Objects and Actions
Learning Representations for Open-World Robotics

Wentao Yuan

A dissertation
submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington
August, 2024

Reading Committee:

Dieter Fox, Chair

Luke Zettlemoyer

Abhishek Gupta

Program Authorized to Offer Degree:

Computer Science and Engineering

University of Washington

Abstract

Objects and Actions
Learning Representations for Open-World Robotics

Wentao Yuan

Chair of the Supervisory Committee:
Dieter Fox
Computer Science and Engineering

Advancing robotics involves enabling systems to generalize across diverse and unseen environments, known as "the open world." Traditional approaches rely on state estimators, while modern learning-based methods develop implicit representations to approximate states. Both approaches require well-designed states or representations for effective generalization. This dissertation investigates learning representations that enhance generalization in robotic systems, focusing on objects and actions.

First, I introduce SORNet (Spatial Object-Centric Representation Network), a framework for learning object-centric representations from RGB images using canonical object views. SORNet generalizes to unseen objects with different shapes and textures, outperforming existing techniques in tasks like spatial relation classification and task planning for sequential manipulation.

Next, I present M2T2, a transformer model that predicts low-level actions for manipulating objects in cluttered scenes. M2T2 reasons about contact points and gripper poses from raw point clouds. Trained on a large-scale synthetic dataset, M2T2 achieves zero-shot sim2real transfer on real robots, surpassing state-of-the-art models in both overall performance and in challenging tasks requiring object re-orientation.

Finally, I introduce RoboPoint, a vision-language model that predicts keypoint affordances from language instructions. Using a synthetic data generation pipeline, RoboPoint trains without real-world data collection or human demonstration. It supports applications such as robot navigation, manipulation, and augmented reality, and outperforms existing models in spatial affordance prediction and task success rates.

The dissertation concludes with a discussion on challenges and future directions for developing foundational models in robotics, aiming to create versatile systems capable of operating in open-world environments.

Acknowledgements

My PhD journey at the University of Washington has been nothing short of extraordinary, a path paved with challenges, discoveries, and unforgettable moments in the enchanting city of Seattle. Over these five years, I have had the privilege of meeting and working alongside remarkable individuals—mentors, colleagues, and friends—whose presence made this journey not only possible but profoundly meaningful.

First and foremost, I extend my deepest gratitude to my advisor, Prof. Dieter Fox. His unwavering guidance and support have been the cornerstone of my PhD experience. Our countless conversations about robotics, learning, and the very essence of research have left an indelible mark on my intellectual pursuit. His insights and approach to inquiry have shaped the way I see the world and will continue to inspire me long into the future.

My heartfelt thanks also go to the brilliant minds at NVIDIA's Seattle Robotics Lab. I am particularly grateful to my mentors—Chris Paxton, Adithya Murali, Arsalan Mousavian, and Valts Blukis—whose dedication and camaraderie were my anchors, even during those intense late-night crunches before conference deadlines. I am equally indebted to Clemens Eppner, Wei Yang, Caelan Garrett, Fabio Ramos, Kaichun Mo, Ankur Handa, and so many others who shared their wisdom and insights, enriching my work and broadening my horizons.

I was fortunate to be surrounded by exceptional labmates who made this journey as collaborative as it was rewarding. Jiafei Duan, with whom I toiled to deploy models on real robots, brought a spark of creativity to every challenge we faced. Karthik Desingh, now a professor, shared in the triumph of turning an almost rejected paper into a Best Systems Paper finalist—a testament to perseverance and the power of collaboration. To Xiangyun Meng, Adam Fishman, Yi Li, Helen Wang, Marius Memmel, Zoey Chen, Aaron Walsman, Mohit Shridhar, Junha Roh, Chris Xie, Daniel Gordon—thank you for your friendship, your intellect, and the joy you brought to our shared journey.

I also had the privilege of interning at Meta Reality Labs Research, where I worked with the extraordinary Tanner Schmidt and Zhaoyang Lv on cutting-edge neural rendering research. Their mentorship and the experience I gained there were invaluable.

A special thanks to my committee members, Abhishek Gupta and Luke Zettlemoyer, whose thoughtful feedback and insights were crucial in refining my thesis.

Lastly, I owe everything to my family. To my parents, who nurtured my dreams and stood by me with unwavering support—you are my foundation, my constant strength. This achievement is as much yours as it is mine.

This journey has been more than an academic pursuit; it has been a chapter of growth, discovery, and connection. As I look back, I am filled with gratitude and pride, not only for what has been achieved but for the people who made it all possible.

Now, let the adventure continue.

Contents

Chapter 1

Introduction

Robotics has seen significant advancements in recent years, with improvements in both hardware and software. However, a critical challenge remains: enabling robots to operate effectively in open-world environments—settings that are unstructured, dynamic, and inherently unpredictable. This dissertation introduces a few efforts towards open-world robotics by developing appropriate *representations* via learning-based methods. More specifically, we focus on representations for *objects* and *actions*, which are fundamental concepts in robotics. These representations are key to enhancing the generalization capabilities of robotic systems in such environments.

Objects and actions are central to robotic operations. A robot's ability to perceive and manipulate objects effectively, and to plan and execute actions, underpins its capacity to function autonomously in a wide range of scenarios. Traditional robotics has relied heavily on explicit state estimators, where understanding is crafted through predefined models and rules. While these methods are effective in controlled environments, they struggle to adapt to the variability and complexity of open-world scenarios. Modern learning-based approaches, on the other hand, aim to develop implicit representations that approximate the state of the environment through data-driven learning processes. By focusing on learning robust representations for objects and actions, we can significantly improve a robot's ability to generalize its behavior and adapt to new, unseen situations.

This dissertation introduces three key contributions—SORNet, M2T2, and RoboPoint—each representing an advancement in how robots perceive and interact with

the world. SORNet is centered on learning object representations, while M2T2 and RoboPoint focus on learning representations for actions. Together, these models push the boundaries of what robots can achieve in open-world environments by enhancing their ability to generalize across different contexts.

1.1 Background

In the field of robotics, enabling robots to perform complex tasks in unstructured environments requires effective methods for representing and reasoning about both objects and actions. This background section explores various approaches in the literature, highlighting the evolution from classical, model-based methods to more recent learning-based approaches that aim to generalize across diverse scenarios.

1.1.1 Evolution of Manipulation Approaches

Traditionally, model-based sequential manipulation [4, 24, 26, 66, 75, 83] has been the cornerstone of robotic systems. In these approaches, robots rely on a model-based state estimator to generate explicit object states—such as bounding boxes or 6D poses—from sensor observations. These states are then used by task and motion planners to produce action sequences that lead to a desired goal. However, the primary limitation of these methods is their reliance on predefined object models, which restricts their ability to handle a wide range of objects, particularly in open-world environments.

In contrast, end-to-end (model-free) manipulation methods [19, 27, 33, 42, 65, 98] bypass the need for explicit object state estimation by learning motor controls directly from raw sensor data. These approaches eliminate dependence on object models, making them more flexible in some contexts. However, they also suffer from significant drawbacks, such as the lack of object-awareness, which limits their ability to reason about tasks involving multiple objects or requiring long-horizon planning.

1.1.2 Advancements in Spatial and Visual Reasoning

The ability to reason about spatial relations between objects is crucial for complex manipulation tasks. In 3D vision, several methods [23, 74, 77] have been proposed to

predict spatial relations from 3D inputs like point clouds or voxels, often assuming complete observation and segmented objects. While effective, these approaches typically require significant preprocessing and are limited by their assumptions about object observability.

In robotics, learning frameworks have been developed to classify spatial relations from sequences of sensor observations, which are then used in planning [41, 54]. Despite these advancements, most of these methods remain constrained by the specific types and numbers of objects they can handle. Recent approaches in visual reasoning have shown promise in addressing these limitations by leveraging transformer networks [90] and object-based attention mechanisms [15, 103]. These models have been applied to spatio-temporal reasoning tasks, often using segmentation models to produce object segments. However, they still rely on certain assumptions that limit their applicability in more complex, real-world scenarios.

SORNet (Spatial Object-Centric Representation Network) [100], introduced in this dissertation, builds on these ideas by learning object-centric representations directly from RGB images without requiring segmentation or object detection modules. This approach enables zero-shot generalization to unseen objects, which is critical for performing spatial reasoning in dynamic environments.

1.1.3 Multi-Task Learning and Action Modes

In robotics, multi-task learning has emerged as a powerful approach to improve sample efficiency and performance across diverse tasks. By training a single model to handle multiple tasks [37], researchers have shown that it is possible to learn common representations that generalize better to new tasks [67]. However, end-to-end language-conditioned policies [5, 38, 79], which have been popular in recent multi-task learning approaches, often struggle with generalizing to out-of-distribution tasks and objects.

M2T2 (Multi-Task Transformer Model) [101], presented in this dissertation, addresses this challenge by providing a common framework for various manipulation skills, such as grasping and object placement. Unlike task-specific models, M2T2 supplies action primitives that work robustly on unseen objects in real-world scenarios, making it a key component of a flexible open-world manipulation system.

Grasping and placement are two fundamental action modes for robotic manipulators. While object grasping has been extensively studied, with many methods now focusing on learning 6-DoF grasp poses from 3D point clouds [22, 59, 60, 85], these approaches typically target single skills. M2T2 extends this capability by incorporating grasping into a broader set of manipulation skills, including object placement, which is less studied but equally important. By predicting placement poses that consider both the object and the gripper, M2T2 advances the state of the art in multi-task learning for robotics.

1.1.4 Affordance Prediction and Vision-Language Models

Affordance prediction—the ability to predict the functions of an object based on its appearance—ties visual observations to potential actions, making it a critical component of robotic manipulation. Various methods have been developed to predict affordances, ranging from part segmentation [16, 35, 69] to keypoint prediction [52, 57, 70]. In this dissertation, the focus is on using a 2D keypoint representation, which can be readily converted into language format, making it compatible with vision-language models.

Recent advancements in zero-shot language models for robotics have shown that language models can be effective planners for robotic tasks [2, 45, 81], particularly when combined with visual reasoning capabilities. However, these models often rely on predefined action primitives or external models for detecting relevant objects, which limits their flexibility. RoboPoint [102], another key contribution of this dissertation, overcomes this limitation by directly predicting keypoint affordances from language instructions. This approach allows for more fine-grained action predictions and greater scalability, making it a valuable addition to the toolkit for open-world robotics.

1.2 Challenges

Despite recent advancements, several challenges remain in the field of robotics, particularly when it comes to generalization and scalability in open-world environments:

Generalization Beyond Training Data: Current end-to-end learning approaches, including behavior cloning, have shown limited success in generalizing beyond their training data distribution. These models often excel in controlled environments but struggle to adapt to new, unseen scenarios, particularly when encountering objects or tasks outside of their training experience.

Object Representation: Traditional model-based approaches rely on explicit object models, which restrict their ability to handle a diverse range of objects. Meanwhile, model-free approaches often lack object-awareness, making it difficult for robots to reason about tasks involving multiple objects or requiring long-term planning.

Action Representation: The ability to generalize action representations across different tasks and environments remains a significant challenge. Current methods often focus on specific manipulation skills, such as grasping, but struggle to extend these capabilities to more complex tasks involving object placement or multi-step manipulation.

Scalability and Flexibility: The reliance on predefined models or action primitives limits the scalability of current approaches. Additionally, the need for extensive training data and expert demonstrations presents significant challenges for deploying these models in real-world scenarios.

1.3 Our Approach and Outline

To address the challenges of generalization in open-world environments, this dissertation presents several approaches that explore the design space for learning-based object and action representations. The focus is on enhancing generalization and enabling robust performance across diverse and unseen scenarios. Below is an outline of the document.

Chapter 2 introduces SORNet (Spatial Object-Centric Representation Network), a novel framework for learning object-centric representations directly from RGB images. Unlike traditional methods that rely on segmentation or object detection,

SORNet enables zero-shot generalization to unseen objects by learning from simple image patches representing objects. This capability is critical for tasks requiring spatial reasoning and sequential manipulation, where understanding the spatial configuration of objects is essential for successful execution. SORNet's ability to generalize to novel objects makes it a powerful tool for complex robotic manipulation tasks.

Chapter 3 presents M2T2 (Multi-Task Masked Transformer), a transformer model designed to address the challenge of action representation. M2T2 provides a unified framework for various manipulation skills, including grasping and object placement, by reasoning about contact points and gripper poses from raw point clouds. This transformer-based model generalizes across different action modes and demonstrates strong zero-shot transfer to real-world scenarios. M2T2 is a key component in building a flexible open-world manipulation system capable of handling diverse tasks with high precision and reliability.

Chapter 4 introduces RoboPoint, a vision-language model (VLM) that predicts keypoint affordances from language instructions, addressing the limitations of current VLMs that rely on predefined action primitives. By directly predicting action points based on language inputs, RoboPoint enables more fine-grained and scalable action predictions. This capability is crucial for complex tasks such as object rearrangement and navigation, where precise and context-aware actions are required. RoboPoint's integration of real-world VQA data with synthetic datasets allows it to generate accurate and flexible action points, demonstrating its potential for broader applications in robotics and beyond.

Chapter 5 concludes this dissertation by highlighting the collective advancements made through SORNet, M2T2, and RoboPoint towards developing autonomous robotic systems capable of operating in open-world environments. However, the journey towards creating a GPT-like foundation model for robotics that functions seamlessly across diverse environments continues. Future work will focus on scaling data collection and exploring end-to-end learning models that replace traditional hierarchical systems. The integration of mobility and manipulation, coupled with advancements in synthetic data generation, will be crucial in overcoming current limitations and pushing the boundaries of what autonomous robotic systems can achieve.

Chapter 2

SORNet: Spatial Object-Centric Representations for Sequential Manipulation

Robots must comprehend objects and their relationships to execute complex multi-step tasks effectively. Consider a task where a robot needs to dismantle a tower of blocks and rebuild it in a different order. The robot must determine whether each block is accessible, devise strategies to make each block accessible, and understand the consequences of its actions as each block is moved.

To address such long-horizon sequential manipulation tasks, it is common to employ a state estimator followed by a task and motion planner [4, 24, 26, 66, 75, 83]. These model-based systems are robust at reasoning and can be applied to various tasks with different goal conditions. However, their effectiveness is limited by the state estimator, which outputs explicit object states, such as 6D poses, that are challenging to estimate precisely from raw data and are not optimized for downstream tasks. Despite the existence of several powerful approaches for explicitly estimating the state of objects in the environment [14, 43, 84, 92], generalizing these approaches to an arbitrary collection of objects remains difficult. Furthermore, manipulation scenes often involve contact and occlusion, where state estimation methods tend to fail [93, 95].

Fortunately, explicit states, such as the exact poses of objects, are not always

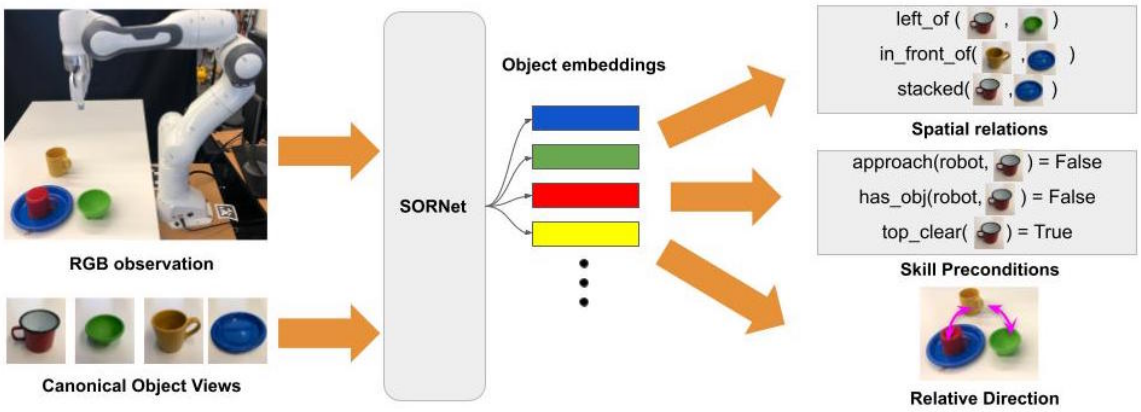


Figure 2.1: We propose **SORNet** (**S**patial **O**bject-Centric **R**epresentation **Net**work), a method which learns object embeddings from RGB observation given a set of object queries called canonical object views. SORNet adapts to novel objects without the need for annotating new data and finetuning. SORNet embeddings can be used to solve a variety of spatial reasoning tasks such as classifying spatial relations and regressing relative directions between objects.

necessary for manipulation tasks. An alternative approach involves learning motor controls directly from raw sensor data, such as RGB images and joint encoder readings. Known as the end-to-end approach, these methods utilize powerful neural network backbones that extract low-level embedding vectors from high-dimensional images and optimize directly for downstream tasks. Although recent methods in this category [19, 27, 33, 42, 65, 98] can perform complex manipulation tasks, they often lack transferability to new tasks and goals, particularly those involving long-horizon planning. The primary limitation is that these methods use scene-level representations (i.e., a single embedding vector per image) that do not facilitate object-level reasoning.

In this chapter, we introduce an object-centric representation learning framework that incorporates explicit notions of objects but uses implicit, learnable embeddings that generalize to novel objects, can be optimized for downstream tasks, and support object-level reasoning for a variety of goal-conditioned tasks. Specifically, we propose a neural network backbone called **SORNet** (**S**patial **O**bject-Centric **R**epresentation **Net**work). SORNet extracts object embeddings from raw RGB observations conditioned on a set of object queries, referred to as canonical object views. The design of SORNet, depicted in Fig. 2.1, enables it to generalize to scenes with novel objects

without any parameter modifications. The object-centric embeddings produced by SORNet can be integrated with readout networks to provide a task and motion planner with crucial spatial relations needed to formulate a plan for goal-directed sequential manipulation tasks, such as logical preconditions for primitive skills or continuous 3D directions between object centers.

We empirically evaluate SORNet on the classification of logical predicates and the regression of relative 3D directions between entities in manipulation scenes. Our results demonstrate the advantage of SORNet's object-centric representation over the scene-level representation produced by state-of-the-art pretraining methods on these spatial reasoning tasks. Additionally, we test SORNet on held-out objects that did not appear in the training data, showing that SORNet generalizes to common household objects from novel categories in a zero-shot manner.

2.1 Object Embedding Network

Our object embedding network, SORNet (Fig.2.2), processes an RGB image along with an arbitrary number of object queries to produce an embedding vector for each object query. The object queries are represented as image patches containing a single object, referred to as *canonical object views*. It is important to note that these canonical object views are not crops from the input image but are arbitrary views of the objects of interest, which may not match the objects' appearance in the scene. The model is designed to recognize objects despite significant changes in lighting, pose, and occlusion (examples of canonical object views used in our experiments can be seen in Fig.2.3). Essentially, the canonical object views function as queries, with the RGB image serving as the context from which spatial relations are extracted.

The network architecture is based on the Vision Transformer (ViT) [17]. The input image (or images) is divided into fixed-sized patches, termed *context patches*. These context patches are concatenated with the canonical object views to form a patch sequence. Each patch is first flattened and then linearly projected into a token vector. We add a series of learnable vectors, matching the token vector dimensions, as positional embeddings. The positional-embedded tokens are then processed through a transformer encoder, comprising multiple layers of multi-head self-attention [90]. The transformer encoder outputs a sequence of embedding vectors, from which we

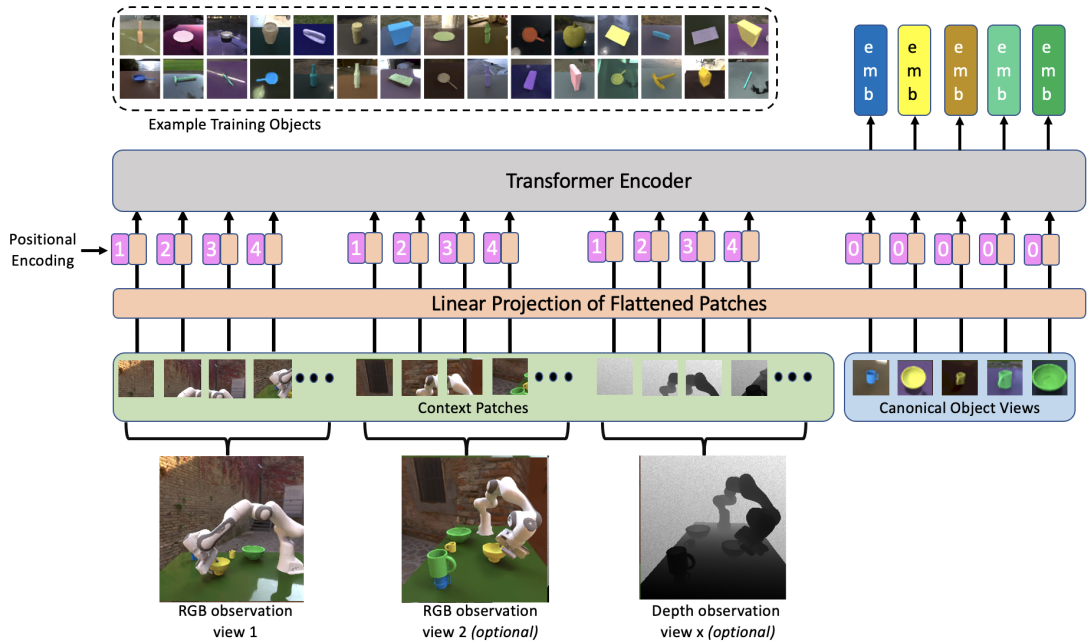


Figure 2.2: **SORNet** architecture. Input to the network is an RGB image and canonical views of the object queries. The RGB image is broken into context patches which have the same size as the canonical object views. The sequence of patches are flattened, position-encoded and passed through a multi-layer transformer to obtain a sequence of embedding vectors. The embeddings corresponding to the canonical object views are used for downstream tasks such as relation prediction. Additional views and modes of observations such as depth can be optionally added to the network. The top left inset shows examples of canonical object views used during training.

discard the embeddings corresponding to the context patches, retaining only those from the canonical object views as the output object embeddings.

There are two noteworthy implementation details:

Increased Spatial Resolution: To enhance spatial resolution, it is advantageous to use smaller patch sizes in the transformer's input sequence, even smaller than the size of the canonical object views. This results in multiple tokens in the output sequence corresponding to the same object query. We concatenate the tokens from the same object to form the representation for that object.

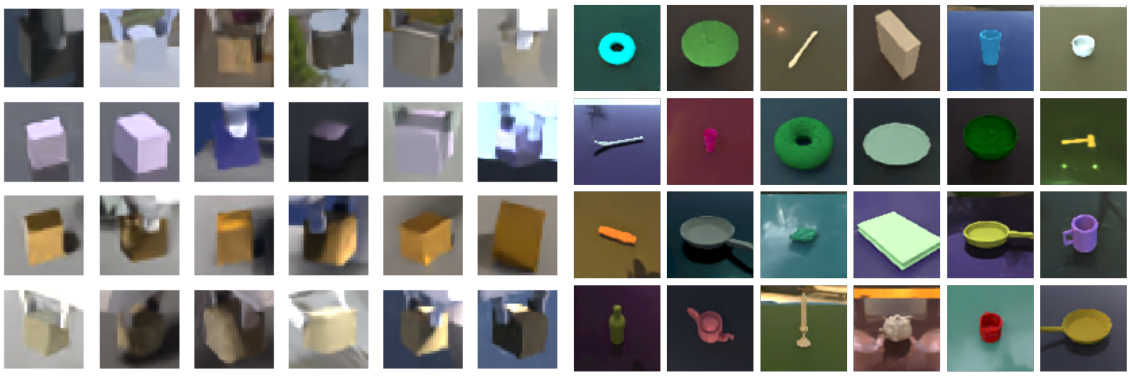


Figure 2.3: Examples of canonical object views in the Leonardo (left) and Kitchen (right) dataset. Lighting, texture and object poses vary across different views. Sometimes there is also occlusion by the robot.

Permutation Equivariance: The positional embedding provides the transformer with the absolute position of the patch in the image. However, there is no inherent order among the set of object queries. To ensure that the output object embeddings are permutation equivariant, we assign the same set of position embedding vectors to each object (indicated by the 0s preceding input object tokens in Fig. 2.3). This allows us to input an arbitrary number of canonical object views in any order without modifying the model parameters during inference.

2.2 Readout Networks

The readout networks (Fig. 2.4), as their name implies, are designed to "read out" object relations using the embedding vectors generated by the embedding network. These relations can be logical statements, such as determining if the blue block is stacked on top of the green block, or continuous quantities, like determining the direction the end effector should move to reach the red block. The readout networks consist of a collection of 2-layer MLPs, each tailored to a specific type of relation. Each readout network takes a number of object embeddings as inputs and outputs a quantity relevant to the input object arguments.

Our framework primarily focuses on unary and binary relations, though it can be extended to handle relations involving more than two objects. Unary readout networks take a single object embedding as input and output a quantity involving that

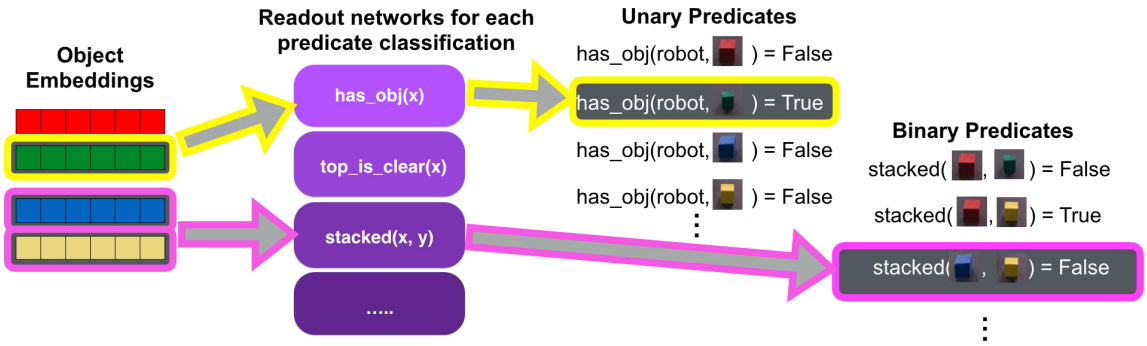


Figure 2.4: Architecture of the readout networks, which uses the object embeddings from **SORNet** to predict spatial relations, such as logical statements that can serve as skill preconditions or continuous 3D directions. The readout network is flexible to accommodate any number of input object embeddings without changing its parameters.

single object, optionally in relation to an environmental element such as the robot or a region on the table. For instance, the `top_is_clear` classifier, when given the embedding conditioned on the blue block, outputs whether there is an object on top of the blue block. Binary readout networks take pairs of object embeddings, created by concatenating the embeddings of two objects, and output quantities involving the pair, such as determining if the blue block is on top of the red block.

A key feature of our readout network design is that the parameters of the readout networks are independent of the number of object inputs. Given N input object embeddings, each unary readout network will output N relations, and each binary readout network will output $N(N-1)$ relations. The number of output relations dynamically adjusts based on the number of inputs. This design enables our model to generalize in a zero-shot manner to scenes with different numbers of objects from those seen during training.

2.3 Data Generation

To ensure that our network learns spatial relations between objects rather than overfitting to irrelevant factors like color and texture, it is crucial to train on a large-scale dataset with diverse object appearances and scene layouts. Existing robotics datasets are often limited in object variety (e.g., only containing YCB objects [8]) and

scene complexity, while existing vision datasets lack robot or object layouts pertinent to manipulation tasks (e.g., stacked objects). Therefore, we created our own tabletop environment where a Franka arm manipulates a set of randomly colored objects.

In our dataset, the robot is provided with a goal formulated as a list of logical predicates to be satisfied. A simple task and motion planner [66] is used to devise a plan based on the ground truth object states in simulation. Fig. 2.7 illustrates some example tasks included in the dataset. The robot then executes the plan in simulation, and we use NVISII [58] to render RGB and depth images each time a predicate value changes. This approach ensures that we capture the critical moments when the robot transitions to a different primitive skill. To enhance visual diversity, we applied domain randomization, including random lighting, backgrounds, and perturbations to the camera position during rendering. Ground-truth logical states are computed and recorded alongside the rendered frames.

We created two datasets, Leonardo and Kitchen, to serve different purposes:

Leonardo Dataset: This dataset includes a single object shape (blocks) but features more complex scene layouts, such as towers of four stacked blocks. This setup allows the network to learn intricate spatial relationships within a simplified object context.

Kitchen Dataset: This dataset contains a wide variety of object shapes, featuring common household items from the ACRONYM subset [21] of ShapeNet [9]. This diversity in object shapes helps the network generalize across different object types and appearances.

Further details about the datasets are elaborated below.

2.3.1 Leonardo Dataset

The first of our two datasets is the Leonardo blocks stacking dataset. The training data comprises over 130K sequences, each depicting a single task: stacking four blocks into a tower. Block colors are randomly selected from the xkcd color palette (https://xkcd.com/color/rgb). The testing dataset consists of 1.6K sequences, each containing 4-7 blocks. For testing, block colors are chosen from seven colors

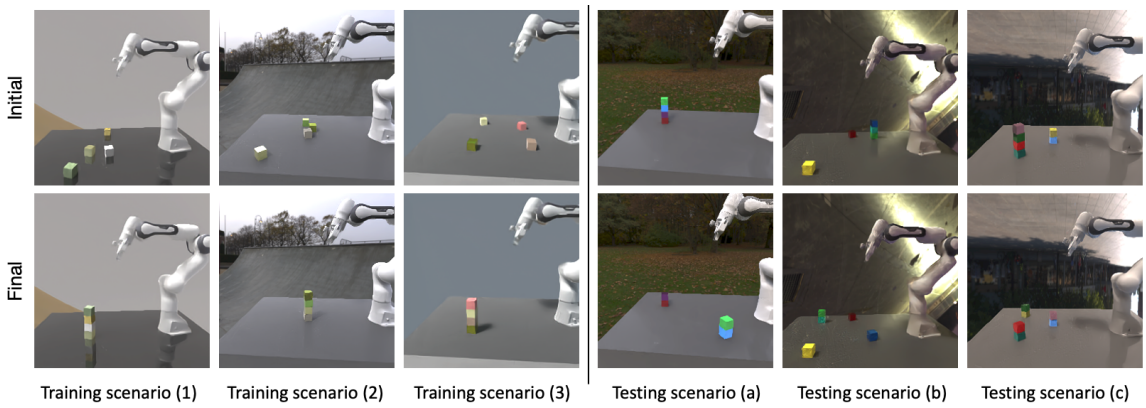


Figure 2.5: Sample scenes from training and testing scenarios in the Leonardo dataset. Top row shows the initial configuration of a sequence and the bottom row shows the goal configuration. The training scenarios contain 4 blocks with a single goal condition. The testing scenarios contain 4-7 blocks with heldout colors and various goal conditions involving multi-tower stacking.

Figure 2.6: Sample scenes from the kitchen dataset. The top and bottom rows show two different views. SORNet can leverage additional views to improve performance, but does not require multiple views.

that are not present in the training data: *red, green, blue, yellow, aqua, pink, purple*. Additionally, we excluded all training colors that contain the test color names (e.g., *light_red*), resulting in a total of 405 unique training objects.

The testing tasks are designed to be different from the single-tower-stacking task in the training data and comprise eight distinct super-tasks:

1. Building a single tower with two blocks.
2. Building two towers, each with two blocks.
3. Ensuring a specific block is not on the table.
4. Ensuring two blocks are not on the table.
5. Building a tower with three blocks.
6. Placing a block in a specific location.
7. Ensuring a block is not on the table while building a tower with two blocks.
8. Building a tower in a designated area of the table.

Fig. 2.7 shows examples of these tasks from the dataset.

2.3.2 Kitchen Dataset

The second dataset is the Kitchen object rearrangement dataset. Objects in this dataset are selected from the ShapeNet objects [9] used in the ACRONYM grasp dataset [21], featuring 330 object shapes across 33 categories typically found in a home, such as books and video game controllers. The Kitchen dataset uses the same train/test color split as Leonardo and further excludes the mug and bowl categories from the training data, ensuring they only appear in the test data. This setup demonstrates SORNet's ability to generalize not only to unseen object instances but also to entirely new object categories. Additionally, a shelf is optionally included beside the table to diversify the environment.

Unlike the Leonardo dataset, which samples specific high-level goals, the Kitchen dataset samples high-level action traces of picking and placing random objects. Objects can be placed in specific regions on the table, on a shelf (if available), or atop other objects. The training dataset contains over 24K sequences, while the testing dataset includes 1.6K sequences. Each sequence features 3 to 7 objects appearing simultaneously. Fig. 2.6 provides example frames from the dataset.

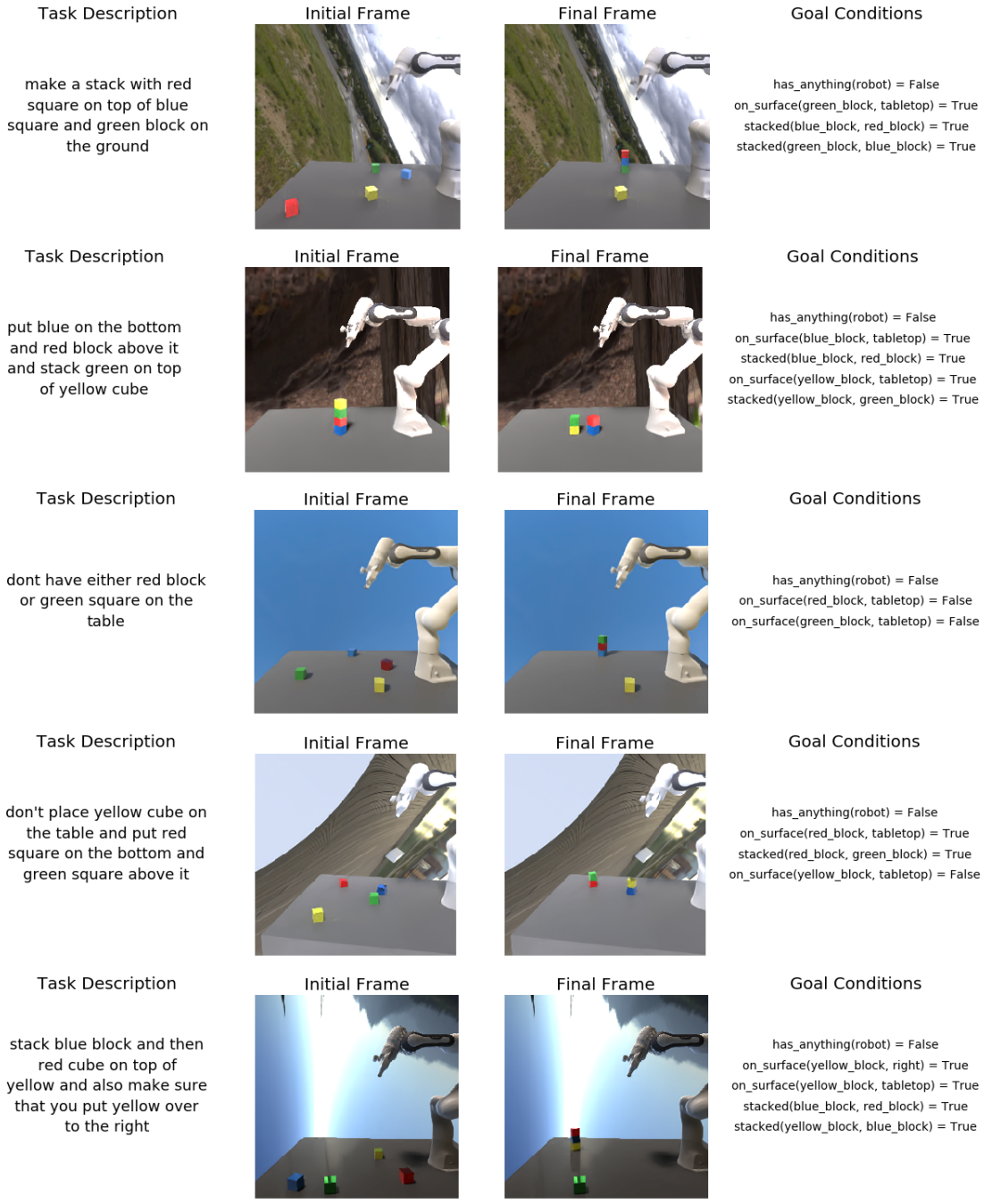


Figure 2.7: Visualization of some of the different tasks in the Leonardo test set. We used a wide variety of tasks to generate interesting data, taking the form of building one or more towers of varying heights, ensuring that certain objects were or were not on the table, and combinations of the above.

2.4 Experimental Results

In this section, we compare SORNet's object-centric embeddings with the scene-level embeddings learned by state-of-the-art representation learning techniques. We also demonstrate SORNet's unique ability to generalize to novel objects without any labels.

2.4.1 Spatial Relation Classification

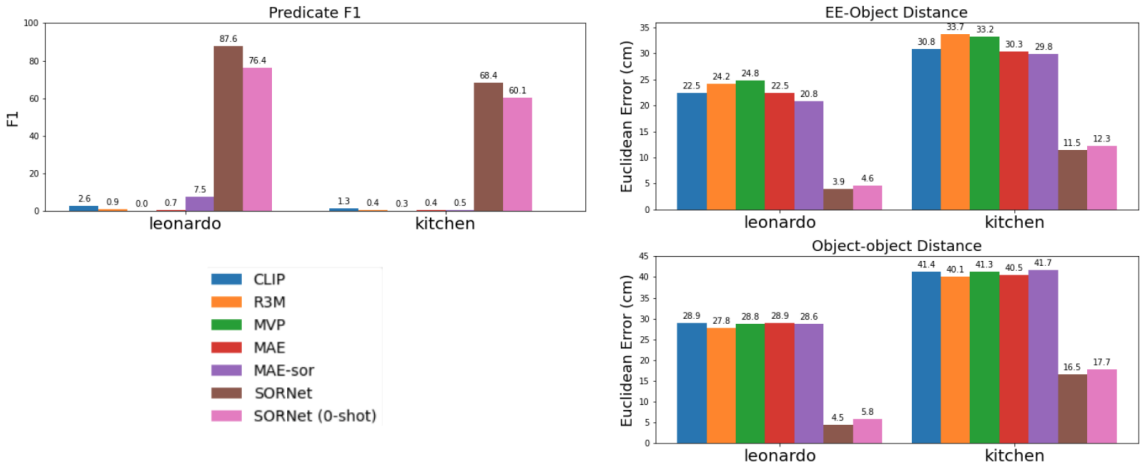
First, we evaluate SORNet on the task of predicting spatial relations among objects in manipulation scenes using logical predicates, e.g., `left_of(red_mug, green_bowl)`. The models are provided with an RGB image of the scene and tasked with predicting a list of binary (True or False) predicates. Please refer to the supplementary materials for a complete list of predicates.

We benchmark our model against these state-of-the-art pretraining methods:

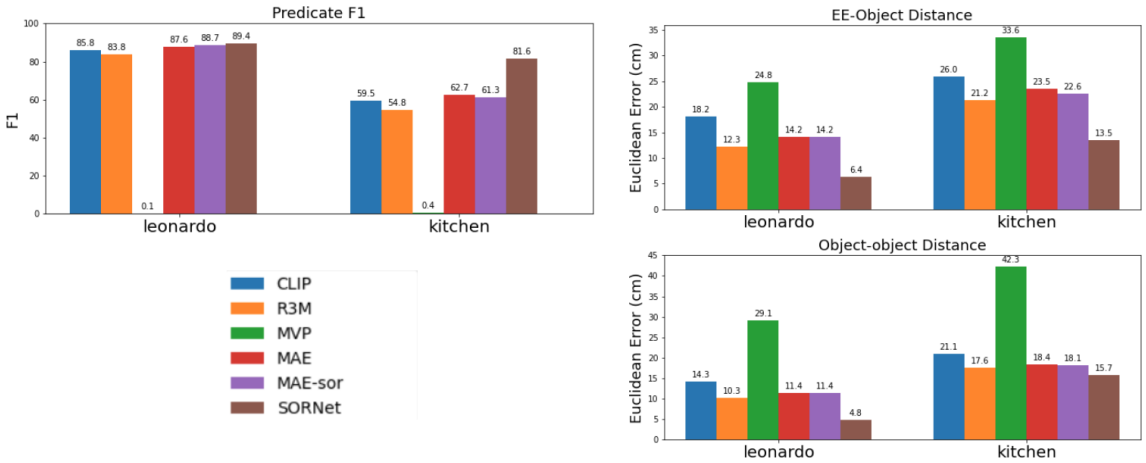
- **CLIP** [71] uses contrastive pretraining on a large-scale, non-public image-caption dataset and has achieved state-of-the-art results on tasks such as ImageNet classification.
- **R3M** [62] pretrains a ResNet [31] on egocentric videos from the Ego4D dataset [29] using time contrastive loss and video-language alignment.
- **MVP** [94] pretrains vision transformers [17] on egocentric videos from a combination of datasets (e.g., Epic Kitchens [13]) using masked autoencoding.
- **MAE** [32] pretrains vision transformers on the ImageNet data using reconstruction from masked input (referred to as masked autoencoding) and has demonstrated state-of-the-art transfer results to image classification.
- **MAE-sor** is a variant of MAE trained on our data (Leonardo and Kitchen) using the masked autoencoding objective.

We design two evaluation protocols:

- **Frozen Embedding:** The pretrained image embedding network is frozen, and only the readout networks (2-layer MLPs) are trained. This protocol assesses how much spatial information the pretrained embedding encodes.
- **Finetuned Embedding:** Both the embedding network and the readout net-



(a) Frozen Embedding



(b) Finetuned Embedding

Figure 2.8: Predicate classification (left) and relative direction regression (right) results on Leonardo and Kitchen data. SORNet clearly outperforms other pre-training methods on reasoning about spatial relations in scenarios with complex object interactions. In addition, the 0-shot model that has not seen any labeled data for the test objects also performs reasonably well.

works are trained. This protocol evaluates whether the network can learn spatial information when provided with appropriate supervision.

To ensure a fair comparison, we trained all baselines and SORNet on 10K sequences, with objects sampled from a fixed repository, and tested them on 1.6K sequences. Evaluations were conducted on both Leonardo (blocks only) and Kitchen data (common household objects). In the Leonardo data, the repository contains 7 objects, with each scene featuring 4-7 objects. In the Kitchen data, the repository contains 14 objects, with each scene featuring 3-8 objects. Both the baselines and SORNet are given ground truth object identities. Baselines always predict predicates for all objects in the repository, and only the predicates related to objects appearing in the current scene are evaluated. For SORNet, we provide the canonical patches corresponding to the objects in the scene.

We report the average F-1 score in Fig. 2.8. In the frozen embedding scenario, the baselines perform significantly worse than SORNet, indicating that existing pretraining methods do not work in a plug-and-play fashion for spatial reasoning tasks. The learned representations from these methods do not contain the necessary information for predicting spatial relationships among objects. In the finetuned embedding scenario, baselines perform much better, but SORNet still outperforms all baselines, especially on the Kitchen data, which contains more complex objects and scenes. This demonstrates the advantage of an object-centric network.

Moreover, SORNet's design allows it to be applied zero-shot, i.e., without any additional annotation, to objects unseen during training. This capability is reflected by the **SORNet (0-shot)** model in Fig. 2.8. None of the baselines can achieve this since their representations are not object-specific, requiring additional training or fine-tuning to work on novel objects. Fig. 2.9 presents qualitative results of the zero-shot model on the Kitchen data, where both the color and shape (category) of all objects do not appear in the training set.

2.4.2 Relative Direction Regression

In addition to predicting logical relations, the object embeddings learned by SORNet can also be used to predict continuous spatial information. To demonstrate this, we apply SORNet to predict the relative 3D direction between entities. Specifically,

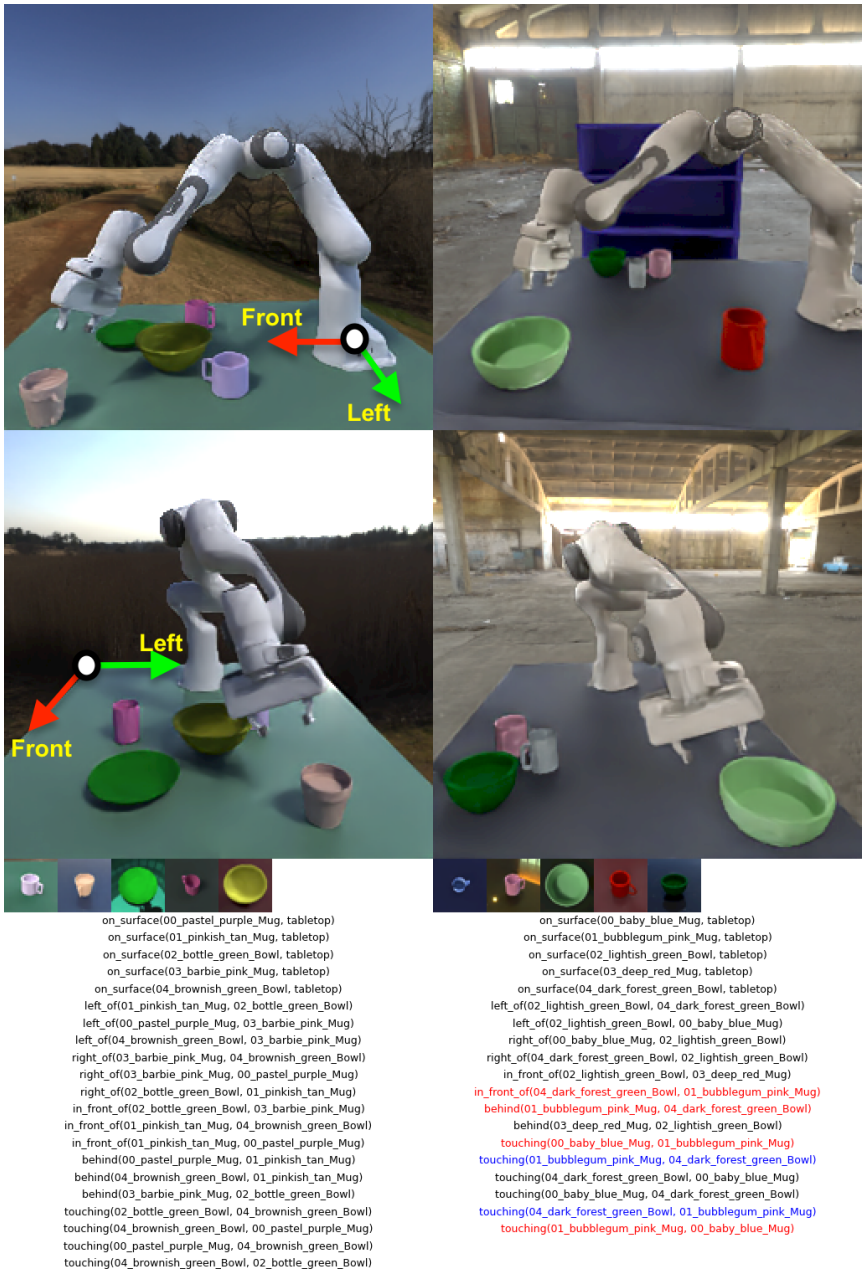


Figure 2.9: Qualitative predicate classification results with SORNet trained on the simulated kitchen dataset and tested on held-out objects. Each column is a different scenario. The first and second row shows the side and front view of the scene respectively, followed by the canonical views of 5 query objects. Black text denotes correctly labeled true predicates; blue text denotes false positive predictions; and red text denotes false negatives. True negatives are not shown due to limited space.

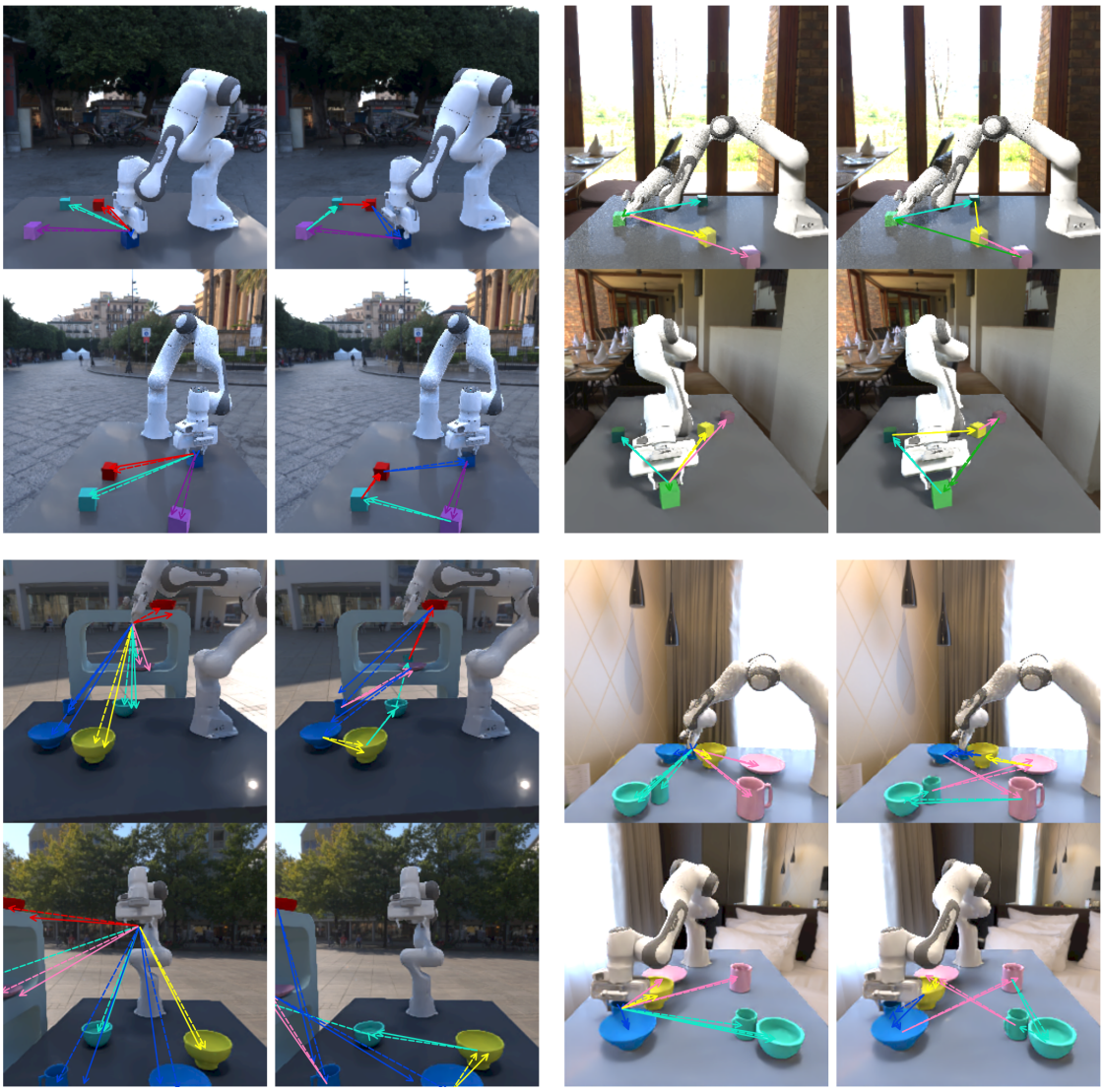


Figure 2.10: Relative Direction Prediction using pretrained SORNet embeddings. The color of the arrow corresponds to the color of the target. Solid arrows are the predictions and dashed arrows are the ground truths. Not all predictions are visualized in order to keep the plot clean. The first and third columns show the relative direction from the robot's end effector to the object centers. The second and fourth columns show the relative direction between object centers.

	MDETR	MDETR-oracle	**sornet(ours)**
ValA Accuracy	84.950	97.944	**99.006**
ValB Accuracy	59.627	98.052	**98.222**

Table 2.1: Zero-shot relation classification accuracy on CLEVR-CoGenT [39]. The MDETR-oracle model has seen all the objects during training, where as MDETR [40] and SORNet have only see objects in condition A. SORNet takes canonical views as queries whereas MDETR takes text queries.

we trained a regressor (using the same architecture as the classifier) to predict the continuous vector between two object centers (Obj-Obj) or the direction the end effector should move to reach a certain object (EE-Obj). The x, y, z components of the continuous vector are treated as "predicates" with continuous values and trained with L2 loss. The regressor predicts both the directions from the robot's end-effector to the object centers (EE-Obj), and the directions between the object centers (Obj-Obj).

Results are summarized in Fig.2.8. We compare to the same baselines and employ the same two evaluation protocols, frozen and finetuned embedding. SORNet is able to outperform competing pretraining methods (CLIP, MVP, R3M, and MAE) in a similar fashion. This demonstrates that SORNet's representation transfers much better to manipulation scenarios where more precise spatial information is crucial. We can also apply SORNet in a zero-shot fashion to scenes with novel objects. Fig. 2.10 shows some qualitative predictions of the zero-shot SORNet model.

2.4.3 Compositional Generalization on CLEVR-CoGenT

We also evaluate our approach on a variant of the CLEVR dataset [39], a well-established benchmark for visual reasoning. CLEVR contains rendered RGB images with up to 10 objects per image. There are 96 different objects in total (2 sizes, 8 colors, 2 materials, 3 shapes). Each image is labeled with 4 types of spatial relations (right, front, left, behind) for each pair of objects.

Specifically, we use the CoGenT version of the dataset, which stands for Compositional Generalization Test, where the data is generated in two different conditions. In condition A, cubes are gray, blue, brown, or yellow, and cylinders are red, green, purple, or cyan. Condition B is the opposite: cubes are red, green, purple, or cyan,

and cylinders are gray, blue, brown, or yellow. Spheres can be any color in both conditions. The models are trained on condition A and evaluated on condition B. The training set (trainA) contains 70K images, and the evaluation set (valB) contains 15K images. Several prior works [40, 96] show a significant generalization gap on CLEVR-CoGenT caused by the visual model learning strong spurious biases between shape and color.

We generate a question for each spatial relation in the image, e.g., "Is the large red rubber cube in front of the small blue metal sphere?" We filter out any ambiguous queries, e.g., if there were two large red cubes, one in front and one behind the small blue sphere. This results in approximately 2 million questions for both valA and valB sets. We compare against MDETR [40], which reports state-of-the-art zero-shot results on CLEVR-CoGenT, i.e., no fine-tuning on any example from condition B. The results are summarized in Table 2.1. Our model performs drastically better in classifying spatial relations of unseen objects and shows a much smaller generalization gap between valA and valB sets.

Unlike MDETR, which takes text queries, our model takes visual queries in the form of canonical object views (i.e., 2 canonical views for the objects mentioned in the question). To eliminate the influence of those factors, we report the performance of the MDETR model trained on the full CLEVR dataset, denoted as MDETR-oracle. Although SORNet is only trained on condition A, it achieves similar performance to MDETR-oracle. The zero-shot generalization ability of our model can potentially be combined with other reasoning pipelines to improve generalization performance on various types of queries.

Fig. 2.11 shows examples of spatial relation classification on CLEVR-CoGenT. These examples demonstrate that SORNet can identify objects not only using color cues but also shape (e.g., blue sphere vs. blue cylinder in the topmost example), size (e.g., small cyan cube vs. big cyan cube in the second from top example), and material (e.g., small purple metal cube vs. small purple rubber cube in the third from top example). We also visualize the relevant canonical object views provided to the model, which can have a very different appearance from the corresponding objects in the input image. It is more appropriate to consider these canonical views as a visual replacement for natural language rather than the result of object detection or segmentation.

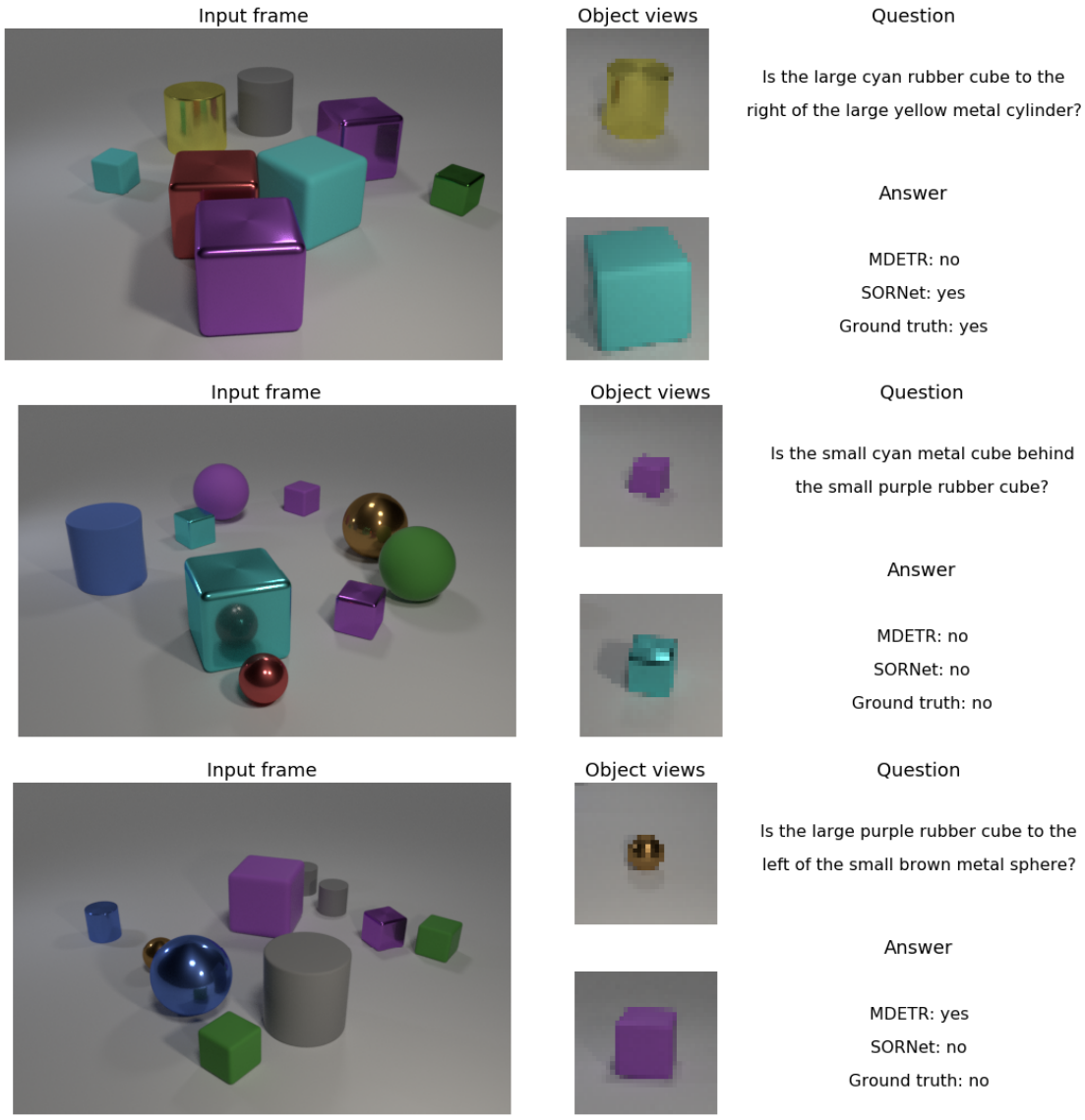


Figure 2.11: Relation classification in CLEVR-CoGenT. In addition to color, SORNet is able to distinguish objects based on shape, size and material. It is also able to deal with heavy occlusion. Note that the object patches provided to SORNet can have very different appearance than the corresponding objects in the input frame.

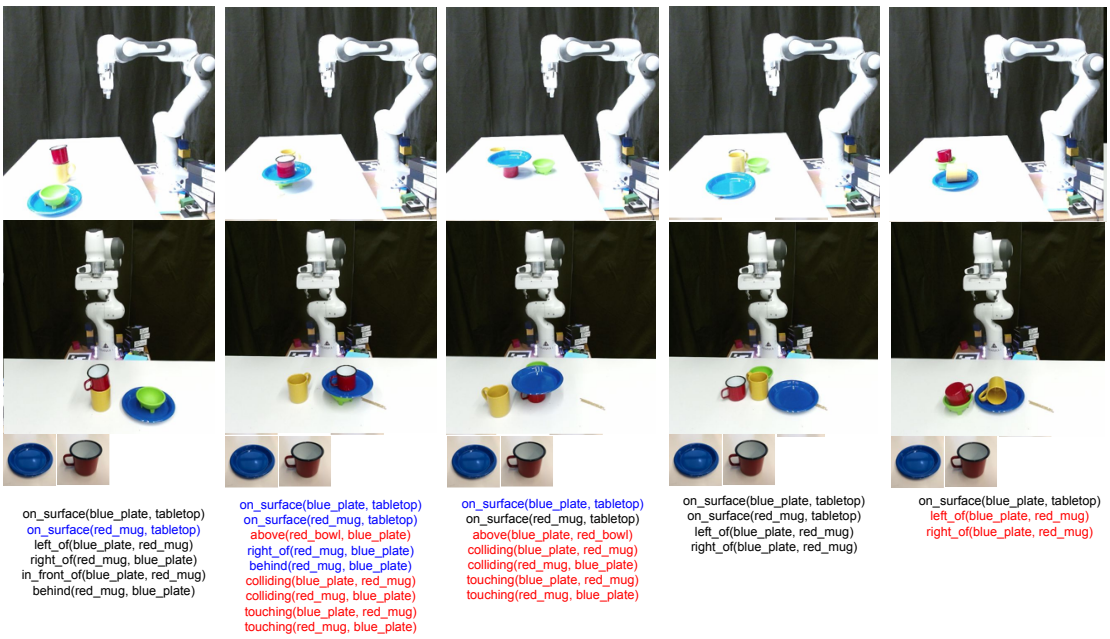


Figure 2.12: Predicate classification by SORNet trained on the simulated kitchen dataset and tested on held-out real-world objects. Each column is a different scenario. The first and second row shows the side and front view of the scene respectively, followed by the canonical views of the query objects ("blue_plate and "red_mug"). Black text denotes correctly labeled true predicates; blue text denotes false positive predictions; and red text denotes false negatives. We can see that SORNet produces accurate labels in many scenes, and shows strong transfer even to scenes and objects that are very out of its training distribution.

2.4.4 Sim-to-real Transfer

SORNet embeddings can transfer from simulation to the real world without any fine-tuning. To demonstrate this capability, we conducted a set of experiments capturing various scenes containing novel, unseen objects in previously unseen colors from our held-out test object set. In each scene, we selected a random pair of objects and compared manually-labeled predicates with those predicted by SORNet.

Fig. 2.12 presents examples of scenes from the real-world testing dataset. Importantly, these objects were randomly chosen from the real world. Two cameras, placed without any extrinsic calibration, captured side and front views of the scene. SORNet, trained solely on RGB data with two views, was used for this experiment. The canonical views were taken using the "square" mode on a smartphone. Each column shows a particular state of the world along with the predicate classification by SORNet. For simplicity, we display the predicates of only two objects: "red_mug" and "blue_plate." In the visualization, black text indicates true positive predictions; blue text indicates false positive predictions, and red text indicates false negatives.

Despite never being trained on real-world observations, SORNet achieves accurate classification in many scenes, demonstrating its robust transfer capability even to scenes and objects outside of its training distribution. This highlights SORNet's potential for real-world applications in complex environments without the need for additional training.

2.4.5 Attention Visualization

Fig. 2.13 visualizes the attention weights learned by the visual transformer model to obtain the object-centric embeddings. Specifically, we extract the normalized attention weights from the tokens corresponding to the canonical object views to the tokens corresponding to the context patches and convert these weights into a colormap. The intensity of the colormap indicates the magnitude of the attention weight over that patch. We then overlay the colormap onto the input image.

From this visualization, we observe that while the model places the highest attention on the patch containing the object of interest, it also learns to attend to the robot arm and other relevant objects, while effectively ignoring the irrelevant background. Additionally, we display the canonical object patches provided to the

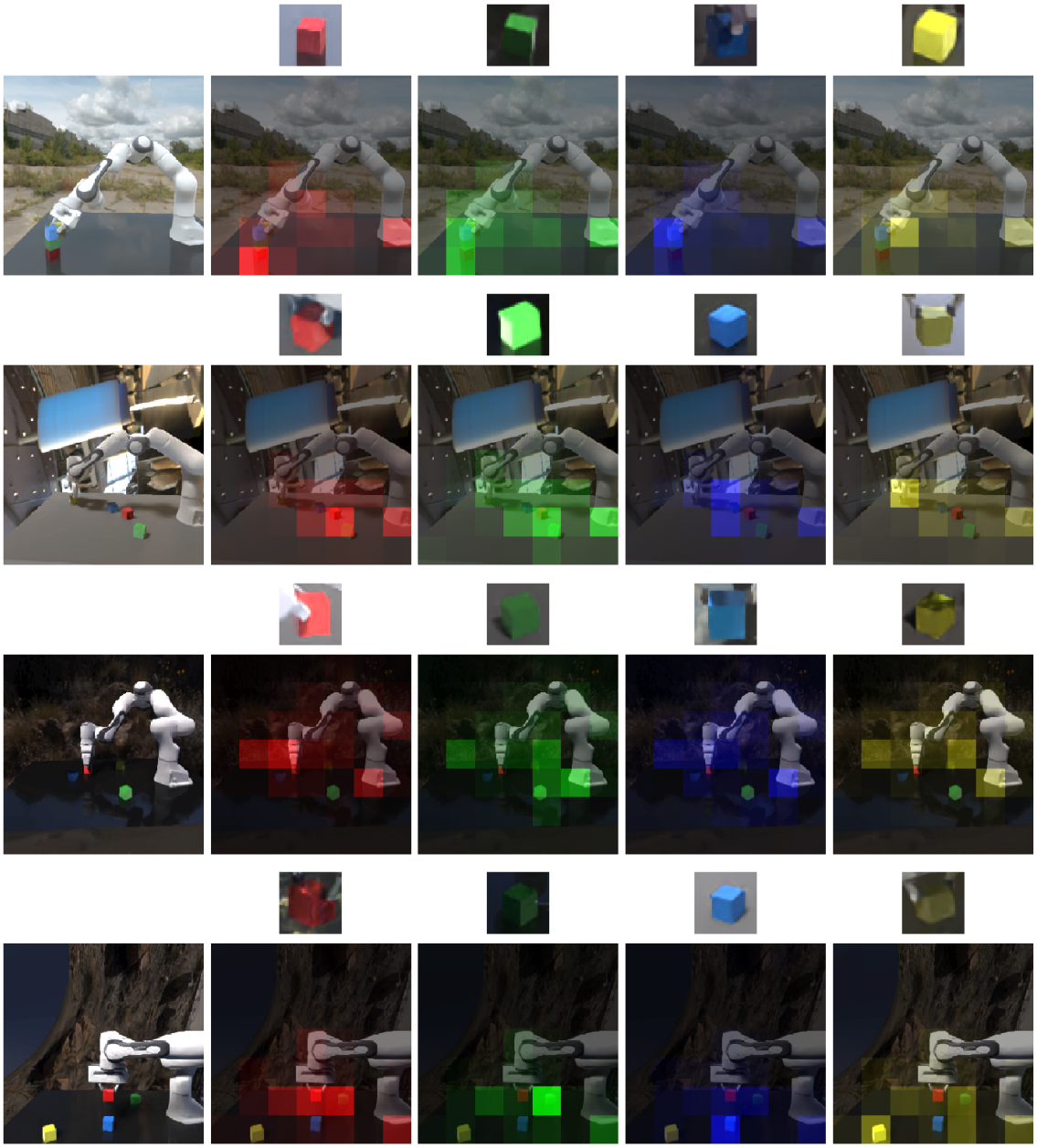


Figure 2.13: Visualization of the attention learnt by SORNet. Leftmost column is the input frame and remaining columns visualize the attention weights of the object tokens over the context patches. The corresponding canonical object view is shown on top of each attention visualization. We can see that SORNet learns to attend to not only the object of interest but also the robot and other objects as well, while ignoring irrelevant background.

model. These canonical patches can look significantly different from the same objects in the image due to variations in lighting conditions and occlusions. The model must associate the canonical view with the input view of the object despite these differences.

This visualization demonstrates the model's capability to focus on pertinent elements in the scene and establish connections between canonical and input views of objects under varying conditions, showcasing its robust understanding of spatial relations and object-centric embeddings.

2.5 Limitations

SORNet currently only takes visual queries using canonical object views. This limits its ability to perform complex queries that involves language and also non-rigid objects that are difficult to capture with a single view. An interesting future direction is to explore the integration of SORNet with modern vision-language models to handle more complex reasoning.

2.6 Summary

We proposed **SORNet** (**S**patial **O**bject-Centric **R**epresentation **Net**work), which learns object-centric representations from RGB images. The object embeddings produced by SORNet effectively capture spatial relations, enabling their use in downstream tasks such as spatial relation classification, skill precondition classification, and relative direction regression. Our method operates on scenes with an arbitrary number of unseen objects in a zero-shot fashion. Through real-world robot experiments, we demonstrate SORNet's applicability in manipulating novel objects, showcasing its potential for real-world applications without requiring additional training.

Chapter 3

M2T2: Multi-Task Masked Transformer for Object-centric Pick and Place

The successful completion of many complex manipulation tasks, such as object rearrangement, relies on robust action primitives capable of handling a wide variety of objects. Recently, significant progress has been made in open-world object manipulation [2, 38, 44, 80] by using language models for high-level planning. However, these methods are often restricted to scenes with a limited number of fixed object shapes due to the limited capability of low-level skills such as picking and placing. Concurrently, task-specific models [60, 82, 85] excel in a particular skill across a wide range of objects. This raises the question: is it possible to have a single model that can robustly handle different action primitives across diverse objects?

We propose the **M**ulti-**T**ask **M**asked **T**ransformer (M2T2), a unified model for learning multiple action primitives. As illustrated in Fig. 3.1, given a point cloud of the scene, M2T2 predicts collision-free gripper poses for various types of actions, including 6-DoF grasping and placing, thus eliminating the need for different methods for different actions. M2T2 can generate a diverse set of goal poses, providing ample options for low-level motion planners. Additionally, it can generate more specific goal poses conditioned on language input. By combining high-level task planners with the robust action primitives from M2T2, robots can solve many complex tasks.

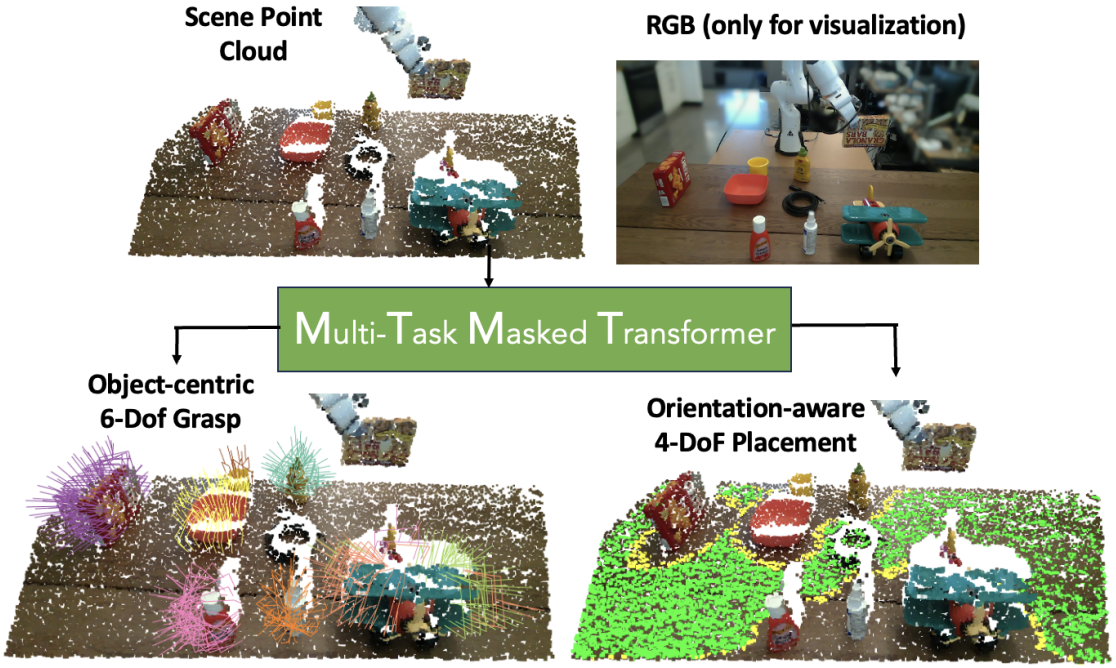


Figure 3.1: We propose M2T2, a unified model for learning multiple action primitives. M2T2 takes a raw 3D point cloud and predicts 6-DoF grasps per-object (lower left) and orientation-aware placements (lower right, where green means the object can fit in any orientation and yellow means only a subset of orientations are possible). Colors on the point clouds are for visualization only.

3.1 Problem Definition

The M2T2 framework is designed to predict target gripper poses for a variety of action primitives. This section focuses on two fundamental manipulation actions: picking and placing.

Object-centric 6-DoF Grasping: The system takes a single 3D point cloud of the scene, typically obtained from standard depth sensors, as input. It then generates a set of grasp proposals for objects within the scene. Each proposal consists of a 6-DoF (degrees of freedom) grasp pose, encompassing 3-DoF for rotation and 3-DoF for translation. These poses specify the exact positioning and orientation required for the robot's end effector to successfully grasp an object.

Orientation-aware Placing: This mode operates on a 3D point cloud of the scene in conjunction with a partial 3D point cloud of the object to be placed. The output is a collection of 6-DoF placement poses, which indicate the precise positioning and orientation the end effector should achieve to release the object in a stable manner, avoiding any collisions. M2T2 ensures the correct orientation of the object to fit into cluttered spaces, thus facilitating the stable and precise placement of objects.

The fundamental concept behind M2T2 is its ability to reason about contact points. In the context of picking, the robot's empty gripper makes contact with the target object. For placing, the robot maneuvers the object held in its gripper to make contact with the target surface.

3.2 Network Architecture

Scene Encoder: The scene encoder processes a 3D point cloud of the scene to generate multi-scale feature maps, which provide context for the contact decoder. Specifically, it produces four feature maps scaled to $1/64, 1/16, 1/4, 1$ of the original input size respectively. Each feature vector within these maps is associated with a corresponding point in the input point cloud. For our implementation, we adapt a PointNet++ [68] model, originally designed for semantic segmentation. However, any network capable of generating multi-resolution feature maps from 3D point clouds could serve as the scene encoder.

The PointNet++ [68] model we use has 4 multi-resolution set abstraction layers

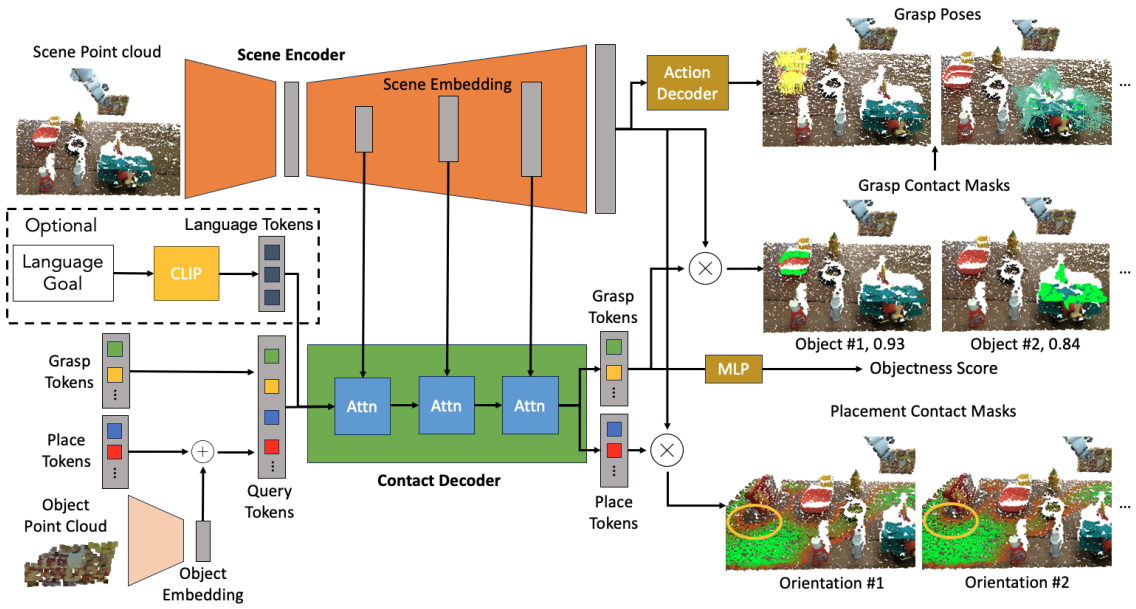


Figure 3.2: M2T2 generates valid gripper poses for grasping and placing with a single model. First, a 3D network (scene encoder) takes the scene point cloud and produces multi-scale feature maps. Then, the features are cross-attended with learnable query tokens via a transformer (contact decoder). Finally, the output tokens are multiplied with per-point features and generate contact masks and gripper poses for each object (for grasping) and each orientation (for placing). For grasping, addition MLPs are applied to the output tokens and per-point features to predict objectness scores (to filter out non-object proposals) and grasp parameters (to reconstruct gripper poses). Optionally, the contact decoder can take a set of tokens encoding language goals to produce goal-conditioned grasping and placing poses.

as the encoder and 4 feature propagation layers as the decoder. The input point cloud is subsampled to 16384 points. Each set abstraction layer will select $N/4$ seed points using furthest point sampling where N is the size of the input pointwise feature map. Then, local features are computed around each seed point and propagated with an MLP. As a result, the scene encoder produces 4 feature maps of decreasing resolution, with 16384, 4096, 1024 and 256 points respectively. We use the first per-point feature map for prediction and the remaining 3 for cross-attention.

Contact Decoder: The contact decoder, implemented as a transformer, predicts contact points for both grasping and placing actions. For grasping, we utilize the grasp representation from [85], where each grasp is centered around a visible point on the object that contacts the gripper. The model also predicts additional parameters to define the relative transformation of the grasp concerning the contact point. We extend this approach for placing by defining the contact point as the location where the center of the object's point cloud projects onto the surface, such as a table.

Our method leverages insights from image segmentation. We modify the masked transformer architecture [11] to predict contact masks. The transformer processes a set of learnable query tokens through multiple attention layers. Feature maps from the scene encoder, at different resolutions, are integrated via cross-attention at various stages. The output tokens from each layer interact with the per-point feature map from the scene encoder to generate interim masks. These interim masks guide the attention mechanism in subsequent layers, focusing it on relevant regions (hence the term "masked transformer"). The final output consists of G grasping masks and P placing masks, where G represents the maximum number of graspable objects and P denotes the number of possible placement orientations.

Objectness MLP: An MLP (Multi-Layer Perceptron) processes the grasp tokens to produce an objectness score for each token. This scoring mechanism filters out non-object tokens, as the number of graspable objects in a scene can vary.

Object Encoder: The object encoder, implemented using PointNet++ [68], encodes a 3D point cloud of the object to be placed into a single feature vector. This feature vector is then added to the place query tokens, enriching them with

object-specific information.

Action Decoder: The action decoder is composed of a 3-layer MLP. It utilizes the per-point feature map from the scene encoder to predict a 3D approach direction, a 3D contact direction, and a 1D grasp width for each point. These predictions are used in conjunction with the contact points to reconstruct grasp poses.

3.3 Training and Inference

3.3.1 Training Objective

Grasping: The training objective for grasping is composed of three components: the objectness loss (L_{obj}), the mask loss (L_{mask}), and the ADD-S loss ($L_{\text{ADD-S}}$).

Since the number of objects N in the scene is unknown, we set the number of grasp tokens G to a large value. M2T2 produces G scalar objectness scores o_i and G per-point masks $M^{\text{grasp}}i$. Hungarian matching is employed to select N masks that best align with the ground truth. The cost for each prediction $(o_i, M^{\text{grasp}}i)$ and ground truth mask $M^{\text{gt}}j$ is computed as follows:

$$C_{ij} = 1 - o_i + \text{BCE}(M_i^{\text{pred}}, M_j^{\text{gt}}) + \text{DICE}(M_i^{\text{pred}}, M_j^{\text{gt}}) \tag{3.1}$$

where BCE is the binary cross-entropy loss and DICE is the DICE loss [55]. Hungarian matching is then applied to the $G \times N$ cost matrix C to find the set of indices $\mathcal{M} = m_i$ that minimize the total cost $\sum j = 1^N C m_j j$. The objectness loss is computed by labeling all matched tokens as positive and others as negative:

$$L_{\text{obj}} = \frac{1}{G} \sum_{i=1}^{G} - [\mathbb{1}(i \in \mathcal{M}) \log(o_i) + (1 - \mathbb{1}(i \in \mathcal{M})) \log(1 - o_i)] \tag{3.2}$$

The mask loss between the matched masks and the ground truth masks is calculated as:

$$L_{\text{mask}} = \frac{1}{N} \sum_{j=1}^{N} \text{BCE}(M_{m_j}^{\text{pred}}, M_j^{\text{gt}}) + \text{DICE}(M_{m_j}^{\text{pred}}, M_j^{\text{gt}}) \tag{3.3}$$

In practice, we find that computing the BCE only for the points with the top k largest

losses improves performance, where $k = 512$ for grasping and $k = 1024$ for placing. This approach addresses the issue of large class imbalance (over 90% of points are not contact points).

The ADD-S loss, introduced by [85], is essential for accurate grasp confidence estimation. To compute this loss, we first define five key points $\mathbf{v}k$ on the gripper. For each pair of predicted grasp and ground truth grasp, the total distance between the transformed key points is computed as:

$$d_{ij} = \sum_{k=1}^{5} \|(R_i^{\text{pred}}\mathbf{v}_k + \mathbf{t}_i^{\text{pred}}) - (R_j^{\text{gt}}\mathbf{v}_k + \mathbf{t}_j^{\text{gt}})\| \tag{3.4}$$

Next, the closest ground truth for each prediction is found as $n_i = \text{argmin}jdij$, and the ADD-S loss is computed as:

$$L_{\text{ADD-S}} = \frac{1}{|\mathcal{C}^{\text{pred}}|} \sum_{i \in \mathcal{C}^{\text{pred}}} s_i d_{in_i} \tag{3.5}$$

where $\mathcal{C}^{\text{pred}}$ is the set of contact points of predicted grasps, and s_i is the grasp confidence, a scalar between 0 and 1 derived from the contact masks before thresholding. By weighting the loss with s_i, predicted grasps far from any ground truth grasp incur a larger penalty on confidence, enhancing contact point estimation.

Placing: The placing objective is defined as a combination of BCE and DICE [55] losses between the predicted and ground truth placement masks:

$$L_{\text{placing}} = \frac{1}{P} \sum_{i=1}^{P} \text{BCE}(M_i^{\text{pred}}, M_i^{\text{gt}}) + \text{DICE}(M_i^{\text{pred}}, M_i^{\text{gt}}) \tag{3.6}$$

This is the sole loss for placing, as no additional learnable parameters are required to reconstruct the placement poses.

3.3.2 Model Inference

Grasp Pose Prediction: During inference, we first select the contact masks with an objectness score greater than 0.5. For each point $\mathbf{p}$ within the selected contact mask, the corresponding grasp parameters from the action decoder are used to reconstruct

a 6-DoF grasp pose $(R_{\text{grasp}}, \mathbf{t}\text{grasp}) \in \text{SE}(3)$ as follows:

$$\mathbf{t}_{\text{grasp}} = \mathbf{p} + \frac{w}{2}\mathbf{c} + d\mathbf{a} \tag{3.7}$$

$$R_{\text{grasp}} = \begin{bmatrix} | & | & | \\ \mathbf{c} & \mathbf{c} \times \mathbf{a} & \mathbf{a} \\ | & | & | \end{bmatrix} \tag{3.8}$$

where $\mathbf{c}$ is the unit 3D contact direction, $\mathbf{a}$ is the unit 3D approach direction, w is the 1D grasp width, and d is the fixed distance from the gripper base to the grasp baseline (the line between the two fingers). For further details on this formulation, refer to the Contact-Grasp-Net paper [85].

Placement Pose Prediction: The P placement contact masks indicate valid placement locations when the object in the gripper is rotated by P discrete planar rotations R_{planar}. To recover the placement poses, we first compute the bottom center $\mathbf{b}$ of the object point cloud, which serves as the reference point for contact. Using forward kinematics, we obtain the current pose of the gripper $(R_{\text{ee}}, \mathbf{t}\text{ee})$. The 6-DoF placement pose $(R\text{place}, \mathbf{t}\text{place})$ is then computed as:

$$\mathbf{t}_{\text{grasp}} = \mathbf{p} + R_{\text{planar}}(\mathbf{t}_{\text{ee}} - \mathbf{b}) \tag{3.9}$$

$$R_{\text{grasp}} = R_{\text{planar}} R_{\text{ee}} \tag{3.10}$$

3.4 Data Generation

We constructed a synthetic dataset containing 130K cluttered scenes for training and evaluating 6-DoF picking and placing methods. This dataset includes 64K training scenes and 1K test scenes for both picking and placing. Each scene contains between 1 and 15 objects scattered on a table of varying size, which is equipped with a Franka Emika robot arm. The objects are sampled from the ACRONYM dataset [21], comprising 8.8K object models, each annotated with 2K grasps. These objects span 252 different categories, with 12 categories excluded from training. Half of the test scenes contain only objects from these 12 unseen categories. For each scene, we generate a 512×512 depth image from a random viewpoint above the table to create the scene point cloud.

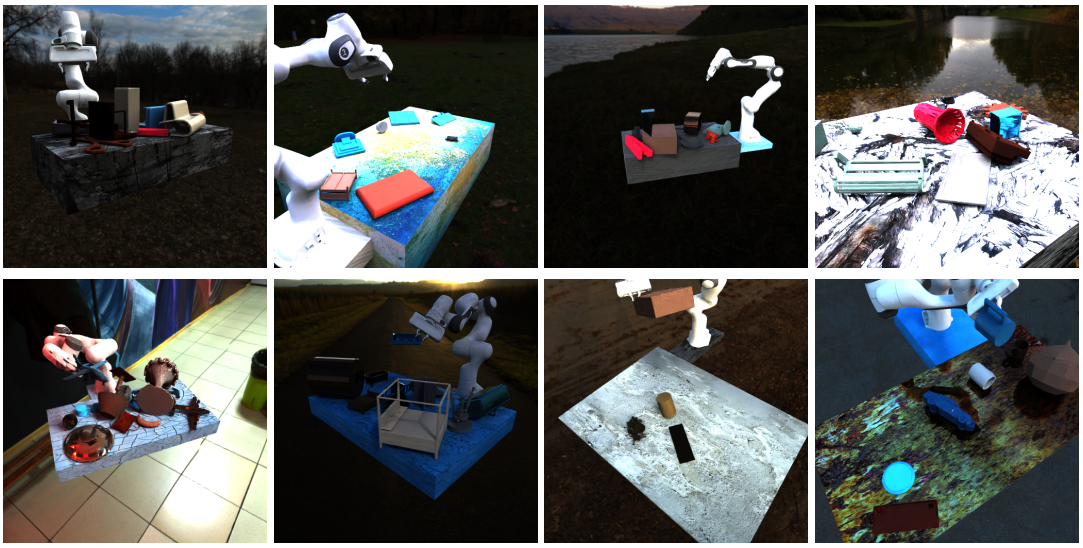


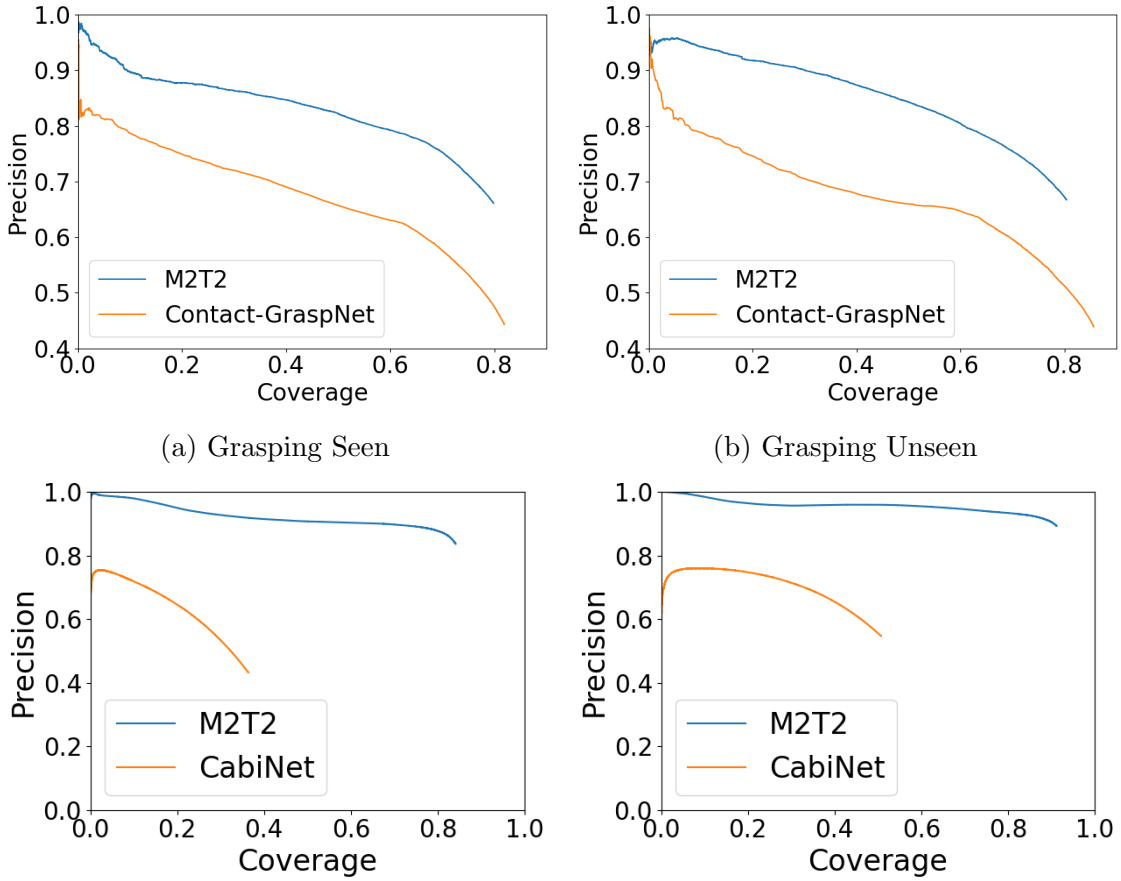
Figure 3.3: Examples for our large-scale synthetic dataset, for the grasping (top) and placing (bottom) tasks respectively. Objects are randomly sampled from ACRONYM [21]. Each scene can contain up to 15 objects, which creates many very cluttered scenes. We also include robot in the observation to simulate realistic occlusion by the robot arm.

We procedurally generated a large-scale synthetic dataset for training M2T2, as shown in Fig 3.3. In each scene, we randomly place 1-15 objects from the ACRONYM dataset [21] on the table. Each object in ACRONYM are labeled with 2000 grasps. We transform these grasps by the object pose and filter out colliding ones. The camera pose is randomized around the entire hemisphere above the table, making M2T2 very robust to viewpoint changes.

3.5 Experimental Evaluation

3.5.1 Evaluation in Simulation

Evaluation Metric: We use the precision-coverage curve as the metric for our evaluation in simulation. To plot this curve, we start with a confidence threshold of 1 and incrementally lower the confidence threshold until 0.5, adding grasps/placements to the set of predictions. During this process, we track two key metrics: precision, defined as the percentage of successful grasps/placements, and coverage, defined



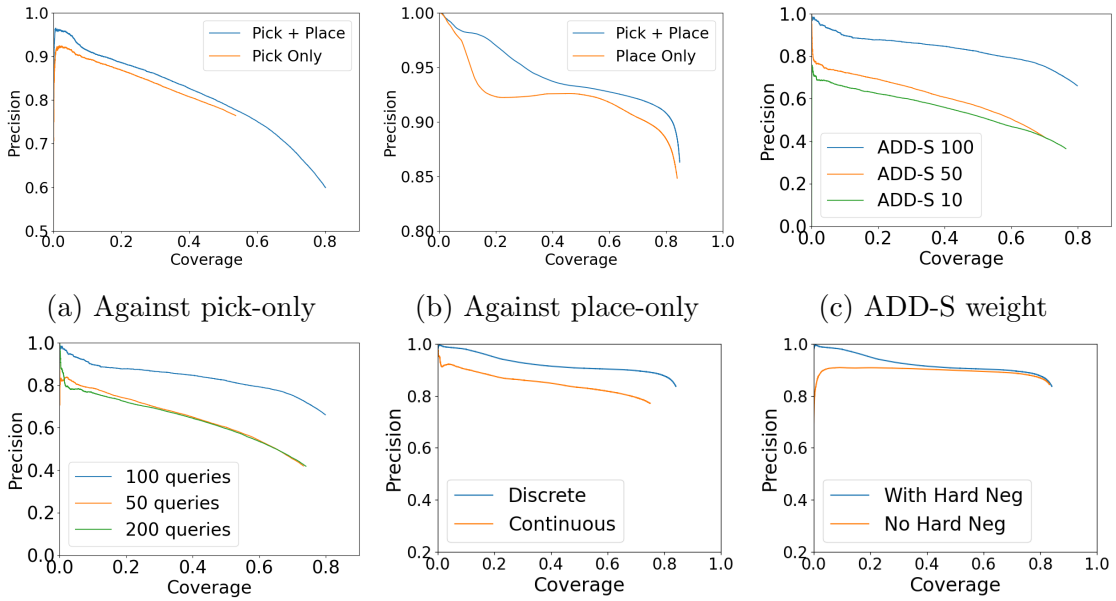
(a) Grasping Seen

(b) Grasping Unseen

(c) Placing Seen

(d) Placing Unseen

Figure 3.4: M2T2 outperforms task-specific models – Contact-GraspNet [85] for grasping and CabiNet [61] for placing – on objects from seen categories (a,c) and unseen categories (b,d).



(a) Against pick-only (b) Against place-only (c) ADD-S weight

(d) Number of tokens (e) Discrete rotation (f) Hard negative mining

Figure 3.5: Ablation studies

as the percentage of ground truth grasps/placements that are within 5 cm of any predicted pose. The coverage is plotted on the x-axis and precision on the y-axis. In practice, we have found that this curve serves as a reliable indicator of a model's performance in real-world scenarios.

A grasp is deemed successful if the gripper does not collide with the scene (including occluded parts) and the grasp remains stable. We evaluate the stability of a grasp by shaking the grasped object for 5 seconds in the Isaac Gym simulator [51] using the *PhysX* physics engine, following the evaluation protocol used in ACRONYM [21]. A placement is considered successful if both the gripper and the object are collision-free and the bottom of the object is less than 5 cm from the correct placement surface.

M2T2 vs. Specialized Baseline Models: We compare M2T2 against two state-of-the-art specialized models: Contact-GraspNet [85] for grasping and CabiNet [61] for placing. The results, summarized in Fig. 3.4, show that M2T2 outperforms both models by a significant margin, especially in the placement task. This demonstrates the advantage of M2T2's orientation reasoning for placement. In many scenarios, achieving a good placement pose requires rotating the object in hand, which M2T2

handles more effectively than the specialized models.

3.5.2 Ablations

Comparison Against Single-task Model: We have trained our model to only perform a single task. These task-specialized models are worse than our multi-task model (see Fig. 3.5a, 3.5b). This shows the importance to formulate both picking and placing under the same framework.

Choice of ADD-S: We find that assigning a larger weight to the ADD-S loss significantly impacts grasping performance. As shown in Fig. 3.5c, reducing the ADD-S weight increases grasp coverage but at the expense of precision, which is not desirable.

Number of Grasp Queries: We experimented with different numbers of grasp query tokens and found 100 to be an optimal number. The results, illustrated in Fig. 3.5d, support this finding.

Discrete vs. Continuous Rotation: We compared our model with a variant where only a single placement mask is used, and placement rotations are regressed similarly to the grasp parameters. Fig. 3.5e demonstrates that having a set of placement masks corresponding to discrete rotations is more effective. Since multiple orientations of an object can be valid for a given placement location, regressing to a single rotation fails to capture the multi-modality of placement orientations.

Importance of Hard Negative Mining: We employ hard negative mining by applying the mask loss to the 1024 points with the largest loss. Without this technique, the quality of the most confident placements deteriorates significantly, as shown in Fig. 3.5f.

3.6 Limitations

M2T2's performance is constrained by the visibility of contact points. For instance, it cannot predict grasps on the side of a box that is not visible to the camera. M2T2

also cannot directly predict actions without contact points, such as lifting. During placing, M2T2 requires segmentation of the object in the gripper to estimate the distance between the gripper and the contact point on the placement surface. Grasp predictions for each token can sometimes be spread across multiple objects in close proximity. Currently, M2T2 is trained and evaluated only on tabletop scenes; however, this could be enhanced by training with more diverse procedurally generated synthetic data as in [25, 61].

3.7 Summary

In this paper, we present Multi-Task Masked Transformer (M2T2), an object-centric transformer model for picking and placing unknown objects in cluttered environments. We train M2T2 on a large-scale synthetic dataset of 130K scenes and deploy it on a real robot without using any real-world training data. M2T2 outperforms state-of-the-art specialized models in 6-DoF grasping [85] and placing [61], achieving an approximately 19% higher overall success rate in real-world scenarios. M2T2 is particularly adept at re-orienting objects for precise, collision-free placements. In future work, we plan to integrate M2T2 with language-conditioned task planners [44, 80] to develop an open-world manipulation system capable of operating in everyday scenes with out-of-distribution objects.

Chapter 4

RoboPoint: A Vision-Language Model for Spatial Affordance Prediction

Spatial reasoning is fundamental to all intellectual processes [89]. Beyond its importance in understanding geometry, science, and architecture [86], spatial reasoning significantly impacts our everyday lives. Even mundane tasks, such as purchasing groceries, require us to identify the vacant space in our shopping carts to load more items. One critical mechanism through which we communicate plans that involve navigation and manipulation is by *pointing*. Studies in developmental psychology demonstrate that infants and adults alike point to share information about their environment [88]. In robotics, pointing has been operationalized through waypoints for navigation and task execution. Roboticists have found that when robots use waypoints effectively, it mimics human pointing, leading to more intuitive plans [18].

Recent explorations have shifted away from pointing in favor of language instructions with the advent of large vision-language models (VLMs)[1, 3, 48]. Trained on large datasets of images and language, VLMs can provide powerful visual semantic understanding and useful guidance for robotic tasks, such as identifying which object a manipulator should pick up or which goal a mobile robot should reach[2, 45, 81]. However, language alone is not precise enough to successfully guide robot behavior. Even the most recent and powerful VLMs, such as GPT-4o [64], have limited accuracy

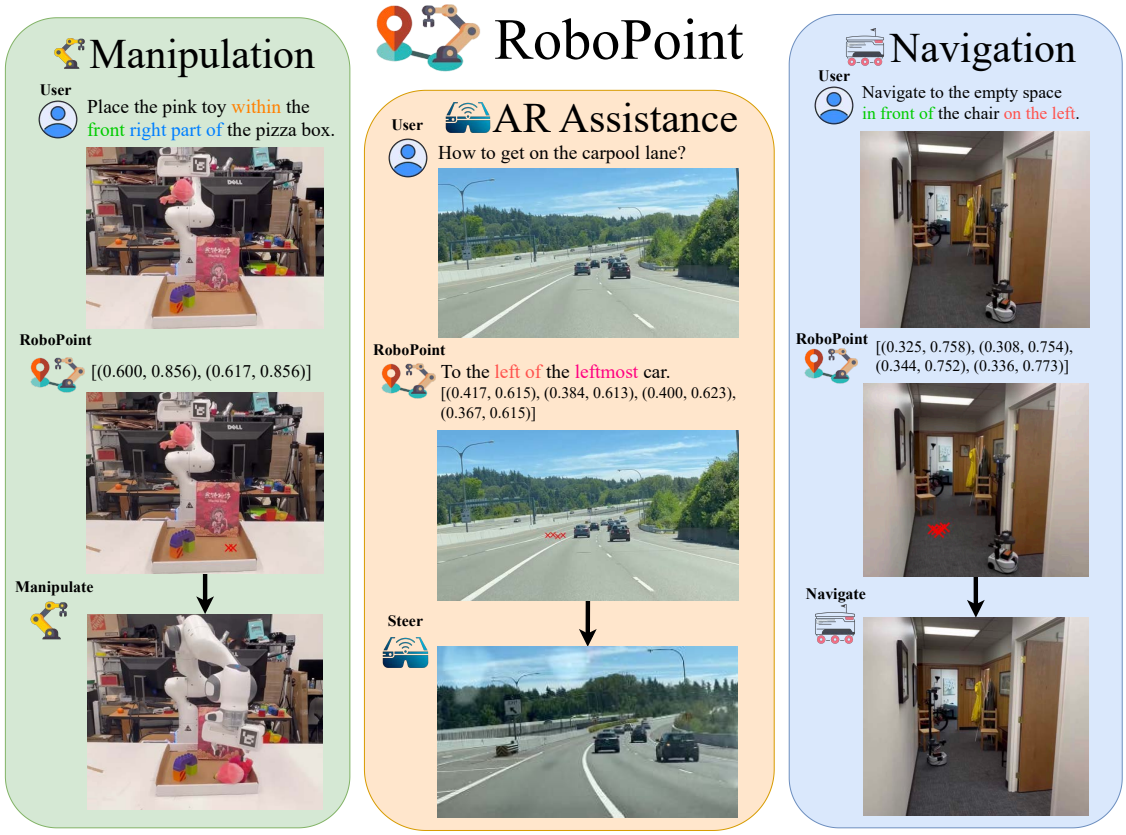


Figure 4.1: RoboPoint is a **Vision-Language Model** that predicts **affordance points** based on language instructions. It is able to generate precise actions (red crosses in the image) which satisfy spatial relations in the instruction. RoboPoint is a generic VLM that can be applied to many domains such as manipulation, augmented reality and navigation.

in real robot execution, especially when language commands involve spatial relations to identify objects or object-free locations, such as "place the cup next to the plate."

In this chapter, we introduce RoboPoint, an open-source VLM instruction-tuned to *point*. Specifically, the instruction tuning process fine-tunes a pre-trained language model to perform *spatial affordance prediction*, the task of pointing at where to act according to language instructions. Two key features differentiate RoboPoint from other VLMs for robotics: a **point-based action space** and a **scalable data pipeline**. First, inspired by prior works [28, 57, 78], actions are specified via points in the RGB image and then transformed to 3D using depth information, eliminating the need for predefined action primitives [45, 81], external object detectors [36, 46], or iterative visual prompting [63]. Second, we design a fully autonomous pipeline that generates a large, diverse dataset of ground truth action points by computing spatial relations from the camera's perspective and sampling points within object masks and object-surface intersections. Compared to approaches that require expensive human demonstration data [5, 87], our pipeline is much easier to scale. Even though we only added data containing simulated images along with templated language, the resulting model's performance improves on real images with natural language commands.

Our results show that RoboPoint significantly outperforms various powerful VLMs such as GPT-4o [64], LLaVA-NeXT [49], Qwen-VL [3], and SpatialVLM [10] on tasks involving relational object reference, free space reference, and object rearrangement in cluttered, real-world environments, without losing accuracy on standard visual question answering (VQA) benchmarks. To evaluate relational free space reference, we collect Where2Place, a manually annotated, challenging real-world benchmark. We also show promising results beyond robotic applications in an interactive augmented reality (AR) setting, where RoboPoint provides visual action suggestions, effectively guiding users through tasks by predicting target points based on common sense.

4.1 The RoboPoint Pipeline

RoboPoint is instruction-tuned from Vicuna-v1.5-13B [12] using a combination of synthetic and real-world data for spatial affordance prediction. This section will cover three critical aspects of the tuning pipeline: 1) the problem formulation, 2) the instruction tuning procedure, and 3) the curation of the data mix.

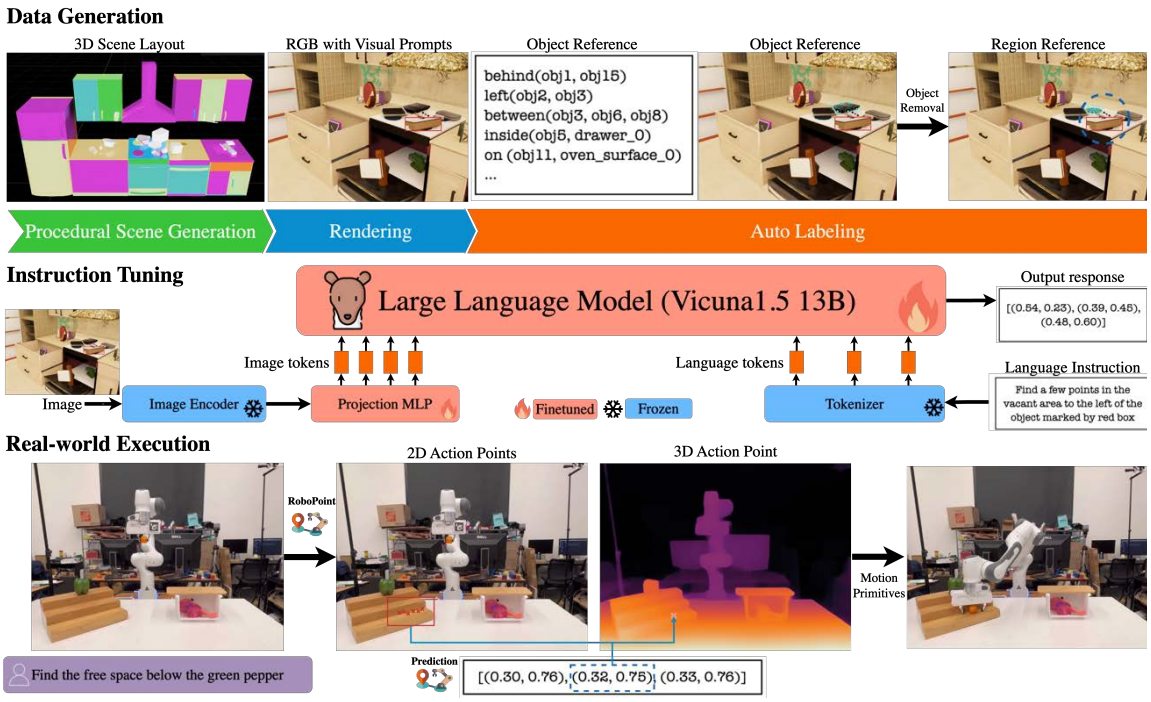


Figure 4.2: **Overview of RoboPoint pipeline.** An RGB image is rendered from a procedurally generated 3D scene. We compute spatial relations from the camera's perspective and generate affordances by sampling points within object masks and object-surface intersections. These instruction-point pairs fine-tune the language model. During deployment, RoboPoint predicts 2D action points from an image and instruction, which are projected into 3D using a depth map. The robot then navigates to these 3D targets with a motion planner.

4.1.1 Spatial Affordance Prediction

We formulate the problem of spatial affordance prediction as predicting a set of target point coordinates $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$ in image space that satisfy the relations indicated by a language prompt. This formulation has several advantages. First, compared to fuzzy language actions such as "place the apple in the drawer," which require detecting the apple and drawer before execution, a point prediction is much more precise and can be directly converted to actions. Most VLMs are trained to predict bounding boxes. However, as shown in Fig. 4.3, bounding boxes often include a lot of undesirable clutter due to camera perspective and are not as specific as point outputs. Second, our formulation is general enough to enable various robotic tasks. For example, the predicted points can be interpreted as waypoints for navigation, contact points for grasping, or region proposals for placement. This versatility allows the model to perform multiple tasks and means it can be trained with multi-task data. Finally, the spatial relations in the language instruction force the model to perform reasoning rather than relying solely on visual features. Half of our synthetic dataset contains examples where the target points are in empty regions without distinct visual cues, ensuring the model cannot rely only on visual features distinct from the background.

4.1.2 Instruction Fine-tuning

Min et al. [56] has shown that in-context learning [7] works by activating patterns from the training data rather than learning new tasks. Therefore, instead of mining patterns from the non-public training dataset, we build our own dataset and fine-tune the language model's parameters. Specifically, we follow the instruction tuning pipeline outlined in Liu et al. [48]. As shown in Fig. 4.2, the model consists of an image encoder, an MLP projector, a language tokenizer, and a transformer language model. The image encoder processes the image into a set of tokens, which are then projected by a 2-layer MLP into the same space as the language tokens. The multimodal tokens are concatenated and passed through the language transformer. All modules are initialized with pre-trained weights, but only the projector and transformer weights are updated, while the vision encoder and tokenizer weights are frozen. The model is autoregressive, and the objective is to predict the response tokens and a special

token delineating the boundary between instruction and response. Our results show that our instruction-tuned model achieves much higher precision than baselines using in-context learning [10, 63].

4.1.3 Co-finetuning with Synthetic Data

Providing the appropriate mix of data is crucial to the model's performance on downstream tasks. As observed by Brohan et al. [6], co-training with a mix of robotic data and internet data ensures the model does not forget the knowledge it has learned during pre-training. Our dataset for fine-tuning consists of four different sources, as illustrated in Table 4.1. The VQA data is a mix of 665K conversations from [47], where the model is asked to answer questions in natural language based on the input image. This ensures the model can reason in language. The LVIS data is converted from [30], where the model is asked to predict bounding box center and dimensions for all instances of a certain category. This teaches the model how to ground language to image regions. The last two data sources, object reference and free space reference, are from our synthetic data pipeline, where the objective is to identify points on an object or in a vacant region that satisfy certain spatial relations. These data enable the VLM to generate precise action points. We formulate the different data sources into the same format and co-train with all of them. Table 4.5 evaluates the importance of each component in our data mix. Notably, VLMs are surprisingly good at combining knowledge learned from different data sources. For example, although our synthetic data uses visual markers to indicate reference objects for spatial affordance prediction, our model can perform the task on real data without markers.

4.2 The RoboPoint Dataset

We generate a diverse dataset in simulation by procedurizing scene layouts, objects, and camera viewpoints. A novel aspect of our pipeline is generating affordance in free space, allowing the model to detect regions without distinct visual cues.

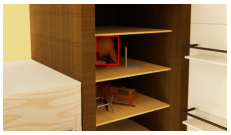
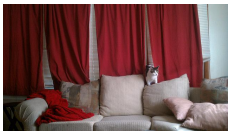
Source	Object Reference	Free Space Reference	VQA	LVIS
				
Quantity	347K	320K	665K	100K
Query	Locate several points on an item situated beneath the bordered item.	Find some spots within the vacant space situated between the marked items.	What is the person feeding the cat?	Find all instances of cushions.
Answer	[(0.56, 0.69), (0.53, 0.76), (0.45, 0.72), ...]	[(0.57, 0.48), (0.58, 0.49), (0.56, 0.45), ...]	The person is feeding an apple to the cat.	[(0.49, 0.38, 0.08, 0.06), (0.53, 0.42, 0.07, 0.05)]

Table 4.1: **Our dataset for instruction-tuning** combines object and space reference data with VQA [47] and object detection data [30]. RoboPoint leverages spatial reasoning, object detection, and affordance prediction from these diverse sources, enabling it to generalize combinatorially. Some answers are not shown in full due to space.

4.2.1 Procedural Scene Generation in Simulation

To train RoboPoint, we generate a large photorealistic dataset in simulation annotated with affordance points. Most existing robotics datasets [20, 53, 76, 91] only have a handful of fixed artist-designed scene layouts, which limits the types of relations that can be generated. We create a diverse dataset by procedurally randomizing several aspects of the scene: the 3D layouts, objects, and camera viewpoints. The scene is represented as an articulated body, including revolute (e.g., fridge, dishwasher doors) and prismatic joints (e.g., cabinet drawers). Objects are sampled from a large repository [21] with over 8K instances and 262 categories. The objects can be placed on any support surface, allowing our model to learn relations in a truly 3D environment.

To ensure that our model generalizes to everyday indoor scenes, we sample assets commonly found in typical kitchen environments (e.g., dishwasher, hood, table, fridge) and use heuristics to place them in random but semantically meaningful layouts. Once the furniture assets are added to the scene, we use a large object dataset sampled from ACRONYM [21]. Object positions are randomly sampled on support surfaces (e.g., countertops, tables), and their orientations are determined by their stable poses. Poses resulting in object collisions with the existing scene are rejected. We place

cameras randomly in the scene and select those with at least three visible objects (visibility is defined as having more than 100 points within the segmentation mask) and at least one valid relationship between a pair of visible objects. The diverse view distribution allows RoboPoint to maintain consistent predictions across different viewpoints. Around 660K (image, relation) pairs are generated from 10K scenes.

4.2.2 Generating Affordance

Once the 3D scene is created, we compute spatial relations among the objects and render an image for each relation from a diverse set of viewpoints in parallel. The relations include left, right, in front, behind, above, below, next to, and in between objects; on, on the left, right, front, and back parts of a surface; and inside a container. Although these relations are templated, the model fine-tuned on this data can generalize to new types of relations. For each relation, there is a target object plus one or two reference objects. We generate two data instances for each relation, one for object reference and one for free space reference.

For object reference, the ground truth is a set of points sampled within the mask of the target object. We then remove the target object and re-render the image for free space reference. The ground truth for free space reference consists of points sampled within the mask of the supporting surface of the target object in the re-rendered image with the target object removed. Instances where the supporting surface is not visible due to occlusion are filtered out. We use furthest point sampling starting from a random point in the mask, resulting in 1 to 50 ground truth points sampled per instance. These sampled points are converted to a list of image coordinates normalized between 0 and 1, which serve as the ground truth response.

A key novelty in our data pipeline is generating affordance in free space. This allows RoboPoint to detect regions without distinct visual cues, such as "the left part of the pizza box" in Fig. 4.5, which an off-the-shelf object detector cannot detect. We employ a simple yet effective strategy: we first compute relations between a target object and another object or surface, then remove the target object, re-render the image, and sample points inside the intersection of the target object's mesh and the supporting surface. This creates affordance labels in free space relative to other entities in the scene.

One caveat of these procedurally generated scenes is that the objects do not have rich text descriptions; most objects only have a category name. We address this by adding visual prompts to the rendered images. Specifically, we draw colored bounding boxes around the objects referenced in the language instruction. Thus, a typical instruction in the synthetic data might be: "There is an object surrounded by a red rectangle in the image. Find some places in the free area to the left of the marked object." Note that we do not add these visual prompts during testing and therefore do not require object detection. The idea is that the model learns to detect objects from other data sources (e.g., LVIS [30]), and it focuses on relational reasoning when dealing with the object and space reference data.

Table 4.2 shows more examples from our procedurally generated synthetic dataset for object reference and free space reference.

4.3 Experimental Results

We demonstrate that RoboPoint achieves superior accuracy in spatial affordance prediction and real-world language-conditioned manipulation than state-of-the-art VLMs [49, 64] and visual prompting methods [10, 63]. Its viewpoint-consistent prediction and conversational ability also enables application to navigation and augmented reality.

4.3.1 Spatial Affordance Prediction

RoboPoint significantly outperforms baselines in terms of accuracy on pointing to objects and free space referred by language. In addition, it generalizes to novel relation types, respects physical constraints, maintains common sense knowledge and produces view-consistent predictions.

Benchmarks We evaluate spatial affordance prediction on two problems: object reference and free space reference. The object reference data is a 750-image subset of RoboRefIt [50]. Unlike human-centered dataset such as RefCoco [97], RoboRefIt features cluttered images with similar-looking objects that can only be distinguished by relational references.

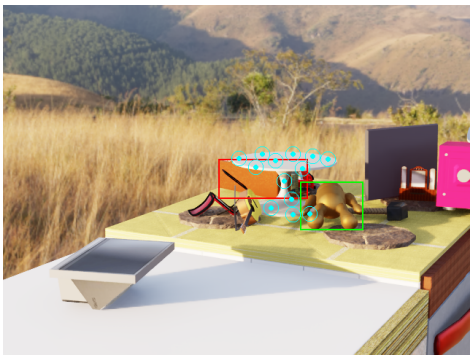
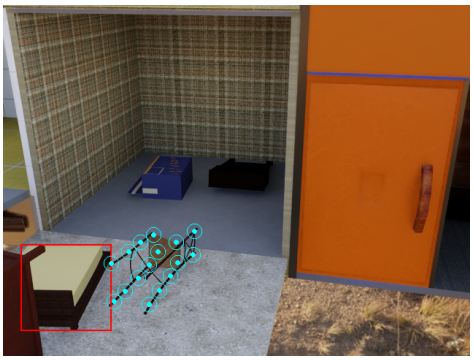
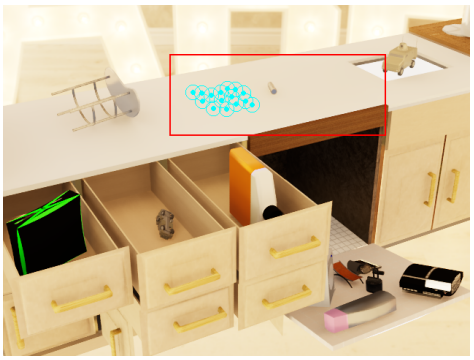
Relation	Between	Inside
		
Prompt	In the image, there is an item framed by a red rectangle and another item encased within a green rectangle. Locate several points upon the item situated between the two highlighted items.	The image depicts a container delineated by a red rectangular border. Pinpoint several spots within the vacant area enclosed by the outlined container.
Relation	Right	On left part
		
Prompt	The image features an object outlined by a red rectangle. Locate several points on an item that is situated on the right side of the marked item.	The image showcases an area demarcated by a red rectangle. Locate a few points within a vacant area on the left side of the marked surface.

Table 4.2: Examples from the synthetic dataset used to teach RoboPoint relational object reference and free space reference. The red and ground boxes are visual prompts to indicate reference objects and the cyan dots are the visualized ground truth (not included in the image inputs to the model).

(a) pepsi can on the middle shelf (b) on the rightmost white plate

Figure 4.3: **Visualization of spatial affordance prediction on objects and free space.** RoboPoint can generalize to (a) unseen relations and (b) scenarios with physical constraints.

Unlike object reference, no existing dataset addresses identifying *free space*. Therefore, we collect Where2Place, a dataset of 100 real-world images from homes and offices in the wild. To minimize bias, we ask one group to label each image with an instruction describing a vacant region relative to other entities, and a different group to label masks according to the instruction. As shown in Fig. 4.3, Where2Place features diverse and challenging scenes with clutter. A subset of 30 examples (Where2Place (h)) contain relation types not in our synthetic data.

Baselines We compare RoboPoint against 3 state-of-the-art VLMs, Qwen-VL [3], LLaVA-NeXT [49], GPT-4o [64] as well as SpaceLLaVa [72], a community implementation of SpatialVLM [10]. We employ a zero-shot visual prompting strategy effective for pretrained VLMs. We label the input image with axes indicating its dimensions and ask the model to output a bounding box (top-left and bottom-right corners) of the target object/region, then sample evenly within the bounding box. For GPT-4o, we also tested in-context learning (GPT-4o-ICL) by providing 14 input-output pairs from our synthetic dataset as context before the query. In-context learning achieved zero accuracy for Qwen-VL and LLaVA-Next, likely because point outputs were not part of their training data.

	RoboRefIt	Where2Place	Where2Place (h)
Qwen-VL	24.08 ± 0.85	10.49 ± 0.77	9.90 ± 0.22
LLaVA-NeXT-34B	19.91 ± 0.92	15.02 ± 0.88	14.76 ± 2.42
SpaceLLaVA	21.30 ± 0.87	11.84 ± 0.73	12.10 ± 1.36
GPT-4o	15.28 ± 1.27	29.06 ± 1.33	27.14 ± 1.47
GPT-4o-ICL	9.01 ± 6.45	14.46 ± 6.38	14.83 ± 4.68
RoboPoint	$\mathbf{49.82 \pm 0.52}$	$\mathbf{46.77 \pm 0.45}$	$\mathbf{44.48 \pm 1.35}$

Table 4.3: **Quantitative comparisons on object reference (RoboRefIt) and free space reference (Where2Place).** RoboPoint outperforms state-of-the-art VLMs by a significant margin, even on examples where the spatial relations are unseen during fine-tuning (Where2Place (h)). The metric is percentage of predicted points within the target mask.

	LLaVA-13B	RoboPoint
GQA	63.24	**63.28**
MME	1522.59	**1524.78**
POPE	85.92	**86.01**
RefCoco	31.99	**32.16**
SEED	67.06	**67.52**
TextVQA	**48.73**	47.31
VizWiz	56.65	**60.37**
VQA-v2	**78.26**	77.83

Table 4.4: **Quantitative evaluation on standard VQA benchmarks.** RoboPoint performs on par with state-of-the-art VLM, maintaining the common sense knowledge learned from pretraining.

No VQA	No LVIS	No Object Ref	No Space Ref	10% Data	All
28.28 ± 2.08	34.27 ± 0.62	42.23 ± 2.28	13.21 ± 1.04	15.71 ± 0.77	$\mathbf{46.77 \pm 0.45}$

Table 4.5: **Ablation on the data composition.** Results on Where2Place show that best results are achieved when all of the data sources are combined during instruction-tuning.

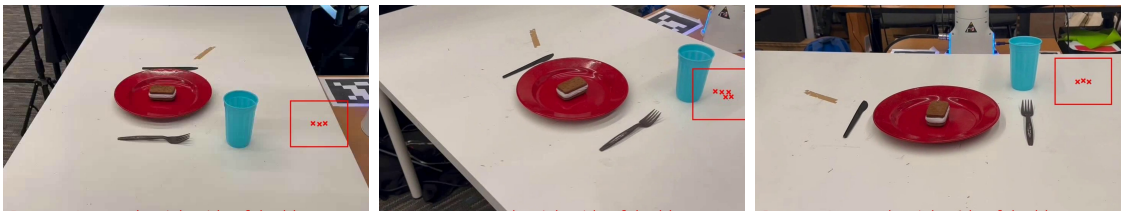


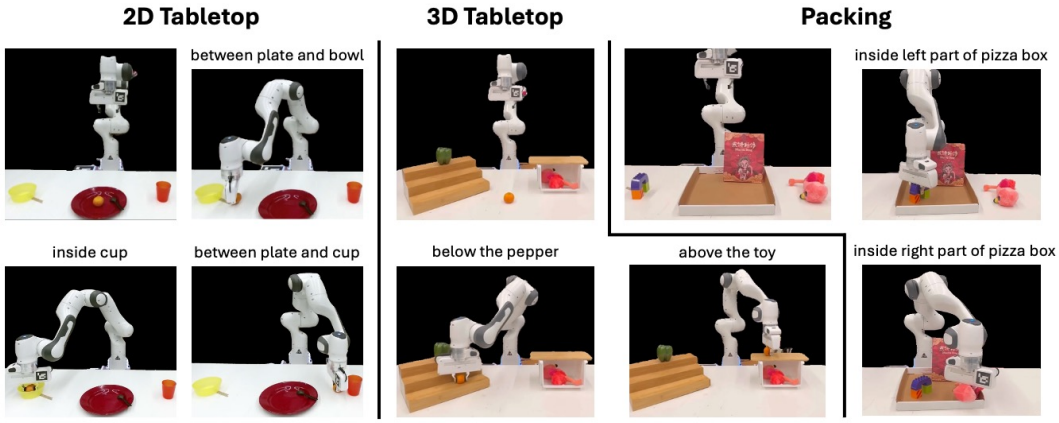
Figure 4.4: **RoboPoint's prediction is consistent across different viewpoints.** Red cross shows RoboPoint's response to "find free space right of the blue cup" in different views.

Results In Table 4.3, we report the average prediction accuracy for RoboPoint and the baselines along with standard deviation computed from 3 different runs. The accuracy is calculated as the percentage of predicted points within the ground truth target mask. We can see that RoboPoint achieves significantly higher accuracy than all baselines, demonstrating the power of RoboPoint in spatial reasoning and precise target generation. Some results are visualized in Fig. 4.3.

Does RoboPoint generalize to unseen relation types? The synthetic dataset we constructed contains templated language and a fixed set of relations. Nevertheless, RoboPoint is able to produce accurate predictions for novel relation types such as in the middle, rightmost etc. that are not in the fine-tuning dataset (Fig. 4.3a). It is also able to maintain its advantage over baselines on these novel relations (Table 4.3).

Does RoboPoint respect physical constraints? RoboPoint's outputs not only satisfy the spatial relations but also respect physical constraints. The target points generated by RoboPoint avoid obstacles such as the the bowl in Fig. 4.3b, whereas the baselines fail to do so.

Does RoboPoint keep common sense knowledge? We evaluate RoboPoint's performance on VQA benchmarks and summarize the results in Table 4.4. RoboPoint performs on-par with LLaVA-v1.5-13B [47], a VLM trained on the same pre-trained weights as RoboPoint on VQA data. This shown that RoboPoint serves a generic VLM rather than a domain-specific model.



	Qwen-VL	GPT-4V	PIVOT	RoboPoint
inside cup	3 / 10	4 / 10	1 / 10	**6** / 10
plate ↔ bowl	2 / 10	1 / 10	1 / 10	**7** / 10
plate ↔ cup	1 / 10	1 / 10	0 / 10	**5** / 10
below pepper	0 / 10	4 / 10	2 / 10	**8** / 10
above toy	1 / 10	2 / 10	2 / 10	**5** / 10
inside left part	2 / 10	1 / 10	1 / 10	**4** / 10
inside right part	2 / 10	0 / 10	0 / 10	**6** / 10

Figure 4.5: **Real-world manipulation evaluation.** We created 7 language-conditioned manipulation tasks to measure RoboPoint's capability on real robot. RoboPoint outperforms the best baseline by 39.5% on average success rate, which depends critically on the alignment between the point predictions and the language.

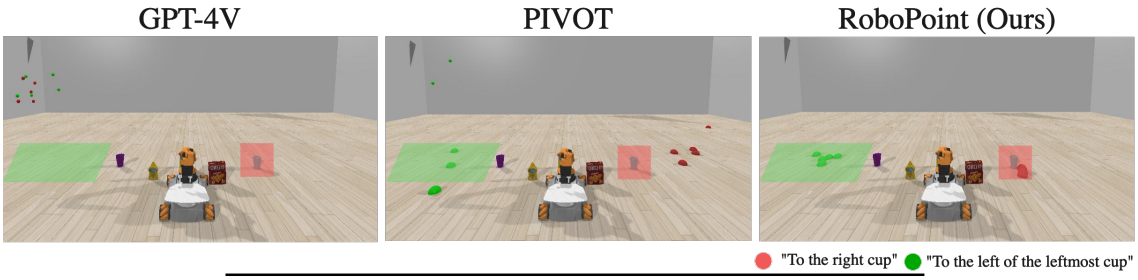
How important is each component in the data mix? In Table 4.5, we evaluated the importance of each data component on the Where2Place benchmark. Each data component – VQA on real images, object detection from LVIS, object and free space reference on synthetic images – significantly contributes to overall accuracy. This highlights the value of a general problem formulation that incorporates diverse data sources. Additionally, data quantity is crucial, as the model's performance drops significantly when fine-tuned on only 10% of the data.

Are RoboPoint's predictions consistent across views? As shown in Fig. 4.4, RoboPoint maintains consistent predictions with camera movement. This makes it particularly suitable for mobile platforms and AR, where RoboPoint provides consistent action suggestions with moving cameras.

4.3.2 Downstream Applications

To assess RoboPoint's capabilities on downstream robotics and vision tasks, we curated various scenarios for manipulation, navigation and AR assistance. We demonstrate RoboPoint's superior performance against state-of-the-art baselines on these tasks.

Real-World Manipulation We set up 3 manipulation environments with 7 tasks (Fig. 4.5). The robot processes image observations and language commands through RoboPoint, which returns 2D point targets. These targets are converted to 3D points using a depth map. The robot's end-effector pose is computed from these 3D points plus an offset. A motion planner then executes the trajectory to the target pose. Success is determined by collision-free execution and accurate placement of the target object as per the language instruction. We conducted 10 trials per task and compared RoboPoint against zero-shot VLMs like Qwen-VL [3] and GPT-4V [1], as well as iterative prompting methods such as PIVOT [63]. RoboPoint surpasses GPT-4V, the best-performing baseline, by a margin of 39.5% on average success rate. It also enables new capabilities. For instance, in the packing scene, RoboPoint's relational reasoning allowed the robot to differentiate regions within a pizza box, fitting multiple objects accurately.



	GPT-4V	PIVOT	RoboPoint
left grill	0 / 5	2 / 5	**5** / 5
front grill	0 / 5	3 / 5	**5** / 5
right grill	0 / 5	2 / 5	**5** / 5
left of left cup	0 / 5	3 / 5	**5** / 5
right cup	0 / 5	4 / 5	**5** / 5
fridge ↔ oven	0 / 5	**3** / 5	1 / 5
oven ↔ drawer	0 / 5	**2** / 5	1 / 5

Figure 4.6: **Application to navigation.** RoboPoint predicts accurate goal point based on language, leading to higher target reaching rate than GPT-4V and PIVOT. Ground truths are drawn as colored masks and predictions are drawn as colored spheres.

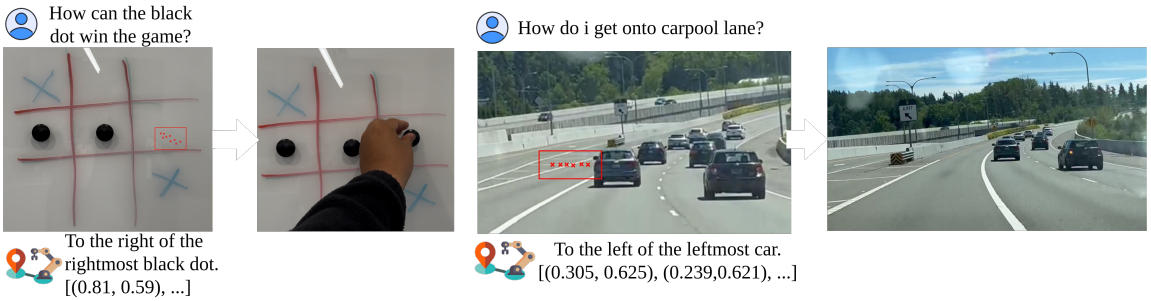


Figure 4.7: **Application to Augmented Reality.** Given a user query, RoboPoint first generates natural language response using common sense and then provide visual guidance using spatial affordance prediction, which the user can execute with greater ease than language guidance.

Navigation To evaluate RoboPoint's spatial affordance predictions beyond tabletop scenarios, we created 3 room scenes using the YouBot mobile manipulation platform in CoppeliaSim [73], where the robot is tasked to navigate to a target region with respect to certain entities in the scene. Fig. 4.6 shows the distribution of affordance generated by RoboPoint, PIVOT [63] and GPT-4V [1] and the success rate of navigating to the correct region using the predicted points with a simple path planner. RoboPoint outperforms PIVOT and GPT-4V in 2 out of 3 scenarios, demonstrating its effectiveness in large-scale room environments for navigation.

Augmented Reality RoboPoint, which is co-trained with VQA data, retains conversational capabilities in natural language. As demonstrated in Fig. 4.1, users can interact with RoboPoint through language and receive action suggestions visually with the predicted affordance. In addition to the *set a formal dining table* task in Fig. 4.1. We demonstrate two more real-world scenarios-*win tic-tac-toe* and *get to carpool lane*-in Fig. 4.7, where RoboPoint gives visual guidance to solve the tasks by predicting the correct spatial affordance points.

4.3.3 Part Affordance Prediction

RoboPoint can also be applied zero-shot to locate object parts based on their function (affordance). Fig. 4.8 shows some examples. (a) is from ManipVQA [35] and (b) is from AffordanceLLM [69].

4.3.4 Failure Mode Analysis

Fig. 4.9 provides additional qualitative comparisons of RoboPoint against baselines on RoboRefIt [50] and Where2Place. It also highlights two common failure modes. First, RoboPoint may fail to detect the correct reference object amidst challenging distractors. As seen in Fig. 4.9e, RoboPoint mistakes the black case for the black lid and incorrectly outputs points between the airpods and the black case. Second, RoboPoint may produce points that meet spatial relations but lack common sense, such as in Fig. 4.9f, where it places points "underneath the monitors" on the floor instead of on the table where a human would logically place them.

(a) Find points on the handle of the power drill. (b) Locate places to sit on the motorcycle.

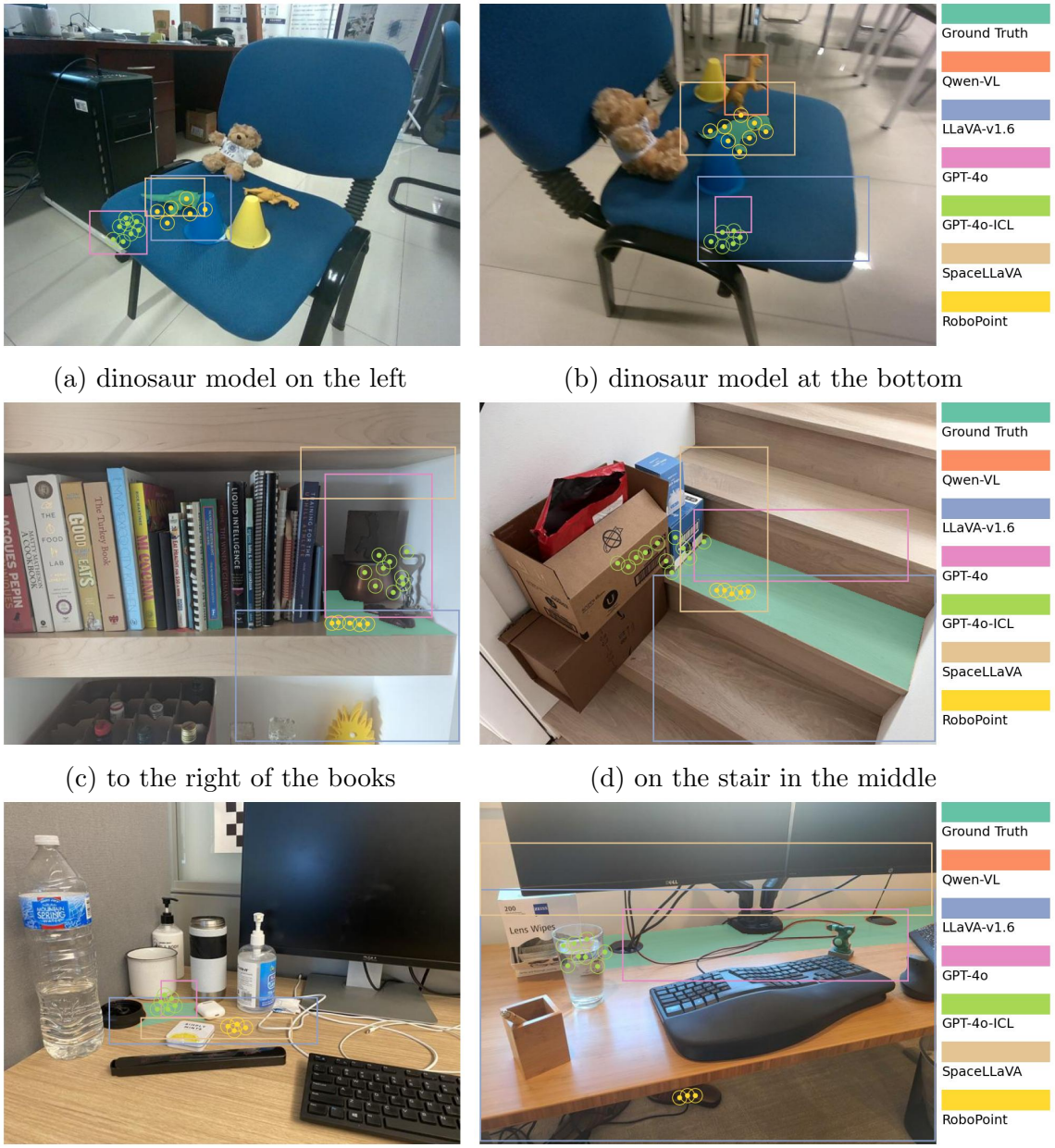
Figure 4.8: Some examples on part-based affordance prediction from (a) ManipVQA [35] and (b) AffordanceLLM [69].

4.4 Limitations

RoboPoint currently does not provide confidence estimates for its point predictions, which can be crucial for assessing the reliability of the suggested actions. Additionally, the number of output points is not controllable, which may limit the flexibility required for specific tasks. Moreover, the predicted points do not include directional information for actions, which could be particularly useful when manipulating articulated objects, as explored in recent works [34, 99]. These are valuable directions to explore in future work to enhance RoboPoint's capabilities.

4.5 Summary

We introduce RoboPoint, a novel Vision-Language Model (VLM) designed to predict spatial affordances in images based on relational language instructions. By integrating real-world Visual Question Answering (VQA) data with automatically generated synthetic data, RoboPoint can generate precise action points that adhere to spatial and physical constraints. This approach overcomes the limitations of current VLMs in robotics, which often depend on pre-defined motion primitives or large-scale expert demonstrations. Experimental results demonstrate RoboPoint's superior performance



(a) dinosaur model on the left

(b) dinosaur model at the bottom

(c) to the right of the books

(d) on the stair in the middle

(e) between AirPods and the black lid

(f) underneath the monitors

Figure 4.9: Qualitative results on RoboRefIt (a, b) and Where2Place (c, d, e, f), including cases with relations unseen during training (d, f) and where GPT-4o performs better (e, f).

in complex tasks, such as relational free space reference and object rearrangement in cluttered environments, compared to state-of-the-art methods. Furthermore, RoboPoint's versatility extends its applicability to augmented reality and robot navigation, showcasing its potential for broader applications in robotics.

Chapter 5

Conclusion

In this dissertation, we presented three innovative models—SORNet, M2T2, and RoboPoint—each addressing different challenges in robotics and computer vision by leveraging advanced spatial reasoning, multi-task learning, and vision-language integration.

SORNet (Spatial Object-Centric Representation Network) focuses on extracting object-centric embeddings from RGB images, enabling precise spatial reasoning for tasks such as spatial relation classification, skill precondition classification, and relative direction regression. SORNet's ability to generalize to novel objects in zero-shot scenarios highlights its robustness and adaptability, making it a valuable tool for complex robotic manipulation tasks.

M2T2 (Multi-Task Masked Transformer) offers a unified solution for handling diverse action primitives in cluttered environments. By predicting collision-free gripper poses for 6-DoF grasping and placing, M2T2 eliminates the need for multiple specialized models. Its superior performance, particularly in re-orienting objects for precise placement, demonstrates the effectiveness of this model in real-world robotic applications. M2T2's success in simulation and real-world deployment underscores the potential of multi-task learning for improving robotic manipulation.

RoboPoint introduces a novel approach to predicting spatial affordances in images based on relational language instructions. By instruction-tuning a vision-language model and integrating a mix of real-world and synthetic data, RoboPoint excels in generating precise action points that satisfy spatial and physical constraints.

Its versatility extends beyond robotic manipulation to applications in augmented reality and navigation, positioning RoboPoint as a powerful tool for a wide range of tasks in robotics and beyond.

Collectively, these models demonstrate significant advancements in the ability to understand and act upon complex spatial relationships in diverse environments. Each model addresses critical gaps in current methodologies, offering new capabilities that enhance the precision, adaptability, and generalization of robotic systems. As we continue to explore these technologies, future work will focus on expanding the models' abilities, such as providing confidence estimates, integrating directional cues, and scaling to even more diverse environments and tasks. The combined potential of SORNet, M2T2, and RoboPoint marks a substantial step forward in the development of intelligent, versatile, and autonomous robotic systems.

5.1 Future Work

The advancements presented in SORNet, M2T2, and RoboPoint illustrate the potential of achieving out-of-distribution generalization through carefully designed object and action representations for learning. However, to build a GPT-like foundation model for robotics that can truly operate in the open world, several challenges remain, particularly in the realm of data collection and model scaling.

Scaling Data Collection A critical aspect of advancing these models is scaling up data collection, particularly in terms of diversity. As we aim to generalize robotic systems to a wide range of environments, it becomes essential to gather data from diverse settings. A mobile platform, combined with capable robotic arms, is indispensable for this purpose. Mobility allows the robot to navigate various environments, while manipulation enables it to interact with those environments in meaningful ways. However, collecting real-world data on a large scale is resource-intensive and costly.

An alternative approach is to leverage video generation models, which are becoming increasingly sophisticated. For example, models like Luma's Dream Machine are beginning to show promise in generating high-quality, albeit imperfect, synthetic data. Despite some inconsistencies—such as objects disappearing or changing unex-

pectedly—these models offer a glimpse into the potential for creating diverse and extensive datasets at a fraction of the cost of real-world data collection. As video generation technology continues to improve, it could play a pivotal role in supplementing real-world data, enabling more scalable and diverse training datasets.

Moving Towards End-to-End Models With access to diverse, large-scale data, the next step is to explore the replacement of traditional hierarchical robotic systems with end-to-end models. While end-to-end models have shown promise, there are still areas where they fall behind traditional systems, particularly in terms of reliability, precision, and handling complex, multi-step tasks. However, with the right problem formulation, careful data curation, and increased scale, these challenges could be addressed.

By focusing on scaling data diversity and harnessing the power of end-to-end learning, future work can bridge the gap between current capabilities and the goal of developing a truly open-world, autonomous robotic system. The integration of mobility and manipulation, along with advances in synthetic data generation, will be crucial in this journey. As we continue to push the boundaries, we remain optimistic that the challenges of today can be overcome, paving the way for a new era of robotics powered by scalable, adaptable, and intelligent models.

Bibliography

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 4, 4.3.2, 4.3.2

[2] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 1.1.4, 3, 4

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 4, 4.3.1, 4.3.2

[4] Stephen Balakirsky, Zeid Kootbally, Craig Schlenoff, Thomas Kramer, and Satyandra Gupta. An industrial robotic knowledge representation for kit building applications. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1365–1370. IEEE, 2012. 1.1.1, 2

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 1.1.3, 4

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 4.1.3

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 4.1.2

[8] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel,

and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015. 2.3

[9] 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. 2.3, 2.3.2

[10] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. *arXiv preprint arXiv:2401.12168*, 2024. 4, 4.1.2, 4.3, 4.3.1

[11] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2former for video instance segmentation. *arXiv preprint arXiv:2112.10764*, 2021. 3.2

[12] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/. 4.1

[13] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4125–4141, 2020. 2.4.1

[14] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking. *IEEE Transactions on Robotics*, 2021. 2

[15] David Ding, Felix Hill, Adam Santoro, and Matt Botvinick. Object-based attention for spatio-temporal reasoning: Outperforming neuro-symbolic models with flexible distributed architectures. *arXiv preprint arXiv:2012.08508*, 2020. 1.1.2

[16] Thanh-Toan Do, Anh Nguyen, and Ian Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *International Conference on Robotics and Automation (ICRA)*, 2018. 1.1.4

[17] 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. 2.1, 2.4.1

[18] Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. Legibility and predictability of robot motion. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 301–308. IEEE, 2013. 4

[19] Yan Duan, Marcin Andrychowicz, Bradly C Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *NIPS*, 2017. 1.1.1, 2

[20] Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Manipulathor: A framework for visual object manipulation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4.2.1

[21] 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. 2.3, 2.3.2, 3.4, 3.3, 3.5.1, 4.2.1

[22] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453, 2020. 1.1.3

[23] Severin Fichtl, Andrew McManus, Wail Mustafa, Dirk Kraft, Norbert Krüger, and Frank Guerin. Learning spatial relationships from 3d vision using histograms. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 501–508. IEEE, 2014. 1.1.2

[24] Richard E Fikes and Nils J Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971. 1.1.1, 2

[25] Adam Fishman, Adithyavairavan Murali, Clemens Eppner, Bryan Peele, Byron Boots, and Dieter Fox. Motion policy networks. In *Conference on Robot Learning*, pages 967–977. PMLR, 2023. 3.6

[26] Maria Fox and Derek Long. Pddl2. 1: An extension to pddl for expressing temporal planning domains. *Journal of artificial intelligence research*, 20:61–124, 2003. 1.1.1, 2

[27] Ali Ghadirzadeh, Atsuto Maki, Danica Kragic, and Mårten Björkman. Deep predictive policy training using reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2351–2358. IEEE, 2017. 1.1.1, 2

[28] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pages 694–710. PMLR, 2023. 4

[29] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 2.4.1

[30] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019. 4.1.3, 4.1, 4.2.2

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2.4.1

[32] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021. 2.4.1

[33] De-An Huang, Suraj Nair, Danfei Xu, Yuke Zhu, Animesh Garg, Li Fei-Fei, Silvio Savarese, and Juan Carlos Niebles. Neural task graphs: Generalizing to unseen tasks from a single video demonstration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8565–8574, 2019. 1.1.1, 2

[34] Siyuan Huang, Haonan Chang, Yuhan Liu, Yimeng Zhu, Hao Dong, Peng Gao, Abdeslam Boularias, and Hongsheng Li. A3vlm: Actionable articulation-aware vision language model. *arXiv preprint arXiv:2406.07549*, 2024. 4.4

[35] Siyuan Huang, Iaroslav Ponomarenko, Zhengkai Jiang, Xiaoqi Li, Xiaobin Hu, Peng Gao, Hongsheng Li, and Hao Dong. Manipvqa: Injecting robotic affordance and physically grounded information into multi-modal large language models. *arXiv preprint arXiv:2403.11289*, 2024. 1.1.4, 4.3.3, 4.8

[36] 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. 4

[37] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. 1.1.3

[38] 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. 1.1.3, 3

[39] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei,

C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. 2.1, 2.4.3

[40] Aishwarya Kamath, Mannat Singh, Yann LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. Mdetr–modulated detection for end-to-end multi-modal understanding. *arXiv preprint arXiv:2104.12763*, 2021. 2.1, 2.4.3

[41] Kei Kase, Chris Paxton, Hammad Mazhar, Tetsuya Ogata, and Dieter Fox. Transferable task execution from pixels through deep planning domain learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10459–10465. IEEE, 2020. 1.1.2

[42] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016. 1.1.1, 2

[43] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018. 2

[44] 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. *arXiv preprint arXiv:2209.07753*, 2022. 3, 3.7

[45] 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. 1.1.4, 4

[46] Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-vocabulary robotic manipulation through mark-based visual prompting. *arXiv preprint arXiv:2403.03174*, 2024. 4

[47] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 4.1.3, 4.1, 4.3.1

[48] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 4, 4.1.2

[49] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next. 4, 4.3, 4.3.1

[50] Yuhao Lu, Yixuan Fan, Beixing Deng, Fangfu Liu, Yali Li, and Shengjin Wang. Vl-grasp: a 6-dof interactive grasp policy for language-oriented objects in cluttered indoor scenes. In *2023 IEEE/RSJ International Conference on*

Intelligent Robots and Systems (IROS), pages 976–983. IEEE, 2023. 4.3.1, 4.3.4

[51] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021. 3.5.1

[52] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. In *The International Symposium of Robotics Research*, pages 132–157. Springer, 2019. 1.1.4

[53] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J Davison. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2678–2687, 2017. 4.2.1

[54] Toki Migimatsu and Jeannette Bohg. Grounding predicates through actions. *arXiv preprint arXiv:2109.14718*, 2021. 1.1.2

[55] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016. 3.3.1, 3.3.1

[56] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022. 4.1.2

[57] Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6813–6823, 2021. 1.1.4, 4

[58] Nathan Morrical, Jonathan Tremblay, Yunzhi Lin, Stephen Tyree, Stan Birchfield, Valerio Pascucci, and Ingo Wald. Nvisii: A scriptable tool for photorealistic image generation. *arXiv preprint arXiv:2105.13962*, 2021. 2.3

[59] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2901–2910, 2019. 1.1.3

[60] Adithyavairavan Murali, Arsalan Mousavian, Clemens Eppner, Chris Paxton, and Dieter Fox. 6-dof grasping for target-driven object manipulation in clutter. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6232–6238. IEEE, 2020. 1.1.3, 3

[61] Adithyavairavan Murali, Arsalan Mousavian, Clemens Eppner, Adam Fishman,

and Dieter Fox. Cabinet: Scaling neural collision detection for object rearrangement with procedural scene generation. *arXiv preprint arXiv:2304.09302*, 2023. 3.4, 3.5.1, 3.6, 3.7

[62] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022. 2.4.1

[63] Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. *arXiv preprint arXiv:2402.07872*, 2024. 4, 4.1.2, 4.3, 4.3.2, 4.3.2

[64] OpenAI. Hello gpt-4o, May 2024. URL https://openai.com/index/hello-gpt-4o. 4, 4.3, 4.3.1

[65] Chris Paxton, Yonatan Bisk, Jesse Thomason, Arunkumar Byravan, and Dieter Foxl. Prospection: Interpretable plans from language by predicting the future. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6942–6948. IEEE, 2019. 1.1.1, 2

[66] Chris Paxton, Nathan Ratliff, Clemens Eppner, and Dieter Fox. Representing robot task plans as robust logical-dynamical systems. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5588–5595. IEEE, 2019. 1.1.1, 2, 2.3

[67] Lerrel Pinto and Abhinav Gupta. Learning to push by grasping: Using multiple tasks for effective learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2161–2168. IEEE, 2017. 1.1.3

[68] 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. 3.2, 3.2

[69] Shengyi Qian, Weifeng Chen, Min Bai, Xiong Zhou, Zhuowen Tu, and Li Erran Li. Affordancellm: Grounding affordance from vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7587–7597, 2024. 1.1.4, 4.3.3, 4.8

[70] Zengyi Qin, Kuan Fang, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Keto: Learning keypoint representations for tool manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7278–7285. IEEE, 2020. 1.1.4

[71] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

2.4.1

[72] Salma Remyx AI (Mayorquin and Terry) Rodriguez. Spacellava, 2024. URL https://huggingface.co/remyxai/SpaceLLaVA. 4.3.1

[73] Eric Rohmer, Surya PN Singh, and Marc Freese. V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pages 1321–1326. IEEE, 2013. 4.3.2

[74] Benjamin Rosman and Subramanian Ramamoorthy. Learning spatial relationships between objects. *The International Journal of Robotics Research*, 30(11): 1328–1342, 2011. 1.1.2

[75] Francesco Rovida, Bjarne Grossmann, and Volker Krüger. Extended behavior trees for quick definition of flexible robotic tasks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6793–6800. IEEE, 2017. 1.1.1, 2

[76] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019. 4.2.1

[77] Mohit Sharma and Oliver Kroemer. Relational learning for skill preconditions. *arXiv preprint arXiv:2012.01693*, 2020. 1.1.2

[78] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2022. 4

[79] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023. 1.1.3

[80] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. *arXiv preprint arXiv:2209.11302*, 2022. 3, 3.7

[81] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530. IEEE, 2023. 1.1.4, 4

[82] Shuran Song, Andy Zeng, Johnny Lee, and Thomas Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations.

Robotics and Automation Letters, 2020. 3

[83] Zhiqiang Sui, Lingzhu Xiang, Odest C Jenkins, and Karthik Desingh. Goal-directed robot manipulation through axiomatic scene estimation. *The International Journal of Robotics Research*, 36(1):86–104, 2017. 1.1.1, 2

[84] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 699–715, 2018. 2

[85] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, 2021. 1.1.3, 3, 3.2, 3.3.1, 3.3.2, 3.4, 3.5.1, 3.7

[86] Holly A Taylor and Barbara Tversky. Spatial mental models derived from survey and route descriptions. *Journal of Memory and language*, 31(2):261–292, 1992. 4

[87] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. 4

[88] Michael Tomasello, Malinda Carpenter, and Ulf Liszkowski. A new look at infant pointing. *Child development*, 78(3):705–722, 2007. 4

[89] Barbara Tversky. What do sketches say about thinking. In *2002 AAAI Spring Symposium, Sketch Understanding Workshop, Stanford University, AAAI Technical Report SS-02-08*, volume 148, page 151, 2002. 4

[90] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 1.1.2, 2.1

[91] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11097–11107, 2020. 4.2.1

[92] 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. 2

[93] Yu Xiang, Christopher Xie, Arsalan Mousavian, and Dieter Fox. Learning rgb-d feature embeddings for unseen object instance segmentation. *arXiv preprint arXiv:2007.15157*, 2020. 2

[94] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022. 2.4.1

[95] Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. The best of both modes: Separately leveraging rgb and depth for unseen object instance segmentation. In *Conference on robot learning*, pages 1369–1378. PMLR, 2020. 2

[96] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *arXiv preprint arXiv:1810.02338*, 2018. 2.4.3

[97] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016. 4.3.1

[98] Tianhe Yu, Pieter Abbeel, Sergey Levine, and Chelsea Finn. One-shot hierarchical imitation learning of compound visuomotor tasks. *arXiv preprint arXiv:1810.11043*, 2018. 1.1.1, 2

[99] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. *arXiv preprint arXiv:2401.11439*, 2024. 4.4

[100] Wentao Yuan, Chris Paxton, Karthik Desingh, and Dieter Fox. Sornet: Spatial object-centric representations for sequential manipulation. In *Conference on Robot Learning*, pages 148–157. PMLR, 2022. 1.1.2

[101] Wentao Yuan, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. M2t2: Multi-task masked transformer for object-centric pick and place. *arXiv preprint arXiv:2311.00926*, 2023. 1.1.3

[102] Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024. 1.1.4

[103] Honglu Zhou, Asim Kadav, Farley Lai, Alexandru Niculescu-Mizil, Martin Renqiang Min, Mubbasir Kapadia, and Hans Peter Graf. Hopper: Multi-hop transformer for spatiotemporal reasoning. *arXiv preprint arXiv:2103.10574*, 2021. 1.1.2