CLIP-Seg: Open-World Semantic Segmentation for Robotics Manipulation

Lirui Wang*
MIT CSAIL
liruiw@mit.edu

Boyuan Chen*
MIT CSAIL
boyuanc@mit.edu

Abstract

Manipulating interested objects in diverse environments is an important skill for future household robots. Past approaches involve training object detectors on a fixed labelled dataset followed by planning algorithms. However, these methods are limited in the real world since the object class of interest can be absent in the dataset. In this work, we propose CLIP-Seg that uses free-form language to query the segmentation mask of a particular object. CLIP-Seg uses a two-steam model, proposed by a recent work, to combine the spatial information fined with supervision and the semantic information pre-trained in CLIP. We conduct extensive training and evaluations in both simulation and the real world datasets. For both domains, we show that CLIP-Seg can generalize to unseen descriptions such as attributes, spatial locations, and relational descriptions of interested object, as well as novel test set images. Finally, we apply the language-conditioned segmentation method for object pick-and-place in a simulated environment. [1].

1 Introduction

To enable future intelligent robots to interact with the unstructured and unseen environments, researchers often establish object-centric abstractions to bridge raw perception and planning algorithms. (Dogar and Srinivasa, 2011; Wang et al., 2020). With the surge of learning methods and the increasing availability of annotated datasets (Lin et al., 2014; Krizhevsky et al., 2012), detection (Duan et al., 2019; Redmon et al., 2016) and segmentation (He et al., 2018) methods are widely adopted in robotic pipelines. Some of these methods are developed with a large number of object classes in mind. However, due to the nature of how these dataset are set up, these methods heavily rely on the notion of pre-defined object classes. Such reliance has several severe limitations for general-purpose deployments in the real life. First, our interested object may not fit in one of the classes pre-defined in the dataset. In that case, the query of interested object by label is intractable. Second, the pre-defined labels are not fine-grained enough and cannot be modified with additional attributes for down-stream purpose. For example, we may want to query the segmentation mask of "red apple" in a given image, but a pre-trained model may give us that of all apples, instead of just red ones if it's trained with a class called "apple".

The transfer learning paradigm is recognized as a key discovery that has pushed the state-of-the-arts in many learning domains. Through learning from massive data, large-scale visual-language pre-trained models such as CLIP (Radford et al., 2021) capture expressive visual and language features. CLIP applies contrastive learning on image and language pairs of a large number of objects and their properties described by text, and learns to encodes semantic and contextual information such as priors of object spatial locations and associations. From a language perspective, the model is trained on open vocabulary as compared to a fixed set of prompt formats. Finally, compared to more standard supervised learning baselines such as ResNet pretrained on ImageNet (Krizhevsky et al., 2012), CLIP has shown to have impressive robustness and generalization that are central in prediction tasks.

To distill the large-scale image-language knowledge for robotics, we follow a recent architecture proposed by Cliport (Shridhar et al., 2021) for single-stage referring segmentation task. The architecture adopts two stream inputs: the spatial sensor stream that has been shown to improve segmentation (Xie et al., 2021; Xiang et al., 2020) and semantic stream that maps visual contexts to embedding vectors. The language input is used as a

[1] Our presentation slide can be found at this link

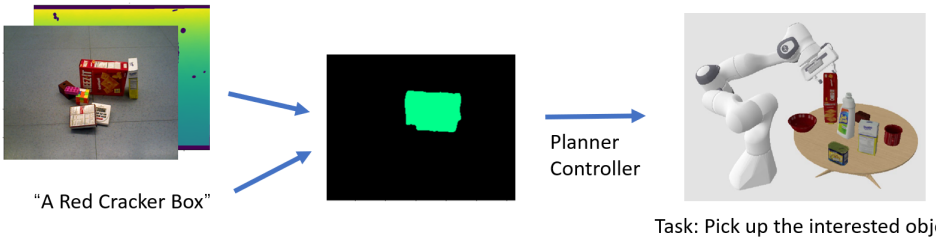

Figure 1: Task Illustrations. With Clip-Seg, the user can refer to an object with a phrase, and the segmented mask with depth inputs can be used to retrieve the point cloud of the target object. The point cloud of the target object can then be fed into the policy for manipulation tasks.

query token to the implicit scene representation for an object mask output.

We train and test separately on table-top scenes in simulation (Xie et al., 2021) and real-world dataset (Suchi et al., 2019; Wang et al., 2021a) and achieve good generalizations. We show that our model surpasses the performance of the baselines on Ref-OCID dataset. We also conduct ablation study on the depth stream, multi-modalness, and image overfitting. Finally we apply our segmentation method to robotic manipulation tasks. Qualitatively, we show that our model is able to predict sharp segmentation mask for query specified by language input. The model shows generalization to unseen language query which include spatial, attribute, and relational expressions and unseen objects that vary distinctly in appearance and shape.

2 Related Works

2.1 Referring Expression Segmentation

Referring expression segmentation refers to the task of segmentation of object from pixels conditioned on language expression. This task focus on the segmentation of an interest instance of object in a scene when multiple instances of the same class object are present. In particular, these expressions many describe both the semantic (such as color) and relational (such as relative position in the image) properties of the object. However, the standard datasets (Yu et al., 2016) for Referring Expression Segmentation are still developed on top of existing image segmentation datasets like MSCOCO (Lin et al., 2015), who pre-defines a fixed set of categories. The dataset simply adds instance-description annotations to such existing dataset. This means the methods (Ding et al., 2021)(Kamath et al., 2021)(Feng et al., 2021)(İlker Kesen et al., 2021)(Jing et al., 2021) trained on the Referring Expression Segmentation datasets are only trained on text containing existing objects categories that are already in the MSCOCO classes. For example, the MSCOCO dataset doesn't have a class called ipod so these methods are never trained on phrases like "an silver ipod on the left". Therefore, the previous work in Refering Expression Segmentation can only achieve instance or attribute level generalization rather than class level generalization. However, with pre-trained language embedding that provides similar embeddings for semantically similar name of objects, Clip-Seg may achieve generalization for open-text and unseen categories. For example, because "ipod" may have similar embeddings as "phone", and because they also look visually similar, the frozen CLIP model can potentially map these two image-language pairs to similar points in the latent space.

2.2 Large-Scale Unsupervised Learning on Internet Data

Clip (Radford et al., 2021) is a large pretrained language-vision proposed by OpenAI that aligns vision and language representations by training on millions of image-caption pairs. CLIP employs contrastive learning such that the inner product of embeddings of image and text respectively estimates the probability that they match with each other. It shows impressive result for language conditioned image generation. The contrastive encoders also provide semantic information about pixels and caption of countless object classes that cannot be learned from a segmentation dataset with a small number of categories. Researchers recently have also proposed the idea of foundation model (Bommasani et al., 2021) that focuses on large-scale pretrained models and their usages in downstream applications.

2.3 CLIP adaptation for downstream tasks

CLIP has been used recently in affordance learning for 2D pick and place tasks (Shridhar et al., 2021). Several works (Gu et al., 2021; Zakka, 2021) have

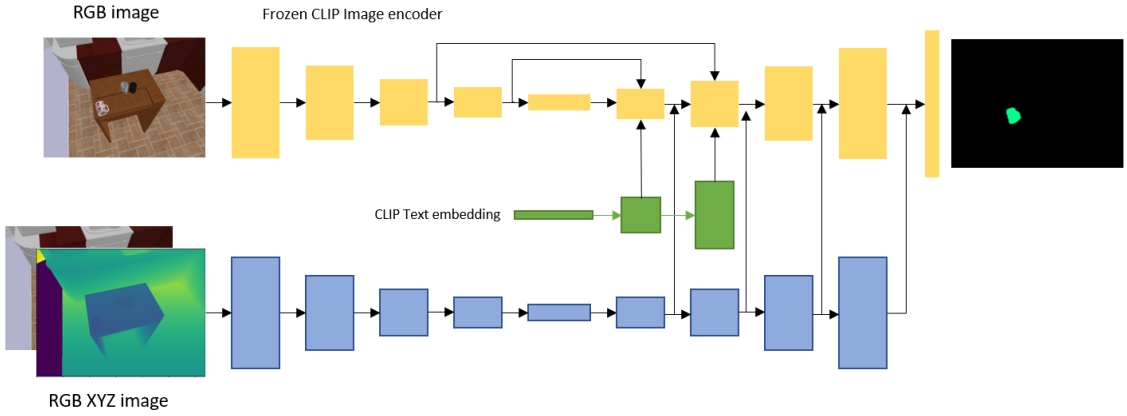

Figure 2: Network architecture. We use a two-stream architecture (Shridhar et al., 2021) for the referring expression segmentation task. We use a frozen CLIP network (Radford et al., 2021) as the base vision-language model for fusing the text and language embedding. Further, the latent features in the spatial stream is skip connected with the CLIP features. The overall model resembles a U-Net to predict a dense segmentation mask for the queried object.

also attempted to combine language features for object detection and show impressive zero-shot results on novel categories. (Zakka, 2021) uses a simple approach to detect objects by scoring pairs of image patches with language prompts. (Gu et al., 2021) proposes a novel network architecture and achieves comparable performance with supervised learning baselines on several benchmarks. Concurrent work (Zhou et al., 2021; Rao et al., 2021) proposes to use CLIP embeddings for dense pixel-wise tasks, and surpasses SOTA transductive zero-shot semantic segmentation methods. Other downstream vision tasks, e.g., text-driven image manipulation (Patashnik et al., 2021), image captioning (Gu et al., 2021), view synthesis (Kato et al., 2019) have also attempted to exploit such features for improved generality and robustness. In contrast, our method focuses on using CLIP along with a spatial stream for robotic manipulation.

2.4 Object segmentation for robotics

Segmenting unseen objects in cluttered scenes is an important skill that robots need to acquire in order to perform tasks in new environments. Mask-RCNN (He et al., 2018) is a standard segmentation model that has been applied to many segmentation benchmarks. Recently (Xie et al., 2021; Xiang et al., 2020) propose to use depth information for unseen object segmentation on table-top setting, whose data is directly accessible from rgbd cameras on robots. The more convenient way for a user to specify an object of interest is through natural language (Hu et al., 2016). In a robotics setting, a segmentation mask is easier to acquire than full state information. With only a segmented for the target object, a closed-loop robotic pipeline can be used to manipulate objects (Wang et al., 2021c,b). In practice, language is one of the most natural input candidates for users to specify target (Wang et al., 2021a).

3 Methodology

We consider the task of refering segmentation in robotic environments. The goal is to learn a function f that maps input X, a pair of language L that describes the interested object and a RGB-D image I of the scene, to output Y, a binary mask of that object. Using a standard supervised learning pipeline, we use separate encoders for language and image inputs and train on labelled datasets.

3.1 Network Architecture

In order to fuse both semantic information for text query and spatial information, we adopt a two-stream architecture (Shridhar et al., 2021). The spatial stream is a ResNet50 (He et al., 2016) that takes in normalized rgb pixels along with xyz coordinate map calculated from camera intrinsic matrix and depth image. The semantic stream uses a frozen CLIP ResNet50 image encoder that encodes RGB input into a spatial latent and the fixed language encoder that encodes the input token. The spatial latents are skip connected and fused with CLIP language embeddings along with latent layers from the spatial stream. Finally, the fused latent is passed through another resnet decoder to decode a dense binary segmentation mask for the queried object.

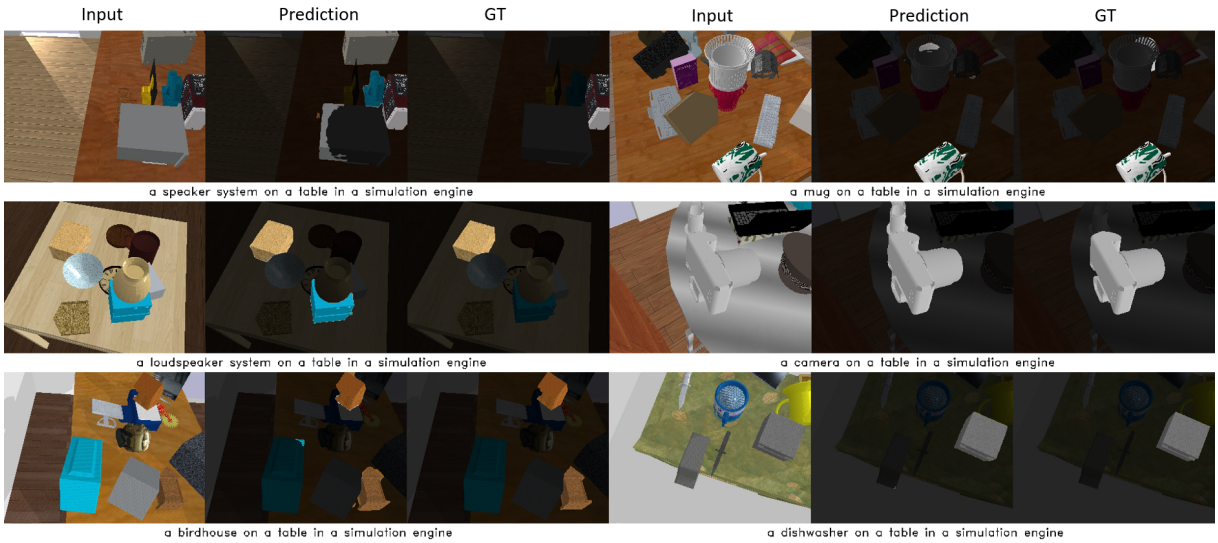

Figure 3: We train our method on the TOD dataset that features randomly generated cluttered scenes in simulation using shapenet objects. The above figure visualizes the segmentation on a test set featuring unseen images and unseen text descriptions. We obtained a test IOU of 0.76 on TOD dataset. The samples visualized are randomly selected.

3.2 Training Setup

There are two domains where a perception system can be trained in robotics: simulation and the real world. Simulation can easily generate large data with accurate groundtruth but not necessarily contain the diversity of the real world. Real-world data is more expensive to acquire. In both scenarios, we assume the dataset contains rgbd images of cluttered scene, refering language prompts, and the query prompt-segmentation mask. For each view in a scene, we sample a target object and generate a prompt using a template such as "a picture of [object] in simulation engine" or "The cereal box behind and on the bottom-left of the sponge". Multiple data points can be generated from the same view of the same scene with reference to different selected objects. Such generation scheme ensures our network to condition on the text embedding rather than making a caption-agnostic guess conditioned on the xyz spatial input. The training data for real-world dataset is scarce in the image level while the simulation data has poor rendering quality and narrow language descriptions. We observe certain overfitting issues in our dataset, which is potentially due the added RGB-D stream and the fusion since we stop the gradient flowing to the CLIP backbone. An interesting future direction is to incorporate inductive bias to improve the model.

3.3 Weighted Mask Losses

Since we modify the traditional multi class segmentation into the query based binary segmentation problem, we encounter the issue of unbalanced number of positive and negative entries for mask. The ground truth segmentation mask of our interested object is just a small proportion of the total number of pixels. Therefore, we use weighted binary cross loss to train the segmentation outputs. For each ground truth binary segmentation mask, we may calculate the proportion of positive and negative entries. We then scale the the total loss for each class by the inverse proportion of the mask pixels so the loss is balanced for positive and negative segmentation mask entries. In the case where the refering objects are not unique, we would generate mask labels for all the objects. The weighted binary cross entropy loss on the foreground (Xie et al., 2021) is defined as follows $l_{fg} = \sum_i w_i(Y_i, \bar{Y}_i)$ where $Y_i, \hat{Y}_i$ are the predicted and ground truth probabilities of pixel i in the image, respectively, and l_{ce} is the cross-entropy loss. We experiment with different thresholds for the binary predictions, and find 0.5 to be stable.

3.4 Robot Policy Pipeline

Previous works (Wang et al., 2021c,b) have shown that robot policy can be used to manipulate objects given a segmented point cloud of the scene. In our method, we plan to generate language prompt for an object in a cluttered scene to retrieve the associated mask. Then we can use a pre-trained policy in the pipeline to grasp the object. The clipseg method can be used as an user interface to refer to the object of interest, or can also be combined into the policy training step as in Cliport (Shridhar et al., 2021).

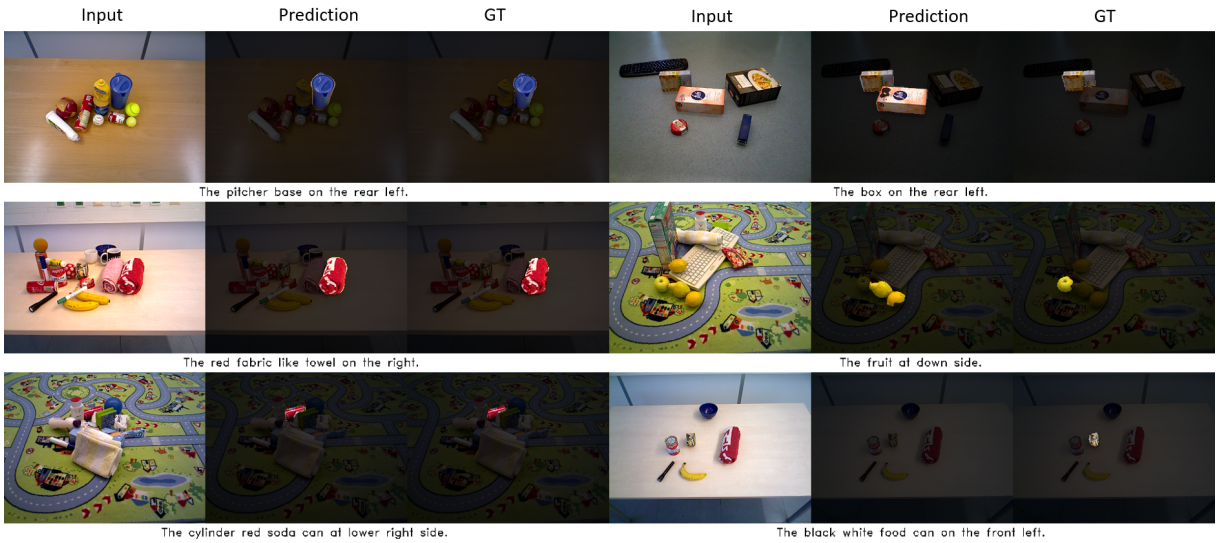

Figure 4: We further evaluate out method on the test split of OCID dataset, featuring real-world clutter scenes of YCB objects. We obtained a test IOU of 0.66. The samples visualized are randomly selected. Note that there are certain failure cases for fine-grained spatial reference (bottom right).

4 Experiments

4.1 Simulation Dataset Experiment

There exists many real-world datasets for image segmentation such as MSCOCO (Lin et al., 2014) and its language-referenced version RefCOCO. However, robotics settings often can have large scale RGB-D training data to train object segmentation networks. In addition, the main focus of this project is generalization instead of segmentic segmentation itself. Therefore, we propose to use readily available depth image in addition to the rgb images for our task. We propose to use Tabletop Object Dataset (Xie et al., 2021) that comprises of 40k synthetic scenes of cluttered ShapeNet objects on a (ShapeNet) tabletop in SUNCG home environment. The dataset filters ShapeNet object classes to roughly 25 classes of objects that could potentially be on a table. Example classes include: jar, mug, helmet, and pillow. Each scene has 5 views on a table top of cluttered objects with diverse 6D poses. We downscale the image size to be 320×240 for clip input and use a batch size of 12. The training split contains 40000 scenes, each of which will generate multiple training data points. We use 1000 scenes as the test split. The language prompt is designed to a template in the form "a _ on a table in simulation engine" where the blank is filled with the object class name. We run a small experiment to pick this prompt shown in Fig. 2. We run a small experiment to test what the template should be. For this image, we compute language embeddings for these three different prompts "some objects on a white table in a rendered image", "some objects on a white table in simulation engine", "some objects on a white table in pybullet". The zero-shot language-image matching probability are $0.32, 0.61, 0.07$. Thus, we pick the last one as our template. It is surprising that for CLIP model, we can simply use the phrase " in a simulation engine " to achieve the domain adaptations.

We interpret the model performance on the TOD dataset 3. There are several limitations of experiments on this dataset. First, the rendering quality in the TOD dataset is poor. Therefore, in many of the cases, even humans cannot correct identify the target object given the prompt due to insufficient texture of the objects. From the language perspective, we cannot generate meaningful and scalable language expressions using only the ShapeNet class names. The model has a high IOU performance but sometimes tends to over-segment the interested objects when there are multiple instances. However, since TOD dataset is generated using synthetic data, we will have abundant variations in terms of input images. The allow us us to evaluate the learned model on a lot of unseen scenes with thousands of shapenet objects. As shown in table 2, CLIP-Seg shows similar performance on train and test set, indicating high performance in generalize to unseen clutter scenes.

4.2 Real-World Dataset Experiment

Furthermore, we experiment with the recent real-world RGBD dataset OCID-ref (Wang et al., 2021a)

Figure 5: We show that our method can generalize to unseen text descriptions of target objects and correctly identify the referred objects when multiple instances of same object class are present in the same scene. CLIP-Seg is able to generalize to descriptions of different attributes including color and non-relational spatial locations. The prompts used in above figures are inputted by humans who have no prior knowledge about prompts in training set of OCID.

Figure 6: We found that CLIP-Seg is able to generalize to unseen names for the a target object class. For example, CLIP-Seg coorectly identifies the white bottle, which is likely some kind of detergent given class names like shampoo or detergent. The red bag in the second row is likely, though not necessarily ramen and CLIP-Seg is able to segment it out according to human intention. The prompts used in above figures are inputted by humans who have no prior knowledge about prompts in training set of OCID.

in the robotic settings OCID-ref (Wang et al., 2021a) has around 100 objects and 2,300 scenes with 305,694 referring expressions with depth in the real world. Compared to TOD dataset, it has language inputs generated from humans that are more rich and less likely to have ambiguity. Contrast to standard segmentation benchmark with only RGB input, the depth information often allows more sharp mask output. However, we do find that CLIP has limited capacity in learning the complex relational structure in the dataset, and also sometimes overfit in the spatial reference. For instance, when testing on another dataset such as YCB-depth and the simulation rendering, the output is not as satisfactory. Overall, as shown in table 1, we found that CLIP-Seg still shows great performance on the test split of the OCID-ref dataset with a test IOU of 0.728.

We further qualitatively evaluate the generalization ability of Clip-Seg from different perspectives. As stated in out motivation, we hope CLIP-Seg can 1. Segment out different instances of objects of same class but different attributes like colors. 2. Segment out objects of same class but different spatial locations. 3. Segment out same instance of object but with different possible names of it. To do so, we ask a human tester with no prior knowledge about training prompts to input prompts to test CLIP-Seg from these perspectives. In figures 7,6,5, we show show samples drawn from these tests. F

Figure 5 shows that CLIP-Seg is able to generalize to attributes very well, even if the human tester inputs something very different from prompts from the training set. In particular, CLIP-Seg has robust performance to color as attributes. We also observe that CLIP-Seg can sometimes generalize to unseen colors in prompts. For example, the color indigo is similar to blue, and humans can use this color to query segmentation for a blueish object and CLIP-Seg can output the correct segmentation zero-shot. In addition, we also ask human testers to deliberately query some instances with wrong attributes. For example, if a scene only contains red bottles and human queries green bottles, we found CLIP-Seg will correctly output nothing.

In figure 7, the human tester is asked to describe a particular instance of object with spatial locations like on the left or on the top right. CLIP-Seg can successfully differentiate objects based on different

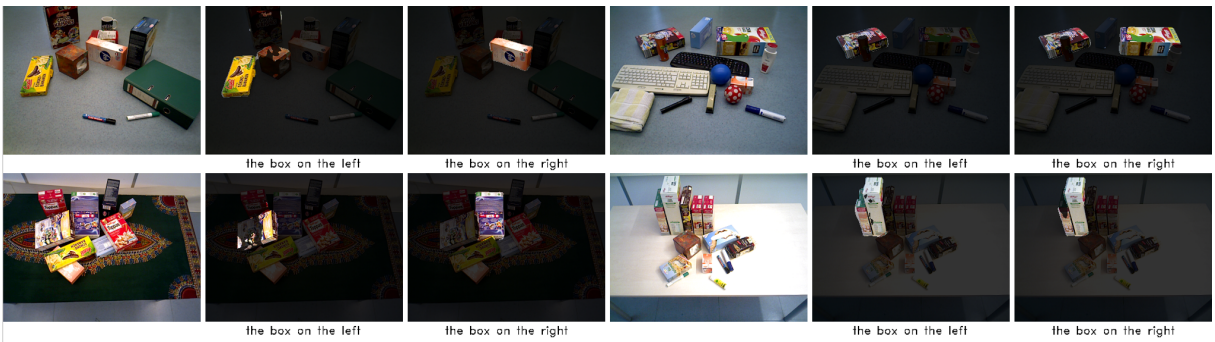

Figure 7: CLIP-Seg is able to generalize to non-relational spatial descriptions like on the left or on the bottom right. The visualization shows some cases of success and failure. In scenes where multiple boxes are present, users can specify absolute locations of objects and CLIP-Seg can parse it accordingly. However, we expect CLIP-Seg to have poor generalization to relational descriptions since we didn't train on relational descriptions of locations. The prompts used in above figures are inputted by humans who have no prior knowledge about prompts in training set of OCID.

Figure 8: Robot Experiments.

positional descriptions. However, we can still see some errors present in the results. We don't expect CLIP-Seg to generalize to relational position descriptions because we removed such querys at training time for our robotics appilication.

Finally, in figure 6, the human tester tries to refer to the same object instance with multiple possible class names. For example, in the first row, there is a white bottle with letters "soft scrub" on it. However, the human tester cannot see those letters clearly and described it as "shampoo" or "detergent" or simply "white bottle". We found that CLIP-Seg is still able to find the correct segmentation mask even though the names are different from the true class name. In the second row, the human describes a red bag of something as ramen. Though we don't really know what it is, CLIP-Seg can still find it using prompts related to ramen.

4.3 Ablation Study

Since we adopted a lot of new components for our downstream robotics application, we hope to understand the influence of these changes compared to traditional settings in referral expression segmentation. There are two components that we hope to study. First is the character of depth information and spatial map that we feed into the second branch of our network. To investigate the importance of the depth input and the training overfitting issue, we run training experiments on the both TOD and OCID dataset with setup identical to main experiments, except we mask out the xyz spatial map. For the OCID dataset, we observed that surprisingly, training without depth improves the segmentation performance, as demonstrated in Table 1. Connecting this with our ablation study, we suspect that there are some levels of image-level overfitting in the training. Because of this, the network memorizes the data and less input information leads to less overfitting.

However, we find that it is not the case in the TOD dataset, perhaps due to the fact that abundance of data in TOD eliminates the problem of overfitting. We further conduct ablation on single-modal predictions. Instead of training our network to output segmentation mask for a all object that satisfy descriptions at a time, we train it to predict any of the objects that satisfy the description, one at a time during training. This is also useful in the robotics setting to avoid confusion for later on planning netowrks. As shown in figure 2, the

Performance/Method	OCID-2D	OCID-3D	OCID-Fusion	CLIP-Seg	CLIP-Seg (No Depth)
Train IOU	0.512	0.588	0.634	0.661	**0.728**
Test IOU	0.501	0.580	0.637	0.677	**0.720**

Table 1: Segmentation performance on the train and test split in OCID-ref (Wang et al., 2021a) dataset. We show that our simple model is able to outperform several baselines proposed in the original paper.

Performance/Method	No Depth Stream	Single Modal	CLIP-Seg
Train IOU	0.727	0.74	**0.8**
Test IOU	0.654	0.68	**0.76**

Table 2: Ablation study on segmentation performance on train and test set in the TOD dataset.

single modal column indicates the train and test IOU in the single Modal setting. The IOU is lowered but CLIP-Seg is still able to achive reasonable performance.

4.4 Robot Experiments

We use the CLIP-Seg trained on the OCID dataset in this robot table-top grasping experiment. With a segmented mask, we can apply a pretrained policy to grasp a target object. The segmentation not only specifies the target object but also allows us to treat other objects as obstacles. We use (Wang et al., 2021b) as the closed-loop policy to fetch an interested object while avoiding collisions from other obstacles. In this experiment, we only use the inference at the start of the episode and execute the policy by transforming the point cloud scenes. The policy is able to grasp four different objects given language prompts.

5 Conclusion

Overall, we introduce a method to distill large language-vision model for open-text segmentation in robotic settings. Our preliminary results on both simulation and the real world suggest that we can leverage large pretrained language-vision model to provide accurate masks given user prompt. We demonstrate a robot grasping experiments with our segmentation method, and this opens many interesting directions to apply the learned segmentation method for robotic applications. However, the method is prone to overfit to images and it still has limitations on the expression complexity. Future directions include incorporating better inductive bias and generating large dataset for segmentation.

References

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. 2021. Vision-language transformer and query generation for referring segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*.

Mehmet Dogar and Siddhartha Srinivasa. 2011. A framework for push-grasping in clutter. *Robotics: Science and systems VII*, 1.

Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. 2019. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6569–6578.

Guang Feng, Zhiwei Hu, Lihe Zhang, and Huchuan Lu. 2021. Encoder fusion network with co-attention embedding for referring image segmentation.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. 2021. Zero-shot detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2018. Mask r-cnn.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer.

Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. 2016. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4555–4564.

Ya Jing, Tao Kong, Wei Wang, Liang Wang, Lei Li, and Tieniu Tan. 2021. Locate then segment: A strong pipeline for referring image segmentation.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. 2021. Mdetr – modulated detection for end-to-end multi-modal understanding.

Naoki Kato, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2019. Zero-shot semantic segmentation via variational mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0.

İlker Kesen, Ozan Arkan Can, Erkut Erdem, Aykut Erdem, and Deniz Yuret. 2021. Modulating bottom-up and top-down visual processing via language-conditional filters.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft coco: Common objects in context.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. 2021. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. 2021. Denseclip: Language-guided dense prediction with context-aware prompting. *arXiv preprint arXiv:2112.01518*.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. 2021. Cliport: What and where pathways for robotic manipulation. *arXiv preprint arXiv:2109.12098*.

Markus Suchi, Timothy Patten, David Fischinger, and Markus Vincze. 2019. Easylabel: A semi-automatic pixel-wise object annotation tool for creating robotic RGB-D datasets. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, pages 6678–6684.

Ke-Jyun Wang, Yun-Hsuan Liu, Hung-Ting Su, Jen-Wei Wang, Yu-Siang Wang, Winston Hsu, and Wen-Chin Chen. 2021a. Ocid-ref: A 3d robotic dataset with embodied language for clutter scene grounding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5333–5338.

Lirui Wang, Yu Xiang, and Dieter Fox. 2020. Manipulation trajectory optimization with online grasp synthesis and selection. In *Robotics: Science and Systems (RSS)*.

Lirui Wang, Yu Xiang, Xiangyun Meng, and Dieter Fox. 2021b. Hierarchical policies for cluttered-scene grasping with latent plans. *arXiv preprint arXiv:2107.01518*.

Lirui Wang, Yu Xiang, Wei Yang, Arsalan Mousavian, and Dieter Fox. 2021c. Goal-auxiliary actor-critic for 6d robotic grasping with point clouds. In *The Conference on Robot Learning (CoRL)*.

Yu Xiang, Christopher Xie, Arsalan Mousavian, and Dieter Fox. 2020. Learning rgb-d feature embeddings for unseen object instance segmentation. In *Conference on Robot Learning (CoRL)*.

Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. 2021. Unseen object instance segmentation for robotic environments. *IEEE Transactions on Robotics (T-RO)*.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. 2016. Modeling context in referring expressions.

Kevin Zakka. 2021. Clip detection. `https://github.com/kevinzakka/clip_playground.git`.

Chong Zhou, Chen Change Loy, and Bo Dai. 2021. Denseclip: Extract free dense labels from clip. *arXiv preprint arXiv:2112.01071*.