# Efficient Alignment of Unconditioned Action Prior for Language-conditioned Pick and Place in Clutter

**Kechun Xu**[1,2], **Xunlong Xia**[2], **Kaixuan Wang**[1], **Yifei Yang**[1],
**Yunxuan Mao**[1], **Bing Deng**[2], **Jieping Ye**[2], **Rong Xiong**[1], **Yue Wang**[1]
[1]Zhejiang University, [2]Alibaba Group
{kcxu, kaixuanwang, yfyang2018, maoyunxuan, rxiong, ywang24}@zju.edu.cn
{xunlong.xxl, dengbing.db, jieping.yjp}@alibaba-inc.com

**Abstract:** We study the task of language-conditioned pick and place in clutter, where a robot should grasp a target object in open clutter and move it to a specified place. Some approaches learn end-to-end policies with features from vision foundation models, requiring large datasets. Others combine foundation models in a zero-shot setting, suffering from cascading errors. In addition, they primarily leverage vision and language foundation models, focusing less on action priors. In this paper, we aim to develop an effective policy by integrating foundation priors from vision, language, and action. We propose $A^2$, an action prior alignment method that aligns unconditioned action priors with 3D vision-language priors by learning one attention layer. The alignment formulation enables our policy to train with less data and preserve zero-shot generalization capabilities. We show that a shared policy for pick and place actions enhances performance for each task, and introduce a policy adaptation scheme to accommodate the multi-modal nature of actions. Extensive experiments in simulation and the real-world show that our policy achieves higher task success rates with fewer steps, effectively generalizing to unseen objects and language instructions.

**Keywords:** Language-conditioned Pick and Place, Action Prior Alignment, Foundation Models for Robotic Manipulation

## 1 Introduction

The ability to pick and place objects is essential for robotic manipulation [1, 2, 3, 4, 5, 6]. Consider a scenario where a robot is commanded with language instructions to grasp a target object in open clutter, and move it to a specified place. The target object may be partially or fully occluded, posing challenges for object grounding and grasping. In such scenarios, multiple pick and place actions may be needed to clear obstacles for object rearrangement.

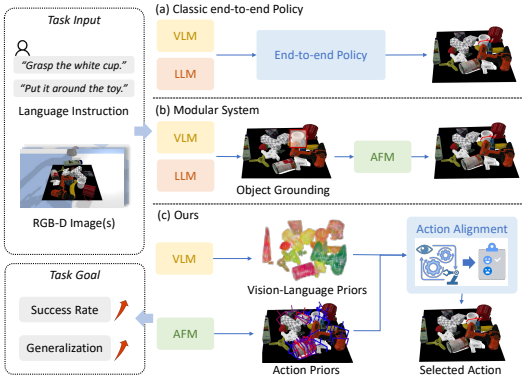



Figure 1: Compared to previous methods (a) classic end-to-end policies and (b) modular systems, our method integrates foundation priors from vision, language, and action through alignment by one attention layer, which enables more efficient policy learning and better task performance.

A common way to construct a policy for such tasks is to predict 6-DoF actions directly from raw sensory information, as in classic end-to-end policies. Recently, these policies have achieved promising performances by incorporating features of pre-trained foundation models, *e.g.*, vision-language models (VLM) and large language models (LLM) [7, 8, 9, 10, 11, 12]. However, they require large amounts of demonstration

data for policy learning, particularly for tasks involving cluttered environments. In addition, one has to deal with generalization issues to deploy these policies in real-world applications.

In contrast, other methods harness the zero-shot generalization capabilities of foundation models by developing modular systems. Many works investigate visual representations for object grounding, followed by rule-based action planners for object manipulation [13, 14, 15, 16, 17, 18]. For example, LERF-TOGO [19] builds 3D scene representations by distilling features from vision-language models, then performs object grounding to filter candidate actions generated by an action foundation model. These approaches are mostly learning-free, showcase zero-shot generalization, and utilize action candidates as priors. Nevertheless, they demand high accuracy in visual grounding, which remains challenging in cluttered settings. Even with correct grounding, the target in clutter may be ungraspable. Some works employ large language models as planners to decide the object grasp order in clutter, but still suffer from cascading errors across individual modules [20, 21, 22].

In general, end-to-end methods require large datasets to effectively learn a policy with substantial network parameters and pay less attention to action priors, whereas modular systems struggle with cascading errors when combining several foundation models in a zero-shot setting. Considering that action foundation models can provide action priors that are unconditioned on specific tasks, we raise a question: *Given unconditioned action priors, is there a policy that can improve performance while learning fewer network parameters?*

To leverage unconditioned action priors in specific tasks, we adopt the idea of *alignment* with a reward model, inspired by the RLHF technique in large language model training [23]. Taking action foundation models as generators, we build a probabilistic policy upon the generated actions for reward alignment. For specific pick and place tasks, the reward model can be defined as a simple binary function. Then, expert demonstrations can be extended into state-action pairs with binary scores. In this way, we can learn the policy that aligns with the task reward by maximizing the probabilities of demonstrated actions through imitation learning.

Guided by the insights, we propose $A^2$, an **A**ction Prior **A**lignment method that aligns unconditioned action priors based on task-conditioned vision-language priors by *learning one attention layer* (Figure 1). Action foundation models, such as GraspNet [24], generate action candidates, providing unconditioned action priors and largely reducing the action space. For vision and language input, we construct 3D zero-shot representations combining vision-language foundation priors from the vision-language model MaskCLIP [25]. Based on these priors, we perform alignment by a cross-attention layer to predict action probabilities for planning. In this way, our policy is to learn one-dimensional probabilities over action priors, requiring less training data and preserving zero-shot generalization capabilities. To learn such a policy, we construct a score-based dataset from expert demonstrations. We use shared network parameters for pick and place tasks, improving performance simultaneously for each task. We also propose a fast policy adaptation scheme, allowing fine-tuning for action multi-modality modeling. At inference time, our policy aligns actions across the scene to predict a sequence of grasps to remove obstacles for target grasping, and ultimately place the target at the specified location. A wide range of experiments in both simulation and real-world settings show that our policy achieves higher task success rates with fewer planning steps, with zero-shot generalization to unseen objects and language instructions. Our main contributions are:

- We propose $A^2$, an efficient action prior alignment method that allows learning one attention layer for language-conditioned pick and place in clutter.
- We leverage the vision-language model to construct 3D vision-language priors that indicate task information with zero-shot generalization capability.
- We conduct alignment of unconditioned action priors based on vision-language priors from foundation models, and develop fast policy adaptation for action multi-modality modeling.
- The learned policy is evaluated on a series of scenarios with seen and unseen objects and language instructions in both simulated and real-world settings, of which the results validate the effectiveness and generalization.

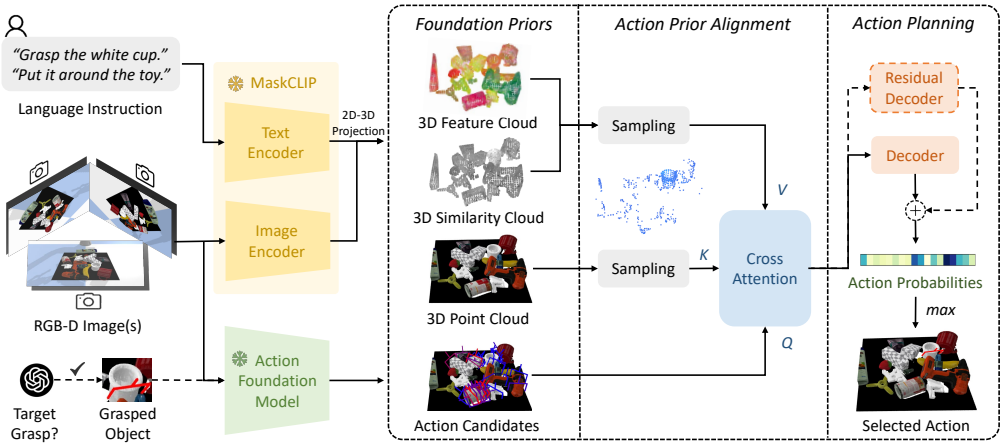



Figure 2: **Overview**. Given the language instruction and RGB-D image(s), the vision-language model MaskCLIP [25] extracts dense patch-level features, which are projected into 3D representations, including a feature cloud, a similarity cloud, and a point cloud. In addition, the action foundation model generates action candidates. Based on these foundation priors, our policy conducts alignment for action planning.

## 2 Overview

**Unconditioned Action Priors based Policy.** Given the RGB-D image(s) $\mathcal{I} = \{I_i\}_{i=0,1,\ldots,M}$ and the language instruction $\mathcal{L}$, we leverage foundation models to extract vision, language, and action priors. Consider an action foundation model that generates $L$ candidate actions from image(s) as action priors $\mathcal{A}_L(\mathcal{I}) = \{a_k\}_{k=0,1,\ldots,L}$, $L$ generally has a controllable upper limit. These priors, distilled from a wide range of unconditioned data, provide feasible action patterns for downstream tasks and largely narrow the action space. Upon these priors, we construct a probabilistic policy $\pi$.

$$\begin{aligned} \pi\left(a|\mathcal{I},\mathcal{L}\right) &= \sum_{k=1}^{L} \omega\left(a_k|\mathcal{I},\mathcal{L}\right)\delta\left(a-a_k\right) \\ \text{s.t.} \quad & \sum_{k=1}^{L} \omega\left(a_k|\mathcal{I},\mathcal{L}\right) = 1 \end{aligned} \tag{1}$$

where $\omega\left(a_k|\mathcal{I},\mathcal{L}\right)$ is the probability of $a_k$ conditioned on the vision and language information.

**Alignment with Reward.** Modular systems obtain $\omega$ with rule-based filtering upon visual grounding results, which demands high visual accuracy. Instead, we propose to learn the $\omega$ to align unconditioned action priors based on vision-language priors. In this way, our policy is to learn one-dimensional probabilities over action priors, largely alleviating data demands. Consider this alignment problem via RL objective, let $r(a,\mathcal{I},\mathcal{L})$ denote the reward function, then the optimal policy is to maximize the expected sum of future rewards. For pick and place tasks, $r(a,\mathcal{I},\mathcal{L})$ can be easily defined as

$$r(a,\mathcal{I},\mathcal{L}) = \begin{cases} 1, & \text{pick or place successfully} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

**Alignment by Imitation Learning.** By employing expert planners, we can collect demonstrations $\mathcal{D} = \{\mathcal{I}_d, \mathcal{L}_d, a_d\}$, where $a_d \in \mathcal{A}_L(\mathcal{I}_d)$. Then we have $r(a_d,\mathcal{I}_d,\mathcal{L}_d) = 1$. Therefore, we can augment each demonstration into score-based samples by labeling $a_d$ as 1, with the remaining ones in $\mathcal{A}_d$ as 0. In this way, we can learn the policy $\pi$ that aligns with the reward by maximizing the likelihood of $a_d$ for $\mathcal{I}_d, \mathcal{L}_d$ through imitation learning.

$$\max_{a_d \in \mathcal{A}_L(\mathcal{I}_d)} \omega\left(a_d|\mathcal{I}_d,\mathcal{L}_d\right) \tag{3}$$

**Architecture of A$^2$.** Figure 2 presents our pipeline. For vision-language input, our method extracts dense patch-level features using MaskCLIP [25]. Then the features are projected into 3D representations, including a 3D point cloud, a 3D feature cloud, and a 3D similarity cloud. Each coordinate

in the point cloud corresponds to a visual feature and a task-relevant vision-language similarity. Additionally, we utilize the action foundation model to yield a set of action candidates. Based on the vision, language, and action priors, we propose to conduct action prior alignment for action planning. We first sample points with higher similarity to create a more compact representation. Then, a cross-attention transformer takes action features as queries, 3D position features as keys, and 3D vision-language features as values to align action priors with vision-language information. The output fusion features are fed into a decoder to get the probabilities of candidate actions.

As a system, our policy first receives the language instruction to grasp the target object, and predicts a sequence of grasp actions by closed-loop action alignment. If the target object is not grasped, our policy will remove the grasped obstacles and proceed regrasping. Once the target is grasped, our policy takes the language instruction of placement along with the grasped object for place prediction, finally placing the grasped object in the assigned location.

# 3 Foundation Priors

## 3.1 3D Vision-Language Priors

We leverage the zero-shot generalization capability of foundation models to construct 3D visual representations that convey semantic and task-relevant information.

**Generalizable Visual-Language Features.** We extract features through the pre-trained vision-language model CLIP [26] that maps visual and language embeddings by training on millions of image-text data. However, CLIP originally generates image-level features. To obtain denser features, we apply MaskCLIP [25] reparameterization trick to extract patch-level features from CLIP. To further get more fine-grained features, we crop each RGB image into several sub-images to extract patch-level features, and concatenate them together to form the final visual feature map.

**3D Representations.** Given RGB-D image(s) $\mathcal{I} = \{I_i\}_{i=0,1,...,M}$ from one or more cameras with fixed viewpoints, we first extract a 3D point cloud $\mathbf{p}$ within the workspace using the camera parameters. For each point $p_j$ of $\mathbf{p}$, we project it back to $i$th camera viewpoint as the pixel $u_j^i$, and get its visual feature $f_j^i$ by interpolation. Following [18], we compute weights for each camera according to the visibility and distance of $p_j$ in the corresponding camera. Finally, we fuse features from all camera viewpoints using a weighted sum, denoted as $f_j$. More details can be accessed in Appendix.

**3D Feature Cloud.** Each point $p_j$ within the workspace paired with its feature $f_j$, forms the 3D feature cloud $\mathbf{f}$. This representation implies the visual information of the scene, which is semantic and zero-shot generalizable.

**3D Similarity Cloud.** To represent the task-relevant information, we further utilize the vision-language similarity property of MaskCLIP. Specifically, the language instruction is encoded by the MaskCLIP text encoder. For each point $p_j$, we compute the cosine similarity between the language embedding and the visual feature $f_j$ to get a similarity value $s_j$, resulting in a 3D similarity cloud $\mathbf{s}$. This representation reflects the degree of task relevance of each point.

## 3.2 Unconditioned Action Priors

**Action Foundation Models.** We employ different action foundation models to yield candidate actions for pick and place respectively. For object picking, we adopt the pre-trained GraspNet [24] to generate 6-DoF grasp poses that demonstrate feasible grasp actions for all objects across the whole scene. For object placement, we first obtain all the object region proposals, then place poses are sampled in and around each object region without overlapping with each other.

**Action Candidates.** By utilizing action foundation models, we yield a set of $L$ candidate actions $\mathcal{A}_L(\mathcal{I}) = \{a_k\}_{k=0,1,...,L}$. $L$ generally has a controllable upper limit and is variable in different scenarios. These candidate actions provide unconditioned priors of the way to manipulate objects and largely narrow the action space into a limited set, facilitating efficient policy learning.

# 4 Action Prior Alignment

Based on foundation priors vision, language, and action, we propose to conduct action prior alignment by learning one attention layer.

## 4.1 Alignment Architecture

Considering that directly taking complete 3D representations is sample-inefficient, we first conduct prioritized sampling to get a more compact representation. To be specific, we sample $N$ points with higher similarities to generate sampled 3D representations $\mathbf{p}_N$, $\mathbf{f}_N$ and $\mathbf{s}_N$. Note that $N$ is a hyperparameter closely related to the total number of 3D points in the representations. Empirically, we sample half of the points from the workspace. Given the sampled 3D visual representations and the generated action candidates, we perform action prior alignment via cross-attention to obtain fusion features, followed by a decoder to predict the action probabilities. Finally, the action with the highest probability is selected for execution.

## 4.2 Cross Attention

We propose to align unconditioned action priors based on task-conditioned vision-language priors. To be specific, we employ transformer's attention mechanism [27]: $\text{Attention}(Q, K, V) = \text{Softmax}\left(QK^T\right)V$, where $Q, K, V$ denote query, key and value respectively.

We weight the 3D visual features $\mathbf{f}_N$ with the similarity values $\mathbf{s}_N$, which capture the vision-language information. We encode $L$ action pose features by an MLP to generate action features. The 3D points $\mathbf{p}_N$ are projected into a nonlinear space using positional embedding as in [28], followed by an MLP to encode position features. To align action features based on vision-language information, the cross-attention transformer takes $L$ action pose features as queries, $N$ position features as keys, and $N$ vision-language features as values, outputting $L$ fusion features $\mathcal{F}_L$. We use RoPE [29] to encode relative position embeddings for keys and values.

$$Q = \text{MLP}_1\left(\mathcal{A}_L\right),\ K = \text{RoPE}\left(\text{MLP}_2\left(\mathbf{p}_N\right)\right),\ V = \text{RoPE}\left(\mathbf{f}_N \circ \mathbf{s}_N\right) \tag{4}$$

## 4.3 Policy Learning

**Shared Policy for Pick and Place.** We train the policy with demonstration data collected by model-based expert planners, and propose to train a policy for pick and place with shared parameters. That is, after generating the 3D representations and candidate actions, pick and place share the same information for action alignment. This is because there is strong common information between pick and place actions. In cluttered scenes, both pick and place tasks require the policy to focus on the regions close to the target.

**Policy Adaptation for Multi-modality.** For both pick and place tasks, the action distribution is inherently multi-modal. In particular, for place tasks, the multi-modal characteristic is more significant, *e.g.* when placing around an object, there may be several feasible actions. However, due to the difficulty of executing all actions in each step, the demonstration data labels only one action as the ground truth. This potentially misleads the policy and degenerates the multi-modality modeling of actions. To address this issue, we propose a policy adaptation scheme using a residual block:

$$\Omega_L = \text{Decoder}\left(\mathcal{F}_L\right),\ \Omega_L^r = \text{Decoder}^r\left(\mathcal{F}_L\right),\ \Omega_L' = \alpha\Omega_L + (1-\alpha)\Omega_L^r \tag{5}$$

where $\Omega_L = \{\omega_k\}_{k=1}^L$ represents the original predicted action probabilities of $\mathcal{A}_L$, $\Omega_L^r$ is the residual output of probabilities, and $\Omega_L'$ is the weighted sum of $\Omega_L$ and $\Omega_L^r$. By fine-tuning the policy with a small set of multi-labeled data of place tasks, we can further improve the policy performance.

# 5 Experiments

In this section, we carry out a series of experiments to evaluate our policy. The goals of the experiments are: 1) to validate the effectiveness of our policy in both language-conditioned pick and place

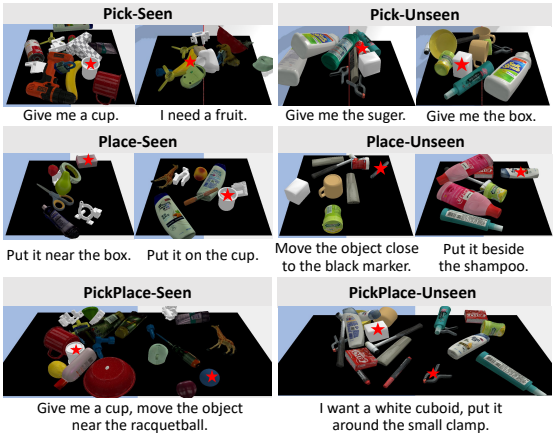



Figure 3: Example test cases in simulation. The target and reference objects are labeled with stars.

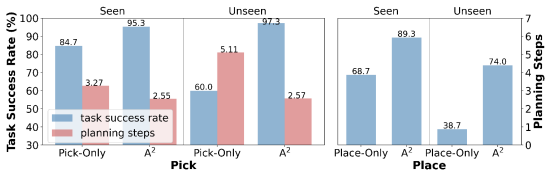



Figure 4: Ablation studies of shared policy.

| Category | Method | Seen | Unseen |
|---|---|---|---|
| Pick | LERF-TOGO | 83.3/3.37 | 76.0/2.01 |
| | GraspSplats | 58.0/**2.05** | 37.3/**1.67** |
| | VLG | 74.3/4.11 | 78.7/3.98 |
| | ThinkGrasp | 84.7/2.55 | 57.3/4.11 |
| | A$^2$-G-Pick | 83.3/3.78 | 84.7/3.85 |
| | A$^2$ | **95.3**/2.55 | **97.3**/2.57 |
| Place | VLP | 40.0 | 20.0 |
| | A$^2$-G-Place | 32.3 | 29.3 |
| | A$^2$ | **89.3** | 74.0 |
| | A$^2$-PA | 89.0 | **76.0** |
| Pick-n-Place | Act3D | 0.0/– | 0.0/– |
| | RVT-2 | 0.0/– | 0.0/– |
| | Act3D$^\dagger$ | 0.0/– | 0.0/– |
| | RVT-2$^\dagger$ | 0.83/4.00 | 0.0/– |
| | 3D Diffuser$^\dagger$ | 1.67/6.13 | 0.0/– |
| | A$^2$-G | 30.7/2.42 | 28.3/**2.00** |
| | A$^2$ | 87.5/2.45 | 71.7/3.02 |
| | A$^2$-PA | **91.3/2.06** | **76.7**/3.22 |

* Metrics of pick and pick-n-place are presented as Task Success Rate / Planning Steps.

Table 1: Simulation Results on All Categories

tasks in clutter; 2) to demonstrate the efficiency of our policy; 3) to validate the zero-shot generalization performance of our policy on unseen objects and language instructions; 4) to test whether our policy can successfully transfer to the real world.

### 5.1 Experimental Setup

**Test Settings.** We first conduct test experiments in simulation with a series of test cases, which can be categorized into three folds: pick, place, and pick-n-place. Each category includes cases of arrangements with both seen and unseen objects during A$^2$ training. For place, some cases of unseen objects pair with unseen relations. Example cases are visualized in Figure 3.

**Evaluation Metrics.** We evaluate the methods with a series of test cases. Each test contains $y = 15$ runs measured with 2 metrics:

- **Task Success Rate**: the average percentage of task success rate over $y$ test runs. For pick, if the robot picks up the target object within 8 action attempts, the task is considered successful and completed. For place, the robot succeeds if placing the object in the correct region with 1 action attempt. For pick-n-place, the robot should simultaneously succeed in both pick and place tasks.
- **Planning Steps**: the average pick or pick-n-place number per task completion. Note that this metric is only evaluated in the categories of pick and pick-n-place.

### 5.2 Baselines

We compare our policy A$^2$ to various baselines, including both modular systems and classic end-to-end policies. For modular systems, we compare to neural field based pick policies (LERF-TOGO [19], GraspSplats [14]), object-centric pick and place polices (VLG [30], ThinkGrasp [22] for picking and VLP [31] for placing), and 3D visual grounding pick and place policies. For end-to-end policies, we compare to 3D policies Act3D [11], RVT-2 [12] and 3D Diffuser [32].

### 5.3 Comparison to Baselines

**Pick.** Results in Table 1 indicate that our policy outperforms all baselines. Although LERF-TOGO and GraspSplats can obtain fine-grained scene representations via time-consuming (>1min) test-

time training, the grounding accuracy is hindered in clutter, leading to cascaded errors in action planning. Therefore, they demonstrate unsatisfactory performances. Other methods support real-time inference. VLG gets object awareness by incorporating object-centric representation, but suffers from detection noise. ThinkGrasp utilizes GPT-4o as the planner based on object-centric crops, which inherits the reasoning capability of LLM. Nevertheless, it operates in a stage-by-stage manner, affected by the accuracy of segmentation and LLM planning, resulting in more planning steps for some fuzzy concepts. $A^2$-G-Pick relies on the similarity cloud for grounding, and ignores the probability of moving away obstacles. In contrast, action prior alignment enables our policy to directly score actions based on task-relevant vision-language features. In this way, our policy avoids over-reliance on accurate visual representations and can remove obstacles for target grasping.

**Place.** We show the performances of place with seen and unseen objects in Table 1, further demonstrating the advantages of our action prior alignment paradigm. The performance of VLP depends heavily on the capability of CLIP, which frequently fails when facing similar visual information or text words. $A^2$-G-Place struggles to distinguish "in" and "around" relation, as it directly grounds the highest point that fits both requirements of reference and relation.

**Pick-n-Place.** As shown in Table 1, Act3D, RVT-2 fail in all cases when employing their pre-trained models, revealing poor generalization to novel objects, backgrounds, and camera viewpoints. Even when trained on our dataset, Act3D$^\dagger$, RVT-2$^\dagger$, and 3D Diffuser$^\dagger$ still struggle to acquire the necessary information to complete tasks, likely due to insufficient data quantity. By further leveraging action foundation priors and aligning them based on zero-shot vision-language priors, our policy achieves higher efficiency and generalization.

**Generalization.** All policies are tested with objects seen and unseen during $A^2$ training. Overall, our policy achieves the highest task success rates in unseen objects, particularly excelling in pick tasks. Thanks to our design of action prior alignment desgin, we effectively preserve the generalization capabilities of the foundation models to a large extent.

### 5.4 Ablation Studies

We conduct extensive ablation studies to elucidate the effectiveness of individual designs within our method. For a fair comparison, all the learning-based methods are trained with the same process.

**Shared Policy.** We first test the effectiveness of the shared policy for pick and place. Let Pick-Only denote the policy trained only with pick samples and Place-Only as the policy trained only with place samples. Results in Figure 4 demonstrate the shared policy boosts performances in both tasks by a large margin. This indicates strong commonalities between pick and place tasks, as both require focus on or around the target region.

**Policy Adaptation.** To validate the policy adaptation scheme, we compare the performances of our policy before ($A^2$) and after adaptation (namely $A^2$-PA) with only 100 multi-labeled place samples. It can be seen from Table 1 that our policy adaptation scheme can improve the generalization performances in both place and pick-n-place tasks. It is interesting to note that $A^2$-PA outcomes fewer planning steps for pick-n-place tasks involving seen objects, suggesting that policy adaptation on place data also facilitates the efficiency of picking. This might be because fine-tuning with multi-labeled data brings multi-modal characteristics, better fitting the true action distribution. And multi-modality is a commonality of pick and place actions.

**Novel Camera Viewpoints.** We vary the camera viewpoints at test time and present the results in Figure 5. It is shown that our policy can generalize to novel camera viewpoints. This benefits from our zero-shot 3D representation, which does not impose strict constraints on camera viewpoints. As a result, our policy remains effective and practical for deployment in new scenarios with varying camera configurations, offering greater flexibility for real-world applications.

**Generalization to More Objects.** To demonstrate the generalization to more object distractors and denser clutter, we double the number of objects for testing as $A^2$-2#O in Table 2. Notably, there is no

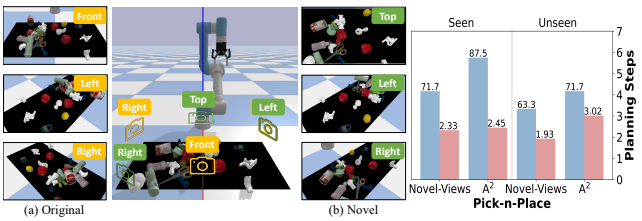



Figure 5: Ablation studies of (a) original camera viewpoints and (b) novel camera viewpoints.

| | Pick | | Place | |
|---|---|---|---|---|
| | Seen | Unseen | Seen | Unseen |
| $A^2$-2#O | 81.3/3.89 | 78.7/3.18 | 81.3 | 56.7 |
| $A^2$-2#D | 97.3/2.36 | 97.3/2.31 | 92.0 | 79.3 |

* Metrics of pick are presented as Task Success Rate / Planning Steps.

Table 2: Scaling to Double Number of Objects and Data

Get a cute toy puppet, put it on the blue doraemon plate.

Grasp the avocado, place it on top of the white bread.

I need the banana, move it next to the bread and the egg tart.

Give me the yellow block, place it into the hello kitty bowl.

Pick up the red pepper, leave it surrounding the two bowls.

Figure 6: Test cases in real world. Each case contains 21∼22 objects that are mostly unseen during training. Target or reference objects are labeled with stars.

retraining of policy. Results show that our policy outperforms most baselines (tested with original object number) even with double objects, validating effectiveness in more complex settings.

**Scaling to More Data.** We double the training data to test the scalability, and report results as $A^2$-2#D in Table. 2, which verifies effective improvements in both tasks when scaling to more data.

### 5.5 Real-world Experiments

**Experiment Setup.** Our real-world setup involves a UR5 robot arm equipped with a ROBOTIQ-85 gripper, and an Intel RealSense L515 capturing RGB-D images at a resolution of 1280×720. Notably, the camera viewpoint in the real-world setup is unseen during training. The workspace is divided into pick and place workspaces, where the robot is supposed to grasp the target object within the pick workspace, and place it within place workspace. Our test cases include 5 scenarios shown in Figure 6. Each of them contains 21∼22 objects that are mostly unseen during training. There are in total of 38 objects for real-world testing, including 10 seen objects and 28 unseen objects.

**Comparison to Baselines.** We compare our policy with $A^2$-G, as it performs better in simulation. We test 10 runs for each case, in a total of 50 times testing. All the policies are zero-shot transferred from simulation to the real world. Test results are reported in Table 3. In general, our policy achieves much better performance in task success rate. Though $A^2$-G demonstrates fewer planning steps, it gets a low task success rate at 56%. This is due to the fact that $A^2$-G cannot afford errors in visual grounding. Instead, our policy assesses the probabilities of feasible actions conditioned on vision-language cues, reducing reliance on visual grounding accuracy. By further injecting multi-modality characteristics, $A^2$-PA improves both task success rate and planning efficiency.

**Generalization.** Results in Table 3 further verify our generalization to camera number, camera viewpoints, and novel objects. This is not merely because we utilize foundation models, but also because our alignment design effectively integrates the priors of multiple foundation models through a lightweight network, all while preserving the knowledge embedded in the pre-trained models.

## 6 Conclusion

| Method | Task Success | Planning Steps |
|---|---|---|
| $A^2$-G | 56.0 | **3.04** |
| $A^2$ | <u>76.0</u> | 3.95 |
| $A^2$-PA | **80.0** | <u>3.68</u> |

Table 3: Real-world Results

We propose $A^2$, an action prior alignment method for language-conditioned pick and place in clutter. Using foundation models, we construct a 3D zero-shot visual representation, and generate candidate actions that provide feasible action patterns. Conditioned on these foundation priors, we conduct alignment by learning one attention layer to score the candidate actions for downstream tasks. Our policy requires less training data, supports fast adaptation, and achieves better task performances. Additionally, our method shows zero-shot generalization to unseen objects and language instructions.

## References


[1] K. Xu, H. Yu, Q. Lai, Y. Wang, and R. Xiong. Efficient learning of goal-oriented push-grasping synergy in clutter. *IEEE Robotics and Automation Letters*, 6(4):6337–6344, 2021.

[2] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

[3] A. H. Qureshi, A. Mousavian, C. Paxton, M. C. Yip, and D. Fox. Nerp: Neural rearrangement planning for unknown objects. In *Robotics: Science and Systems (RSS)*, 2020.

[4] A. Goyal, A. Mousavian, C. Paxton, Y.-W. Chao, B. Okorn, J. Deng, and D. Fox. Ifor: Iterative flow minimization for robotic object rearrangement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14787–14797, 2022.

[5] B. Tang and G. S. Sukhatme. Selective object rearrangement in clutter. In *Conference on Robot Learning*, pages 1001–1010. PMLR, 2023.

[6] K. Xu, Z. Zhou, J. Wu, H. Lu, R. Xiong, and Y. Wang. Grasp, see and place: Efficient unknown object rearrangement with policy structure prior. *IEEE Transactions on Robotics*, 2024.

[7] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[8] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *6th Annual Conference on Robot Learning*, 2022.

[9] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan. Vima: robot manipulation with multimodal prompts. In *Proceedings of the 40th International Conference on Machine Learning*, pages 14975–15022, 2023.

[10] Y. Ze, G. Yan, Y.-H. Wu, A. Macaluso, Y. Ge, J. Ye, N. Hansen, L. E. Li, and X. Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. In *Conference on Robot Learning*, pages 284–301. PMLR, 2023.

[11] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. In *7th Annual Conference on Robot Learning*, 2023.

[12] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox. Rvt-2: Learning precise manipulation from few demonstrations. In *Robotics: Science and Systems (RSS)*, 2024.

[13] W. Goodwin, S. Vaze, I. Havoutis, and I. Posner. Semantically grounded object matching for robust robotic scene rearrangement. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 11138–11144. IEEE, 2022.

[14] M. Ji, R.-Z. Qiu, X. Zou, and X. Wang. Graspsplats: Efficient manipulation with 3d feature splatting. In *8th Annual Conference on Robot Learning*, 2024.

[15] O. Shorinwa, J. Tucker, A. Smith, A. Swann, T. Chen, R. Firoozi, M. D. Kennedy, and M. Schwager. Splat-mover: multi-stage, open-vocabulary robotic manipulation via editable gaussian splatting. In *8th Annual Conference on Robot Learning*, 2024.

[16] Y. Zheng, X. Chen, Y. Zheng, S. Gu, R. Yang, B. Jin, P. Li, C. Zhong, Z. Wang, L. Liu, et al. Gaussiangrasper: 3d language gaussian splatting for open-vocabulary robotic grasping. *IEEE Robotics and Automation Letters*, 2024.

[17] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, A. Maalouf, S. Li, G. S. Iyer, S. Saryazdi, N. V. Keetha, et al. Conceptfusion: Open-set multimodal 3d mapping. In *Robotics: Science and Systems (RSS)*, 2023.

[18] Y. Wang, M. Zhang, Z. Li, T. Kelestemur, K. R. Driggs-Campbell, J. Wu, L. Fei-Fei, and Y. Li. $D^3$fields: Dynamic 3d descriptor fields for zero-shot generalizable rearrangement. In *8th Annual Conference on Robot Learning*, 2024.

[19] A. Rashid, S. Sharma, C. M. Kim, J. Kerr, L. Y. Chen, A. Kanazawa, and K. Goldberg. Language embedded radiance fields for zero-shot task-oriented grasping. In *7th Annual Conference on Robot Learning*, 2023.

[20] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, 2022.

[21] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. In *Conference on Robot Learning*, pages 540–562. PMLR, 2023.

[22] Y. Qian, X. Zhu, O. Biza, S. Jiang, L. Zhao, H. Huang, Y. Qi, and R. Platt. Thinkgrasp: A vision-language system for strategic part grasping in clutter. In *8th Annual Conference on Robot Learning*, 2024.

[23] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[24] H.-S. Fang, C. Wang, M. Gou, and C. 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.

[25] C. Zhou, C. C. Loy, and B. Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, pages 696–712, 2022.

[26] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[28] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, pages 405–421, 2020.

[29] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[30] K. Xu, S. Zhao, Z. Zhou, Z. Li, H. Pi, Y. Zhu, Y. Wang, and R. Xiong. A joint modeling of vision-language-action for target-oriented grasping in clutter. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11597–11604. IEEE, 2023.

[31] Z. Xu, K. Xu, R. Xiong, and Y. Wang. Object-centric inference for language conditioned placement: A foundation model based approach. In *2023 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 203–208, 2023.

[32] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *8th Annual Conference on Robot Learning*, 2024.

[33] J. E. King, M. Cognetti, and S. S. Srinivasa. Rearrangement planning using object-centric and robot-centric action spaces. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3940–3947. IEEE, 2016.

[34] K. Xu, H. Yu, R. Huang, D. Guo, Y. Wang, and R. Xiong. Efficient object manipulation to an arbitrary goal pose: Learning-based anytime prioritized planning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7277–7283. IEEE, 2022.

[35] H. Tian, C. Song, C. Wang, X. Zhang, and J. Pan. Sampling-based planning for retrieving near-cylindrical objects in cluttered scenes using hierarchical graphs. *IEEE Transactions on Robotics*, 39(1):165–182, 2022.

[36] K. Xu, R. Chen, S. Zhao, Z. Li, H. Yu, C. Chen, Y. Wang, and R. Xiong. Failure-aware policy learning for self-assessable robotics tasks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9544–9550. IEEE, 2023.

[37] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *The International Journal of Robotics Research*, 41(7): 690–705, 2022.

[38] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. 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.

[39] K. Fang, Y. Bai, S. Hinterstoisser, S. Savarese, and M. Kalakrishnan. Multi-task domain adaptation for deep learning of instance grasping from simulation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3516–3523. IEEE, 2018.

[40] M. Danielczuk, J. Mahler, C. Correa, and K. Goldberg. Linear push policies to increase grasp access for robot bin picking. In *2018 IEEE 14th international conference on automation science and engineering (CASE)*, pages 1249–1256. IEEE, 2018.

[41] A. Kurenkov, J. Taglic, R. Kulkarni, M. Dominguez-Kuhne, A. Garg, R. Martín-Martín, and S. Savarese. Visuomotor mechanical search: Learning to retrieve target objects in clutter. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8408–8414. IEEE, 2020.

[42] Y. Yang, H. Liang, and C. Choi. A deep learning approach to grasping the invisible. *IEEE Robotics and Automation Letters*, 5(2):2232–2239, 2020.

[43] G. Zhai, D. Huang, S.-C. Wu, H. Jung, Y. Di, F. Manhardt, F. Tombari, N. Navab, and B. Busam. Monograspnet: 6-dof grasping with a single rgb image. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1708–1714. IEEE, 2023.

[44] G. Zhai, X. Cai, D. Huang, Y. Di, F. Manhardt, F. Tombari, N. Navab, and B. Busam. Sg-bot: Object rearrangement via coarse-to-fine robotic imagination on scene graphs. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4303–4310. IEEE, 2024.

[45] A. D. Vuong, M. N. Vu, H. Le, B. Huang, H. T. T. Binh, T. Vo, A. Kugi, and A. Nguyen. Grasp-anything: Large-scale grasp dataset from foundation models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14030–14037. IEEE, 2024.

[46] M. Jia, H. Huang, Z. Zhang, C. Wang, L. Zhao, D. Wang, J. X. Liu, R. Walters, R. Platt, and S. Tellex. Open-vocabulary pick and place via patch-level semantic maps. *arXiv preprint arXiv:2406.15677*, 2024.

[47] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[48] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, pages 1–31, 2024.

[49] Gpt-4v(ision) system card. 2023. URL https://api.semanticscholar.org/CorpusID:263218031.

[50] S. Huang, Z. Jiang, H. Dong, Y. Qiao, P. Gao, and H. Li. Instruct2act: Mapping multi-modality instructions to robotic actions with large language model. *arXiv preprint arXiv:2305.11176*, 2023.

[51] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. In *Conference on Robot Learning*, pages 1199–1210. PMLR, 2023.

[52] Y. Yang, H. Yu, X. Lou, Y. Liu, and C. Choi. Attribute-based robotic grasping with data-efficient adaptation. *IEEE Transactions on Robotics*, 40:1566–1579, 2024.

[53] W. Yuan, A. Murali, A. Mousavian, and D. Fox. M2t2: Multi-task masked transformer for object-centric pick and place. In *7th Annual Conference on Robot Learning*.

[54] D. Shim, S. Lee, and H. J. Kim. Snerl: Semantic-aware neural radiance fields for reinforcement learning. In *International Conference on Machine Learning*, pages 31489–31503. PMLR, 2023.

[55] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu. Synergies between affordance and geometry: 6-dof grasp detection via implicit representations. 2021.

[56] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022.

[57] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. 2023.

[58] Y. Deng, J. Wang, J. Zhao, J. Dou, Y. Yang, and Y. Yue. Openobj: Open-vocabulary object-level neural radiance fields with fine-grained understanding. *IEEE Robotics and Automation Letters*, 2024.

[59] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola. Distilled feature fields enable few-shot language-guided manipulation. In *Conference on Robot Learning*, pages 405–424. PMLR, 2023.

[60] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. 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.

[61] S. H. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor. Chatgpt for robotics: Design principles and model abilities. *IEEE Access*, 2024.

[62] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, and K. Ikeuchi. Gpt-4v (ision) for robotics: Multimodal task planning from human demonstration. *IEEE Robotics and Automation Letters*, 2024.

[63] Y. Hu, F. Lin, T. Zhang, L. Yi, and Y. Gao. Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*.

[64] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2023.

[65] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[66] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[67] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, et al. Openvla: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*, 2024.

[68] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[69] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. $\pi_0$: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[70] S. Yenamandra, A. Ramachandran, K. Yadav, A. S. Wang, M. Khanna, T. Gervet, T.-Y. Yang, V. Jain, A. Clegg, J. M. Turner, et al. Homerobot: Open-vocabulary mobile manipulation. In *Conference on Robot Learning*, pages 1975–2011. PMLR, 2023.

[71] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021.

[72] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[73] Openai. gpt-4o: Openai's multimodal vision-language system. 2023. Accessed: 2024-06-05. URL https://openai.com/research/gpt-4o.

[74] Z. Zhou, Y. Yang, Y. Wang, and R. Xiong. Open-set object detection using classification-free object proposal and instance-level contrastive learning. *IEEE Robotics and Automation Letters*, 8(3):1691–1698, 2023.

# A Related Works

## A.1 Target-oriented Pick and Place in Clutter

Robotic pick and place in clutter has been a topic of interest in manipulation for decades. Traditional approaches [33, 34, 35, 36] are in the context of task and motion planning (TAMP) under the assumption of known object models and states. These methods struggle in open real scenarios, where obtaining precise object models and states is challenging. More recent research studies target-oriented unknown object grasping in clutter by first clearing obstacles [37, 38, 39], or retrieving the target object through non-prehensile actions [40, 41, 42, 1]. [2, 4, 6][43, 44] step forward to build unknown object pick and place systems in cluttered environments, which are promising for real applications. However, these works still require images to specify target objects. Instead, language instructions are more flexible in open-world applications. By cooperating with foundation models, policies are capable of dealing with open-vocabulary objects in scattered scenes [7, 45, 9, 46]. In this paper, we aim to develop a policy for open-vocabulary pick and place in clutter, with the target specified with language instructions.

## A.2 Foundation Models for Language-conditioned Manipulation

Foundation models in the field of CV and NLP have demonstrated powerful performance [26, 47, 48, 49], and have been explored to facilitate robotic manipulation in open-world applications. A common way to utilize foundation models is to directly ground their capabilities into robotic scenarios. A series of approaches [13, 20, 50] uses vision foundation models for object grounding from flexible language instructions. Among them, some works explore object-centric representations for better scene understanding [13, 30, 51, 52, 53]. Other methods build 3D scene representations capturing both semantic and geometric information [54, 55, 56, 57, 32, 58]. For example, several approaches distill 3D neural feature fields from 2D foundation models [59, 19], requiring dense camera views and time-consuming training for high-quality rendering. This hinders real-time interaction in real-world scenarios. And efforts to overcome these limitations include introducing 3D Gaussian Splatting [16, 14, 15] and using sparse-view 3D representations [18, 17]. There are also methods [60, 21, 61, 22, 62, 63] utilizing the reasoning capability of large language models to build systems for planning. However, the performance of these policies largely depends on the capability of foundation models, and suffers from cascaded errors across individual modules. Another line of works [7, 8, 10, 9, 11, 12, 32] integrates features from vision foundation models into end-to-end policies. Despite promising results, these works consume extensive demonstration data and take plenty of training steps for convergence. In addition, one has to face the generalization issue if the tested objects or scenes are significantly different from those in the training data.

Recently, researchers have tried to learn action foundation models from large-scale robot data. For instance, AnyGrasp [64] is a grasp foundation model capable of generating grasp actions for open scenes. More generally, efforts are made to develop large Vision-Language-Action models (VLA) for general tasks and even embodiments [65, 66, 67, 68, 69]. However, leveraging priors from these action foundation models is much less explored. Some methods simply deploy pre-trained grasp models to generate grasp actions after object grounding, essentially paying less attention to the action planning side [19, 70]. In this paper, our policy aims to integrate priors from vision, language, and action foundation models to improve task performance.

# B 3D Representation Details

Given image(s) $\mathcal{I} = \{I_i\}_{i=0,1,\dots,M}$ of one or more RGB-D camera(s), we extract 2D patch-level features $\mathcal{W}_i$ by MaskCLIP [25], including visual patch-level features $\mathcal{W}_i^f$ and vision-language similarity information $\mathcal{W}_i^s$ denoting cosine similarities between language embeddings and $\mathcal{W}_i^f$.

We generate a 3D point cloud $\mathbf{p}$ within the workspace using the camera parameters. For each point $p_j$ of $\mathbf{p}$, we project it back to the $i$th camera viewpoint as the pixel $u_j^i$, and get its visual feature $f_j^i$

by interpolation:

$$f_j^i = \mathcal{W}_i^f[u_j^i] \tag{1}$$

Following [18], we compute weights for each camera according to the visibility and distance of $p_j$ relative to the $i$th camera. We denote the distance from $p_j$ to the $i$th camera viewpoint as $l_i$, and compute the depth by interpolating the corresponding depth image $I_i^d$ as $l_i' = I_i^d[u_j^i]$. Then the truncated depth difference is defined as:

$$d_i = l_i - l_i', \quad d_i' = \max(\min(d_i, \mu), -\mu), \tag{2}$$

where $\mu = 0.02$ represents the truncation threshold for the Truncated Signed Distance Function (TSDF). The visibility of $p_j$ in the $i$th camera viewpoint can be represented as $v_i = \mathbb{1}_{d_i < \mu}$. Here $\mathbb{1}$ is the indicator function. We compute the weight for the $i$th camera viewpoint as:

$$\beta_i = \exp\left(\frac{\min\left(\mu - |d_i|, 0\right)}{\mu}\right). \tag{3}$$

where $\beta_i$ decays as $|d_i|$ increases. Then, we can obtain the semantic feature $f_j$ by fusing features from $M$ camera viewpoints:

$$f_j = \frac{\sum_{i=1}^{M} \beta_i v_i f_j^i}{\epsilon + \sum_{i=1}^{M} v_i} \tag{4}$$

where $\epsilon = 1 \times 10^{-6}$ is to avoid numeric issues.

Similarly, we can get the similarity value $s_j$ for $p_j$ in the same way upon $\mathcal{W}_i^s$. Finally, we get a 3D feature cloud $\mathbf{f} = \{f_j\}$ indicating the visual features and a 3D similarity cloud $\mathbf{s} = \{s_j\}$ indicating the task-relevant information.

We employ the checkpoint of MaskCLIP ViT-L/14 to generate visual features and crop the raw image into 12 sub-images for more fine-grained features. We exclude the table points from the 3D representations for pick tasks while retaining them for place tasks. This is because the policy does not require the feature information of the table for pick action planning, and the filtering helps the policy focus on the objects. Specifically, table points are removed by height filtering of the point cloud in world coordinates.

# C Training Details

## C.1 Simulation Environment

We collect demonstration data by model-based expert planners with a UR5 arm in PyBullet [71]. There are three statically mounted cameras ($M = 3$) overlooking the tabletop as shown in Fig. 2: one positioned 45° downward from the front, one 50° downward from the anti-diagonal perspective and one 50° downward from the diagonal perspective, referred to as the front, left and right cameras respectively. For each camera, we adopt the same camera intrinsics as those of Intel RealSense L515. Our object models are from GraspNet-1Billion [24].

## C.2 Data Collection

**Data Collection Settings.** For both pick and place, we collect data from 5k episodes, among which the success steps are recorded as demonstrations. This results in around 6.5k successful samples in total, with approximately 3.4k for pick and 3.1k for place. During data collection of pick, 15 objects are randomly dropped into the workspace to form a cluttered scene, and the model-based pick expert planner chooses the nearest grasp of the target objects. For placement, to ensure adequate space, there are 8 objects in the workspace whose center positions are at least 0.1m from one another. The model-based place expert planner identifies the valid place region based on the reference object and the relation, and then randomly chooses a place within this region.

**Flexible Language Instructions.** During each rollout of data collection, we randomly sample a language template along with keywords (target for pick tasks, reference and relation for place tasks)

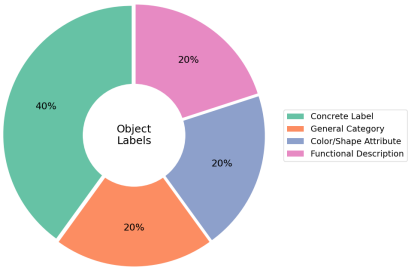
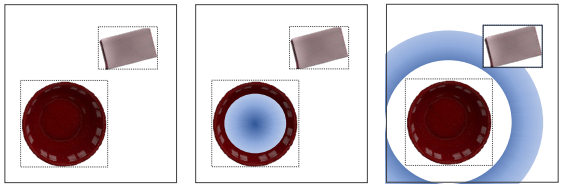



Figure 7: Diversity of object labels in language instructions.



(a) on (b) around

Figure 8: Example generated place regions for (a) "on" and (b) "around" the red bowl.

to form a complete language instruction. For pick, there are five language templates: "Give me the {target}", "I need a {target}", "Grasp a {target} object", "I want a {target} object", "Get something to {target}", where {target} can be a concrete label (*e.g.* banana), a general category (*e.g.* fruit), or the attribute of color (e.g. red), shape (e.g. round), or even a functional description (*e.g.* hold other things). For place tasks, the language instructions are similar, but with additional spatial relation words: "Put it {relation} the {reference}", "Place this {direction} the {reference}", "Move the object {direction} the {reference}". Here {relation} specifies the spatial relationship respective to the {reference}. {reference} is analogous to {target}, while {relation} can be words indicating "on" or "around" relations relative to reference}. For instance, words like "on top of", and "into" belong to "on" relation, and others such as "next to", and "near" belong to "around" relation. There is a total of 66 object models for data collection, with 36 language keywords categorized into four types: concrete labels, general categories, attributes of color or shape, and functional descriptions. The four types of object label follow a 4:2:2:2 distribution, as shown in Fig. 7. For spatial relations, there are 6 choices for "on" or "around" relations.

**Model-based Experts.** We collect data with model-based expert planners. The model-based pick expert planner selects the grasp nearest to the target objects from candidates generated by GraspNet [24]. The model-based place expert planner determines valid place regions based on the reference object and the relation. Specifically, we first obtain object region proposals from the mask image in Pybullet [71], where each pixel donates the index of the object visible in the camera. Object regions are identified as bounding boxes of pixels with the same index, and regions whose size is smaller than $5 \times 5$ are discarded. Then the valid place region is generated within the reference object for the "on" relation, or around the reference object for the "around" relation. Note that the generated "around" region should not overlap with any object regions. Fig. 8 shows example place regions for the "on" and "around" relations.

**Visual Representation Filtering.** We exclude the table points from the visual representations (*i.e.* 3D feature cloud and 3D similarity cloud) for pick tasks while retaining them for place tasks. Specifically, table points are removed by height filtering of the point cloud in world coordinates. This is because the policy does not require the feature information of the table for pick action planning, and the filtering helps the policy focus on the objects.

### C.3 Training

**Network Architecture.** We adopt the transformer architecture of the text encoder in [26], with width of 768, head of 8, and layer of 1. The action decoder is a 3-layer MLP. The network parameters of MaskCLIP and action models are fixed during training.

**Hyperparameters.** We set the sample number $N = 500$. For the action candidate number $L$, we sample 6 place poses for each object (3 for "on" relation and 3 for "around" relation), while for pick, $L$ depends on the output of GraspNet. We use $\alpha = 0.2$ during policy adaptation.

**Imitation Learning Setting.** Regarding Eqn. 3, our goal is to maximize the likelihood of the successful action $a_d$ among the candidate actions $\mathcal{A}_L(\mathcal{I}_d)$ for each demonstration $\mathcal{D} = \{\mathcal{I}_d, \mathcal{L}_d, a_d\}$. We formulate this as a maximum likelihood estimation (MLE) problem, which is optimized via the

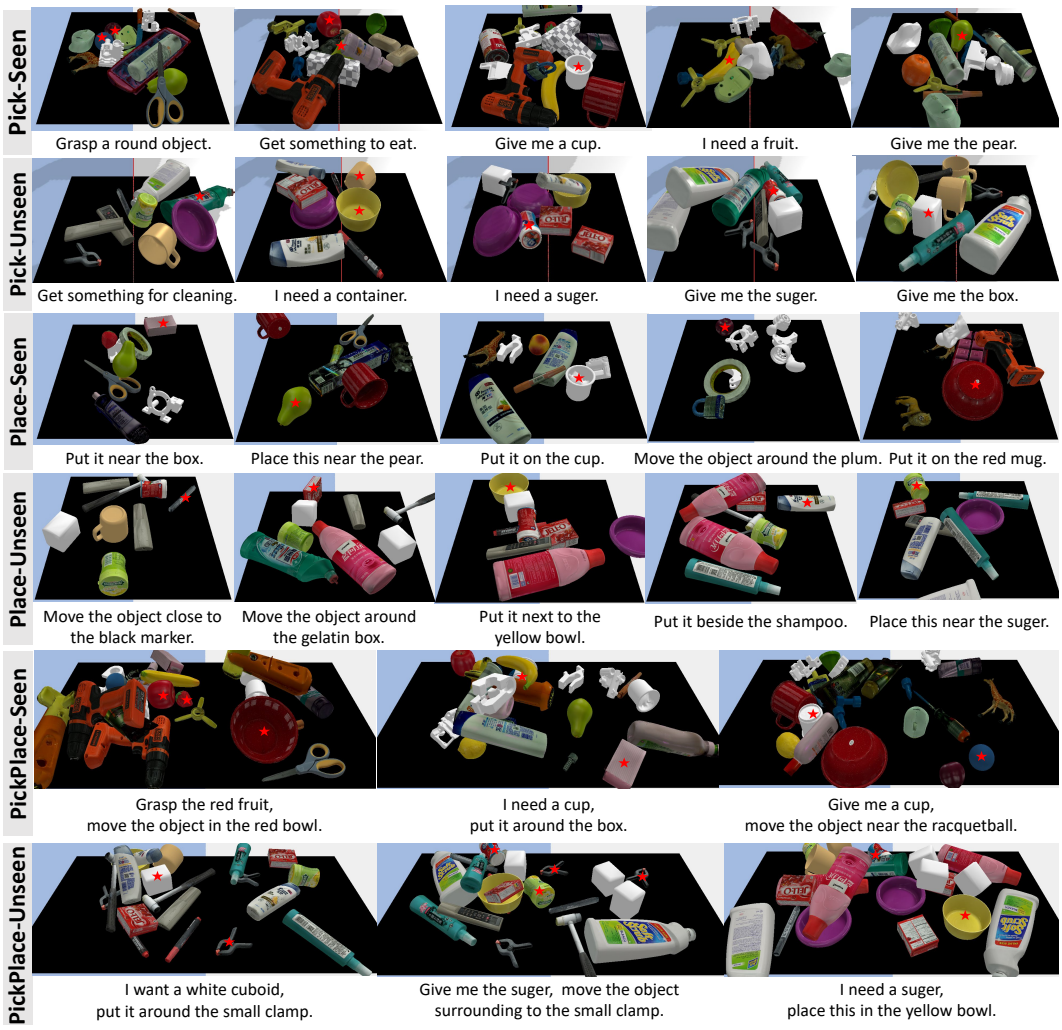



Figure 9: More example test cases in simulation. Target or reference objects are labeled with stars.

cross-entropy loss. To be specific, the policy is trained through cross-entropy loss for 200 epochs. During fine-tuning, the policy is trained with only 100 multi-labeled place data using binary cross-entropy loss for 200 epochs, consuming around 2 minutes.

$$\mathcal{L}_{\text{CE}} = -\log \omega(a_d|\mathcal{I}_d, \mathcal{L}_d) \tag{5}$$

# D Evaluation Details

## D.1 Test Cases

We collect test cases with 66 seen objects and 17 unseen objects. For pick, each case contains 15 objects to form adversarial clutter where the robot might need to grasp away other obstacles for target grasping. All pick policies are evaluated on $x$=10 arrangements of seen objects and $x = 5$ of unseen objects. For place, each case contains 8 objects to preserve some free space surrounding the reference object for placement. For place policies, performances are tested on $x$=20 arrangements of seen objects and $x$=10 of unseen objects. Given that placement is a one-step task, we add variances to action candidates in each test run of each case to evaluate robustness. For pick-n-place, we test policies with $x = 8$ seen objects cases and $x = 4$ unseen objects cases. Note that in these cases, we divide the workspace into a pick workspace (left) and a place workspace (right). To determine the success of target grasping, we use environment feedback in simulation. In real world, target


Figure 10: Example test cases of double object number in simulation. The target objects or reference objects are labeled with stars.

is regarded as grasped if CLIP similarity of language and grasped object crop (filtered by depth) exceeds a threshold.

More example test cases across all categories are presented in Fig. 9. For cases of more objects (30 objects in a scene) in Sec. 2, Fig. 10 shows some example cases, demonstrating more complex settings with frequent occlusion and dense clutter than those in Fig. 9.

### D.2 Baseline Implementations

**Neural Field based Pick Policies.** These include LERF-TOGO [19] and GraspSplats [14]. LERF-TOGO [19] is a NeRF-based method that distills feature fields from CLIP [26], while GraspSplats [14] reconstructs 3D feature fields from CLIP by 3D Gaussian Splatting [72]. In the experiments, we train the feature fields on each step of action planning at test time. With the feature fields, they first locate the target object from language instructions, and select the corresponding grasp from GraspNet [24] generated grasps. We follow the number of camera viewpoints in their papers to guarantee a fair comparison. Specifically, we add a circle of camera viewpoints around the workspace to provide sufficient information. LERF-TOGO trains its feature field with 53 posed RGB images, while GraspSplats uses 23, and both of the inputs include the 3 RGB-D images used by our method.

**Object-centric Pick and Place Policies.** There are two object-centric pick policies. VLG [30] leverages object-centric representation to jointly model vision, language, and action information. ThinkGrasp [22] is an approach that develops a vision-language system with GPT4o [73] to plan the object grasp sequence, followed by object segmentation and grasp planning. For placement, we implement a method similar to [31], namely VLP, which grounds reference objects and spatial relations respectively. For a fair comparison, CLIP is not fine-tuned in VLP as in [31].

**3D Visual Grounding Pick and Place Policies.** We implement variant methods that directly conduct visual grounding using our 3D visual representations, named $A^2$-G-Pick and $A^2$-G-Place for pick and place tasks respectively. These methods select the action nearest to the region with the highest average similarity of K-nearest neighbors (K=0.05$M$). In addition, $A^2$-G-Pick can combine with $A^2$-G-Place as a 3D visual grounding pick-n-place policy, denoted as $A^2$-G.

**3D End-to-end Pick and Place Policies.** We compare to 3D end-to-end policies Act3D [11], RVT-2 [12] and 3D Diffuser [32], which leverage multi-view CLIP features to predict 3D actions. We use pre-trained models of Act3D and RVT-2, as well as the models trained on our data (referred to as Act3D$^\dagger$, RVT-2$^\dagger$, and 3D Diffuser$^\dagger$) for evaluation. Note that the setting of the pre-trained model of 3D Diffuser is distinct from our setting, thus cannot be directly employed.

| RL | IL | Res | Data | Seen | Unseen |
|---|---|---|---|---|---|
| ✓ | | | 1500 | 36.7 | 28.0 |
| ✓ | | ✓ | 1500 | 89.7 | 68.0 |
| | ✓ | | 100 | 56.3 | 19.7 |
| | ✓ | ✓ | 100 | 89.0 | 76.0 |

Table 4: Ablation Studies of Different Policy Adaptations

| TE | LE | RoPE | RGB | Pick | | Place | |
|---|---|---|---|---|---|---|---|
| | | | | Seen | Unseen | Seen | Unseen |
| | | | ✓ | 90.7/2.41 | 80.0/3.20 | 69.0 | 52.0 |
| | | ✓ | | 60.0/4.58 | 73.3/3.55 | 26.7 | 42.7 |
| | | ✓ | ✓ | 95.3/2.55 | 97.3/2.57 | 89.3 | 74.0 |
| ✓ | | ✓ | ✓ | 92.7/2.84 | 78.7/3.06 | 74.0 | 39.3 |
| | ✓ | ✓ | ✓ | 94.0/2.48 | 88.0/2.39 | 71.3 | 43.3 |

* Metrics of pick are presented as Task Success Rate / Planning Steps.

Table 5: Ablation Studies of Network Architecture

### D.3 Real-world Experiment

**Setups.** In our real-world experiments, we initially adopted a single Intel RealSense L515 camera and observed that our policy achieves a good task performance. This result demonstrates that our method generalizes well to limited-view settings, which are common in practical robotic deployments. We also experimented with multi-camera setups but encountered depth interference caused by overlapping structured-light laser patterns. This interference resulted in noisy or unstable depth maps, which affected downstream modules such as GraspNet, whose grasp predictions rely on accurate depth information. As a result, the decision to use a single camera represents a deliberate trade-off between broader observation coverage and depth sensing reliability.

At inference, the policy first receives the pick language instruction, and plans actions upon the point cloud within the grasp workspace. Once the target object is grasped, the place language instruction is fed into the policy for action planning, with the point cloud within the place workspace. For the place action model, we employ a pre-trained model to generate object region proposals [74], which is trained on data from GraspNet-1Billion [24] with $mAP = 70.70$ for seen objects and $mAP = 34.53$ for unseen objects.

**Example Sequences.** Figure 15 illustrates some execution sequences in real-world experiments. In a cluttered environment, by scoring the candidate actions through alignment, our policy displays the ability to gradually remove obstacle objects, grasp the target object, and finally place it at the specified location.

## E More Ablation Studies

**Different Policy Adaptations.** We compare different ways of policy adaptations, including different learning paradigms (RL, SL), data amounts and architectures (with or without the residual block). Performance comparisons on place tasks are reported in Table 4. It is shown that fine-tuning with a residual block significantly outperforms full fine-tuning. This is because the residual block effectively injects multi-modality without losing much information of the pre-trained policy. With the residual block, both RL and SL can achieve comparable performances, whereas RL consumes much more data and demonstrates poorer generalization. This might be because place is a one-step task, where the lack of sequential decision making limits the effectiveness of RL optimization.

**Network Architecture.** We compare our method with some variant methods to evaluate the architecture design. Testing results are shown in Table 5. Removing RoPE causes notable drops: $17.3\%$

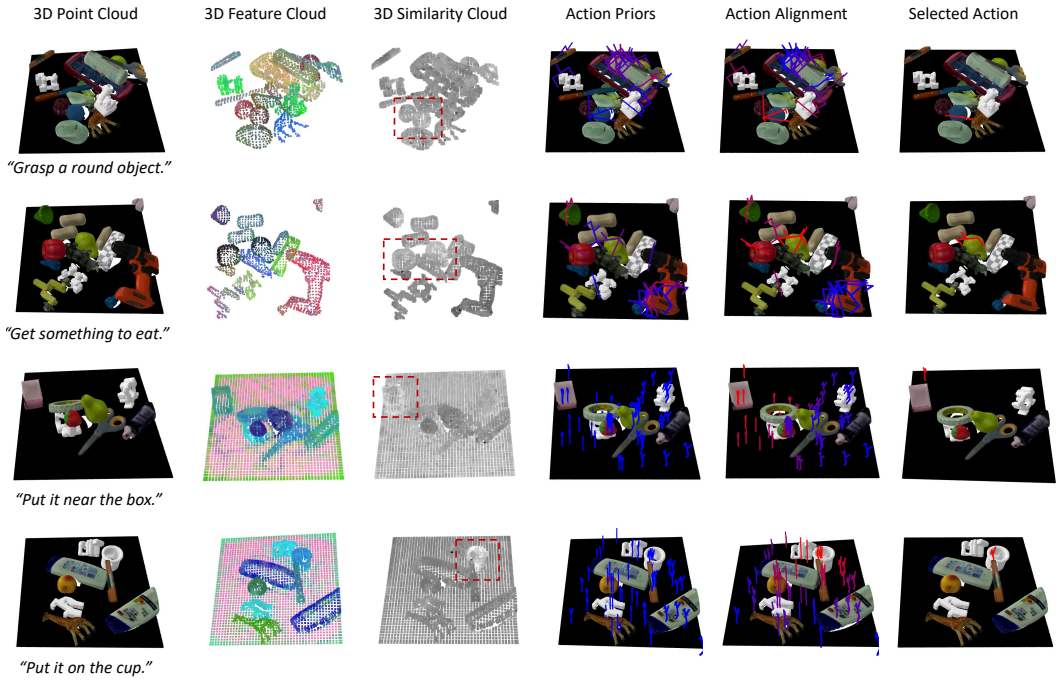



Figure 11: Case studies. For each case, we show the 3D representations (*i.e.* 3D point cloud, 3D feature cloud, and 3D similarity cloud), the action priors from action foundation models, the alignment results, and the final selected action. Notably, in the similarity cloud, regions with high similarity are highlighted with red rectangles. For each action, the labeled color indicates the action probability, with the color shifting toward red as the probability increases.

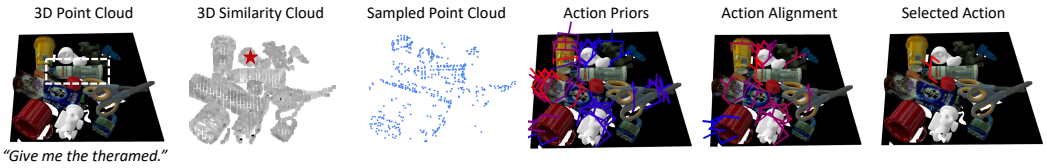



Figure 12: Case visualization where the visual grounding fails, yet our policy selects the correct grasp via alignment. The white rectangle in the 3D point cloud marks the target, while the red star in the similarity cloud marks the direct visual grounding result. For each action, the labeled color indicates the action probability, with the color shifting toward red as the probability increases.

for unseen pick tasks and over $20\%$ for place tasks, highlighting its importance for generalization. In addition, using sampled point features as values in cross-attention instead of image features degrades performance, showing the benefit of foundation model features for effectiveness and generalization. Also, adding task-specific embeddings (TE) to action features harms performance, likely by hindering shared representations between pick and place. Finally, directly feeding the language embedding (LE) into cross-attention instead of weighting visual features with similarities weakens CLIP priors and reduces success rates.

**Case Studies.** Figure 11 shows several cases to illustrate the 3D representations, action priors, and alignment results of our policy. Given language instructions, the similarity cloud can highlight the task-relevant regions, and our policy aligns action priors based on these representations. Figure 12 further shows a case where visual grounding alone fails, but our alignment still enables correct grasp selection through alignment. This indicates that while we take similarity-sampled points, we evaluate actions guided by visual grounding rather than being determined by it. Figure 13 visualizes some typical failure modes, including heavy occlusion and visual ambiguity of target objects, as well as semantic ambiguity in language instruction, *i.e* ambiguous word "cylinder".

**Failure Modes.** Fig. 13 visualizes some typical failure modes, including heavy occlusion and visual ambiguity of target objects, as well as the semantic ambiguity in language instruction. In the left


Figure 13: Example failure modes, including heavy occlusion, visual ambiguity, and semantic ambiguity.

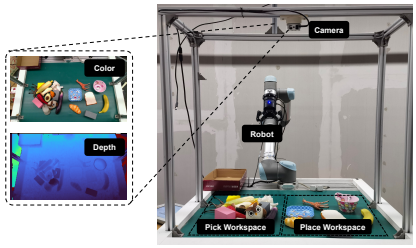



Figure 14: Real-world platform.

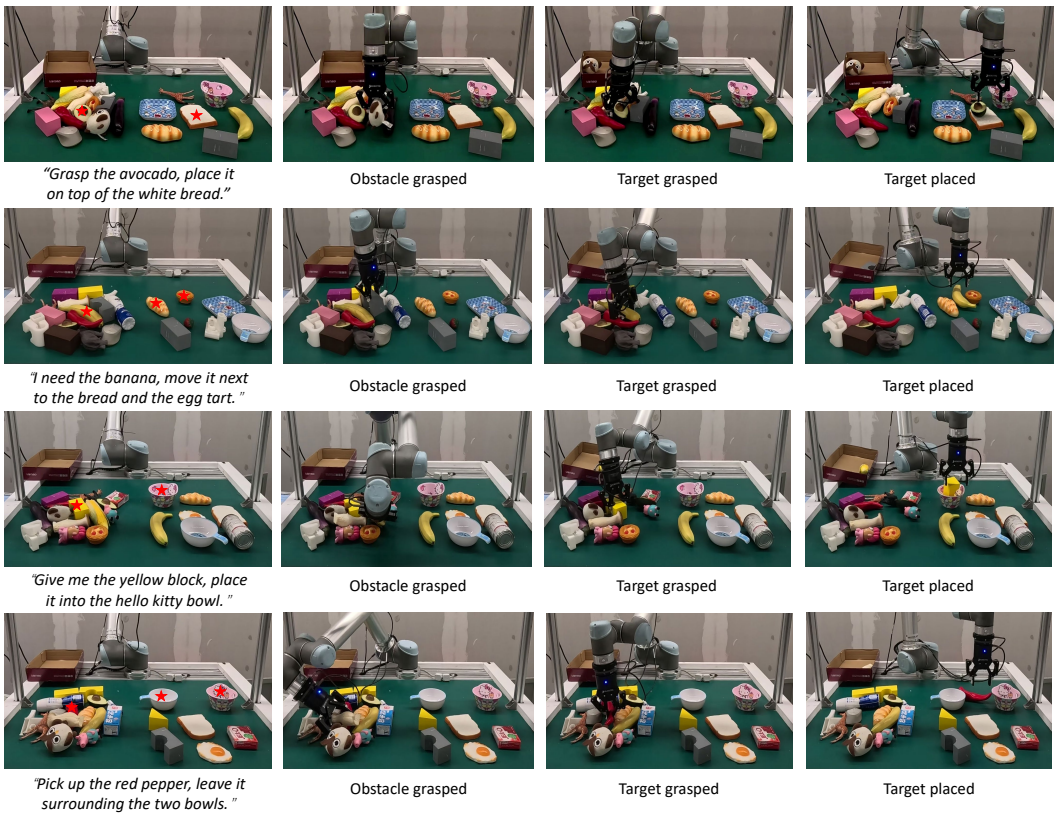



Figure 15: Example testing sequences. The camera viewpoint and most of the objects are unseen during training. Taking the language instructions for pick and place, our policy is able to gradually remove obstacles, grasp the target object, and finally place it at the target location.

case, the target object "strawberry" is largely occluded by other distractors, demonstrating a heavily cluttered scene. In such cases, the policy struggles to pick up the target within limited planning steps. In the middle case, the target object "darlix toothpaste" shares a similar visual appearance with the distractor "darlix box", which misleads the policy during selection. The right case illustrates semantic ambiguity in the language instruction. Although the phrase "into a cylinder" suggests that the target object is container-like, the expression lacks specificity and may lead the policy to select other cylindrical objects that do not afford containment.