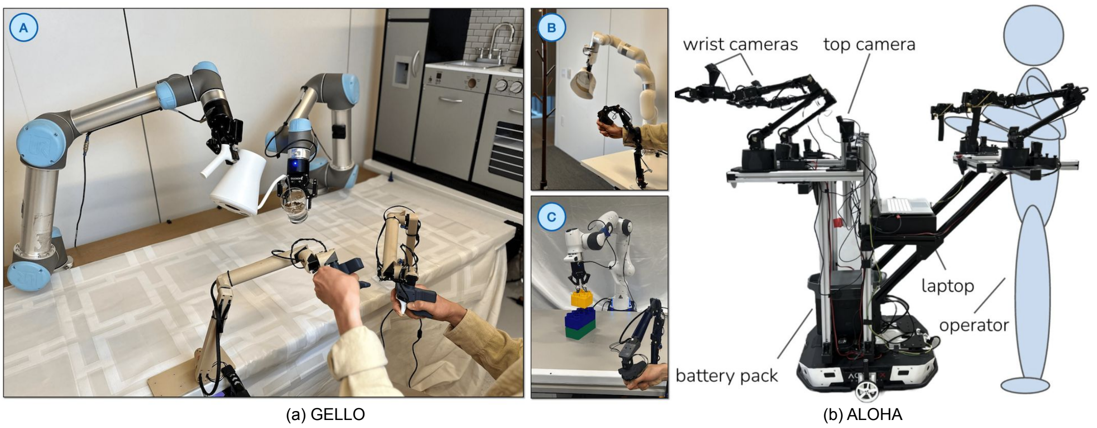


Figure 8.1: Examples of recent work about data collection platforms for embodied intelligence. (a) GELLO [206]. Reprinted image: ©2023, IEEE. (b) ALOHA [207]. Reprinted image: ©2024, IEEE.

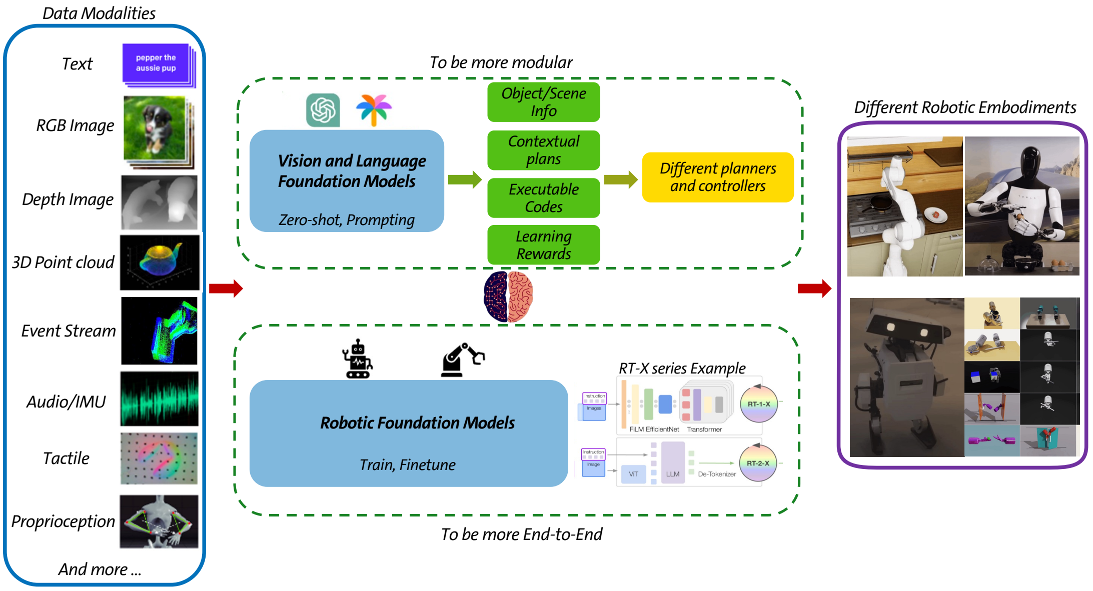


Figure 8.2: The system of embodied intelligence models for general multimodal dexterous manipulation.

robotic tasks can significantly hinder the effectiveness of transfer learning in real-world experiments.

- **Optimizing the diffusion learning into the robot tabletop rearrangement:** Current robot rearrangement work does not consider explicit collision avoidance mechanisms in the denoising process. We can integrate it with a collision-checking network to make the generated layout more plausible. Besides, we currently use the same number of inference steps, and we wonder whether it is possible to determine a more accurate number of denoising steps by assessing the level of mess in the present scene, which may enhance efficiency.

- **Employing multi-fingered hand to do dexterous manipulation tasks:** Multi-fingered hands are typically equipped with multiple degrees of freedom per finger, allowing for fine-motor control akin to that of a human hand. This capability is critical for tasks that involve complex object manipulation such as assembly in manufacturing, surgical operations in healthcare, or domestic tasks in personal robotics. Therefore, to adapt robot learning strategies for use in humanoid service robots, we can utilize deep learning and reinforcement learning models to acquire dexterous manipulation skills such as playing the piano, which is beyond the capabilities of a parallel gripper. More importantly, Fig. 8.1 shows some recent work of data collection platforms for the training of embodied intelligence. To construct a big VLA dataset of dexterous manipulation, we also devote ourselves to designing a teleoperation hardware platform to collect human demonstration data for the robotic community. Finally, we aim to develop a foundation model to process all kinds of modality information and achieve general dexterous manipulation skills for different robotic embodiments, described in Fig. 8.2.

Appendix A

List of Abbreviations

AIGC artificial intelligence-generated content

ARSNN Attention-based Residual Spiking Neural Network

CMOS Complementary Metal–oxide–semiconductor

CD Chamfer Distance

DDPG Deep Deterministic Policy Gradien

EMD Earth Mover's Distance

FLOPs Floating-point Operations

FPS Farthest Point Sampling

GSAM Grounded Segment Anything Model

GPD Grasp Pose Detection

GQCNN Grasping Quality Convolutional Neural Network

IL Imitation learning

LIF Leaky Integrate-and-fire

LLMs Large Language Models

MSE Mean Squared Error

MLP Multi-Layer Perception

MPA Multi-peg-assembly

PLIF Parametric Leaky Integrate-and-Fire

POMDP Partially Observable Markov Decision Process

PPO Proximal Policy Optimization

PCA Principle Component Analysis

RL Reinforcement Learning

RCLS Repeated Closed-loop Smooth

SNN Spiking Neural Network

SAC Soft Actor-Critic

TSDF Truncated Signed Distance Function

TOG Task-oriented Grasping

VGN Volumetric Grasping Network

VAE Variational Autoencoder

Appendix B

Publications

The following first-author publications constitute a significant part of my PhD thesis. The list is organized chronologically.

- Wenkai Chen, Changming Xiao, Ge Gao, Fuchun Sun, Changshui Zhang, and Jianwei Zhang. DreamArrangement: Learning Language-conditioned Robotic Rearrangement of Objects via Denoising Diffusion and VLM Planner. IEEE Transactions on Systems, Man and Cybernetics: Systems, 2024. (Minor Revision)

- Wenkai Chen, Shang-Ching Liu, and Jianwei Zhang. EHoA: A Benchmark for Task-oriented Hand-Object Action Recognition via Event Vision. IEEE Transactions on Industrial Informatics, 2024.

- Wenkai Chen, Chao Zeng, Hongzhuo Liang, Fuchun Sun, and Jianwei Zhang. Multimodality driven impedance-based sim2real transfer learning for robotic multiple peg-in-hole assembly. IEEE Transactions on Cybernetics, 2023.

- Wenkai Chen, Chao Zeng, Fuchun Sun, and Jianwei Zhang. Learning Multiple Peg-in-hole Assembly Skills Using Multimodal Representations and Impedance Control. International Conference on Robotics and Automation Workshop (ICRA Workshop), 2023.

- Hongzhuo Liang*, Wenkai Chen*, Shang-Ching Liu, Fuchun Sun, and Jianwei Zhang. Learning Multimodal Multifingered Dexterous Manipulation Skills from Human Demonstration. Science China Information Sciences, 2023. *(*Revision)

- Wenkai Chen, Hongzhuo Liang, Zhaopeng Chen, Fuchun Sun, and Jianwei Zhang. Improving object grasp performance via transformer-based sparse shape completion. Journal of Intelligent & Robotic Systems, 104(3):1–14, 2022.

- Wenkai Chen, Hongzhuo Liang, Zhaopeng Chen, Fuchun Sun, and Jianwei Zhang. Learning 6-dof task-oriented grasp detection via implicit estimation and visual affordance. In 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 762–769. IEEE, 2022.

Appendix C

Acknowledgements

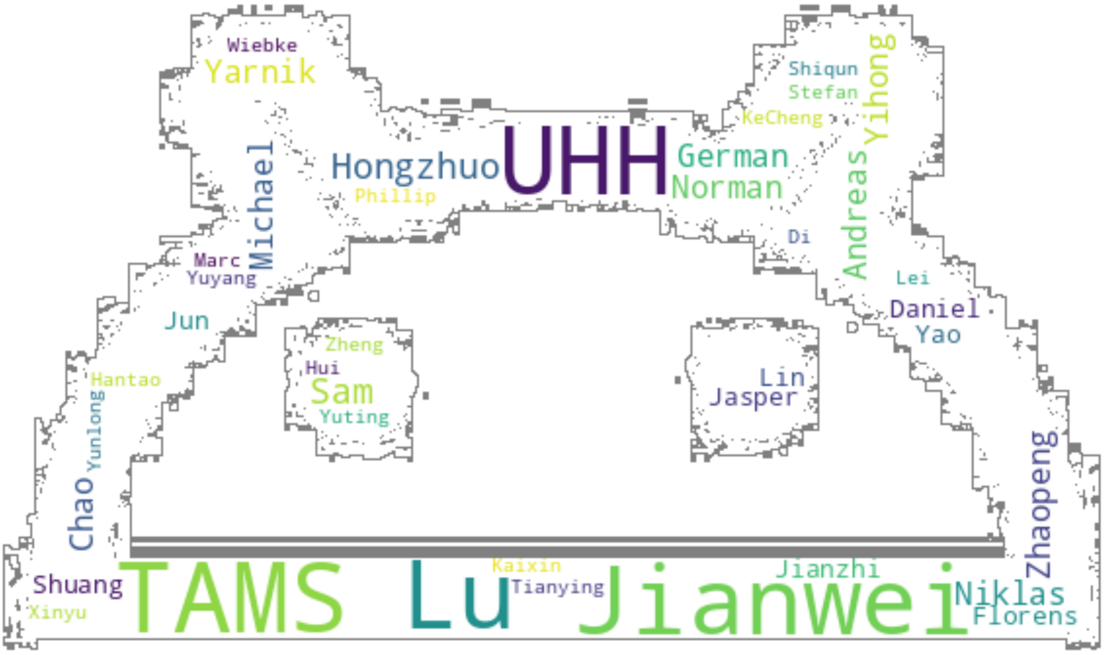


Figure C.1: Acknowledgement from a Minibot: I sincerely thank my professor, colleagues, and friends for their selfless support and guidance. Your invaluable suggestions and assistance have greatly contributed to my development as a robotic researcher throughout my four-year PhD study. Special thanks to the AI tool UHHGPT provided by the University of Hamburg, which assisted me in spell-checking and optimizing expressions in the Abstract and Introduction sections (The input prompt is: *Can you help me make a spell-checking and expression optimizing for the provided sentence?*).

Bibliography

[1] Linda Smith and Michael Gasser. The Development of Embodied Cognition: Six Lessons from Babies. *Artificial Life*, 11(1-2):13–29, January 2005.

[2] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[3] Wenkai Chen, Hongzhuo Liang, Zhaopeng Chen, Fuchun Sun, and Jianwei Zhang. Improving Object Grasp Performance via Transformer-Based Sparse Shape Completion. *Journal of Intelligent & Robotic Systems*, 104(3):1–14, 2022.

[4] Hongzhuo Liang, Xiaojian Ma, Shuang Li, Michael Görner, Song Tang, Bin Fang, Fuchun Sun, and Jianwei Zhang. PointNetGPD: Detecting Grasp Configurations from Point Sets. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3629–3635. IEEE, 2019.

[5] Wenkai Chen, Hongzhuo Liang, Zhaopeng Chen, Fuchun Sun, and Jianwei Zhang. Learning 6-DoF Task-oriented Grasp Detection via Implicit Estimation and Visual Affordance. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 762–769. IEEE, 2022.

[6] Wenkai Chen, Shang-Ching Liu, and Jianwei Zhang. EHoA: A Benchmark for Task-oriented Hand-Object Action Recognition via Event Vision. *IEEE Transactions on Industrial Informatics*, 2024.

[7] Wenkai Chen, Chao Zeng, Hongzhuo Liang, Fuchun Sun, and Jianwei Zhang. Multimodality Driven Impedance-Based Sim2Real Transfer Learning for Robotic Multiple Peg-in-Hole Assembly. *IEEE Transactions on Cybernetics*, 2023.

[8] Wenkai Chen, Changming Xiao, Ge Gao, Fuchun Sun, Changshui Zhang, and Jianwei Zhang. DreamArrangement: Learning Language-conditioned Robotic Rearrangement of Objects via Denoising Diffusion and VLM Planner. *Authorea Preprints*, 2024.

[9] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1316–1322. IEEE, 2015.

[10] Fu-Jen Chu, Ruinian Xu, and Patricio A Vela. Real-world Multi-object, Multi-grasp Detection. *IEEE Robotics and Automation Letters*, 3(4):3355–3362, 2018.

[11] Tarik Tosun, Daniel Yang, Ben Eisner, Volkan Isler, and Daniel Lee. Robotic Grasping through Combined Image-Based Grasp Proposal and 3D Reconstruction. *arXiv preprint arXiv:2003.01649*, 2020.

[12] Michel Breyer, Jen Jen Chung, Lionel Ott, Roland Siegwart, and Juan Nieto. Volumetric Grasping Network: Real-time 6 DOF Grasp Detection in Clutter. *arXiv preprint arXiv:2101.01132*, 2021.

[13] Chaozheng Wu, Jian Chen, Qiaoyu Cao, Jianchi Zhang, Yunxin Tai, Lin Sun, and Kui Jia. Grasp Proposal Networks: An End-to-End Solution for Visual Learning of Robotic Grasps. *Advances in Neural Information Processing Systems*, 33:13174–13184, 2020.

[14] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp Pose Detection in Point Clouds. *The International Journal of Robotics Research*, 36(13-14):1455–1473, 2017.

[15] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017.

[16] Wenkai Chen, Hongzhuo Liang, Zhaopeng Chen, Fuchun Sun, and Jianwei Zhang. Improving Object Grasp Performance via Transformer-Based Sparse Shape Completion. *Journal of Intelligent & Robotic Systems*, 104(3):45, 2022.

[17] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-DOF GraspNet: Variational Grasp Generation for Object Manipulation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2901–2910, 2019.

[18] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: Point Completion Network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018.

[19] T Groueix, M Fisher, VG Kim, BC Russell, and M Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. *arXiv preprint arXiv:1802.05384*, 11, 2018.

[20] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and Sampling Network for Dense Point Cloud Completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11596–11603, 2020.

[21] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. GRNet: Gridding Residual Network for Dense Point Cloud Completion. In *European Conference on Computer Vision*, pages 365–381. Springer, 2020.

[22] Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. Shape Completion Enabled Robotic Grasping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2442–2447, 2017.

[23] Andrew T Miller and Peter K Allen. Graspit! A Versatile Simulator for Robotic Grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, 2004.

[24] Jens Lundell, Francesco Verdoja, and Ville Kyrki. Robust Grasp Planning Over Uncertain Shape Completions. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1526–1532. IEEE, 2019.

[25] Jens Lundell, Francesco Verdoja, and Ville Kyrki. Beyond Top-Grasps Through Scene Completion. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 545–551. IEEE, 2020.

[26] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics. *arXiv preprint arXiv:1703.09312*, 2017.

[27] James J Gibson. The theory of affordances. the ecological approach to visual perception. In *The People, Place and, Space Reader*, pages 56–60. Routledge New York and London, 1979.

[28] Shengheng Deng, Xun Xu, Chaozheng Wu, Ke Chen, and Kui Jia. 3D AffordanceNet: A Benchmark for Visual Object Affordance Understanding. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1778–1787, 2021.

[29] Yuke Zhu, Alireza Fathi, and Li Fei-Fei. Reasoning about Object Affordances in a Knowledge Base Representation. In *2014 European Conference on Computer Vision (ECCV)*, pages 408–424. Springer, 2014.

[30] Austin Myers, Ching L Teo, Cornelia Fermüller, and Yiannis Aloimonos. Affordance detection of tool parts from geometric features. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1374–1381. IEEE, 2015.

[31] Thanh-Toan Do, Anh Nguyen, and Ian Reid. AffordanceNet: An End-to-End Deep Learning Approach for Object Affordance Detection. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5882–5889. IEEE, 2018.

[32] Anh Nguyen, Dimitrios Kanoulas, Darwin G Caldwell, and Nikos G Tsagarakis. Object-based affordances detection with Convolutional Neural Networks and dense Conditional Random Fields. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5908–5915. IEEE, 2017.

[33] Dan Song, Carl Henrik Ek, Kai Huebner, and Danica Kragic. Task-Based Robot Grasp Planning Using Probabilistic Inference. *IEEE Transactions on Robotics*, 31(3):546–561, 2015.

[34] Kuan Fang, Yuke Zhu, Animesh Garg, Andrey Kurenkov, Viraj Mehta, Li Fei-Fei, and Silvio Savarese. Learning task-oriented grasping for tool manipulation from simulated self-supervision. *The International Journal of Robotics Research*, 39(2-3):202–216, 2020.

[35] Nikolaus Vahrenkamp, Leonard Westkamp, Natsuki Yamanobe, Eren E Aksoy, and Tamim Asfour. Part-based grasp planning for familiar objects. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 919–925. IEEE, 2016.

[36] Safoura Rezapour Lakani, Antonio J Rodríguez-Sánchez, and Justus Piater. Towards affordance detection for robot manipulation using affordance for parts and parts for affordance. *Autonomous Robots*, 43(5):1155–1172, 2019.

[37] Weiyu Liu, Angel Daruna, and Sonia Chernova. CAGE: Context-Aware Grasping Engine. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2550–2556. IEEE, 2020.

[38] Ruinian Xu, Fu-Jen Chu, Chao Tang, Weiyu Liu, and Patricio A Vela. An Affordance Keypoint Detection Network for Robot Manipulation. *IEEE Robotics and Automation Letters*, 6(2):2870–2877, 2021.

[39] Martin Hjelm, Carl Henrik Ek, Renaud Detry, and Danica Kragic. Learning Human Priors for Task-Constrained Grasping. In *International Conference on Computer Vision Systems*, pages 207–217. Springer, 2015.

[40] Paola Ardón, Eric Pairet, Ronald PA Petrick, Subramanian Ramamoorthy, and Katrin S Lohan. Learning Grasp Affordance Reasoning through Semantic Relations. *IEEE Robotics and Automation Letters*, 4(4):4571–4578, 2019.

[41] Zhenyu Jiang, Yifeng Zhu, Maxwell Svetlik, Kuan Fang, and Yuke Zhu. Synergies Between Affordance and Geometry: 6-DoF Grasp Detection via Implicit Representations. *arXiv preprint arXiv:2104.01542*, 2021.

[42] Adithyavairavan Murali, Weiyu Liu, Kenneth Marino, Sonia Chernova, and Abhinav Gupta. Same Object, Different Grasps: Data and Semantic Knowledge for Task-Oriented Grasping. *arXiv preprint arXiv:2011.06431*, 2020.

[43] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. CIFAR10-DVS: An Event-Stream Dataset for Object Classification. *Frontiers in Neuroscience*, 11:309, 2017.

[44] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau,

Marcela Mendoza, et al. A Low Power, Fully Event-Based Gesture Recognition System. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7243–7252, 2017.

[45] Gianluca Scarpellini, Pietro Morerio, and Alessio Del Bue. Lifting Monocular Events to 3D Human Poses. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1358–1368, 2021.

[46] Lorenzo Berlincioni, Luca Cultrera, Chiara Albisani, Lisa Cresti, Andrea Leonardo, Sara Picchioni, Federico Becattini, and Alberto Del Bimbo. Neuromorphic Event-based Facial Expression Recognition. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4108–4118, 2023.

[47] Gregor Lenz, Sio-Hoi Ieng, and Ryad Benosman. Event-Based Face Detection and Tracking Using the Dynamics of Eye Blinks. *Frontiers in Neuroscience*, 14:587, 2020.

[48] Guang Chen, Hu Cao, Jorg Conradt, Huajin Tang, Florian Rohrbein, and Alois Knoll. Event-Based Neuromorphic Vision for Autonomous Driving: A Paradigm Shift for Bio-Inspired Visual Sensing and Perception. *IEEE Signal Processing Magazine*, 37(4):34–49, 2020.

[49] Gaurvi Goyal, Franco Di Pietro, Nicolo Carissimi, Arren Glover, and Chiara Bartolozzi. MoveEnet: Online High-Frequency Human Pose Estimation with an Event Camera. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4023–4032, 2023.

[50] Viktor Rudnev, Vladislav Golyanik, Jiayi Wang, Hans-Peter Seidel, Franziska Mueller, Mohamed Elgharib, and Christian Theobalt. EventHands: Real-Time Neural 3D Hand Pose Estimation from an Event Stream. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12365–12375, 2021.

[51] Guang Chen, Wenkai Chen, Qianyi Yang, Zhongcong Xu, Longyu Yang, Jörg Conradt, and Alois Knoll. A Novel Visible Light Positioning System With Event-Based Neuromorphic Vision Sensor. *IEEE Sensors Journal*, 20(17):10211–10219, 2020.

[52] Eric Hunsberger and Chris Eliasmith. Spiking Deep Networks with LIF Neurons. *arXiv preprint arXiv:1510.08829*, 2015.

[53] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating Learnable Membrane Time Constant to Enhance Learning of Spiking Neural Networks. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2661–2671, 2021.

[54] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. Attention Mechanisms for Object Recognition with Event-Based Cameras. In *2019*

IEEE Winter Conference on Applications of Computer Vision (WACV), pages 1127–1136. IEEE, 2019.

[55] Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li. Temporal-wise Attention Spiking Neural Networks for Event Streams Classification. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10221–10230, 2021.

[56] Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *arXiv preprint arXiv:2209.13929*, 2022.

[57] Juntao Jian, Xiuping Liu, Manyi Li, Ruizhen Hu, and Jian Liu. AffordPose: A Large-scale Dataset of Hand-Object Interactions with Affordance-driven Hand Pose. In *2023 Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14713–14724, 2023.

[58] Hu Cao, Guang Chen, Zhijun Li, Yingbai Hu, and Alois Knoll. NeuroGrasp: Multimodal Neural Network With Euler Region Regression for Neuromorphic Vision-Based Grasp Pose Estimation. *IEEE Transactions on Instrumentation and Measurement*, 71:1–11, 2022.

[59] Bin Li, Hu Cao, Zhongnan Qu, Yingbai Hu, Zhenke Wang, and Zichen Liang. Event-Based Robotic Grasping Detection With Neuromorphic Vision Sensor and Event-Grasping Dataset. *Frontiers in Neurorobotics*, 14:51, 2020.

[60] Hezhen Hu, Weilun Wang, Wengang Zhou, and Houqiang Li. Hand-Object Interaction Image Generation. *arXiv preprint arXiv:2211.15663*, 2022.

[61] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 409–419, 2018.

[62] Samarth Brahmbhatt, Cusuh Ham, Charles C Kemp, and James Hays. ContactDB: Analyzing and Predicting Grasp Contact via Thermal Imaging. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8709–8719, 2019.

[63] Samarth Brahmbhatt, Chengcheng Tang, Christopher D Twigg, Charles C Kemp, and James Hays. ContactPose: A Dataset of Grasps with Object Contact and Hand Pose. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 361–378. Springer, 2020.

[64] Ruolin Ye, Wenqiang Xu, Zhendong Xue, Tutian Tang, Yanfeng Wang, and Cewu Lu. H2O: A Benchmark for Visual Human-human Object Handover Analysis. In *2021 Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15762–15771, 2021.

[65] Hongzhuo Liang, Lin Cong, Norman Hendrich, Shuang Li, Fuchun Sun, and Jianwei Zhang. Multifingered Grasping Based on Multimodal Reinforcement Learning. *IEEE Robotics and Automation Letters*, 7(2):1174–1181, 2021.

[66] Bojan Nemec, Leon Žlajpah, and Aleš Ude. Door opening by joining reinforcement learning and intelligent control. In *2017 18th International Conference on Advanced Robotics (ICAR)*, pages 222–228. IEEE, 2017.

[67] Peter Englert and Marc Toussaint. Learning manipulation skills from a single demonstration. *The International Journal of Robotics Research*, 37(1):137–154, 2018.

[68] Serkan Cabi, Sergio Gómez Colmenarejo, Alexander Novikov, Ksenia Konyushkova, Scott Reed, Rae Jeong, Konrad Zolna, Yusuf Aytar, David Budden, Mel Vecerik, et al. Scaling data-driven robotics with reward sketching and batch reinforcement learning. *arXiv preprint arXiv:1909.12200*, 2019.

[69] Xiapeng Wu, Dapeng Zhang, Fangbo Qin, and De Xu. Deep reinforcement learning of robotic precision insertion skill accelerated by demonstrations. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pages 1651–1656. IEEE, 2019.

[70] Tadanobu Inoue, Giovanni De Magistris, Asim Munawar, Tsuyoshi Yokoya, and Ryuki Tachibana. Deep Reinforcement Learning for High Precision Assembly Tasks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 819–825. IEEE, 2017.

[71] Gerrit Schoettler, Ashvin Nair, Jianlan Luo, Shikhar Bahl, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Deep Reinforcement Learning for Industrial Insertion Tasks with Visual Inputs and Natural Rewards. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5548–5555. IEEE, 2020.

[72] Wenzhou Lv, Tianyu Wu, Luolin Xiong, Liang Wu, Jian Zhou, Yang Tang, and Feng Qi. Hybrid Control Policy for Artificial Pancreas via Ensemble Deep Reinforcement Learning. *arXiv preprint arXiv:2307.06501*, 2023.

[73] Nan Lin, Linrui Zhang, Yuxuan Chen, Yujun Zhu, Ruoxi Chen, Peichen Wu, and Xiaoping Chen. Reinforcement learning for robotic safe control with force sensing. In *2019 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*, pages 148–153. IEEE, 2019.

[74] Tie Zhang, Meng Xiao, Yanbiao Zou, and Jiadong Xiao. Robotic constant-force grinding control with a press-and-release model and model-based reinforcement learning. *The International Journal of Advanced Manufacturing Technology*, 106(1):589–602, 2020.

[75] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.

[76] Cristian C Beltran-Hernandez, Damien Petit, Ixchel G Ramirez-Alpizar, and Kensuke Harada. Variable Compliance Control for Robotic Peg-in-Hole Assembly: A Deep-Reinforcement-Learning Approach. *Applied Sciences*, 10(19):6923, 2020.

[77] Marius Hebecker, Jens Lambrecht, and Markus Schmitz. Towards real-world force-sensitive robotic assembly through deep reinforcement learning in simulations. In *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1045–1051. IEEE, 2021.

[78] Garrett Thomas, Melissa Chien, Aviv Tamar, Juan Aparicio Ojea, and Pieter Abbeel. Learning Robotic Assembly from CAD. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3524–3531. IEEE, 2018.

[79] Zheng Wu, Wenzhao Lian, Vaibhav Unhelkar, Masayoshi Tomizuka, and Stefan Schaal. Learning Dense Rewards for Contact-Rich Manipulation Tasks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6214–6221. IEEE, 2021.

[80] Michelle A Lee, Yuke Zhu, Peter Zachares, Matthew Tan, Krishnan Srinivasan, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making Sense of Vision and Touch: Learning Multimodal Representations for Contact-Rich Tasks. *IEEE Transactions on Robotics*, 36(3):582–596, 2020.

[81] Chao Zeng, Yanan Li, Jing Guo, Zhifeng Huang, Ning Wang, and Chenguang Yang. A Unified Parametric Representation for Robotic Compliant Skills With Adaptation of Impedance and Force. *IEEE/ASME Transactions on Mechatronics*, 27(2):623–633, 2021.

[82] Chao Zeng, Shuang Li, Zhaopeng Chen, Chenguang Yang, Fuchun Sun, and Jianwei Zhang. Multifingered Robot Hand Compliant Manipulation Based on Vision-based Demonstration and Adaptive Force Control. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[83] Chenguang Yang, Guangzhu Peng, Yanan Li, Rongxin Cui, Long Cheng, and Zhijun Li. Neural Networks Enhanced Adaptive Admittance Control of Optimized Robot–environment Interaction. *IEEE Transactions on Cybernetics*, 49(7):2568–2579, 2018.

[84] Linghuan Kong, Wei He, Chenguang Yang, Zhijun Li, and Changyin Sun. Adaptive Fuzzy Control for Coordinated Multiple Robots With Constraint Using Impedance Learning. *IEEE Transactions on Cybernetics*, 49(8):3052–3063, 2019.

[85] Xinbo Yu, Wei He, Yanan Li, Chengqian Xue, Jianqiang Li, Jianxiao Zou, and Chenguang Yang. Bayesian Estimation of Human Impedance and Motion Intention for Human–Robot Collaboration. *IEEE Transactions on Cybernetics*, 51(4):1822–1834, 2019.

[86] Yiming Jiang, Yaonan Wang, Zhiqiang Miao, Jing Na, Zhijia Zhao, and Chenguang Yang. Composite-Learning-Based Adaptive Neural Control for Dual-Arm Robots With Relative Motion. *IEEE Transactions on Neural Networks and Learning Systems*, 33(3):1010–1021, 2020.

[87] Xinbo Yu, Bin Li, Wei He, Yanghe Feng, Long Cheng, and Carlos Silvestre. Adaptive-constrained Impedance Control for Human–robot Co-transportation. *IEEE Transactions on Cybernetics*, 52(12):13237–13249, 2021.

[88] Jianlan Luo, Eugen Solowjow, Chengtao Wen, Juan Aparicio Ojea, Alice M Agogino, Aviv Tamar, and Pieter Abbeel. Reinforcement Learning on Variable Impedance Controller for High-Precision Robotic Assembly. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3080–3087.

[89] Miroslav Bogdanovic, Majid Khadiv, and Ludovic Righetti. Learning Variable Impedance Control for Contact Sensitive Tasks. *IEEE Robotics and Automation Letters*, 5(4):6129–6136, 2020.

[90] Padmaja Kulkarni, Jens Kober, Robert Babuška, and Cosimo Della Santina. Learning Assembly Tasks in a Few Minutes by Combining Impedance Control and Residual Recurrent Reinforcement Learning. *Advanced Intelligent Systems*, 4(1):2100095, 2022.

[91] Quantao Yang, Alexander Dürr, Elin Anna Topp, Johannes A Stork, and Todor Stoyanov. Variable Impedance Skill Learning for Contact-Rich Manipulation. *IEEE Robotics and Automation Letters*, 7(3):8391–8398, 2022.

[92] Oren Spector and Miriam Zacksenhouse. Learning Contact-Rich Assembly Skills Using Residual Admittance Policy. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6023–6030. IEEE, 2021.

[93] Korntham Sathirakul and Robert H Sturges. Jamming conditions for multiple peg-in-hole assemblies. *Robotica*, 16(3):329–345, 1998.

[94] Michael E Came, Tomás Lozano-Pérez, and Warren P Seering. Assembly strategies for chamferless parts. In *1989 Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 472–477, 1989.

[95] Yanqiong Fei and Xifang Zhao. An Assembly Process Modeling and Analysis for Robotic Multiple Peg-in-hole. *Journal of Intelligent and Robotic Systems*, 36(2):175–189, 2003.

[96] Zhimin Hou, Haiming Dong, Kuangen Zhang, Quan Gao, Ken Chen, and Jing Xu. Knowledge-driven deep deterministic policy gradient for robotic multiple peg-in-hole assembly tasks. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 256–261. IEEE, 2018.

[97] Jing Xu, Zhimin Hou, Wei Wang, Bohao Xu, Kuangen Zhang, and Ken Chen. Feedback deep deterministic policy gradient with fuzzy reward for robotic multiple peg-in-hole assembly tasks. *IEEE Transactions on Industrial Informatics*, 15(3):1658–1667, 2018.

[98] Xinwang Li, Juliang Xiao, Wei Zhao, Haitao Liu, and Guodong Wang. Multiple peg-in-hole compliant assembly based on a learning-accelerated deep deterministic policy gradient strategy. *Industrial Robot: the International Journal of Robotics Research and Application*, 2021.

[99] Zhimin Hou, Zhihu Li, Chenwei Hsu, Kuangen Zhang, and Jing Xu. Fuzzy Logic-driven Variable Time-scale Prediction-based Reinforcement Learning for Robotic Multiple Peg-in-hole Assembly. *IEEE Transactions on Automation Science and Engineering*, 2020.

[100] Stefanie Tellex, Nakul Gopalan, Hadas Kress-Gazit, and Cynthia Matuszek. Robots That Use Language. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:25–55, 2020.

[101] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[102] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A Benchmark for Language-Conditioned Policy Learning for Long-Horizon Robot Manipulation Tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

[103] Oier Mees, Jessica Borja-Diaz, and Wolfram Burgard. Grounding Language with Visual Affordances over Unstructured Data. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11576–11582. IEEE, 2023.

[104] Yiding Jiang, Shixiang Shane Gu, Kevin P Murphy, and Chelsea Finn. Language as an Abstraction for Hierarchical Deep Reinforcement Learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[105] Dipendra Misra, John Langford, and Yoav Artzi. Mapping Instructions and Visual Observations to Actions with Reinforcement Learning. *arXiv preprint arXiv:1704.08795*, 2017.

[106] Corey Lynch and Pierre Sermanet. Language Conditioned Imitation Learning over Unstructured Data. *arXiv preprint arXiv:2005.07648*, 2020.

[107] Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-Conditioned Imitation Learning for Robot Manipulation Tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.

[108] Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots That Ask For Help: Uncertainty Alignment for Large Language Model Planners. *arXiv preprint arXiv:2307.01928*, 2023.

[109] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents. In *Proceedings of the 39th International Conference on Machine Learning*, pages 9118–9147. PMLR, 2022.

[110] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *Conference on Robot Learning*, pages 287–318. PMLR, 2023.

[111] Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, et al. Socratic Models: Composing Zero-Shot Multimodal Reasoning with Language. *arXiv preprint arXiv:2204.00598*, 2022.

[112] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as Policies: Language Model Programs for Embodied Control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023.

[113] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models. *arXiv preprint arXiv:2307.05973*, 2023.

[114] Alexander Quinn Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[115] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*, 2020.

[116] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022.

[117] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen Video: High Definition Video Generation with Diffusion Models. *arXiv preprint arXiv:2210.02303*, 2022.

[118] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[119] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[120] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. DiffEdit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.

[121] Chaerin Kong, DongHyeon Jeon, Ohjoon Kwon, and Nojun Kwak. Leveraging Off-the-shelf Diffusion Model for Multi-attribute Fashion Image Manipulation. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 848–857, 2023.

[122] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended Diffusion for Text-driven Editing of Natural Images. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18208–18218, 2022.

[123] Ivan Kapelyukh, Vitalis Vosylius, and Edward Johns. DALL-E-Bot: Introducing Web-Scale Diffusion Models to Robotics. *IEEE Robotics and Automation Letters*, 2023.

[124] Weiyu Liu, Chris Paxton, Tucker Hermans, and Dieter Fox. StructFormer: Learning Spatial Structure for Language-Guided Semantic Rearrangement of Novel Objects. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6322–6329. IEEE, 2022.

[125] L Tang, H Liu, H Huang, XR Xie, NL Liu, and M Li. A reinforcement learning method for rearranging scattered irregular objects inside a crate. *IEEE Transactions on Cognitive and Developmental Systems*, 2022.

[126] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: General Robot Manipulation with Multimodal Prompts. *arXiv preprint arXiv:2210.03094*, 2022.

[127] Weiyu Liu, Tucker Hermans, Sonia Chernova, and Chris Paxton. StructDiffusion: Object-Centric Diffusion for Semantic Rearrangement of Novel Objects. *arXiv preprint arXiv:2211.04604*, 2022.

[128] Jacob Varley, Jonathan Weisz, Jared Weiss, and Peter Allen. Generating multi-fingered robotic grasps via deep learning. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4415–4420. IEEE, 2015.

[129] Marcus Gualtieri and Robert Platt. Robotic Pick-and-Place With Uncertain Object Instance Segmentation and Shape Completion. *arXiv preprint arXiv:2101.11605*, 2021.

[130] David Watkins-Valls, Jacob Varley, and Peter Allen. Multi-Modal Geometric Learning for Grasping and Manipulation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7339–7345. IEEE, 2019.

[131] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck Transformers for Visual Recognition. In *2021 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16514–16524, 2021.

[132] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. PCT: Point cloud transformer. *Computational Visual Media*, pages 187–199, 2021.

[133] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3D shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016.

[134] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015.

[135] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[136] Hongwu Kuang, Bei Wang, Jianping An, Ming Zhang, and Zehan Zhang. Voxel-FPN: Multi-Scale Voxel Feature Aggregation for 3D Object Detection from LIDAR Point Clouds. *Sensors*, 20(3):704, 2020.

[137] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning Representations and Generative Models for 3D Point Clouds. In *Proceedings of the 35th International Conference on Machine Learning*, pages 40–49. PMLR, 2018.

[138] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. PF-Net: Point Fractal Network for 3D Point Cloud Completion. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7662–7670, 2020.

[139] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[140] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 605–613, 2017.

[141] Andreas ten Pas and Robert Platt. Using Geometry to Detect Grasp Poses in 3D Point Clouds. In *Robotics Research*, pages 307–324. Springer International Publishing, 2018.

[142] Michael Görner, Robert Haschke, Helge Ritter, and Jianwei Zhang. Moveit! Task Constructor for Task-level Motion Planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 190–196, 2019.

[143] Herbert Edelsbrunner, David Kirkpatrick, and Raimund Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, 1983.

[144] Jeannette Bohg, Matthew Johnson-Roberson, Beatriz León, Javier Felip, Xavi Gratal, Niklas Bergström, Danica Kragic, and Antonio Morales. Mind the gap - robotic grasping under incomplete observation. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 686–693. IEEE, 2011.

[145] Robert Haschke, Jochen J Steil, Ingo Steuwer, and Helge Ritter. Task-oriented quality measures for dextrous grasping. In *2005 International Symposium on Computational Intelligence in Robotics and Automation*, pages 689–694. IEEE, 2005.

[146] Zexiang Li and S Shankar Sastry. Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal on Robotics and Automation*, 4(1):32–44, 1988.

[147] Tanis Mar, Vadim Tikhanoff, Giorgio Metta, and Lorenzo Natale. Self-supervised learning of grasp dependent tool affordances on the iCub Humanoid robot. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3200–3206. IEEE, 2015.

[148] Alexander Stoytchev. Behavior-Grounded Representation of Tool Affordances. In *2005 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3060–3065. IEEE, 2005.

[149] Fu-Jen Chu, Ruinian Xu, and Patricio A Vela. Learning Affordance Segmentation for Real-World Robotic Manipulation via Synthetic Images. *IEEE Robotics and Automation Letters*, 4(2):1140–1147, 2019.

[150] Johann Sawatzky and Jurgen Gall. Adaptive Binarization for Weakly Supervised Affordance Segmentation. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 1383–1391, 2017.

[151] Fu-Jen Chu, Ruinian Xu, Landan Seguin, and Patricio A Vela. Toward Affordance Detection and Ranking on Novel Objects for Real-World Robotic Manipulation. *IEEE Robotics and Automation Letters*, 4(4):4070–4077, 2019.

[152] Kun Qian, Xingshuo Jing, Yanhui Duan, Bo Zhou, Fang Fang, Jing Xia, and Xudong Ma. Grasp Pose Detection with Affordance-based Task Constraint Learning in Single-view Point Clouds. *Journal of Intelligent & Robotic Systems*, 100(1):145–163, 2020.

[153] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. ACRONYM: A Large-Scale Grasp Dataset Based on Simulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6222–6227. IEEE, 2021.

[154] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An Information-Rich 3D Model Repository. *arXiv preprint arXiv:1512.03012*, 2015.

[155] Aditya Grover, Manik Dhar, and Stefano Ermon. Flow-GAN: Combining Maximum Likelihood and Adversarial Learning in Generative Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[156] Ke Li and Jitendra Malik. Implicit Maximum Likelihood Estimation. *arXiv preprint arXiv:1809.09087*, 2018.

[157] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, 27, 2014.

[158] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in Neural Information Processing Systems*, 30, 2017.

[159] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic Graph CNN for Learning on Point Clouds. *Acm Transactions On Graphics (TOG)*, 38(5):1–12, 2019.

[160] Lei Deng, Yujie Wu, Xing Hu, Ling Liang, Yufei Ding, Guoqi Li, Guangshe Zhao, Peng Li, and Yuan Xie. Rethinking the Performance Comparison Between SNNS and ANNS. *Neural Networks*, 121:294–307, 2020.

[161] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11062–11070, 2021.

[162] Xiang Cheng, Yunzhe Hao, Jiaming Xu, and Bo Xu. LISNN: Improving Spiking Neural Networks with Lateral Interactions for Robust Object Recognition. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 1519–1525, 2020.

[163] José Antonio Pérez-Carrasco, Bo Zhao, Carmen Serrano, Begona Acha, Teresa Serrano-Gotarredona, Shouchun Chen, and Bernabé Linares-Barranco. Mapping from Frame-Driven to Frame-Free Event-Driven Vision Systems by Low-Rate Rate Coding and Coincidence Processing–Application to Feedforward ConvNets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2706–2719, 2013.

[164] Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

[165] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-Excitation Networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7132–7141, 2018.

[166] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual Attention Network for Image Classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2017.

[167] Weihua He, YuJie Wu, Lei Deng, Guoqi Li, Haoyu Wang, Yang Tian, Wei Ding, Wenhui Wang, and Yuan Xie. Comparing snns and rnns on neuromorphic vision datasets: Similarities and differences. *Neural Networks*, 132:108–120, 2020.

[168] Yangfan Hu, Huajin Tang, and Gang Pan. Spiking Deep Residual Network. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–6, 2021.

[169] Simone Undri Innocenti, Federico Becattini, Federico Pernici, and Alberto Del Bimbo. Temporal Binary Representation for Event-Based Action Recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10426–10432. IEEE, 2021.

[170] Anh Nguyen, Thanh-Toan Do, Darwin G Caldwell, and Nikos G Tsagarakis. Real-Time 6DOF Pose Relocalization for Event Cameras With Stacked Spatial LSTM Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 0–0, 2019.

[171] Elias Mueggler, Chiara Bartolozzi, and Davide Scaramuzza. Fast event-based corner detection. 2017.

[172] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[173] Christoph Feichtenhofer. X3D: Expanding Architectures for Efficient Video Recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 203–213, 2020.

[174] Janis Arents and Modris Greitans. Smart Industrial Robot Control Trends, Challenges and Opportunities within Manufacturing. *Applied Sciences*, 12(2):937, 2022.

[175] Hong Qiao, Jiahao Chen, and Xiao Huang. A survey of brain-inspired intelligent robots: Integration of vision, decision, motion control, and musculoskeletal systems. *IEEE Transactions on Cybernetics*, 52(10):11267–11280, 2021.

[176] Íñigo Elguea-Aguinaco, Antonio Serrano-Muñoz, Dimitrios Chrysostomou, Ibai Inziarte-Hidalgo, Simon Bøgh, and Nestor Arana-Arexolaleiba. A review on reinforcement learning for contact-rich robotic manipulation tasks. *Robotics and Computer-Integrated Manufacturing*, 81:102517, 2023.

[177] K.P. Valavanis and K.M. Stellakis. A General Organizer Model for Robotic Assemblies and Intelligent Robotic Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(2):302–317, 1991.

[178] Chi-Haur Wu and Myong Gi Kim. Modeling of Part-mating Strategies for Automating Assembly Operations for Robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(7):1065–1074, 1994.

[179] Heping Chen, George Zhang, Hui Zhang, and Thomas A Fuhlbrigge. Integrated robotic system for high precision assembly in a semi-structured environment. *Assembly Automation*, 2007.

[180] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[181] Fengming Li, Qi Jiang, Sisi Zhang, Meng Wei, and Rui Song. Robot skill acquisition in assembly process using deep reinforcement learning. *Neurocomputing*, 345:92–102, 2019.

[182] Zhiming Zheng, Tan Li, Bohu Li, Xudong Chai, Weining Song, Nanjiang Chen, Yuqi Zhou, Yanwen Lin, and Runqiang Li. Industrial Metaverse: Connotation, Features, Technologies, Applications and Challenges. In *Methods and Applications for Modeling and Simulation of Complex Systems: 21st Asia Simulation Conference*, pages 239–263. Springer, 2022.

[183] Aziz Siyaev and Geun-Sik Jo. Neuro-Symbolic Speech Understanding in Aircraft Maintenance Metaverse. *IEEE Access*, 9:154484–154499, 2021.

[184] Saeed Banaeian Far and Azadeh Imani Rad. Applying Digital Twins in Metaverse: User Interface, Security and Privacy Challenges. *Journal of Metaverse*, 2(1):8–16, 2022.

[185] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a

Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.

[186] Yiming Jiang, Chenguang Yang, Yaonan Wang, Zhaojie Ju, Yanan Li, and Chun-Yi Su. Multi-hierarchy interaction control of a redundant robot using impedance learning. *Mechatronics*, 67:102348, 2020.

[187] Peng Song, Yueqing Yu, and Xuping Zhang. A Tutorial Survey and Comparison of Impedance Control on Robotic Manipulation. *Robotica*, 37(5):801–836, 2019.

[188] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018.

[189] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929*, 2020.

[190] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver IO: A General Architecture for Structured Inputs & Outputs. *arXiv preprint arXiv:2107.14795*, 2021.

[191] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A Generalist Agent. *arXiv preprint arXiv:2205.06175*, 2022.

[192] Jun Nakanishi, Rick Cory, Michael Mistry, Jan Peters, and Stefan Schaal. Operational Space Control: A Theoretical and Empirical Comparison. *The International Journal of Robotics Research*, 27(6):737–757, 2008.

[193] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A Modular Simulation Framework and Benchmark for Robot Learning. *arXiv preprint arXiv:2009.12293*, 2020.

[194] Julius Hietala, David Blanco-Mulero, Gokhan Alcan, and Ville Kyrki. Learning Visual Feedback Control for Dynamic Cloth Folding. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1455–1462. IEEE, 2022.

[195] Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, et al. Rearrangement: A Challenge for Embodied AI,. *arXiv preprint arXiv:2011.01975*, 2020.

[196] Alex Irpan, Alexander Herzog, Alexander Toshkov Toshev, Andy Zeng, Anthony Brohan, Brian Andrew Ichter, Byron David, Carolina Parada, Chelsea Finn, Clayton Tan, et al. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *Conference on Robot Learning*, number 2022, 2022.

[197] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. PaLM-E: An Embodied Multimodal Language Model. *arXiv preprint arXiv:2303.03378*, 2023.

[198] Guangyao Zhai, Xiaoni Cai, Dianye Huang, Yan Di, Fabian Manhardt, Federico Tombari, Nassir Navab, and Benjamin Busam. SG-Bot: Object Rearrangement via Coarse-to-Fine Robotic Imagination on Scene Graphs. *arXiv preprint arXiv:2309.12188*, 2023.

[199] Qiuhong Anna Wei, Sijie Ding, Jeong Joon Park, Rahul Sajnani, Adrien Poulenard, Srinath Sridhar, and Leonidas Guibas. LEGO-Net: Learning Regular Rearrangements of Objects in Rooms. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19037–19047, 2023.

[200] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. DiffuScene: Denoising Diffusion Models for Generative Indoor Scene Synthesis. *arXiv preprint arXiv:2303.14207*, 2023.

[201] Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. In *Advances in Neural Information Processing Systems, NeurIPS*, pages 11895–11907, 2019.

[202] Pascal Vincent. A Connection Between Score Matching and Denoising Autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

[203] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks, 2024.

[204] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computational Vision*, 40(2):99–121, 2000.

[205] Yixiang Jin, Dingzhe Li, A Yong, Jun Shi, Peng Hao, Fuchun Sun, Jianwei Zhang, and Bin Fang. RobotGPT: Robot Manipulation Learning from ChatGPT. *IEEE Robotics and Automation Letters*, 2024.

[206] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. GELLO: A General, Low-Cost, and Intuitive Teleoperation Framework for Robot Manipulators. *arXiv preprint arXiv:2309.13037*, 2023.

[207] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile ALOHA: Learning Bimanual Mobile Manipulation with Low-Cost Whole-Body Teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

Erklärung der Urheberschaft

Hiermit versichere ich an Eides statt, dass ich die vorliegende Doctoral thesis im Studiengang Informatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel - insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Ort, Datum Unterschrift

Erklärung zur Veröffentlichung

Ich stimme der Einstellung der Doctoral thesis in die Bibliothek des Fachbereichs Informatik zu.

Ort, Datum Unterschrift