

A Joint Modeling of Vision-Language-Action for Target-oriented Grasping in Clutter

Kechun Xu, Shuqi Zhao, Zhongxiang Zhou, Zizhang Li, Huaijin Pi, Yifeng Zhu, Yue Wang, Rong Xiong

Abstract—We focus on the task of language-conditioned grasping in clutter, in which a robot is supposed to grasp the target object based on a language instruction. Previous works separately conduct visual grounding to localize the target object, and generate a grasp for that object. However, these works require object labels or visual attributes for grounding, which calls for handcrafted rules in planner and restricts the range of language instructions. In this paper, we propose to jointly model vision, language and action with object-centric representation. Our method is applicable under more flexible language instructions, and not limited by visual grounding error. Besides, by utilizing the powerful priors from the pre-trained multi-modal model and grasp model, sample efficiency is effectively improved and the sim2real problem is relieved without additional data for transfer. A series of experiments carried out in simulation and real world indicate that our method can achieve better task success rate by less times of motion under more flexible language instructions. Moreover, our method is capable of generalizing better to scenarios with unseen objects and language instructions. Our code is available at <https://github.com/xukechun/Vision-Language-Grasping>

I. INTRODUCTION

Target-oriented grasping in clutter is an essential task for robotic manipulation, and has been explored for decades [1]–[4]. Common practices to specify the target object are to provide a target object image [1], [5], or collect a set of demonstrations [6]. But providing such information is often infeasible for a user, especially in open-world applications, such as tabletop rearrangement over open-vocabulary objects.

An intuitive idea is to replace visual command with natural language to assign the target object. Recent progress in pre-trained large language models [7], [8] and multi-modal models [9] makes this idea practical and promising. In [10]–[13], the robot is able to learn policies for multi-task (*e.g.* pick-and-place and room rearrangement) by grounding language instructions as task goals with these pre-trained models. However, these works learn raw image based policy with lots of simulated demonstration data. In addition, one has to address sim2real gap to deploy raw image based policy in real world. The cluttered scene may further increase the data requirement and sim2real difficulty.

Instead, object-centric representation endows the robot with awareness of objects in the scene, which enables easier scene understanding, thus more efficient for grasping tasks. In a similar vein, another line of works [14]–[18] decouples the task of language-conditioned target-oriented

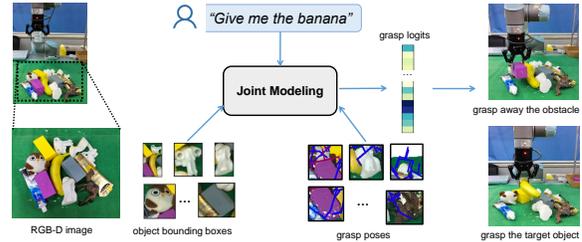


Fig. 1. An example scenario of language-conditioned grasping in clutter. Our method jointly models vision, language and action with object-centric representation (*i.e.* object bounding box), and conducts a sequence of grasps to grasp away obstacles, finally achieving the target object.

grasping into two separate steps: language visual grounding to figure out the target object and grasp planning to pick up that object. In [15]–[17], the robot generates object-centric representations by object bounding boxes, grounds semantics with object labels and visual attributes, and selects an action by a rule-based planner. However, these handcrafted attributes and rules restrict the generalization of language instructions. Furthermore, error in visual grounding results, and disturbance from other objects in the cluttered scene might degenerate the successful grasping, or call for more complex handcrafted rules in planning.

In this paper, we set to investigate an intermediate solution between the previous two lines of works. We propose to jointly model vision, language and action with object-centric representations as input, and grasp poses as output (Fig. 1). Specifically, we utilize the pre-trained visual-language model CLIP [9] to encode object-centric bounding boxes and the language instruction to visual-language features, and deploy the pre-trained graspnet [19] to generate a set of grasp poses, which are encoded to spatial features. We apply a transformer to build the cross-attention features with grasp features as queries, visual-position features as keys and visual-language features as values. Then the cross-attention features are fed into the policy network for action decision. As a whole, our policy is learned through model-free deep reinforcement learning in simulator, which predicts a sequence of grasps to grasp away objects hindering the successful grasping, and finally achieve the goal.

Compared to the first line of works, leveraging pre-trained models with object-centric representation provides powerful priors and object awareness for policy learning, thus improving the sample efficiency. Also, considering that the processes of generating object-centric representation and grasp poses from raw images cause sim2real gap, we deploy pre-trained models learned with massive simulation and real data for these two processes, relieving this problem. Compared to the second line, our policy directly takes the

This work was supported by the National Nature Science Foundation of China under Grant 62173293. Kechun Xu, Shuqi Zhao, Zhongxiang Zhou, Zizhang Li, Huaijin Pi, Yue Wang, Rong Xiong are with Zhejiang University, Hangzhou, China. Yifeng Zhu is with the University of Texas at Austin. Corresponding author, wangyue@iipc.zju.edu.cn.

multi-modal features for grasp selection, which eliminates the explicit design of object labels or visual attributes in planner or language instructions, and avoids relying on visual grounding. Consequently, our method is able to handle more flexible language instructions, and achieve better language generalization. Experiments conducted in simulation and real-world environments demonstrate that our system can achieve higher task success rate by less steps in language-conditioned target-oriented grasping in clutter. We also show that our method is capable of performing better in scenarios with unseen objects and language instructions, validating the generalization. To summarize, our main contributions are:

- We propose a joint modeling of vision, language and action for the task of target-oriented grasping in clutter through model-free deep reinforcement learning.
- We utilize the priors from the pre-trained visual-language model and grasp model to relieve sim2real gap without additional data for transfer, and improve the sample efficiency.
- The learned system 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.

II. RELATED WORKS

A. Target-oriented Grasping in Clutter

Robotic grasping in clutter has been a topic of interest in manipulation for decades. Recent works [20]–[24] have achieved significant improvements in grasping general objects in clutter, and some of them [19], [25]–[27] are capable of generating 6DoF grasp proposals of a clutter of every objects. Another line of works studies target-oriented grasping by assigning an image of the target object [5], or providing a demonstration [6]. Recent works [1]–[4] step forward to explore target-oriented grasping in clutter. [1], [4] achieve the target object grasping by grasping other obstacles away. [28]–[32] conduct some pre-grasping actions such as pushing to singulate the target object. However, these works require an image of the target object, or assume that a target object mask in the scene is given, which is often infeasible, especially in open-world applications, such as tabletop rearrangement. Instead, language instructions are more flexible in open-world applications [12], [33].

B. Language-conditioned Manipulation

Because of the convenience of natural language, language-conditioned manipulation has become a promising research topic in recent years. [34]–[36] study grasp detection based on language instructions in scattered scenes. [37] explores to ground language to some long-horizon manipulation tasks, but with instructions limitation. [33] introduces language as a flexible goal specification with objects of simple shape in scattered scenes. Thanks to the develop of language models [7], [8] and multi-modal models [9], recent works [10]–[13], [38] are able to ground more flexible language instructions into long-horizon manipulation. However, such works consume lots of demonstrations to learn raw image

based policy for a good performance, and take plenty of training steps for convergence. In addition, one has to address sim2real gap to deploy raw image based policy in real world. Other works [14]–[18], [39] focus on language-conditioned grasping by separately visual grounding and grasp planning with object-centric representations. [15], [16], [18] ground object bounding boxes in the scene with pre-defined visual attributes, or object labels, which calls for handcrafted design in planner, and limits the range of language instructions. To overcome visual grounding error and language ambiguity, they need to interact with human by asking questions with pre-defined templates. In this line of work, language-conditioned grasping in clutter is much less explored except that [14], [17]. [14] conducts direct visual grounding by calculating the cosine similarity between text feature and object features, which cannot handle visual grounding error and visual occlusion. Analogous to [16], [17] trains on RefCOCO dataset [40] and requires object labels for grounding.

In this paper, we also focus on the task of language-conditioned grasping in clutter. There are several differences between our methods and the above target-oriented grasping works. Instead of grounding a matching score of each object bounding box from language instructions, we utilize the fusion feature of visual and language from CLIP to learn a grasp policy without handcrafted grounding rules, thus applicable to more flexible language instructions. Also, by jointly modeling vision, language and action and conducting cross attention with grasp queries, our system focuses on the ability of all grasp actions in the scene, rather than relying on the visual grounding to select an object.

III. METHODS

A. System Overview

As shown in Fig. 2, given a language instruction, and a set of object bounding boxes from the detection module, our system generates visual features of all bounding boxes and language feature by pre-trained CLIP [9] image encoder and text encoder. Each bounding box visual feature is fused with language feature to generate box visual-language feature, while fused with the embedding of bounding box center position to generate box visual-position feature. We utilize the pre-trained graspnet [19] to yield a set of grasp poses which are encoded to spatial features. Given these feature embeddings, we propose to jointly model vision, language and grasp action by a cross-attention transformer, which takes grasp features as queries, box visual-position features as keys, box visual-language features as values, and outputs the cross-attention features. Then the policy and critic MLPs take as input the cross-attention features, and output the logits and values of all grasps respectively. At inference time, our policy conducts a sequence of grasp poses to grasp away the obstacles and finally catch the target object.

B. Pre-processing with Pre-trained Models

Object-centric Representation. In this paper, we obtain object bounding boxes for object-centric representation, which are fed into pre-trained models for pre-processing.

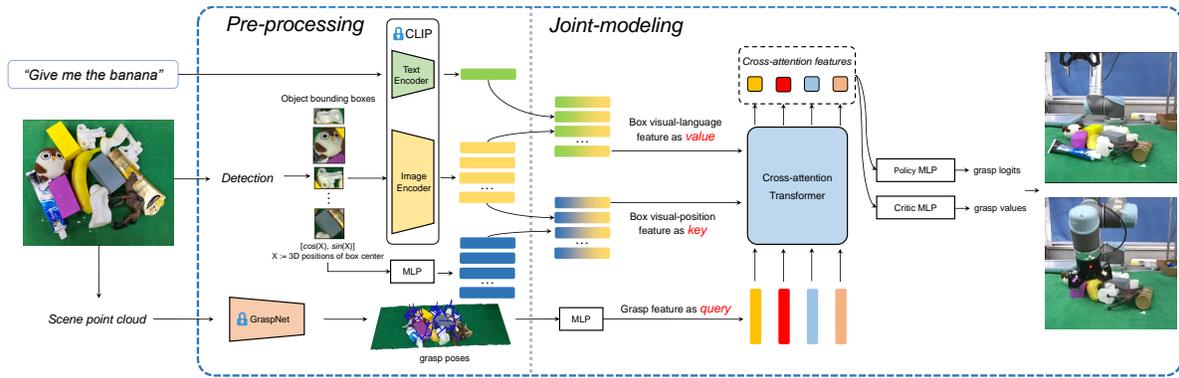


Fig. 2. **System Overview.** Given a language instruction, and object bounding boxes from detection module, our system pre-processes visual-language input by CLIP [9], and generates grasp poses by graspnet [19] (see Sec. III-B). And we jointly model vision-language-action by a cross-attention transformer, of which outputs are fed into the policy and critic MLPs to generate logits and values of all grasps (see Sec. III-C).

Visual-language Model. We propose to yield visual and language features through CLIP [9], which aligns image and language features by training on millions of image-text pairs. Instead of directly applying CLIP on the raw image of a clutter of objects, we propose to process object crops in the scene. Given a raw image with a clutter of M objects, N bounding boxes are extracted from detection models, and N object crops are obtained by cropping the raw image with corresponding bounding boxes. Note that M is not equal to N in most cases due to object occlusion in clutter and the detection uncertainty. Then N object crops are fed into the CLIP image encoder and the language instruction is fed into the CLIP text encoder to get N bounding box visual features and a language feature in CLIP image-text alignment domain.

Grasp Model. We adopt the pre-trained model of graspnet [19] to get a set of grasp poses of the whole scene. Graspnet is a 6DoF grasp detector taking as input a scene point cloud, and predicting K 6DoF grasp poses.

Thanks to pre-trained vision-language and grasp model, the sample efficiency is effectively improved. Also, utilizing CLIP [9] and graspnet [19] trained with massive simulation and real data for pre-processing relieves the sim2real gap.

C. Vision-language-action Joint Modeling

Vision-language-action Cross-attention. Instead of separately conducting visual grounding and selecting a best-matched object to grasp, we propose to jointly model vision, language and action by transformer’s attention mechanism [41]: $\text{Attention}(Q, K, V) = \text{softmax}(QK^T)V$, where Q, K, V donate query, key and value. Specifically, a cross-attention transformer is adapted to conduct cross attention among multi-modal information, and outputs fusion features. We fuse N box visual features with the language feature by element-wise product respectively and get N box visual-language features accordingly. Also, we generate K grasp pose features by MLP. Considering that our goal of joint modeling is to get the fusion vision-language-action features, and to map grasp features into the CLIP image-text feature domain, we set K grasp pose features as queries and N box visual-language features as values. To combine spatial features (*i.e.* grasp features) and visual-language features, we further utilize the box visual-position features as keys. To be specific, 3D positions of the centers of bounding boxes are

projected into a nonlinear space (positional embedding as in [42]), followed by a MLP to encode the box position embeddings. Then N box position embeddings are fused with N box visual features by adding. After vision-language-action cross attention, K cross-attention features will be obtained.

Policy and Critic. In this work, our system is trained through deep reinforcement learning with discrete SAC [43], [44]. We consider the cross-attention features at step t as state s_t , the grasp pose selection at step t as action a_t . Then our policy is defined as $\pi(a_t|s_t)$, and critic outputs Q function $Q(s_t, a_t)$. The policy and critic MLPs both take K cross-attention features as input, and output logits and q values of K grasp poses. Note that our policy and critic are capable of processing variable number of actions by parallel processing K cross-attention features, which corresponds to K grasps.

Language Instructions. By jointly modeling vision-language-action and conducting cross attention with grasp queries, our system focuses on the ability of grasp actions without requirement of object labels or visual attributes for grounding and handcrafted rules for planning. This formulation enables our system to adapt to more flexible language instructions without assigning concrete object labels. Our system is trained with language templates such as “Give me the {keyword}”, “Get something to {keyword}”, where {keyword} can be a concrete label (*e.g.* banana), or a general label (*e.g.* fruit), or the attribute of color (*e.g.* red), shape (*e.g.* round), or even object function (*e.g.* hold other things). Although some of the language instructions are ambiguous, our policy can learn from experience by trial and error.

D. Curriculum Learning Schedule

Directly training a policy in clutter might make the agent struggle with the trade-off between grasping away other obstacles and focusing on the target object. Considering that our final goal is to grasp the target object, we design a two-stage curriculum learning schedule to ease the training:

Stage I. Scattered Scene Training. In this stage, we train our policy in a scattered scene with a objects randomly dropped in the workspace, which aims at guiding the robot to focus on grasping the target object. The number of grasps is limited up to 5, and once the target object is successfully grasped, the episode ends. The reward function is defined as:

$$R_g = \begin{cases} 2, & \text{if successfully grasping the target object} \\ -1, & \text{otherwise} \end{cases}$$

Stage II. Cluttered Scene Training. For this stage, our policy is trained in cluttered environments with b objects randomly dropped in the workspace, where the tight packing or occlusion degenerates the successful grasp of the target object. For each episode, the robot is supposed to grasp away obstacles hindering the grasping of the target object, and finally grasp the target object. The number of grasps is limited up to 8. In this stage, we encourage the robot to grasp the obstacle objects near the target object. Therefore, we implement less punishment if the object is nearer to the target, and we model the reward function as follows:

$$R_g = \begin{cases} 2, & \text{if successfully grasping the target object} \\ -\frac{dist}{dist_{max}}, & \text{if successfully grasping other objects} \\ -1, & \text{otherwise} \end{cases}$$

where $dist$ is the 3D position distance between the grasped object and the target object. If there are more than one target objects, it is the minimize distance from all target objects. $dist_{max}$ is the diagonal length of the workspace. Dividing $dist$ with $dist_{max}$ normalizes the value within $[-1, 0)$.

CLIP Grounding Prior. To optimise CLIP to guide the policy to select grasp actions of the target object, we further explore a CLIP-guided loss. We ground N visual-language pairs into N probabilities with CLIP, which represent the corresponding visual-language matching scores, and multiply the probabilities with the $N \times K$ box-grasping mapping matrix, which assigns grasps to boxes by 3D position distance. In this way, we can get K mapped grasp probabilities as the grounding prior for the policy. Then a guided loss (*i.e.* Kullback-Leibler divergence) can be added for the policy training. Details can be found in Appendix.

E. Implementation Details

We train our system with five language templates: “Give me the {keyword}”, “I need a {keyword}”, “Grasp a {keyword} object”, “I want a {keyword} object”, “Get something to {keyword}” with 66 object models and 36 language keywords, which are categorized into four types: label, general label, shape or color, function. Network parameters of CLIP and graspnet will be fixed during training. We train the first stage with $a = 8$ objects and 500 episodes, and the second stage with $b = 15$ objects and 1500 episodes. If using CLIP-guided loss, the policy loss and the loss of the cross-attention module will be added with the CLIP-guided loss described in Sec. III-D for the former 800 episodes. Other training details of can be accessed in Appendix.

IV. EXPERIMENTS

In this section, we carry out a series of experiments to evaluate our system. The goals of the experiments are: 1) to demonstrate that our policy is effective for the task of language-conditioned grasping in clutter; 2) to evaluate the generalization performance of our policy on unseen objects and language instructions. 3) to indicate the sample efficiency of our method. 4) to test whether our system can successfully



Fig. 3. Test cases in simulation. The top two rows demonstrate the arrangements with seen objects, while the bottom row shows the arrangements with unseen objects. The target objects are labeled with stars.

transfer from simulation to the real world. We compare the performance of our system to the following baselines:

CLIPORT-Grasp is a method which learns actions from raw image. It is a variant of CLIPORT [12], and utilizes CLIP to ground semantic concepts from raw image. Grasp actions are also learned from scratch by predicting pixel-wise affordance on raw image. To be specific, we use the attention stream of [12] for grasping.

CLIP Grounding is a semantic grounding approach which grounds all object bounding boxes with language instruction by CLIP [9]. Specifically, it computes similarities between the CLIP visual features of boxes and the language feature, and selects the bounding box with the highest similarity as the target. Then the policy randomly samples a grasp pose corresponding to that box by the box-grasp mapping matrix described in Sec. III-D. If there is not grasp for that box, the policy will randomly select a grasp pose.

Raw Image is a policy which uses CLIP to process the raw image, and utilizes graspnet to yield grasp poses. This baseline is to demonstrate the performance of the policy without object-centric representation, but with grasp priors. The network architecture is the same as ours.

Raw Image Grids is a policy similar to **Raw Image**. Instead, it uses grid crops of the raw image as CLIP inputs.

For fair comparison, all the baselines are trained with the same schedule with discrete SAC as ours except **CLIP Grounding**, which directly deploys CLIP for grounding.

A. Evaluation Metrics

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

- **Task Success Rate:** the average percentage of task success rate over $i = 15$ test runs. If the robot picks up the target object within $j = 8$ action attempts in a run, then the task is considered successful and completed.
- **Motion Number:** the average motion number per task completion.

B. Simulation Experiments

Our simulation environment is built in PyBullet [45], which involves a UR5 arm, a ROBOTIQ-85 gripper, and a camera of Intel RealSense L515.

Comparisons to Baselines. We conduct test experiments with a series of test cases, including arrangements with seen objects and unseen objects (Fig. 3). Arrangements with

TABLE I
SIMULATION RESULTS ON ALL ARRANGEMENTS

Method	Task Success		Motion Number	
	seen	unseen	seen	unseen
CLIPORT-Grasp	—	—	—	—
CLIP Grounding	59.3	64.0	4.80	4.15
Raw Image	48.1	45.3	4.35	5.37
Raw Image Grids	60.7	70.7	4.35	3.49
Ours	74.3	78.7	4.11	3.98
Ours w/ CLIP-guided loss	77.3	74.7	3.74	3.65

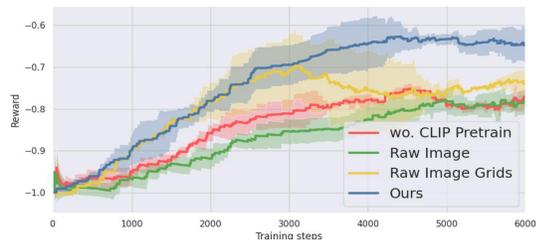


Fig. 4. Compared training performance of our method and three methods.

seen objects involve 10 cases with adversarial clutter where the robot might need to grasp away other obstacles for target object grasping. Compared results reported in Table I demonstrate that our method outperforms all baselines across all metrics. **CLIPORT-Grasp** fails in all cases, which suggests that learning grasp actions from scratch and grounding semantics from raw image might fail to work with just 2000 episodes training. Also, **CLIPORT-Grasp** conducts top-down grasping, which might fail in clutter in many cases. Instead, with object-centric representation and pre-trained grasp model, **CLIP Grounding** can achieve around 60% task success rate, but with the highest motion number. This might due to the fact that **CLIP Grounding** tends to focus on the best-matched object and keep grasping it, and ignores the possibility of grasping away other obstacles, which is effective for grasping in clutter. Also, it can not handle the error of the grounding results. **Raw Image** shows the second worst task success rate, and always cannot fix the target object in the cluttered scenarios. Instead, **Raw Image Grids** gets better performance than **Raw Image**. This performance gain mainly comes from the grid crops, which help the policy easier to focus on the target zone. However, there is little object-level information in grids, which might account for its unsatisfactory performance. By leveraging object-centric bounding boxes and jointly modeling vision, language and action, our policy can focus on the target object without limiting by the grounding error. Also, **Ours w/ CLIP-guided loss** can achieve better performance, especially in motion number, indicating that the CLIP prior helps the policy to be more concerned with the target object.

Results of 5 test cases with unseen objects are also shown in Table I. We observe that our method is capable of generalizing to unseen objects with similar language instructions, and achieves the best task success rate with a bit more motion number. Although **Raw Image Grids** achieves the least motion number, it does not prove its high efficiency since its task success rate is lower. It’s worth noting that **Ours w/ CLIP-guided loss** reports lower task success rate than ours. This might due to the fact that this policy is influenced by CLIP prior to a greater extent. Although it conducts more

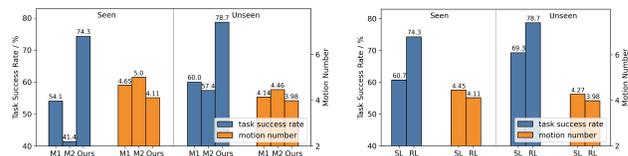


Fig. 5. Ablation on all arrangements for CLIP (left) and RL vs. SL (right).

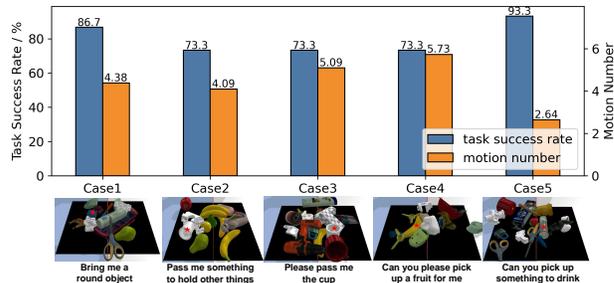


Fig. 6. Testing performance on unseen language templates.

target-driven grasps in some cases, as a trade-off, it is also affected by the CLIP grounding error.

Ablation Studies. We compare our methods with a series of ablation methods to test: 1) whether our techniques can improve sample efficiency; 2) whether CLIP brings benefits to our policy; 3) whether our system gains more by learning with trial and error.

To demonstrate 1), we compare our training efficiency with **Ours w/o CLIP-pretrain**, **Raw Image** and **Raw Image Grids** to indicate whether our method benefits from pre-trained visual-language model and object-centric representation. We present the reward curves versus training steps in Fig. 4. It can be seen that our policy can improve performance at a faster pace and achieve a higher reward. This suggests that the CLIP prior and object-centric representation help our policy to capture vision-language information, thus improving the sample efficiency. **Raw Image Grids** shows better training performance than **Raw Image** by using image crops rather than raw image with a clutter of objects. It is interesting to note that **Raw Image Grids** can keep a similar pace as our method in 2k~3k steps, but fails to improve later. This shows that **Raw Image Grids** can handle scattered scenes (Stage I), but struggles to capture useful information in clutter with simple crops representation.

For 2), we make comparisons between our policy and two ablation methods: **M1: Ours w/o CLIP-pretrain** which trains CLIP from scratch, and **M2: Ours w/ RN50-BERT** which encodes image and text by ImageNet-trained ResNet50 [46] and BERT [7]. Testing results are shown in Fig. 5 (left). It’s obvious that without CLIP prior, **Ours w/o CLIP-pretrain** performs worse than ours, which suggests that pre-trained CLIP is a powerful prior benefiting the policy training. **Ours w/ RN50-BERT** achieves the worst performance, indicating the unfeasibility of simply combining two pre-trained single-modal models. Instead, CLIP aligns visual feature and language feature into a common feature space, thus simplifying the fusion of vision and language and easing the policy learning.

Also, we additionally train our policy with supervised learning, which is trained with the same amount of success-

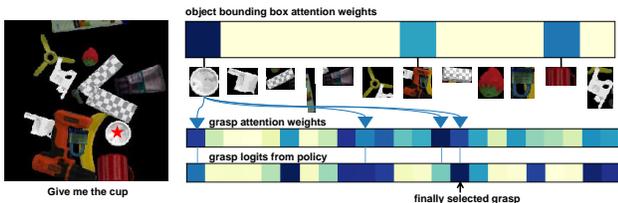


Fig. 7. A case study to show the cross attention weights. Given a scenario where the target object is labeled with a star, we show the attention weights of all bounding boxes (top) and grasps of the box with the highest attention weight (middle). The bottom row shows the grasp logits outputted by policy. Note that grasp poses are assigned to boxes by the box-grasp mapping matrix in Sec. III-D. We mark the grasp poses of the box with triangles, and the selected grasp of the policy with an arrow.

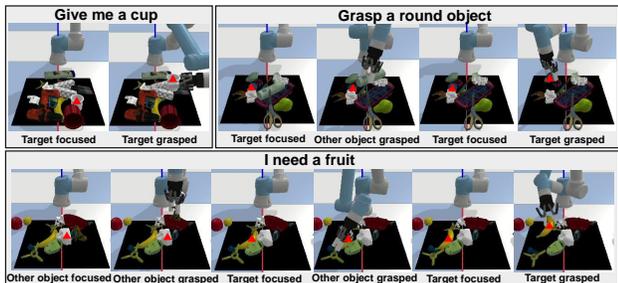


Fig. 8. Example testing sequences in simulation of 3 cases. Objects with the highest attention weights are marked with triangles.

ful action sequences as reinforcement learning, and testing results (Fig. 5 (right)) show the advantage of training by trial and error. This might be because the supervised policy is unable to be competent for scenarios differing much from those in training set. And the training action sequences are not enough to cover the testing domain. Instead, training with RL by trial and error implicitly learns a value function for actions, which might bring better generalization performance.

Novel Language Templates. Furthermore, we test our policy with some novel language templates. As shown in Fig. 6, we evaluate our policy with 5 scenarios containing seen objects, but under unseen language templates such as “Please pass me the {keyword}”, “Can you please pick up a {keyword} for me”. Results suggest that our policy is applicable for some unseen language templates, which further confirms our claim on language generalization.

Case Studies. Fig. 7 visualizes the cross attention weights of a test case. We can see that the target object can be focused with the highest attention weight, followed by two surrounding objects, which means that the cross-attention mechanism is able to catch the target object with spatial information. Also, grasp poses for the target object can achieve high attention weights. Taking the cross-attention features, our policy further adjusts the logits distribution of grasp poses and draws more attention to the target. Also, we report three typical cases in Fig. 8. In the first case, the target object can be focused and grasped successfully, which is the most ideal situation. In more situations (the second case), the policy chooses to grasp the surrounding object for better grasping of the target. When failed to focus on the target at the beginning (the third case), our policy is capable of adjusting by exploring with a sequence of grasp actions.



Fig. 9. Test cases in real world. Target objects are labeled with stars.

TABLE II
REAL-WORLD RESULTS ON TEST ARRANGEMENTS

Method	Task Success	Motion Number
CLIP Grounding	56.0	4.84
Raw Image Grids	48.0	5.33
Ours	72.0	5.13

C. Real-world Experiments

In this section, we evaluate our system in real-world settings. Our real-world platform involves a UR5 robot arm with a ROBOTIQ-85 gripper, and an Intel RealSense L515 capturing RGB-D images of resolution 1280×720 . Object bounding boxes are generated by a pre-trained open-set detection model (details in Appendix), and test cases are shown in Fig. 9. We compare our methods with **Raw Image Grids** and **CLIP Grounding** which have demonstrated better performances than others in simulation experiments. Note that these methods are transferred from simulation to the real world without any retraining. Test results are reported in Table II. Overall, our policy shows much better performance in task success rate, and is capable of generalizing to unseen target objects (*i.e.* purple block, toothpaste and toy puppet). Although **CLIP Grounding** costs the least motion number, it shows a task success rate low at 50~60%. For case 1 and 2, where CLIP grounds with high accuracy, **CLIP Grounding** can focus on the target grasping with less steps. However, when the target is occluded (case 3), **CLIP Grounding** frequently fails to figure out the target. **Raw Image Grids** shows the worst performance in real-world cases, which tends to select grasps in a zone for the same case. Also, it demonstrates a significant performance degradation, which might be because that the sim2real gap (*e.g.* background) affects the crop grids input. Instead, by using bounding boxes, the influence of background can be decreased.

V. CONCLUSIONS

In this work, we study the task of language-conditioned grasping in clutter. Instead of following the cascaded pipeline of visual grounding and grasp detection, we propose to jointly model vision, language and action in a union view, which is not limited by visual grounding error and is applicable with more flexible language instructions. We leverage pre-trained CLIP, a multi-modal model aligning visual and language into a common feature space, and utilize 6DoF graspnet to generate grasps in the scene, which effectively improves the sample efficiency. We train our policy through deep reinforcement learning and evaluate it with a series of experiments in simulation and real world, which indicates that our method can achieve better task success rate with less motion number compared to other methods. Moreover, our method shows better generalization in scenarios with unseen objects and language instructions. It is worth mentioning that our method can also handle some novel language templates. More discussions can be accessed in Appendix.

REFERENCES

- [1] 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*, vol. 41, no. 7, pp. 690–705, 2022.
- [2] 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)*. IEEE, 2020, pp. 6232–6238.
- [3] 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)*. IEEE, 2018, pp. 3516–3523.
- [4] E. Jang, S. Vijayanarasimhan, P. Pastor, J. Ibarz, and S. Levine, “End-to-end learning of semantic grasping,” in *Conference on Robot Learning*. PMLR, 2017, pp. 119–132.
- [5] M. Sun and Y. Gao, “Gater: Learning grasp-action-target embeddings and relations for task-specific grasping,” *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 618–625, 2021.
- [6] M. Laskey, J. Lee, C. Chuck, D. Gealy, W. Hsieh, F. T. Pokorny, A. D. Dragan, and K. Goldberg, “Robot grasping in clutter: Using a hierarchy of supervisors for learning from demonstrations,” in *2016 IEEE international conference on automation science and engineering (CASE)*. IEEE, 2016, pp. 827–834.
- [7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [9] 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*. PMLR, 2021, pp. 8748–8763.
- [10] 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,” *arXiv preprint arXiv:2204.01691*, 2022.
- [11] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar *et al.*, “Inner monologue: Embodied reasoning through planning with language models,” *arXiv preprint arXiv:2207.05608*, 2022.
- [12] M. Shridhar, L. Manuelli, and D. Fox, “Cliport: What and where pathways for robotic manipulation,” in *Conference on Robot Learning*. PMLR, 2022, pp. 894–906.
- [13] K. Zheng, X. Chen, O. C. Jenkins, and X. E. Wang, “Vlmbench: A compositional benchmark for vision-and-language manipulation,” *arXiv preprint arXiv:2206.08522*, 2022.
- [14] J. Hatori, Y. Kikuchi, S. Kobayashi, K. Takahashi, Y. Tsuboi, Y. Unno, W. Ko, and J. Tan, “Interactively picking real-world objects with unconstrained spoken language instructions,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3774–3781.
- [15] M. Shridhar and D. Hsu, “Interactive visual grounding of referring expressions for human-robot interaction,” *arXiv preprint arXiv:1806.03831*, 2018.
- [16] Y. Yang, X. Lou, and C. Choi, “Interactive robotic grasping with attribute-guided disambiguation,” *arXiv preprint arXiv:2203.08037*, 2022.
- [17] H. Zhang, Y. Lu, C. Yu, D. Hsu, X. La, and N. Zheng, “Invigorate: Interactive visual grounding and grasping in clutter,” *arXiv preprint arXiv:2108.11092*, 2021.
- [18] 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)*. IEEE, 2022, pp. 11 138–11 144.
- [19] 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*, 2020, pp. 11 444–11 453.
- [20] L. Pinto and A. Gupta, “Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours,” in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 3406–3413.
- [21] J. Mahler and K. Goldberg, “Learning deep policies for robot bin picking by simulating robust grasping sequences,” in *Conference on robot learning*. PMLR, 2017, pp. 515–524.
- [22] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke *et al.*, “Scalable deep reinforcement learning for vision-based robotic manipulation,” in *Conference on Robot Learning*. PMLR, 2018, pp. 651–673.
- [23] A. Ten Pas and R. Platt, “Using geometry to detect grasp poses in 3d point clouds,” in *Robotics research*. Springer, 2018, pp. 307–324.
- [24] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, “Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics,” *arXiv preprint arXiv:1703.09312*, 2017.
- [25] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, “Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 438–13 444.
- [26] C. Wang, H.-S. Fang, M. Gou, H. Fang, J. Gao, and C. Lu, “Graspness discovery in clutter for fast and accurate grasp detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 964–15 973.
- [27] D. Son, “Grasping as inference: Reactive grasping in heavily cluttered environment,” *arXiv preprint arXiv:2205.13146*, 2022.
- [28] M. Kiatos and S. Malassiotis, “Robust object grasping in clutter via singulation,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1596–1600.
- [29] 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)*. IEEE, 2020, pp. 8408–8414.
- [30] Y. Yang, H. Liang, and C. Choi, “A deep learning approach to grasping the invisible,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2232–2239, 2020.
- [31] 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*, vol. 6, no. 4, pp. 6337–6344, 2021.
- [32] B. Huang, S. D. Han, J. Yu, and A. Boularias, “Visual foresight trees for object retrieval from clutter with nonprehensile rearrangement,” *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 231–238, 2021.
- [33] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor, “Language-conditioned imitation learning for robot manipulation tasks,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 139–13 150, 2020.
- [34] A. B. Rao, K. Krishnan, and H. He, “Learning robotic grasping strategy based on natural-language object descriptions,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 882–887.
- [35] H. Ito, H. Ichiwara, K. Yamamoto, H. Mori, and T. Ogata, “Integrated learning of robot motion and sentences: Real-time prediction of grasping motion and attention based on language instructions,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 5404–5410.
- [36] Y. Chen, R. Xu, Y. Lin, and P. A. Vela, “A joint network for grasp detection conditioned on natural language commands,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4576–4582.
- [37] D. K. Misra, J. Sung, K. Lee, and A. Saxena, “Tell me dave: Context-sensitive grounding of natural language to manipulation instructions,” *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 281–300, 2016.
- [38] E. Stengel-Eskin, A. Hundt, Z. He, A. Murali, N. Gopalan, M. Gombolay, and G. Hager, “Guiding multi-step rearrangement tasks with natural language instructions,” in *Conference on Robot Learning*. PMLR, 2022, pp. 1486–1501.
- [39] H. Ahn, S. Choi, N. Kim, G. Cha, and S. Oh, “Interactive text2pickup networks for natural language-based human-robot collaboration,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3308–3315, 2018.
- [40] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, “Modeling context in referring expressions,” in *European Conference on Computer Vision*. Springer, 2016, pp. 69–85.

- [41] 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*, vol. 30, 2017.
- [42] 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*, 2020, pp. 405–421.
- [43] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.
- [44] P. Christodoulou, "Soft actor-critic for discrete action settings," *arXiv preprint arXiv:1910.07207*, 2019.
- [45] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," <http://pybullet.org>, 2016–2021.
- [46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [47] Z. Liu, W. Liu, Y. Qin, F. Xiang, M. Gou, S. Xin, M. A. Roa, B. Calli, H. Su, Y. Sun *et al.*, "Ocrtoc: A cloud-based competition and benchmark for robotic grasping and manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 486–493, 2021.
- [48] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

A. Details about CLIP Grounding Prior

By calculating the cosine similarity between N box visual feature and the text feature, we can ground N visual-language pairs into N probabilities, each of which represents a matching score of the corresponding object bounding box against the language instruction. We obtain 3D positions (x, y, z) of bounding box centers by transforming pixel coordinates to world coordinates with camera extrinsics for (x, y) and getting the depth of the center pixel of depth image for z . Leveraging the 3D positions of the bounding box centers, 3D distance between each grasp and each bounding box can be calculated. If the distance of a grasp and a bounding box is smaller than a distance threshold $d = 0.05$ (a hyper parameter referring to [47], which uses graspnet [19] to generate grasps and assigns grasps to objects), then the grasp is assigned to that object, which results in a $N \times K$ box-grasp mapping matrix. By multiplying CLIP probabilities and the box-grasping mapping matrix, we can get K mapped probabilities, which can be regarded as a grounding prior for the policy. Then a guided loss (*i.e.* Kullback-Leibler divergence) can be added for the policy training.

B. More Training Details

Our training object models are from GraspNet-1Billion [19]. For simplicity, we generate object bounding boxes from the mask image in Pybullet [45], of which each pixel donates the index of the object visualized in the camera. Specifically, the bounding box of pixels with the same index corresponds to an object-centric representation. And the bounding box whose size is smaller than 15×15 will be abandoned. Note that this representation is not a ground-truth bounding box, but considering object occlusion. During the training stage, we randomly sample a language template and a keyword to form a complete language instruction. And the probability of the keyword sampling is 0.4, 0.2, 0.2, 0.2 for the above four types. For the keyword type of general label and function, which is regarded to form the difficult scenarios, we will drop two target objects in the scene, and successfully catching one is regarded as finishing the task. For the rest type of keywords, one target object is dropped into the workspace, and grasping the target object means task completion.

3D positions of box centers are obtained as described in Appendix. A. And before feeding into the network, the positions are projected into a nonlinear space by positional encoding as in [42]. We adopt the transformer architecture of the text encoder in [9], and conduct multi-head attention with different query, key and value. Parameters of the cross transformer includes width of 512, head of 8 and layer of 1.

We train our method with discrete SAC [44]. The temperature parameter α is initialized as 0.2 with automatic entropy tuning. We define the policy as $\pi(a_t|s_t)$, where s_t donates the cross-attention state at time t , and a_t represents the grasp action at time t . The critic outputs Q function $Q(s_t, a_t)$, then the loss function of actor, critic and α can be formulated as Eqn. 1-3. And the loss of the cross-attention module is the

TABLE III
ABLATION RESULTS FOR ARCHITECTURE DESIGN

Method	Task Success		Motion Number	
	seen	unseen	seen	unseen
Arrangement	57.3	48.0	4.65	4.23
Position as Key	58.7	38.7	4.28	4.67
FiLM Fusion	58.7	38.7	4.28	4.67
Ours	74.3	78.7	4.11	3.98

sum of policy loss and critic loss. All the networks are trained with Adam optimizer using fixed learning rates 3×10^{-4} . Our future discount γ is set as a constant at 0.99.

$$\mathcal{L}_\pi(\theta) = \pi_\theta(a_t|s_t)^T (-Q_\theta(s_t, a_t) + \alpha \log \pi_\theta(a_t|s_t)) \quad (1)$$

$$\begin{aligned} \mathcal{L}_Q(\theta) &= (Q_\theta(s_t, a_t) - y(r_t, s_{t \sim t+1}, a_{t \sim t+1}))^2 \\ y(r_t, s_{t \sim t+1}, a_{t \sim t+1}) &= r_t + \gamma (\pi_\theta(a_t|s_t)^T V_\theta(s_{t+1}, a_{t+1})) \\ V_\theta(s_{t+1}, a_{t+1}) &= Q_\theta(s_{t+1}, a_{t+1}) - \alpha \log \pi_\theta(a_{t+1}|s_{t+1}) \end{aligned} \quad (2)$$

$$\mathcal{L}_\alpha(\alpha) = \pi_\theta(a_t|s_t)^T (-\alpha \log \pi_\theta(a_t|s_t) + \bar{H}) \quad (3)$$

where r_t is the reward at step t , θ donates the network parameters, and \bar{H} is the target entropy.

C. Details about Object Detection Model

We deploy a pre-trained open-set detection model in our real-world experiments. The detection model is trained with data from GraspNet-1Billion [19] with $mAP = 70.70$ for known objects and $mAP = 34.53$ for unknown objects.

D. Ablation Studies of Architecture Design

In this section, we compare our methods with two variants to verify the design of our joint modeling. Compared results are shown in Table III. **Position as Key** is a variant which directly uses the 3D positions of bounding boxes as the keys of cross attention, while **FiLM Fusion** is a variant to fuse the cross attention feature of visual and action with the language feature by FiLM [48], which is a fusion technique for visual-language reasoning. It is obvious that the performance of both variants are unsatisfactory. **Position as Key** tries to simply combine geometry information (*i.e.* grasp pose) and visual-language information by the 3D positions of bounding boxes, which might overfit to seen scenarios. **FiLM Fusion** fails possibly due to the fact that the visual feature and language feature has been aligned into one space by CLIP, but FiLM additionally introduces a inductive bias, which makes the fusion feature out of the distribution of CLIP domain.

E. Example Action Sequences in Real-world

Fig. 10 shows two example testing sequences in real world. In these two cases, the target objects (*i.e.* toy puppet and toothpaste) are unseen during training in simulation. For the first case, the robot is able to grasp away the surrounding object to make space for the target object grasping. For the second case, the target object is occluded by other objects, and fails to extracted by the detection module at the beginning. Instead, the target object is partially included in other objects' bounding box. We observe that in this situation, our policy can also grasp away the near objects

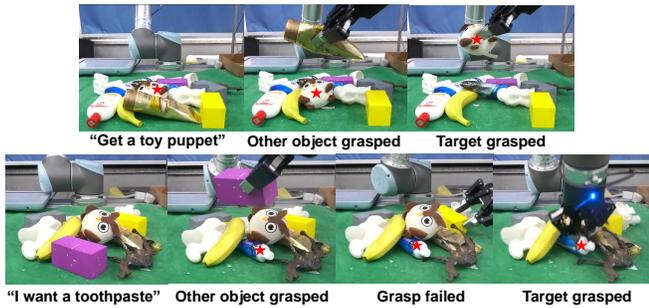


Fig. 10. Example testing sequences in real world. Target objects are labeled with stars.

and finally successfully grasp the target, which demonstrate that our policy might capture the target information in the surrounding bounding boxes, which guides the exploration of the policy.

In real-world experiments, failure happens mainly when the detection model detects a wrong target object, or the graspnet fails to detect a successful grasp of the target object.

F. Limitations and Future Works

In this work, we focus on the grasping task in language-conditioned manipulation. Our method is trained and tested with language instructions containing only one concept. However, we believe compound concepts such as "a red and round object" can be integrated into this work by additional finetuning. Additionally whether grasping the target object is decided by human. In future work, we will consider distinguishing the final state by the model itself. Also, more complex tasks, such as pick-and-place, or tool use, can be extended in the future.