Human-in-the-loop Robotic Grasping using BERT Scene Representation

Yaoxian Song[1,2], **Penglei Sun**[1], **Pengfei Fang**[3], **Linyi Yang**[2,4], **Yanghua Xiao**[1], **Yue Zhang**[2,4]*
[1]School of Computer Science, Fudan University
[2]School of Engineering, Westlake University
[3]School of Engineering, Australian National University
[4] Institute of Advanced Technology, Westlake Institute for Advanced Study
{songyaoxian, yanglinyi, zhangyue}@westlake.edu.cn,
{plsun20, shawyh}@fudan.edu.cn, u5765437@anu.edu.au

Abstract

Current NLP techniques have been greatly applied in different domains. In this paper, we propose a human-in-the-loop framework for robotic grasping in cluttered scenes, investigating a language interface to the grasping process, which allows the user to intervene by natural language commands. This framework is constructed on a state-of-the-art grasping baseline, where we substitute a scene-graph representation with a text representation of the scene using BERT. Experiments on both simulation and physical robot show that the proposed method outperforms conventional object-agnostic and scene-graph based methods in the literature. In addition, we find that with human intervention, performance can be significantly improved. Our dataset and code are available on our project website [1].

1 Introduction

Grasping (Mahler et al., 2019) is a fundamental task for robot systems. It is useful for warehousing, manufacturing, medicine, retail, and service robots. One setting in robotic grasping is to grasp object orderly without disturbing the remaining in cluttered scenes (Chen et al., 2021b; Mees and Burgard, 2020; Zhang et al., 2021a) (called **collision-free grasp**). To solve this problem, a typical method (Zhu et al., 2021) parses the input into a **scene graph** first (Figure. 1(b)), in order to infer the spatial relation between objects and select a collision-free object for grasping. In particular, as shown in Figure. 2(a), nodes in a scene graph represent objects and edges represent their spatial relationship. Such structural knowledge can effectively improve the grasping performance as compared to an end-to-end model without scene structure information (Chu et al., 2018) (Figure. 1(a)).

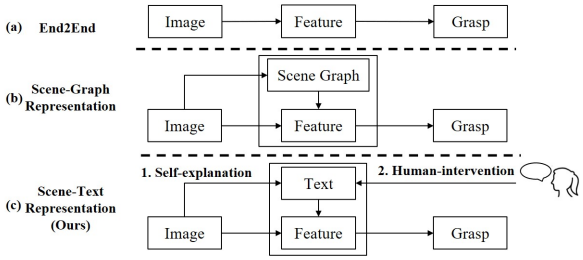


Figure 1: Various model structures for robotic grasping. (a) **End2End** (Chu et al., 2018) outputs an object-agnostic grasping by directly modeling on the input images. (b) **Scene-Graph Representation** fuses a generated scene graph with visual feature to predict grasping. (c) **Scene-Text Representation (ours)** makes use of language scene description and visual feature for achieving collision-free grasping.

As a structured representation of the scene, the scene graph has a few limitations. For example, it can be costly to manually label scene graphs, and the amount of existing labeled data is quite small. In practice, due to variance in the working environment, it can be necessary to calibrate a scene understanding model when deployed on physical robots (Zhu et al., 2021). This problem can be regarded as a domain adaptation task, which requires a certain amount of labeled scene graph data (Xu et al., 2017). In addition, graphs are relatively abstract and thus inconvenient for human-robot interaction.

We consider a natural language representation of the scene for substituting a scene graph structure representation. As shown in Figure. 2(a), small texts such as *"notebook placed under pliers"* and *"apple on notebook"* are used to indicate the recognized objects and their stacking relations. Compared with a scene graph structure, natural language scene representation has the following advantages. First, the cost of manual labeling is relatively lower thanks to the availability of speech recognition systems (Chiu and Raffel, 2018) and the relative independence from labeling GUI (Srivastava et al., 2021). Second, the state-of-the-art pre-trained representation models (Kenton and

* Corresponding Author
[1]https://sites.google.com/view/hitl-grasping-bert

Proceedings of the 29th International Conference on Computational Linguistics, pages 2992–3006
October 12–17, 2022.

Toutanova, 2019; Radford et al., 2019) can be used to improve scene understanding, which contains external knowledge beyond a scene graph structure. Third, online human interaction can be achieved by using human input of the natural language scene representation to replace incorrect robot scene understanding through speech communication[2].

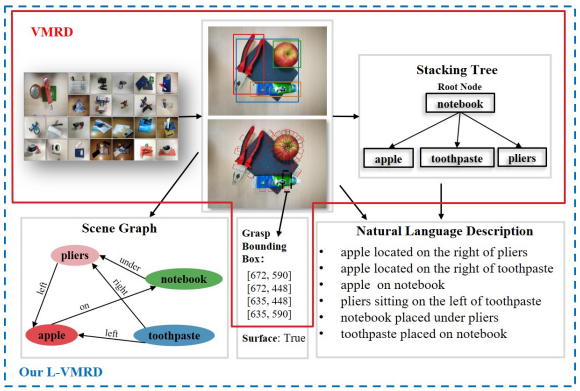
As shown in Figure. 1(c), we adopt the model of Zhu et al. (2021) by substituting the scene graph with a text description of the object to grasp and its spatial context, and using a neural image-to-text model for scene understanding and a pre-trained language model to represent the scene text for visual language grounding in subsequent grasping decisions. We compare the model performance with a dominant two-stage end-to-end planar grasping baseline (Chu et al., 2018) (Figure. 1(a)) and the baseline scene graph model (Figure. 1(b)). For all models, the grasping backbone is implemented using an extended version model of (Chu et al., 2018) with extra scene knowledge input.

For training and evaluation, we make extensions to the Visual Manipulation Relationship Dataset (VMRD) (Zhang et al., 2019b) by manually adding text descriptions and scene graphs to the scenes, resulting in a new dataset **L-VMRD**, as shown in Figure. 2(a). Experimental results show that (1) human language description can be a highly competitive alternative to the scene graph representation, giving better results for grasping; (2) online human language intervention is useful for improving the final grasping results, which is a new form of human-in-the-loop grasping. This indicates the promise of NLP models, especially pre-trained language models, for human-robot interaction. To our knowledge, we are the first to consider explicit textual scene representation and human intervention correction for robot grasping decisions, where BERT (Kenton and Toutanova, 2019) is firstly introduced into the internal structure of a robotic model as a state representation.

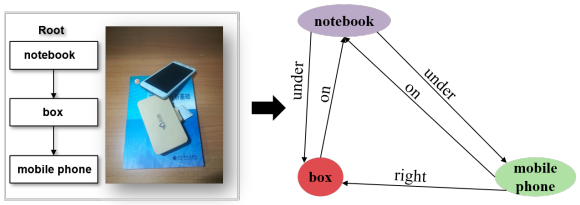
2 Task and Dataset

The input of the robotic grasping task is an image from robotic camera observation and the output is a grasping configuration (a grasping bounding box). As shown in Figure.1(c), we introduce a language description for the scene (image) during the inference process. We take a human-in-the-loop setting, where the language description can be obtained from a scene understanding model (**Self-explanation**) or human (**Human-intervention**).

[2]We adopt the **typed text** to simulate the process here since voice recognition is beyond our research scope



(a) Overview of proposed dataset L-VMRD



(b) Relationship tree & Scene graph.

Figure 2: **(a):** L-VMRD is built on VMRD. We extend **(i)** scene language description, **(ii)** scene graph and **(iii)** **surface** per grasp, including $112,965$ scene object relationship expressions and $21,713$ **surface** attributes paired with grasp bounding boxes. **(b):** relationship tree vs. scene graph.

Existing grasping datasets (e.g., VMRD Zhang et al. (2019b)) cannot be employed directly, because they do not include scene knowledge in human language. Hence, we develop an extended language version of VMRD, named L-VMRD, to evaluate our method. L-VMRD is an integrated dataset, and each sample is organized as a 6-tuple **(image, language descriptions, scene graph, object bounding box, grasping bounding box, surface)** shown in Figure. 2. L-VMRD contains $4,676$ samples, and is split into (train/validate/test) $= (3,740/468/468)$. The detail of dataset generation and usage in modeling are demonstrated in Appendix A.1.1 and A.1.2. Below we describe the main extensions from VMRD (Figure.2(a)).

Language Description in L-VMRD Object pairs in an image are sampled and labeled with a scene description. There exist many factors that affect collision-free grasping, in which stacking is the most significant one (Avigal et al., 2021). We first label the objects with the stacking relationship, e.g., *"apple on notebook"* or *"notebook sitting under pliers"* in Figure. 2(a), and then label

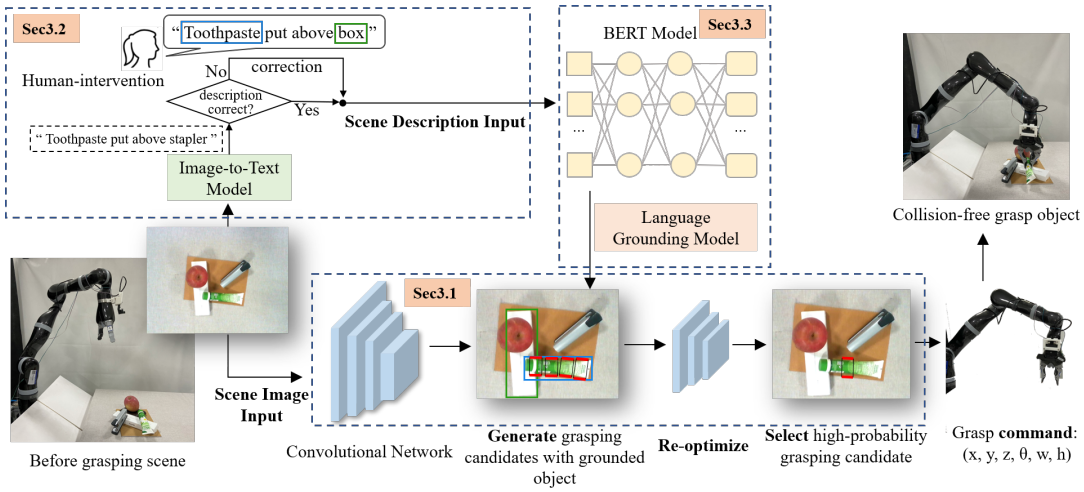


Figure 3: Overview of our proposed model. Firstly, a scene image is given to a human and an image-to-text model to generate a scene description. Secondly, scene description is fed into a BERT-based language grounding model to select a collision-free object. Thirdly, the grounded object is as the internal result fused into the language-based grasping model.

the horizontal relationship between non-stacked. Then considering the distribution of the horizontal relationships (*"left"*, *"right"*, *"front"*, *"back"*) in our dataset, we use ***"left"*** and ***"right"*** to indicate the scene relationships.

Scene Graph The original VMRD dataset includes partial relationships for adjacent objects, encoded by a relationship tree that only reveals the stacking relationship between two objects but is not able to indicate the relationship between objects stacked directly. To facilitate inference, we add a full scene graph to encode the pair-wise relationships of objects per image, where nodes and edges present objects and relationships, respectively. Detail is available in Appendix A.1.3.

Grasp-wise Spatial Attribute We introduce each grasp bounding box with a new attribute named **surface**. It is a binary variable indicating whether the grasped object sits on the top (True – on the top, False – stacked by other objects.). It is a grasp-wise label and can improve the robustness of the grasping model. An example is available in Appendix A.1.4.

3 Our Approach

The overall framework is shown in Figure. 3. The input scene image is fed into convolutional-based grasping model (Sec. 3.1) and scene understanding model (Sec. 3.2; Sec. 3.3), respectively. The scene understanding model is an image-to-text component that produces a sentence that describes the object to grasp and its context, such as *"toothpaste put above box"*. That scene description is then fused with intermediate grasping results to select object-related grasping candidates by pre-trained language model and language grounding model. **The final grasping output** is selected with high probability after re-optimization. A grasp command is sent to a real robot to complete a collision-free grasping operation. For the human-in-the-loop scenario, **an extra conditional input** from human-intervention will be given to correct scene description when the description from the image-to-text model is incorrect.

3.1 Overall of Language-based Grasping Model

Let I denote an image as perceptive information from the environment (i.e., cluttered scenes). Our robot f first identifies the grasp configuration from observation I. A typical 5-dimensional grasp configuration is given by:

$$g_i = f(I) = (x, y, \theta, w, h), \tag{1}$$

where (x, y) is the center position of the grasp rectangle, θ is the orientation angle with the x-axis, and (w, h) is the weight and height of the grasp rectangle. A general robotic grasping is presented by a probability $P(g_i|I)$, where $g_i \in G$ and G is a set of grasping candidates.

To achieve more stable and safe grasping, a joint probability $P(g_i, K_g|I)$ can integrate additional scene knowledge K_g as auxiliary information to guide vision-based grasping. It can be decomposed into conditionally independent two parts, given by:

$$P\left(g_i, K_g|I\right) = P\left(g_i|I, K_g\right) P\left(K_g|I\right), \tag{2}$$

where $P(K_g|I)$ is a scene understanding model. It can be a scene structure parsing model (Zhu et al., 2021; Figure. 1(b)), an image-to-text model (Figure. 1(c); Sec.3.2) with grounding model (Sec.3.3) or direct human intervention with grounding model (Sec.3.3). $P\left(g_i|I, K_g\right)$ is a convolutional network and the details are described in Appendix A.2.

3.2 Grasping Scene Understanding

A state-of-the-art image-to-text model (MMT) (Cornia et al., 2020) is used to generate the scene description in our work, which is a standard encoder-decoder Transformer-based model (Vaswani et al., 2017), that learns a multi-level representation of the relationships between image regions integrating with learned prior knowledge, and uses a mesh-like connectivity at decoding stage to exploit low- and high-level features. More details are in Appendix A.3.

Encoder A set of image regions I as Input is fed into encoding layer following Eq. (3).

$$\begin{aligned} Z &= AddNorm\left(M_{mem}\left(I\right)\right), \\ \tilde{X} &= AddNorm\left(F\left(Z\right)\right), \end{aligned} \tag{3}$$

where $AddNorm$ indicates the composition of a residual connection and layer normalization. M_{mem} is memory-augmented attention operation in Eq. (4). $F\left(Z\right)$ is a position-wise feed-forward layer composed of two affine transformations with a single non-linearity. $\tilde{X} = (\tilde{X}^1, ..., \tilde{X}^i..., \tilde{X}^N)$ is the set of all encoding layers and N is the number of layers.

$$\begin{aligned} M_{en}\left(I\right) &= Attention\left(W_q I, K, V\right), \\ K &= \left[W_k I, M_k\right], \\ V &= \left[W_v I, M_v\right], \end{aligned} \tag{4}$$

where $Attention$ is the self-attention operations used in (Vaswani et al., 2017). W_q, W_k, W_v are matrices of learnable weights. M_k and M_v are learnable prior information.

Decoder The decoder takes an input sequence of vector Y and output layers from encoder $\tilde{X}$, and then outputs sequence $\tilde{Y}$, in Eq. 5.

$$\begin{aligned} Z &= AddNorm\left(M_{de}\left(\tilde{X}\right), Y\right), \\ \tilde{Y} &= AddNorm\left(F\left(Z\right)\right), \end{aligned} \tag{5}$$

where M_{de} is defined in Eq. 6. Y is the input sequence of vector (groundtruth).

$$\begin{aligned} M_{de}\left(\tilde{X}, Y\right) &= \sum_{i=1}^{N} \alpha_i \odot C\left(\tilde{X}^i, Y\right), \\ C\left(\tilde{X}^i, Y\right) &= Attention\left(W_q Y, W_k \tilde{X}^i, W_v \tilde{X}^i\right), \\ \alpha_i &= \sigma\left(W_i\left[Y, C\left(\tilde{X}^i, Y\right)\right] + b_i\right), \end{aligned} \tag{6}$$

where C is cross-attention operation, σ the sigmoid activation function, and $\odot$ element product.

We make use of two adaptations during training. The first replaces the region features in MMT[3] (Cornia et al., 2020) with the concatenation of region features with bounding box features. Secondly, we add an extra score to multiply the CIDEr-D reward (Rennie et al., 2017) during training by maximizing a reinforcement learning based reward, since the description of subject object is usually the grasped one in our task. The score is computed by the correct rate of the subject over all generated sentences for each training batch.

3.3 Language Grounding for Grasping

We make use of visual language grounding models to map a scene description to a specified object. For visual language grounding, let Q represent a query sentence from human or image-to-text model and $I \in R^{H \times W \times 3}$ denote the image of width W and height H. The task aims to find the object region K_g represented by its center point (x_t, y_t) and the object size (w_t, h_t). The overall method can be formulated as a mapping function $(x_t, y_t, w_t, h_t) = \phi(Q, I)$.

In this paper, considering the real-time robotic control, we deploy our task on a one-stage language grounding model[4] (Yang et al., 2019) based on YOLOv3[5] (Redmon and Farhadi, 2018) with different language encoders for the mapping function ϕ. Formally, the scene image I and scene description Q are input to the visual encoder and text encoder, respectively, and the grounding module outputs the grounded object with encoders' features following Eq. 7.

$$\begin{aligned} Z_{vis} &= M_{vis}\left(I\right), \\ Z_{lang} &= M_{lang}\left(Q\right), \\ K_g &= Ground\left(\left[Z_{vis}, Z_{lang}, Z_{spatial}\right]\right), \end{aligned} \tag{7}$$

where M_{vis} is Darknet-53 (Redmon and Farhadi, 2018) pre-trained on COCO object dataset (Lin et al., 2014) and fine-tuned on our proposed L-VMRD. $Ground$ is the same as the output layers of YOLOv3. $Z_{spatial}$ is spatial feature of visual feature defined as follows: $(\frac{i}{W}, \frac{j}{H}, \frac{i+0.5}{W}, \frac{j+0.5}{H}, \frac{i+1}{W}, \frac{j+1}{H}, \frac{1}{W}, \frac{1}{H})$, which denotes top-left, center, bottom-right plane coordinates, sizes of the each pixel in visual feature

[3] https://github.com/aimagelab/meshed-memory-transformer

[4] https://github.com/zyang-ur/onestage_grounding

[5] YOLOv3 is a typical object detection model and derives many multimodal variants.

mapping Z_{vis}, respectively, normalized by the width W and height H of the feature mapping. $i \in \{0, 1, ..., W-1\}$ and $j \in \{0, 1, ..., H-1\}$.

For M_{lang} of the text encoder, we choose BERT (Kenton and Toutanova, 2019)[6] and the encoder of Transformer (Vaswani et al., 2017) (simply named Transformer). BERT is a pre-trained language model and builds on the Transformer network. Each description is fed into M_{lang}, resulting in 768 dimensions embeddings of all the tokens as natural language representations. Transformer[7] can be regarded as randomly initialized BERT without pre-training. Each description is embedded into $1,024$ dimensions embeddings. The model is randomly initialized.

3.4 Training Details

We break an end-to-end grasping training process into three submodules, **(i)** image-to-text (**self-explanation**), **(ii)** language grounding, and **(iii)** language-based grasping successively. The first two models are trained based on the original project configurations. The detail of the last one is described in Appendix A.4.

4 Evaluation

We construct both simulation and physical experiments to investigate four research questions:

Q1: How much natural language (**unstructured**) scene description perform better than scene graph (**structured**) knowledge in collision-free grasping task?

Q2: How much does the pre-trained model perform better than the randomly initialized model in our task?

Q3: How and where does human intervention using NLP improve grasping performance under the proposed human-in-the-loop framework?

Q4: How does our method perform on a physical robot? Does data collection from our proposed human-in-the-loop framework improve efficiency during the fine-tuning process?

4.1 Settings

Scene and Platform Implementation In simulation experiments, we use the test set split in Sec. 2 to evaluate our method. In the physical experiment, we collect objects and set up the placement as similar as possible to the training set, in which 2-6 objects are randomly placed (stacked) on a white table. But the reality gap is usually inevitable, especially in a robotic environment, which also tests the robustness of our method implicitly. Other details of the robot and training framework are described in Appendix A.5.

Evaluation Metrics We take Benchmark Performance and Success Rate as the main evaluation metrics. The first is used in both simulation and physical experiments, and the second one is only used in the physical robot experiment:

• *Benchmark Performance*: In simulation experiment, we evaluate model performances of the collision-free grasp using the object retrieval *top-k recall (R@k)* and *top-k precision (P@k)* metrics to evaluate multi-grasp detection (Hu et al., 2016). Chen et al. (2021b) proposes above metric to evaluate language-based multi-grasping. We do not compare it with our work directly, because: (i) their work (including dataset) is not open-sourced. (ii) it is just a command-based end-to-end grasping method that did not consider language scene understanding with human-in-the-loop. In physical robot experiment, ***accuracy*** is the percentage of correct cases over all test cases. The correct case is defined in Appendix A.6.

• *Success Rate*: In the physical robot experiment, we calculate the percentage of successful collision-free grasps over all grasping trials.

4.2 Models

We compare different pipeline methods visualized in Figure. 1. The baselines include:

End2End We re-train a state-of-the-art end-to-end object-agnostic planar grasp detection model MultiGrasp (Chu et al., 2018) on L-VMRD, shown in Figure. 1(a).

SceneGraph-Rep This is shown in Figure. 1(b) using a structured form of the scene graph generation[8] (IMP) (Xu et al., 2017) encoded with relational graph convolution network (RGCN) (Schlichtkrull et al., 2018) shown in Figure. 3. It replaces the subprocess from Image-to-Text Model to Language Grounding Model in Figure. 3 to select the grouned object. See details in Appendix A.7.1.

Our proposed models (**Scene-Text Representation** in Figure. 1 (c)) include:

SceneText-{BERT, Transformer} They are models using image-to-text MMT (Cornia et al., 2020) with language grounding (Yang et al., 2019) to re-

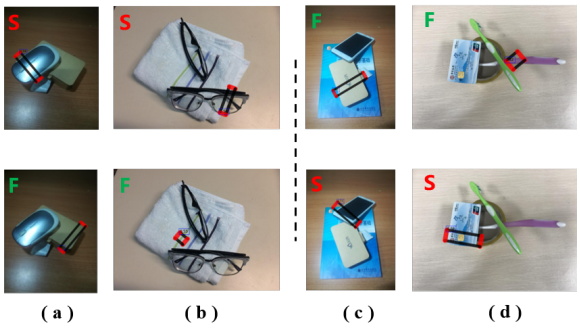
[6] We use bert-base-uncased model in our work.

[7] https://github.com/pytorch/examples/tree/master/word_language_model (**6 encoder_layers implemented**)

[8] https://github.com/jwyang/graph-rcnn.pytorch

Method	R@1	R@3	R@5	R@10	P@1	P@3	P@5	P@10
End2End	42.5	66.2	78.9	88.9	42.5	40.5	40.1	37.5
SceneGraph-Rep	72.2	85.6	90.3	92.2	73.8	70.7	69.0	64.0
SceneText-Transformer	72.2	85.6	88.3	90.3	72.4	68.8	66.2	61.6
SceneText-BERT	**73.7**	**88.9**	**91.8**	**93.1**	**73.9**	**71.9**	**69.2**	**65.3**
With Human-intervention								
SceneText-Interv-Oracle	77.0	89.1	90.8	91.4	78.0	76.4	75.0	71.7
SceneText-Interv-Transformer	75.9	88.8	91.8	92.4	76.3	71.2	69.2	64.9
SceneText-Interv-BERT	**76.9**	**90.3**	**93.0**	**94.9**	**76.3**	**75.3**	**73.6**	**69.1**

Table 1: Results of self-explanation and human-intervention models in the simulation experiment. The best performance is highlighted in bold.



(a) S SceneText-*: mouse placed on box F: End2End: -
(b) S SceneText-*: glasses placed on towel F: End2End : -
(c) F SceneText-*: box placed on mobile phone
S SceneText-Interv-*: mobile phone located on box
(d) F: SceneText-*: toothbrush sitting left of toothbrush
S SceneText-Interv-*: card placed on the left of toothbrush

Figure 4: Visualization of baseline models and our proposed models in our work. S means a successful case, while F means a failure case.

alize explainable grasping (**Self-explanation w/o Human-intervention**).

SceneText-Interv-{BERT, Transformer} They bring in human-intervention shown in Figure. 3. In **SceneText-Interv-Oracle**, the retrieval region of language grounding from the groundtruth is fed directly into the downstream grasp model (**Self-explanation w/ Human-intervention**).

4.3 Simulation Results

Scene Knowledge Table 1 gives the results of the end-to-end, scene-graph based method, and our method on L-VMRD data. In this setting, the input of the model is only the image and models predict a collision-free grasping. Although End2End (Chu et al., 2018) is the state-of-the-art method in object-agnostic grasping, it performs poorly on the cluttered scene grasping tasks. In contrast, by first obtaining a scene graph and then predicting the selected object, the SceneGraph-Rep model gives much improvement, with 72.2% over End2End R@1. This shows the necessity of scene understanding for the collision-free grasping task. As a case study, in Figure. 4(a)(b), SceneText-* generate self-explanation expression and obtain correct collision-free grasping, while End2End predicts incorrect grasping (incorrect object selection and low-quality grasp detection).

Scene Graph vs. Natural Language For **Q1**, compared with SceneGraph-Rep model which has R@1 of 72.2%, our method SceneText-BERT gives a better R@1 of 73.7%. This shows the feasibility of using natural language to replace the scene graph for object selection. Both methods are trained under the same settings, yet a natural language is more useful for achieving expandability in real-time human-robot interaction, thanks to its direct connection to the natural representation of the scene. For **Q2**, among our models, BERT can better parse the generated scene descriptions than Transformer model in both P@k and R@k, which shows the benefit of external pre-training. Note that Transformer alone does not outperform SceneGraph-Rep in Table 1. The results show that pre-training allows a textual representation of the scene to compete with a standard graph representation. More case studies can be found in Appendix A.7.2.

Human-intervention For **Q3**, the last two rows in Table 1 show that the models can take human language descriptions about the cluttered scene as a guidance or error correction for its own textual scene representation. By comparing results in Table 1, we can find that our proposed human-in-the-loop framework improves the performance from SceneText-* to SceneText-Interv-*. For example, compared with SceneText-BERT, SceneText-Interv-BERT improves the R@1 value from 73.7% to 76.9%, which shows that human intervention can be useful in practical scenarios. As a case study shown in Figure. 4(c)(d), the image-to-text model generates an incorrect relationship between *"box"* and *"mobile phone"*, leading to a failed collision-free grasping detection on the stacked box. In contrast, an extra human scene language description

Inter(%)	0	25	50	75	100	baseline (L-VMRD)
MMT	44.6	45.4	49.2	51.7	**55.0**	12.5
LangGr	91.3	91.7	90.8	92.1	**92.9**	79.6

Table 2: Results of image-to-text accuracy and language grounding accuracy @0.5 in the physical experiment. **Inter-** for short of intervention rate, e.g., **Inter-100**.

corrects the self-explanation description and helps the model to obtain a correct collision-free grasping detection. Within the above process, 21.1% scene language description from a robot is incorrect and intervened by human-correction. We also present the results of different human-intervention rates on F1 score, between SceneText-BERT and SceneText-Interv-BERT. As shown in Figure. 6(a), with the increase of human-intervention, the F1 score improves steadily. This shows that human language intervention can compensate for the flaws of scene understanding from the image-to-text models effectively.

4.4 Physical Robot Results

We only investigate the performance of BERT-based models (SceneText-BERT, SceneText-Interv-BERT) in the physical robot experiment. We recruit five graduate students to participate in our real-world experiment, who observe one whole process of a cluttered grasping with human-like grasping object selection each time by turns. When the image-to-text (self-explanation) model outputs an erroneous description, they correct the scene description by text typing decisively. Ultimately we collect 400 samples, including 160 correct output samples without human-intervention (self-explanation) and 240 human corrected (intervention) samples using the model trained on L-VMRD. Each sample contains an image, a language description sentence, and a grounded object bounding box. A pipeline case containing human-intervention is shown in Figure. 5.

Human-in-the-loop Learning Deployment Inspired by Lu et al. (2022), we take 160 samples as the training set to fine-tune our model based on different intervention rates (proportion of human-intervention samples, **Inter** for short). We randomly sample 80 samples from the remaining 240 human-intervention samples (getting rid of training used) as our test set. We repeat the process three times to generate three different test sets. The results in Table 2 show the mean values of models test on three test sets. For **Q4**, Table 2 shows the Benchmark Performance of models fine-tuned on different intervention rate training sets. The values of baseline are from the image-to-text model (**MMT**) and language grounding model (**LangGr**) trained on L-VMRD, respectively, which show the necessity of domain adaptation. For both MMT and LangGr, the models fine-tuned on human-intervention data (intervention rate 100%) achieve the best performance (55.0% and 92.9%). For MMT, the best model improves up to **42.5**% compared to baseline (12.5%) and **10.4**% compared to the model fine-tuned on self-explanation (44.6%, intervention rate 0%). For LangGr, the best model improves up to **13.3**% compared to baseline (79.6%) and **1.6**% compared to the model fine-tuned on self-explanation data (91.3%). This shows that the data collected from human-intervention can achieve better performance in domain adaptation from simulation to physical environment.

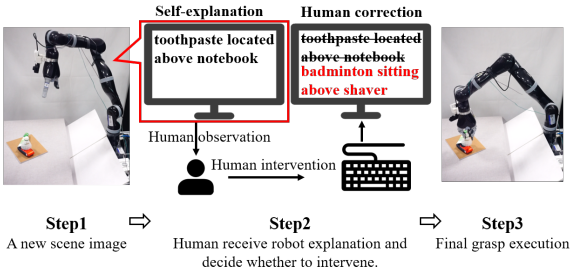


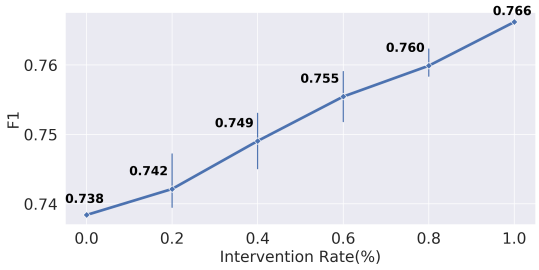
Figure 5: A pipeline of real robot execution with human.

Evaluation on Physical Robot For a final performance test, we conduct extra 80 grasping trials[9] for each model settings corresponding to Table 2. Grasping performance execution by a physical robot is shown in Figure. 6(b). End2End is the result from our baseline model trained on L-VMRD in Figure. 1(a). MMT and MMT+LangGr are models fine-tuning **image-to-text** or both **image-to-text and language grounding** respectively in SceneText-BERT setting. **+human** adds extra human intervention (SceneText-Interv-BERT) with Inter-100 fine-tuning model during grasping.

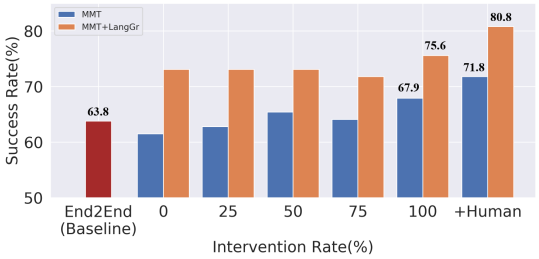
As shown in Figure. 6(b), our proposed method, which fine-tunes on human-intervention collection data, achieves 67.9% and 75.6% success rate compared to our baseline (63.8%). For **Q3** and **Q4**, our proposed methods achieve 71.8%(↑ 3.9%) and 80.8%(↑ 5.2%) success rate with human-intervention (**+human**) compared to without human-intervention (67.9%, 75.6%), in which

[9]We keep 80 object placements for each setting consistent.

46.3% scene language description from a robot is incorrect and intervened by human-intervention. This shows that human language intervention can improve the performance of grasping online on a real robot compared to the only image-to-text (self-explanation) method. Moreover, results from **MMT** and **MMT+LangGr** show that using human-intervention samples to train models can achieve better performance when there are very few to fine-tune the model. This indicates our proposed human-in-the-loop framework is applicable and performs well on the physical robot.



(a) F1 score on different intervention rates.



(b) Results of Grasping Success Rate in physical experiment.

Figure 6: Evaluation results.

5 Related Work

Natural Language and Robotics Natural language has been used with a variety of robot platforms, ranging from manipulators to mobile robots to aerial robots (Ahn et al., 2022; Raychaudhuri et al., 2021; Thomason et al., 2016; Chen et al., 2021a; Scalise et al., 2019). Most existing work is related to language understanding and language generation problems. For human-to-robot, language grounding is the mainstream means to learn the connection between percepts and actions in visual language navigation (Anderson et al., 2018; Ku et al., 2020) and robotic grasping tasks (Can et al., 2019; Zellers et al., 2021; Wang et al., 2021). For robot-to-human, multi-modal natural language generation (NLG) is widely adopted to lessen the communication barriers between humans and robots (Vinyals et al., 2015; Li et al., 2020; Cornia et al., 2020; Yuan et al., 2020; Shi et al., 2021; Zhang et al., 2021b), converting non-verbal data to the language that human can understand (Singh and JV, 2020). For bidirectional human-robot, Yuan et al. (2022) propose an explainable artificial intelligence system in which a group of robots predicts users' values by taking in situ feedback into consideration while communicating their decision processes to users through explanations. Our work is in line with the above work in exploiting information from natural language to facilitate decision-making. This is important because natural language is the most intuitive means of human-robot interaction. However, from the aspect of language, the main difference from existing work is that we take a step forward to not only considering language scene description as *input* for a robot but also as an *interface* of the model for online self-explanation simultaneously.

Grasping in Cluttered Scene Conventional methods (Chu et al., 2018; Mahler et al., 2019; Morrison et al., 2020; Kumra et al., 2020) focus more on object-agnostic grasp points detection (Figure. 1 (a)) missing parsing the object stacking scenario. For cluttered grasping, scene understanding and human instruction are usually considered. Zhu et al. (2021); Zhang et al. (2019a) adopt structured scene understanding (e.g., scene graph or relationship tree) to realize cluttered grasping detection. Mees and Burgard (2020); Chen et al. (2021b) fuse a natural language command and an observation image to detect a grasping in a two-stage and an end-to-end manner, respectively. Shridhar and Hsu (2020); Zhang et al. (2021a) receive natural command and image input, and then grasp a specified object. Existing work exploits human language to specify an object from the clutter, but does not allow human intervention for error correction. While existing method take language as *external* input, our scene language description is an *internal* component of the model. Moreover, we adopt the pre-train language model instead of RNN models in existing work.

6 Conclusion

We investigate language scene representation to robotic grasping, which enables a robot to explain its object selection to the user and allows the user to intervene with the selection by natural language. Experiments show that the proposed explainable textual scene representation outperforms

both object-agnostic and scene-graph based methods. By human language intervention, the performance can be broadly increased. Our results indicate the promise of using NLP models in a robotic system both as a representation and for human intervention. To our knowledge, we are the first to consider textual scene encoding and human correction in robotic grasping tasks, which can improve grasping performance using natural language (vs. w/o human-in-the-loop) and robustness (by pretrained language model).

7 Ethical Statement

Five graduate students who studied electronic engineering are hired to cooperate with a collaborative robot (Kinova) in our real-world experiment. Because of the subject background, they can be easy to decide whether give human language intervention based on human-like grasping behavior each turn. The participants need to annotate the object bounding box for each sample during the data collection stage. All participants have received labor free corresponding to their amount of trials.

Acknowledgements

We would like to thank anonymous reviewers for their insightful comments and suggestions to help improve the paper. This publication has emanated from research conducted with the financial support of the Pioneer and "Leading Goose" R&D Program of Zhejiang under Grant Number 2022SDXHDX0003, Shanghai Science and Technology Innovation Action Plan under Grant Number 19511120400, and Hangzhou City Agriculture and Social Development General Project under Grant Number 20201203B118. Yue Zhang is the corresponding author.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683.

Yahav Avigal, Vishal Satish, Zachary Tam, Huang Huang, Harry Zhang, Michael Danielczuk, Jeffrey Ichnowski, and Ken Goldberg. 2021. Avplug: Approach vector planning for unicontact grasping amid clutter. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 1140–1147. IEEE.

Ozan Arkan Can, Pedro Zuidberg Dos Martires, Andreas Persson, Julian Gaal, Amy Loutfi, Luc De Raedt, Deniz Yuret, and Alessandro Saffiotti. 2019. Learning from implicit information in natural language instructions for robotic manipulations. *NAACL HLT 2019*, page 29.

Feilong Chen, Xiuyi Chen, Can Xu, and Daxin Jiang. 2021a. Learning to ground visual objects for visual dialog. *arXiv preprint arXiv:2109.06013*.

Yiye Chen, Ruinian Xu, Yunzhi Lin, and Patricio A Vela. 2021b. A joint network for grasp detection conditioned on natural language commands. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4576–4582. IEEE.

Chung-Cheng Chiu and Colin Raffel. 2018. Monotonic chunkwise attention. In *International Conference on Learning Representations*.

Fu-Jen Chu, Ruinian Xu, and Patricio A Vela. 2018. Real-world multiobject, multigrasp detection. *IEEE Robotics and Automation Letters*, 3(4):3355–3362.

Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587.

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.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. 2020. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4392–4412.

Sulabh Kumra, Shirin Joshi, and Ferat Sahin. 2020. Antipodal robotic grasping using generative residual convolutional neural network. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9626–9633. IEEE.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer.

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.

Jinghui Lu, Linyi Yang, Brian Namee, and Yue Zhang. 2022. A rationale-centric framework for human-in-the-loop machine learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6986–6996.

Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg. 2019. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26).

Oier Mees and Wolfram Burgard. 2020. Composing pick-and-place tasks by grounding language. In *International Symposium on Experimental Robotics*, pages 491–501. Springer.

Douglas Morrison, Peter Corke, and Jürgen Leitner. 2020. Learning robust, real-time, reactive robotic grasping. *The International journal of robotics research*, 39(2-3):183–201.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel X Chang. 2021. Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. *arXiv preprint arXiv:2109.15207*.

Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2016. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024.

Rosario Scalise, Jesse Thomason, Yonatan Bisk, and Siddhartha Srinivasa. 2019. Improving robot success detection using static object data. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4229–4235. IEEE.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Zhan Shi, Hui Liu, Martin Renqiang Min, Christopher Malon, Li Erran Li, and Xiaodan Zhu. 2021. Retrieval, analogy, and composition: A framework for compositional generalization in image captioning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1990–2000.

Mohit Shridhar and David Hsu. 2020. Interactive visual grounding of referring expressions for human-robot interaction. *The International journal of robotics research*, 39(2-3):217–232.

Yuvaram Singh and Kameshwar Rao JV. 2020. Ai sensing for robotics using deep learning based visual and language modeling. In *Second Grand-Challenge and Workshop on Multimodal Language (Challenge-HML)*, pages 60–63.

Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, et al. 2021. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *5th Annual Conference on Robot Learning*.

Jesse Thomason, Jivko Sinapov, Maxwell Svetlik, Peter Stone, and Raymond J Mooney. 2016. Learning multi-modal grounded linguistic semantics by playing" i spy". In *IJCAI*, pages 3477–3483.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Ke-Jyun Wang, Yun-Hsuan Liu, Hung-Ting Su, Jen-Wei Wang, Yu-Siang Wang, Winston Hsu, and Wen-Chin Chen. 2021. 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.

Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. 2017. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419.

Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. 2019. A fast and accurate one-stage approach to visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4683–4693.

Chenxi Yuan, Yang Bai, and Chun Yuan. 2020. Bridge the gap: High-level semantic planning for image captioning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3157–3167.

Luyao Yuan, Xiaofeng Gao, Zilong Zheng, Mark Edmonds, Ying Nian Wu, Federico Rossano, Hongjing Lu, Yixin Zhu, and Song-Chun Zhu. 2022. In situ bidirectional human-robot value alignment. *Science Robotics*, 7(68):eabm4183.

Rowan Zellers, Ari Holtzman, Matthew Peters, Roozbeh Mottaghi, Aniruddha Kembhavi, Ali Farhadi, and Yejin Choi. 2021. Piglet: Language grounding through neuro-symbolic interaction in a 3d world. *arXiv preprint arXiv:2106.00188*.

Hanbo Zhang, Xuguang Lan, Site Bai, Lipeng Wan, Chenjie Yang, and Nanning Zheng. 2019a. A multi-task convolutional neural network for autonomous robotic grasping in object stacking scenes. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6435–6442. IEEE.

Hanbo Zhang, Xuguang Lan, Site Bai, Xinwen Zhou, Zhiqiang Tian, and Nanning Zheng. 2019b. Roi-based robotic grasp detection for object overlapping scenes. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4768–4775. IEEE.

Hanbo Zhang, Yunfan Lu, Cunjun Yu, David Hsu, Xuguang La, and Nanning Zheng. 2021a. Invigorate: Interactive visual grounding and grasping in clutter. In *Robotics: Science and Systems (RSS)*.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021b. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588.

Yifeng Zhu, Jonathan Tremblay, Stan Birchfield, and Yuke Zhu. 2021. Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs. In *2021 International Conference on Robotics and Automation (ICRA)*, pages 6541–6548. IEEE.

A Appendix

A.1 Details of Dataset

A.1.1 Dataset Generation

VMRD contains $4,683$ samples for 31 classes originally. Each sample has an **image** labeled with **object bounding boxes**, **object class**[10], and **grasp bounding boxes**. Also, stacking relationships between objects in images are provided in a **relationship tree**. Based on this, we further label various **language descriptions** and a **scene graph** for each sample in VMRD[11] based on the spatial information from object bounding box and relationship tree. An auxiliary grasp-wise spatial attribute **surface** is also introduced for better cluttered grasping performance.

A.1.2 Dataset Usage

For the image-to-text model, (image and language descriptions) are used. For the language grounding model, (image, language descriptions, and object bounding box) are used. For scene graph generation and graph-based object selection model, (image, scene graph, object bounding box, grasping bounding box) are used. For grasping model (vanilla), (image, object bounding box, grasping bounding box, surface) are used. All of model are setup in Sec. 4.2.

A.1.3 Scene Graph Example

For example, in Figure.2(b), the relationship tree only shows relationships as *"mobile phone-on-box"* and *"box-on-notebook"*, but cannot encode the relationship between *"mobile phone"* and *"notebook"* (e.g., *"mobile phone-on-notebook"*).

A.1.4 Surface Example

In Figure. 2(a), the *"notebook"* is stacked by an *"apple"*, thus the **surface** corresponding to *"apple"* grasping groundtruth is *"False"*. As for the toothpaste, it is not under any other objects, thereby labeled *"True"* for **surface**. In our task, this attribute can improve grasping performance.

[10]The object class is used to generate language description of the scene and construct the scene graph in our proposed dataset.

[11]Note that we filter seven samples with incorrect labeling in the original VMRD.

A.2 Details of Language-based Grasping Model

A.2.1 Backbone Model

Our backbone model is developed on top of the two-stage grasp detection pipeline (Chu et al., 2018) (a grasp-version Faster RCNN (Ren et al., 2016)), fusing the language knowledge as guidance. In doing so, we propose a Knowledge-guided Grasp Proposal Network (K-GPN) to replace with Region Proposal Network (RPN) in Figure. 7, for fusing the grounded object feature with the visual feature. In our framework, we formulate the grasp detection as three parts: (1) Grasp Proposals, (2) Grasp Orientation Classification and Multi-grasp Detection, and (3) Grasp Stacking Classification, described below.

Finally, the highest-confidence angles are selected for each grasp bounding box, and the grasp bounding box (predicted from proposals) corresponding to the highest confidence (mean of the bounding box confidence and surface confidence) is selected as g_i in Eq. (1) with the selected angle.

A.2.2 Grasp Proposals

The module aims to fuse grounded object feature and visual feature to the grasped object. The visual feature used here is a feature map ($z \in R^{50\times50\times1024}$) of the intermediate layers of ResNet-101, and the grounded object feature ($k \in R^{1\times4}$, also named K_g) is obtained from language grounding model (in Sec.3.3) based on **Self-explanation** or **Human-intervention**. The proposed K-GPN is employed to fuse z and k and output a new feature vector $1 \times 1 \times 512$, which is further fed into a two-layer Multi-Layer Perceptron (MLP) to predict the probability of grasp proposal and region of interest (ROI). Different from RPN used by (Chu et al., 2018), which takes positive and negative proposals with groundtruth over the whole image, the proposed K-GPN samples proposals based on language knowledge and produces ROI related to the **selected object** in Algorithm 1. The ROI features from **K-GPN** is fed into the following module to re-optimize and predict the final grasp, shown in Figure. 7. The details are described in Appendix A.2.3 and A.2.4.

As shown in Algorithm 1, our method takes in visual feature z and grounded object feature k. And then, the grasp proposals G generated from RPN is selected by satisfying iou constraints and $tiou$ constraints (language knowledge). g_{gt} is the

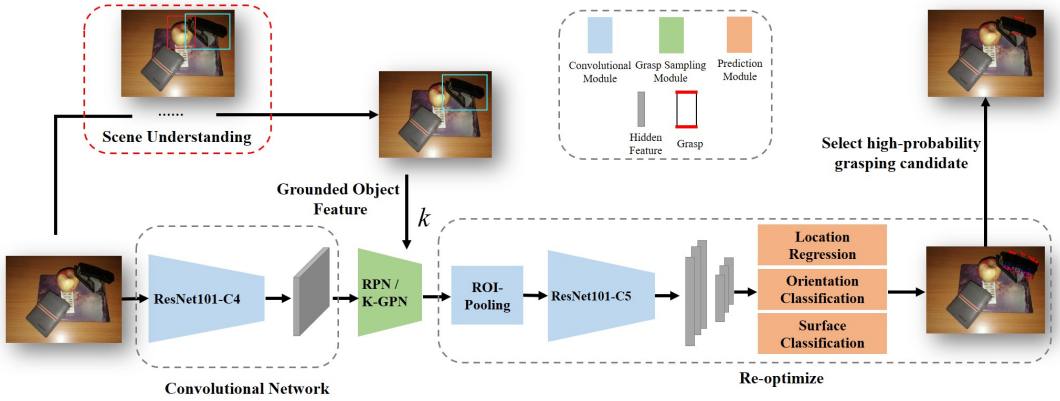


Figure 7: The architecture of Language-based Grasping Model. The input of the scene image is fed into **ResNet101-C4** to extract visual features, which is one of the inputs of our proposed **K-GPN**. The grounded object feature k is another input of **K-GPN**. **K-GPN** predicts grasp proposals and output proposal grasping region (also named ROI) features, which forwards into **ROI-Pooling** and **ResNet101-C5** to obtain 2048 dimensions feature vectors. These feature vectors are used to predict the final grasping location, orientation, and surface. The highest confidence grasping candidate is selected to execute by a real robot.

Algorithm 1 Knowledge-guided Grasp Proposal Network (**K-GPN**)

Input: visual feature z, grounded object feature k
Model: K-GPN
Output:Grasp Proposals G

Global proposal set G_g = RPN(z)
Positive proposal set and Negative proposal set:S_p, S_n
while $size(S_p)$ and $size(S_n)$ are less than the sampling count **do**
  Sampling a grasp proposal g from G_g
  if $iou(g, g_{gt}) > 0.5$ and $tiou(g, k) > 0.5$ **then**
    Put g into Positive proposal set S_p
  else
    Put g into Negative proposal set S_n
  end if
end while
return Grasp Proposals $G = \{S_p, S_n\}$

groundtruth grasp corresponding to the proposed g. $size(\cdot)$ is the number of the set. $iou(\cdot)$ is the conventional Intersection-over-Union (IoU) function (Ren et al., 2016). $tiou(\cdot)$ is a function defined in Eq. (8), used to select knowledge-guided ROI:

$$tiou\,(g,k) = \frac{|g \bigcap k|}{|g|}, \tag{8}$$

where g is the grasp proposal, and k is the grounded object feature (visual-language grounded object bounding box).

The proposal loss is defined in Eq. (9)

$$L_p\left(\left\{(p_c,t_c)_{c=1}^{C}\right\}\right) = \sum_c L_{cls}\,(p_c,p^*) + \lambda_1 \sum_c p^* L_{loc}\,(t_c,t^*), \tag{9}$$

where C is the set of all proposals, L_{cls} is the cross entropy loss of grasp proposal classification (binary classification). L_{loc} is the smooth L1 regression loss of the proposal locations. (p_c, t_c) is the binary class and proposal location to the i-th proposal. p_c is True if a grasp is specified, and False if not. (p^*, t^*) is the groundtruth. λ_1 is the weight coefficient.

A.2.3 Grasp Orientation Classification and Multi-grasp Detection

Our model quantizes the orientation θ into $R+1$ classification problem by discretizing the continuous orientation angles into R values. Another non-grasp case is also considered in the classification problem and in that case, the grasp proposal is considered incorrect. We select the highest score class as the orientation angle value. In practice, we use equal intervals of 10° to discretize the angles and $R + 1 = 19$. The loss function is as follows:

$$L_g\left(\left\{(\rho_l,\beta_l)_{c=0}^{C}\right\}\right) = \sum_c L_{cls}\,(\rho_l,\rho^*) + \lambda_2 \sum_c 1_{c\neq 0}(c) L_{loc}\,(\beta_c,\beta^*), \tag{10}$$

where ρ_l denotes the probability of class l and β_l corresponds to the grasp bounding box. L_{cls} is the cross entropy loss of the orientation angle classification (19-class classification). L_{loc} is the smooth L1 loss of bounding boxes with the weight coefficient λ_2 when angle class $c \neq 0$, where $c = 0$ is short for non-grasp case. (ρ^*, β^*) is the groundtruth.

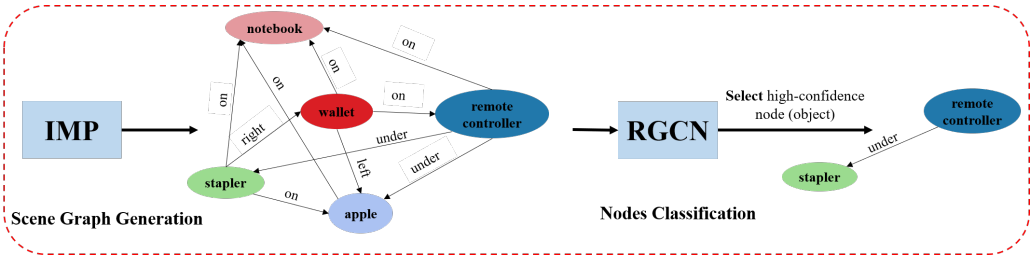


Figure 8: The architecture of scene-graph based scene understanding module.

A.2.4 Surface Classification

We propose a binary classification task to predict whether the grasped object is on the top of a stack of objects. The loss function is as follows:

$$L_s\left(\left\{(p_c)_{c=1}^{C}\right\}\right) = \sum_c L_{cls}\left(s_c, s^*\right), \qquad (11)$$

where the same as Eq. (9), L_{cls} is the cross entropy loss of grasp proposal **surface** classification (binary classification). s_c is False if the grasped object is stacked by others, and True if not. s_* is the groundtruth. Detail is available in Appendix A.1.4.

The total training loss for language-based grasping detection is:

$$L_{total} = L_p + L_g + L_s. \qquad (12)$$

A.3 Image-to-text Model

MMT is shown in Figure. 9. **Input** is the region features and bounding box detected from the robot observed image. **Output** is a description of the spatial relationships of objects in the scene. We hope the subject object can be grasped without collision based on the described spatial relationship.

A.4 Training Details of Language-based Grasping Model

For the baseline model, during ROI sampling in **K-GPN**, the positive and negative sampling counts for loss calculations are both 128. The optimizer is Adam and the learning rate is $1e-4$ for 100 epochs with batch size 8. λ_1 and λ_2 are both 1.0.

It is noted that for grounded object feature k, we first use the groundtruth of the language grounding model to train from scratch and fine-tune the model using the outputs of the language grounding model.

A.5 Hardware and Software Implementation

The grasping execution is taken place on a single-arm Kinova Jaco 7DOF robot under the framework of Robot Operating System (ROS) Kinetic, shown in Figure.3. We use an Intel RealSense SR300 RGB-D camera to obtain RGB-D images mounted on the wrist of the robot. All the computation is completed on a PC running Ubuntu16.04 and Pytorch 1.7 with one Intel Core i7-8700K CPU and one NVIDIA Geforce GTX 1080ti GPU.

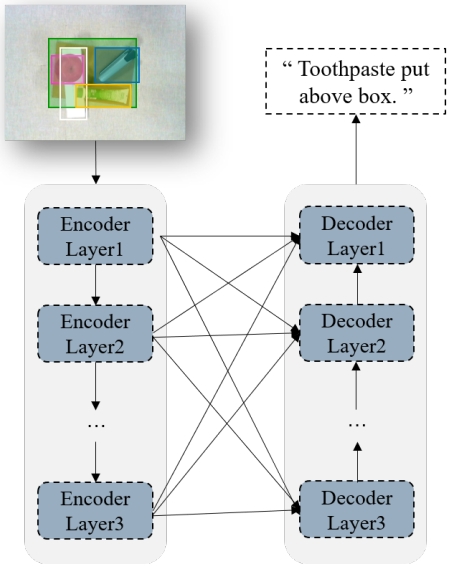


Figure 9: The overview of the image-to-text model applied in our self-explanation pipeline.

A.6 Metric Details

R@k is the percentage of cases where at least one of the top-k detection is correct. P@k is the correct rate for all top-k predictions.

In the simulation setting, a correctly detected grasp has a Jaccard Index greater than 0.25 and the absolute orientation error less than 30° relative to at least one of the groundtruth grasps of the collision-free object (Kumra et al., 2020).

In the physical setting, for the grounding model, the correct label is Intersection-over-Union (IoU) over 0.5, and for the image-to-text model, the correct label is human-annotated.

A.7 Scene Graph based method

A.7.1 Model Structure

The model **SceneGraph-Rep** uses a scene-graph based scene understanding module (shown in Figure. 8) to realize the function of red frame in Figure. 7.

The module is hierarchical including two submodules: **(i)** Iterative Message Passing (IMP)(Xu et al., 2017) to generate a scene graph in our work. **(ii)** Relational Graph Convolution Network (RGCN)(Schlichtkrull et al., 2018) to realize binary classification (can or cannot be grasped for each node). The IMP and RGCN are trained on our proposed L-VMRD same as our proposed method.

The **input** is region features from common object detection model (Faster RCNN (Ren et al., 2016)) using scene image I. IMP outputs the scene graph S_g in the form of triple (i.e., <subject, predicate, object>), which is fed into RGCN to predict the graspability of each node (object). The high-confidence object is selected corresponding with the bounding box. The whole process can be formulated as follows:

$$\begin{aligned} Z, B &= Detecton\,(I)\,, \\ S_g &= IMP\,(Z)\,, \\ (x_t, y_t, w_t, h_t) &= RGCN\,(S_g, B)\,, \end{aligned} \tag{13}$$

where $Detecton$ is a common object detection model. Z and B are the set of region features and the set of bounding boxes for each detected object, respectively. $\{x_t, y_t, w_t, h_t\}$ is the bounding box of selected object, same as definition in Sec.3.3.

A.7.2 Case Study

In Figure. 10, we give two failure cases caused by low-quality scene graph generation, indicating that **SceneGraph-Rep** highly depends on the output quality of the scene graph generation model. In Figure. 10(a), failure is caused by the incorrect scene graph generation. In Figure. 10(b), failure is mainly caused by the error classification from RGCN.

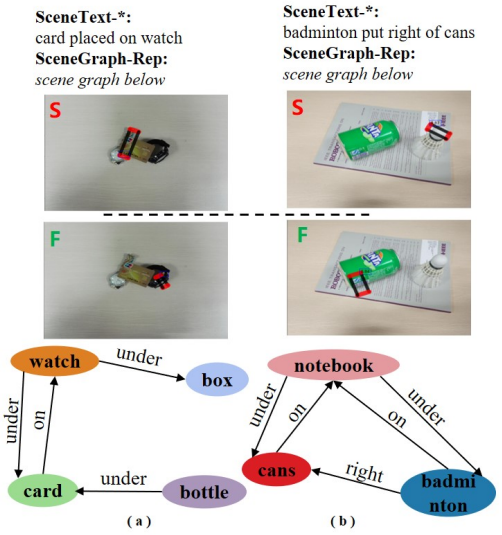


Figure 10: Visualization of MMT (image-to-text) and IMP (scene graph) based self-explanation models. **S** means successful case, while **F** means failure case.