Towards Open-World Interactive Disambiguation for Robotic Grasping

Yuchen Mo, Hanbo Zhang, Tao Kong
ByteDance AI Lab
{moyuchen,zhb,kongtao}@bytedance.com

Abstract: Language-based communications are essential in human-robot interaction, especially for the majority of non-expert users. In this paper, we present SeeAsk, an open-world interactive visual grounding system to grasp specified targets with ambiguous natural language instructions. The main contribution of SeeAsk is that it can robustly handle open-world scenes in terms of both open-set objects and open-vocabulary interactions. Specifically, our SeeAsk is built upon modern large-scale vision-language pre-trained models and traditional decision-making process, and shows promising results to be deployed in real-world scenarios. SeeAsk outperforms previous state-of-the-art algorithms with a clear margin in terms of not only success rate but also asking smarter and more informative questions. User studies also demonstrate its advantages over previous works.

Keywords: Human-Robot Interaction, Multi-Round Dialog

1 Introduction

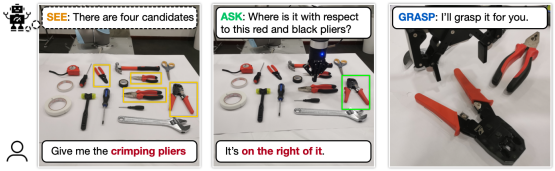
It has been a long dream for human beings to make intelligent robots enter our houses and live together with us. To get closer to non-expert users, robots must learn to understand the world visually, listen and speak in natural languages, and more importantly, connect visual concepts and languages properly for grounded communications. However, it remains super challenging for robots to do so because:

- **Visual uncertainty**: visual perceptions are not always reliable or even worse. How to make decisions with noisy visual inputs is important for robustness.
- **Language ambiguity**: natural languages are always ambiguous even for humans. How to resolve ambiguity is important for interaction efficiency.
- **Open-world concepts**: our daily life scenes usually contain open-world objects that are unknown to robots. Moreover, open-vocabulary instructions are also difficult for robots to understand. How to adapt to such open-world concepts is important for deployment.

Previous works [1, 2, 3, 4] have tried to resolve the first two challenges and achieved promising results. Nevertheless, the final challenge remains completely untouched. Fortunately, recent advances in deep multi-model learning [5, 6, 7, 8, 9] have made impressive achievements in bridging the gap among vision, language, and open worlds. Then a question arises naturally: *how can we push interactive visual grounding algorithms into open worlds by harnessing the large-scale vision-language pre-trained models?*

In this paper, we propose an agent SeeAsk, an interactive visual grounding system designed for open-world scenarios. In SeeAsk, we decompose the interactive visual grounding into three key components: Scene Understanding, Visual Grounding, and Decision Making. The raw observation image is first fed into the Scene Understanding module to output a structured visual representation, containing object categories, attributes, spatial locations, and binary relations. Then, the user instruction (e.g., *Give me the red cup in the table*) and the structured visual representation are used by

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.



(a) Interactive disambiguation process.

(b) Open-world objects used to test SeeAsk.

Figure 1: Our agent SeeAsk first converts the raw visual input to structured representations (See), and tries to ground user instructions and generate disambiguation questions (Ask). Finally, the robot will grasp the target when it is confident enough (Grasp). In SeeAsk, the user defines tasks in the open world using open-vocabulary languages with the robot. With the help of active interaction, the robot can finish tasks robustly even with severe ambiguity and perceptual uncertainty in zero-shot manners.

the Visual Grounding module for grounding user instructions and generating disambiguation questions. The Visual Grounding module takes the image and user's instruction as input, and outputs the initial probabilistic belief of each object being the target. The probabilistic belief is finally fed into the Decision Making module to determine whether it should directly grasp a target, or ask an informative question to resolve the ambiguity.

During the planning of the Decision Making module, SeeAsk maintains a belief over all objects, which will be initialized by the Visual Grounding module and updated continuously using observations from the user. Intuitively, this belief encodes all historical information for decision-making to maximize the cumulative rewards. In each time step, if the uncertainty level is low, our robot will grasp the target directly. Otherwise, the Decision Making module will be invoked to generate the most informative questions using an improved decision-making planner.

To the best of our knowledge, SeeAsk is the first open-world interactive visual grounding system that supports open-set objects and open-vocabulary interactions. It tightly and interpretably combines modern large-scale vision-language pre-trained models and traditional decision-making planners, and shows promising results to be deployed in real-world scenarios. Through experiments, we show that SeeAsk outperforms previous state-of-the-art algorithms with a clear margin in terms of not only robustness but also asking smarter and more informative questions. User studies also demonstrate that they do prefer SeeAsk over other baselines.

2 Related Works

Language grounding has been extensively studied [10]. Recently, researchers have been pushing grounding tasks to scenarios with unrestricted objects and language descriptions [11, 12, 13, 14]. Even though, single-round visual grounding cannot locate language descriptions with ambiguity. Therefore, interactive visual grounding is then introduced into grounding tasks [15, 16, 2, 3, 4, 17]. Particularly, INGRESS [1, 2] builds an interactive system for disambiguation in robot grasping tasks. It generates self-referential and relation descriptions with DenseCap [18], and does grounding and question generation with these descriptions. INVIGORATE [3] extends interactive visual grounding to clutter scenes by leveraging the integration of learning and planning. Attr-POMDP [4] explicitly uses object attributes to guide the planning of human-robot interaction for disambiguation. Nevertheless, these methods can fail in open-world domains. In this paper, we aim to bridge the gap between interactive disambiguation and open-world concepts by leveraging the modern large-scale pre-trained models and traditional planning framework.

Human-robot interaction (HRI) by asking questions is also related to our work. Traditional methods usually rely on templates [19, 20, 21] and plan with probabilistic models [22, 23, 24]. Recently, with the help of deep visual models, robots can generate unrestricted descriptions for objects subject to the training data. To interact with humans, especially for disambiguation, a straightforward question is to ask "Is it ...?" or "Do you mean ...?" for confirmation. To do so, some models [18, 25, 26, 27, 28, 2, 29] are trained on RefCOCO [30, 31], Visual Genome [32], or other vision-language datasets to generate object descriptions fitting the pre-defined question templates. There

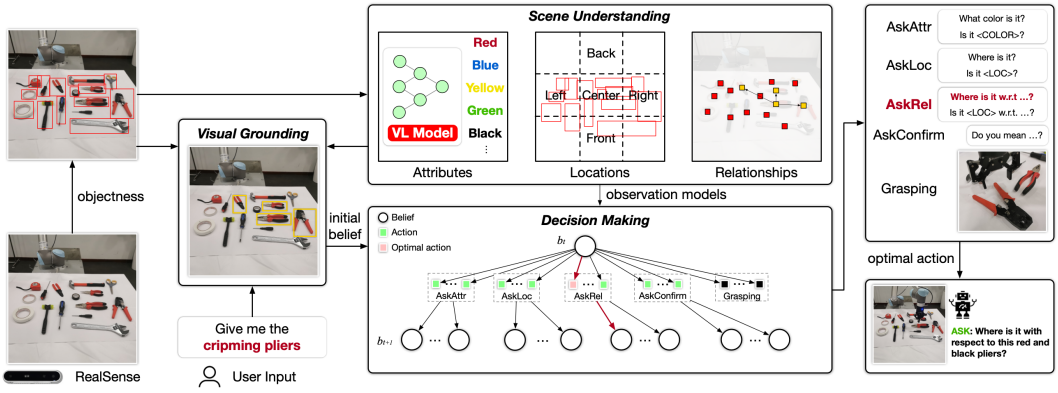


Figure 2: Overview of the interactive disambiguation workflow. First, the scene understanding module generates visual concepts to describe the scene and all objects, e.g., objects, attributes, and relationships. These visual concepts and user input will then be employed by visual grounding module to generate initial belief of each object being the desired target. Finally, the robot will ask various questions and have the next round of dialog with the user, until the robot becomes confident enough and then executes the grasp action.

are also works [33, 34, 13, 35, 36] exploring how to ask useful questions to locate objects. Besides, multi-modal interaction is also proved to be helpful when it is hard to generate descriptions [21], especially in open-world settings. Inspired by these works, we equip the robot with multi-modal interaction ability to disambiguate when meeting ambiguity and develop a user-friendly human-robot interaction system to follow verbal instructions.

3 Problem Formulation

SeeAsk is designed to grasp a specified object following *ambiguous* natural language instructions. Formally, the input includes a visual observation o_0^v and a linguistic observation o_0^l. o_0^v defines the workspace including a set of open-set objects $\{x_i\}_{i=1}^N$. o_0^l defines the initial instruction given by the user, which may be *ambiguous*, i.e., not informative enough to distinguish the ground-truth target x^* from other objects in $\{x_i\}_{i=1}^N$. To disambiguate, the robot is allowed to actively ask questions, and then gets additional linguistic observations, i.e., the answers from the user $\{o_t^l\}_{t=1}^T$. After collecting enough information, it will grasp the target with the highest belief. Following previous works [1, 2], we make the following assumptions in our task: a) The user is trustworthy and does not change the target during interaction; b) There is one and only one target object in the scene.

4 SeeAsk

The overview of SeeAsk is shown in Figure 2. SeeAsk takes an image o_0^v and a user instruction o_0^l as the initial observation. It includes three components, Scene Understanding, Visual Grounding, and Decision Making. The Scene Understanding module is used to parse all possible visual concepts in the input image, including object categories, attributes, spatial locations, and relationships. Based on the parsed results, the Visual Grounding module assigns a matching probability between each detected object with the user instruction. The matching probability is the initial belief over the underlying true state inferred from the initial observation o_0^v and o_0^l. If the uncertainty level after matching is low, the robot will grasp the matched target directly without asking any questions. Otherwise, it invokes the Decision Making module to generate the optimal question which is expected to bring more information. In SeeAsk, it is implemented as a tree planner to search forward for the optimal trajectory with the highest cumulative reward. In each asking round, the robot will get an additional linguistic observation $o_t^l \in \{o_t^l\}_{t=1}^T$. It will then use the observation to update its belief over the underlying true state of the environment, i.e., to determine which object is the desired target.

Table 1: Question types in SeeAsk

Visual Concept	Question Type	Answer List
Self Attribute	What color is it?	13 commonly seen colors
	Is it <COLOR>?	Yes/No
Spatial Location	Where is it?	Left/Right/Back/Front/Center
	Is it <LOC>?	Yes/No
Binary Relation	Where is it w.r.t. <OBJ>?	Left/Right/Back/Front/Itself
	Is it <LOC> w.r.t. <OBJ>?	Yes/No
-	Do you mean <OBJ>?	Yes/No

4.1 Scene Understanding

The Scene Understanding module mainly extracts four types of visual concepts: objects, attributes, spatial locations, and relationships. They are all used to ground user instructions as well as generate diversified questions in case of ambiguity.

Object Detection To detect open-set objects, we apply the most recently proposed open-vocabulary object detector, Detic [7]. It can locate and classify unseen objects using open vocabularies using the large-scale visual-language pre-trained model CLIP [5]. The detected objects $\{x_i\}_{i=1}^N$ will be used throughout all the following steps, including attribute and relationship extraction, visual grounding, and decision-making.

Attribute Extraction For object attributes, we run CLIP [5] for 0-shot classification using its image patch. Specifically, we pre-define a list of possible attributes, and then do prompt-based categorical or binary classification on each of them [5]. For example, "a photo of a bottle that is *(not) red*" for attribute "red", while "a photo of a *red/yellow/.../blue* bottle" for attribute "color" categorical classification.

Spatial Location Object location is important for referring. To extract the spatial location, we split the canvas into 3x3 grids first , and the object location is classified softly by calculating its IoU with each grid cell: $P(p_i|x_i) \propto \mathrm{IoU}(p_i, x_i)$, where p_i represents the spatial location of object x_i.

Relationship We support 4 basic binary relations "on the left", "on the right", "in front of", and "behind" in this part. Concretely, in a horizontal plane, let the shortest vector from a point on object x_j to x_i be $v(x_j, x_i) = (v_x, v_y)$. In case that $i = j$, i.e., $v(x_j, x_i) = (0, 0)$, we set "itself" as an additional relation. To support "itself", we attach an additional dimension to $v(x_j, x_i)$ and set it to 1 for convenience. Then for each relation $r_k \in \{\text{left, right, front, back, itself}\}$, we set a vector $e(r_k)$ as $(0, 1, 0), (0, -1, 0), (1, 0, 0), (-1, 0, 0), (0, 0, \epsilon)$, respectively, where $\epsilon \ll 1$ is positive. Therefore, we empirically have $\mathrm{Rel}(\mathrm{x_i}, \mathrm{x_j}, \mathrm{r_k}) = \max\{v(x_j, x_i) \cdot e(r_k), 0\}$, and $P(r_k|x_i, x_j) \propto \mathrm{Rel}(x_i, x_j, r_k)$ for each relation.

4.2 Visual Grounding

We base our Visual Grounding module on ReCLIP [9]. Given an expression of natural language, ReCLIP resolves the self-referential part via CLIP and binary-relation part via heuristics by parsing the grammar tree with SpaCy [37]. However, original ReCLIP cannot handle spatial locations since they are not captured either by CLIP or binary relations. Therefore, we extend ReCLIP so that the expression can then be parsed and resolved with three parts: self-referential part, spatial location part, and binary-relation part.

Formally, our Visual Grounding module is expected to output the probability of each object being the target $P(x_i = x^*|e)$ given an expression e (abbreviated as $P(x_i|e)$) in a one-shot manner. For any open-vocabulary expression e (e.g., "The blue cup on the left in front of the red cup."), we first decompose it using SpaCy into three parts: $e = e_{self} \cup e_{loc} \cup e_{rel}$, where e_{self}, e_{loc}, and e_{rel} represent self-referential part (e.g., "The blue cup"), spatial location part (e.g., "on the left"), and binary-relation part (e.g., "in front of the red cup") respectively. Then the posterior $P(x_i|e)$ can be

derived as:

$$P(x_i|e) \propto P(x_i)P(e|x_i) \propto P(x_i)P(e_{self}|x_i)P(e_{loc}|x_i)P(e_{rel}|x_i), \tag{1}$$

where $P(x_i|e_{self})$ is from CLIP:

$$P(e_{self}|x_i) \propto P(x_i|e_{self}) \propto \text{CLIP}(e_{self}, x_i) \tag{2}$$

given the prior $P(e_{self})$ is uniform since we do not assume that linguistic prior knowledge is available. $P(e_{loc}|x_i)$ can be derived as following:

$$P(e_{loc}|x_i) = \sum_{p_i} P(e_{loc}|p_i)P(p_i|x_i) \propto \sum_{p_i} P(p_i|e_{loc})P(p_i|x_i) \tag{3}$$

given the prior $P(e_{loc})$ is uniform, meaning that the object will uniformly distribute across all trials. In Equation 3, $P(p_i|e_{loc})$ is calculated using text similarity:

$$P(p_i|e_{loc}) \propto L(p_i, e_{loc}) \tag{4}$$

where $L(\cdot,\cdot)$ represents the text similarity score. In our case, we use SpaCy. Since a binary relation can be represented with a tuple $(x_i, r_{i,j}, x_j)$, and the binary-relation part e_{rel} contains information of both $r_{i,j}$ and x_j, we explicitly decompose $P(e_{rel}|x_i)$ as:

$$\begin{aligned} P(e_{rel}|x_i) &\propto P(x_i|e_{rel}) \\ &= \sum_{x_j} P(x_j|e_{rel}) \sum_{r_{i,j}} P(r_{i,j}|e_{rel})P(x_i|x_j, r_{i,j}) \\ &\propto \sum_{x_j} P(x_j|e_{rel}) \sum_{r_{i,j}} P(r_{i,j}|e_{rel})P(r_{i,j}|x_i, x_j) \end{aligned} \tag{5}$$

The last step assumes the prior distribution $P(r_{i,j})$ is uniform since again, we do not assume any prior knowledge on relational information between objects. In Equation 5, we adopt the relationships extracted in Section 4.1 for $P(r_{i,j}|x_i, x_j)$. For $P(x_j|e_{rel})$ and $P(r_{i,j}|e_{rel})$, we first parse e_{rel} into two separate phrases that contain the information of $r_{i,j}$ and e_j respectively. Then, $P(x_j|e_{rel})$ is derived using the same method introduced in Equation 2 and 3, while $P(r_{i,j}|e_{rel})$ is calculated in the same way as Equation 4 by the text similarity between $r_{i,j}$ and e_{rel}:

$$P(r_{i,j}|e_{rel}) \propto L(r_{i,j}, e_{rel}) \tag{6}$$

4.3 Decision Making

SeeAsk is built upon the extracted various visual concepts with a POMDP (Partially Observable Markov Decision Process) [38] defined by $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{Z}, \mathcal{R})$.

State Space $\mathcal{S}$ State s_t is defined as a one-hot vector over all objects $\{x_i\}_{i=1}^N$ with 1 indicating the target. Since we assume that the desired target does not change, for simplicity, we denote s_t as s. Note that the true state is not available, and must be estimated through observations. Hence, we maintain a *belief* b_t, a categorical distribution over state s_t.

Action Space $\mathcal{A}$ In brief, we allow the robot to ask questions when necessary and grasp the target when confirmed. For asking, each type of concept from Section 4.1 supports a set of questions. We list all possible questions and their corresponding visual concepts in Table 2.

Observation Space $\mathcal{O}$ Observation includes both visual o_0^v and linguistic $\{o_t^l\}_{t=1}^T$. In general, we allow free-form linguistic observations from the user to track the belief b_t.

Transition Model $\mathcal{T}$ The transition model is defined as: $\mathcal{T}(s_{t+1}|s_t, a_t) = P(s_{t+1}|s_t, a_t) = \mathbb{1}_{s_{t+1}=s_t}$ since we assume the user does not change their mind during interaction.

Observation Model $\mathcal{Z}$ Observation model $\mathcal{Z}$ captures the distribution of all possible observations (the possible answers in our case) given the state and action. Because our transition model is identical, the belief is fully updated using $\mathcal{Z}$. To be specific, as shown in Table 2, our observation models $\mathcal{Z}(o_t|s, a_t)$ for different questions are predefined from the probabilistic visual representations in Section 4.1. Notably, however, we assumed that users can provide open-vocabulary answers when facing a question. To achieve so, we assume that the answer o_t^l from the user can have either or both of two parts: *response* α_t (e.g., "Yes" or "No") and *description* d_t (e.g., "No, the other one to the right."). Without loss of generality, we assume that if both parts exist, the response always goes first. Hence, we first parse the answer into these two parts explicitly using SpaCy [37]. Then we play with them sequentially and update the belief accordingly:

$$b_{t+1}(s) \propto b_t(s)\mathcal{Z}(d_t|s)\mathcal{Z}(\alpha_t|s, a_t)$$

Table 2: Observation Models

Question Type	$\mathcal{Z}(o_t \mid s_t, a_t)$	
	o_t=Yes	o_t=No
Is it <COLOR>?	CLIP(<COLOR>, x^*)	CLIP(not <COLOR>, x^*)
Is it <LOC>?	IoU(<LOC>, x^*)	$1 - \text{IoU}(<\text{LOC}>, x^*)$
Is it <LOC> w.r.t. <OBJ>?	$\text{Rel}(x^*, <\text{OBJ}>, <\text{LOC}>)$	$1 - \text{Rel}(x^*, <\text{OBJ}>, <\text{LOC}>)$
Do you mean <OBJ>?	$\mathbf{1}(<\text{OBJ}> = x^*)$	$1 - \mathbf{1}(<\text{OBJ}> = x^*)$
What color is it?	CLIP(<COLOR>, x^*)	
Where is it?	IoU(<LOC>, x^*)	
Where is it w.r.t. <OBJ>?	$\text{Rel}(x^*, <\text{OBJ}>, <\text{LOC}>)$	

Figure 3: (Left) Experiment setting and (Right) scene settings.

For the description d_t, we regard it as an additional instruction and do an additional round of visual grounding with our Visual Grounding module in Section 4.2 to get $\mathcal{Z}(d_t|s)$. For α_t, we derive $\mathcal{Z}(\alpha_t|s, a_t)$ approximately from the observation model $\mathcal{Z}(o_t^l|s, a_t)$ defined in Table 2. Due to no assumption on linguistic priors, we assume that prior distributions of α_t and o_t^l are both uniform. Hence, we have:

$$\mathcal{Z}(\alpha_t|s, a_t) = \sum_{o_t^l} P(\alpha_t|o_t^l)\mathcal{Z}(o_t^l|s, a_t) = P(\alpha_t) \sum_{o_t^l} \frac{P(o_t^l|\alpha_t)\mathcal{Z}(o_t^l|s, a_t)}{P(o_t^l)} \propto \sum_{o_t^l} P(o_t^l|\alpha_t)\mathcal{Z}(o_t^l|s, a_t)$$

Where $P(o_t^l|\alpha_t)$ proportional to their text similarity.

Reward Intuitively, our reward function $\mathcal{R}(s, a_t)$ is empirically designed to encourage the robot to confirm and grasp the target x^* with the fewest questions. $\mathcal{R}(s, a_t)$ is -1 for asking, 10 for successfully grasping x^*, and -10 for wrong object $x_i \neq x^*$.

Planning We directly use the vanilla tree search as our planner [38]. Searching depth is set to 3, meaning that we expect the robot to ask fewer than 3 questions. If the robot decides to ask self-referential questions, it simply uses templates like "Is it *red*?" or "What color is it?". For relational (e.g., "Where is it w.r.t. ...?") or confirming questions (e.g., "Do you mean ...?"), the robot needs to fill in the template with the description of a particular object. We implemented a referring expression generator that auto-regressively adds attributives to the object class name according to their distinctiveness and accuracy. To generate natural descriptions, we set the maximum of attributives to 3.

5 Experiments

In this section we aim to answer 3 questions: (1) Does SeeAsk outperform previous interactive visual grounding methods on open-set objects? (2) Does SeeAsk generate better interactive questions? (3) What components in SeeAsk contribute to its disambiguation performance?

5.1 Experimental settings

Platform Our experimental platform is shown in Figure 3. We deploy SeeAsk as well as all baselines on a UR5 with a GeForce RTX 2070 GPU. Input images are from a RealSense camera with 1280×720 resolution. Interaction volunteers sit on the opposite of the robots.

Interaction Procedure We set up 10 different scenes, shown in Figure 3, with severe ambiguity to test the disambiguation performance for SeeAsk and all baseline algorithms. For each scene,

we recruit 10 volunteers for interaction. Before the interaction, each volunteer will be required to: (1) choose one target at the beginning, and give an instruction to the robot to grasp it; (2) give correct instructions and answers if necessary; (3) give ambiguous instructions if possible. For each scene, each volunteer will interact with the robot twice, one using self-referential instructions and another using relational instructions. We allow the robot to ask at most 5 questions. In total, we have conducted $10\times10\times2$=200 interaction experiments for each algorithm. For INGRESS and INGRESS-POMDP, we only collect 186 valid data points since it sometimes totally fails in open settings.

Metrics To evaluate the performance of different algorithms from comprehensive perspectives, we set up 5 different metrics: (1) **Success Rate (SR)** for the percentage of successfully grasping the correct target specified by the user; (2) **Number of Questions (#Questions)** to complete disambiguation; (3) **Cumulative Reward** for trade-off between asking and grasping; (4) **Information Gain** computed using entropy changes of the belief before and after asking each question to measure the quality of questions; (5) **User Experience** from the volunteers to evaluate how users feel about the interaction.

5.2 Real-robot Experiments

5.2.1 Baselines

We mainly compare our method to 6 baselines: **Greedy Asking** only asks confirming questions "Do you mean ...?" for the object with the highest belief. **ReCLIP** [9] directly invokes ReCLIP for visual grounding and grasps the object with the highest belief. It does not ask questions. **INGRESS** [1] uses DenseCap [18] and UMDRef [39] for generative visual grounding and question-asking. It only supports "Do you mean ...?" questions. **INGRESS-POMDP** [2] re-models INGRESS with a POMDP planner. **Open INGRESS** [1] is the re-implemented version of INGRESS using our open-world visual models. **Open Attr-POMDP** [4] is also a re-implemented version using open-world visual models. It explicitly parses attributes (e.g., color and location) of each object to facilitate question-asking and disambiguation. It is the state-of-the-art algorithm for interactive visual grounding.

5.2.2 Results

All results are shown in Table 3. We can see that SeeAsk has outperformed all baseline algorithms in the disambiguation tasks, i.e., it achieves the highest success rate 92% with only a few questions. By contrast, due to high ambiguity, ReCLIP achieves only 22.0% success rate. It is also noteworthy that both INGRESS and INGRESS-POMDP failed in most cases with a success rate of 15.6% and 13.4% respectively, even lower than ReCLIP, which is non-interactive. The reason is that DenseCap, as an object detector, fails to locate and classify objects in most cases of our open-world settings, and INGRESS relies heavily on what DenseCap has generated for visual grounding. When equipped with open-world visual models, INGRESS gets a much higher performance of 61.5% success rate. However, its performance is still 30.5% lower than SeeAsk due to its limited interaction modes (it only supports asking "Do you mean ...?" questions). Open Attr-POMDP also achieves commendable performance in our open-world settings. Nevertheless, it can hardly make use of object relationships. As a result, when known attributes are not applicable, especially for open-world settings, it degenerates into Greedy Asking because it can only iteratively ask "Do you mean...?" to confirm one by one. We also demonstrate some qualitative examples in Figure 4. Particularly, we show the significance of relationship understanding during interactive visual grounding. With the help of relationship understanding, we can see that the robot can smartly ask dichotomous questions and quickly converge to the target with only a few questions.

Table 3: Compared with previous methods.

Method	Success (%)	#Q
ReCLIP[9]	22.0	-
Greedy Asking	81.5	2.85
INGRESS[1]	15.6	2.10
INGRESS-POMDP[2]	13.4	**1.26**
Open INGRESS*[1]	61.5	1.76
Open Attr-POMDP*[4]	85.0	2.92
Ours	**92.0**	1.94

* means that we replaced all visual models with our open-world models.

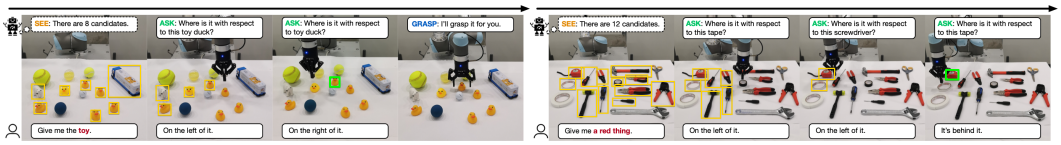


Figure 4: Qualitative examples. Objects with low confidence have been disabled.

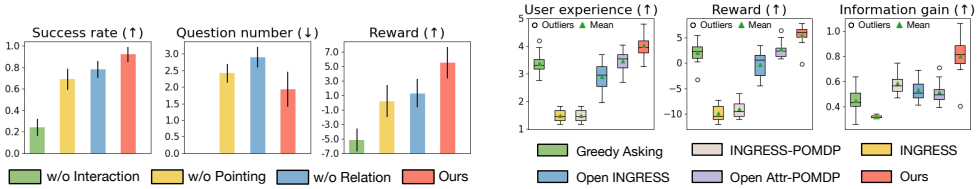


Figure 5: (Left) Ablation on interaction, pointing and relation. (Right) User study over methods.

5.3 Ablations

We also conduct a series of ablation studies to validate the contributions of different components in SeeAsk. **w/o Interaction** removes all asking actions in SeeAsk. Compared to the ReCLIP baseline, it is a modified version introduced in Section 4.1. It is used to validate the significance of interaction for disambiguation. **w/o Pointing** disables pointing actions during the interaction. It is used to validate the efficiency of multi-modal interaction. **w/o Relation** disables object relation parts both in visual grounding and question asking. It is used to validate the importance of object relationship understanding in human-robot interaction.

From the ablation study shown in Table 3, we can conclude that: (1) Interaction is necessary in open-world scenarios since it can actively ask for more information and fix perception errors during the dialogue. (2) Pointing is crucial to improve robustness, especially in complex scenes where it helps the user clearly know which one the object of interest is. (3) Object relationship modeling helps reduce the number of questions being asked and slightly improves the final performance, as it provides a robust and informative choice for disambiguation.

5.4 User Study

We also interviewed our experiment participants to collect their experiences during the interaction. Results, together with each question's reward and average information gain, are shown in Figure 5. We can see that users do prefer SeeAsk over other baselines. Interestingly, we find that the user experience is highly related to the cumulative reward we directly optimize during planning. This phenomenon indicates that to some extent, our algorithm is directly optimizing the user experience.

Besides, we also find that the number and average information gain of questions do not dominate the user experience. Here we list four potential reasons: (a) Users tend to rate highly those agents who ask meaningful and interpretable questions. (b) Users' attitude toward lucky success varies. (c) Pointing actions are less preferred than clear questions. (d) Spatial locations are sometimes confusing and hence less preferred than relational questions. Users tend to use relative location to the same kind of objects rather than the workspace.

6 Conclusions

In this work, we present an interactive disambiguation system, SeeAsk, for open-world robotic grasping tasks. Equipped with large-scale pre-trained vision-language models, SeeAsk can plan to ask various questions (self-referential, relational, or pointing-based) to perform effective interaction, or decide to do grasping then. We evaluate SeeAsk in real-robot settings. With open-world objects and ambiguous instructions, our system achieves 92.0% success rate, which outperforms other baselines with significantly better accuracy and user experience. There are several possible directions for future work: (a) explore further on the modularized integration of modern large-scale models and planning algorithms; (b) embodied interactive visual grounding in open domains; (c) optimize user experience rather than manually designed rewards.

References

[1] M. Shridhar and D. Hsu. Interactive visual grounding of referring expressions for human-robot interaction. *arXiv preprint arXiv:1806.03831*, 2018.

[2] M. Shridhar, D. Mittal, and D. Hsu. Ingress: Interactive visual grounding of referring expressions. *The International Journal of Robotics Research*, 39(2-3):217–232, 2020.

[3] H. Zhang, Y. Lu, C. Yu, D. Hsu, X. La, and N. Zheng. Invigorate: Interactive visual grounding and grasping in clutter. *arXiv preprint arXiv:2108.11092*, 2021.

[4] Y. Yang, X. Lou, and C. Choi. Interactive robotic grasping with attribute-guided disambiguation. *arXiv preprint arXiv:2203.08037*, 2022.

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

[6] A. Zareian, K. D. Rosa, D. H. Hu, and S.-F. Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14393–14402, 2021.

[7] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra. Detecting twenty-thousand classes using image-level supervision. *arXiv preprint arXiv:2201.02605*, 2022.

[8] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022.

[9] S. Subramanian, W. Merrill, T. Darrell, M. Gardner, S. Singh, and A. Rohrbach. Reclip: A strong zero-shot baseline for referring expression comprehension. *arXiv preprint arXiv:2204.05991*, 2022.

[10] Y. Qiao, C. Deng, and Q. Wu. Referring expression comprehension: A survey of methods and datasets. *IEEE Transactions on Multimedia*, 23:4426–4440, 2020.

[11] S. Guadarrama, E. Rodner, K. Saenko, N. Zhang, R. Farrell, J. Donahue, and T. Darrell. Open-vocabulary object retrieval. In *Robotics: science and systems*, volume 2, page 6, 2014.

[12] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018.

[13] J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

[14] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020.

[15] D. Whitney, E. Rosen, J. MacGlashan, L. L. Wong, and S. Tellex. Reducing errors in object-fetching interactions through social feedback. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1006–1013. IEEE, 2017.

[16] J. Hatori, Y. Kikuchi, S. Kobayashi, K. Takahashi, Y. Tsuboi, Y. Unno, W. Ko, and J. Tan. Interactively picking real-world objects with unconstrained spoken language instructions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3774–3781. IEEE, 2018.

[17] P. Pramanick, C. Sarkar, S. Paul, R. dev Roychoudhury, and B. Bhowmick. Doro: Disambiguation of referred object for embodied agents. *IEEE Robotics and Automation Letters*, 7 (4):10826–10833, 2022.

[18] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4565–4574, 2016.

[19] G.-J. M. Kruijff, H. Zender, P. Jensfelt, and H. I. Christensen. Clarification dialogues in human-augmented mapping. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 282–289, 2006.

[20] S. Rosenthal, J. Biswas, and M. M. Veloso. An effective personal mobile robot agent through symbiotic human-robot interaction. In *AAMAS*, volume 10, pages 915–922, 2010.

[21] A. Thomaz, G. Hoffman, and M. Cakmak. Computational human-robot interaction. *Foundations and Trends in Robotics*, 4(2-3):105–223, 2016.

[22] R. Deits, S. Tellex, P. Thaker, D. Simeonov, T. Kollar, and N. Roy. Clarifying commands with information-theoretic human-robot dialog. *Journal of Human-Robot Interaction*, 2(2):58–79, 2013.

[23] D. Simeonov, S. Tellex, T. Kollar, and N. Roy. Toward interpreting spatial language discourse with grounding graphs. In *RSS Workshop on Grounding Human-Robot Dialog for Spatial Tasks*, 2011.

[24] S. Tellex, R. Knepper, A. Li, D. Rus, and N. Roy. Asking for help using inverse semantics. 2014.

[25] L. Yang, K. Tang, J. Yang, and L.-J. Li. Dense captioning with joint inference and visual context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2193–2202, 2017.

[26] J. Liu, L. Wang, and M.-H. Yang. Referring expression generation and comprehension via attributes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4856–4864, 2017.

[27] M. Tanaka, T. Itamochi, K. Narioka, I. Sato, Y. Ushiku, and T. Harada. Generating easy-to-understand referring expressions for target identifications. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5794–5803, 2019.

[28] J. Kim, H. Ko, and J. Wu. Conan: A complementary neighboring-based attention network for referring expression generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1952–1962, 2020.

[29] O. Mees and W. Burgard. Composing pick-and-place tasks by grounding language. In *International Symposium on Experimental Robotics*, pages 491–501. Springer, 2020.

[30] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer, 2016.

[31] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[32] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[33] J. Zhang, Q. Wu, C. Shen, J. Zhang, J. Lu, and A. v. d. Hengel. Asking the difficult questions: Goal-oriented visual question generation via intermediate rewards. *arXiv preprint arXiv:1711.07614*, 2017.

[34] F. Strub, H. De Vries, J. Mary, B. Piot, A. Courville, and O. Pietquin. End-to-end optimization of goal-driven and visually grounded dialogue systems. *arXiv preprint arXiv:1703.05423*, 2017.

[35] P. Shukla, C. Elmadjian, R. Sharan, V. Kulkarni, M. Turk, and W. Y. Wang. What should i ask? using conversationally informative rewards for goal-oriented visual dialog. *arXiv preprint arXiv:1907.12021*, 2019.

[36] W. Pang and X. Wang. Visual dialogue state tracking for question generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11831–11838, 2020.

[37] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. spacy: Industrial-strength natural language processing in python. 2020.

[38] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[39] V. K. Nagaraja, V. I. Morariu, and L. S. Davis. Modeling context between objects for referring expression understanding. In *European Conference on Computer Vision*, pages 792–807. Springer, 2016.