ROBOGROUND: Robotic Manipulation with Grounded Vision-Language Priors

Supplementary Material

The supplementary materials are organized as follows:

1. Implementation details are provided in Appendix A.
2. Data details are presented in Appendix B.
3. A detailed explanation of the grounded perceiver is given in Appendix C.
4. Discussions on limitations and future work are provided in Appendix D.

A. Implementation Details

For the grounded VLM, we fine-tune GLaMM [32] starting from its publicly available checkpoint, pre-trained on the Grounding-anything Dataset, which contains 7.5M unique concepts spanning 810M regions. The fine-tuning process adopts an instruction-following approach to enable grounded conversational capabilities. To augment the model's training data, we incorporate 112K QA pairs generated from simulated data into its existing 277K grounded conversation dataset. The model is fine-tuned using LoRA with a rank of 8. The base learning rate is set to 3e-4 with a WarmupDecayLR scheduler, and the batch size is 20. The fine-tuning runs for 20 epochs, covering 10K training steps, and requires approximately 40 hours on 8 NVIDIA RTX 4090 GPUs.

For the grounded policy network, we re-implement the model architecture of GR-1 [43], omitting the image prediction head due to the unavailability of their video dataset for pre-training. Instead, we train the model from scratch using our simulation data. The training employs a base learning rate of 5e-4 with a cosine annealing schedule, a batch size of 32, and spans 5 epochs. Full training takes approximately 70 hours on 8 NVIDIA RTX 4090 GPUs. For faster experimentation during ablation studies, we use a subset of the data, reducing the training time to 5 hours and evaluation time to 1 hour.

Table 7. **Embedding Similarity of Instructions: Original vs. Generated Data.** A lower mean similarity and higher variance suggest greater diversity in the generated data compared to the original data.

	Mean	Variance
Original Data	0.961	0.00043
Generated Data	0.888	0.00387

B. Data Details

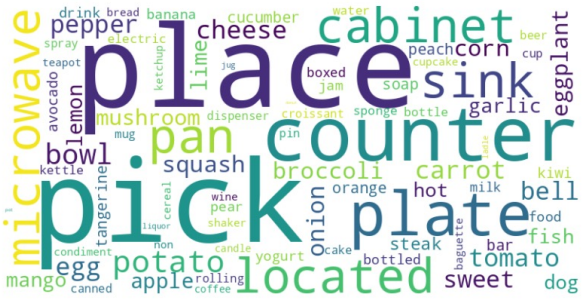
Instruction Diversity. To assess the instruction diversity of our generated data, we first visualize word clouds for pick-and-place tasks in Figure 4, which clearly demonstrate the higher diversity of our generated data. Additionally, we quantitatively evaluate the diversity by using BERT to obtain CLS embeddings for each instruction. We then compute the cosine similarities for all instruction pairs and calculate the mean and variance of the similarity matrix as measures of diversity. The results, presented in Table 7, show that the generated data exhibit lower mean similarity and higher variance, indicating greater diversity.

Prompts for Data Generation. We provide the prompts utilized for GPT-4 in our work. The prompt for filtering kitchen-related objects, shown in Figure 5, includes a list of valid kitchen-related object types and object attributes. GPT-4 is tasked with determining whether a given object is related to the kitchen based on this information. It is worth noting that the type list and object attributes were initially generated by GPT-4 in earlier stages of our process.

The prompt for generating key attributes of objects is illustrated in Figure 6. In this case, GPT-4 is instructed to describe the object's attributes using distinct descriptive words, referred to as key attributes. Similarly, the prompt for generating descriptive phrases is depicted in Figure 7, where GPT-4 is tasked with describing object attributes using detailed descriptive phrases.

The prompt for generating common-sense instructions is presented in Figure 8. In this scenario, GPT-4 is provided with multiple views of the scene, including the target object, the placement area, and other surrounding objects. It is then instructed to generate instructions that require common-sense reasoning.

Simulation Tasks. We adopt the setup of 22 atomic tasks defined in RoboCasa [28], categorizing them into four task types:*Pick and Place*, *Open/Close*, *Press*, and *Turn/Twist*, as summarized in Table 8. Beyond the original dataset of 3,000 generated samples for each task (referred to as "Easy" data), we introduce more complex scenes and instructions specifically for pick-and-place tasks. These enhancements aim to increase diversity by incorporating variations in appearance, spatial relationships, and common-sense reasoning.



(a) Word cloud of original data

(b) Word cloud of generated data

Figure 4. **Comparison of Word Clouds: Original Data (left) vs. Generated Data (right).**

Based on the given information:
1. List of kitchen-related object types:
[alcohol, apple, avocado, bagel, bagged_food, baguette, banana, bar, bar_soap, beer, bell_pepper, bottled_drink, bottled_water, bowl, boxed_drink, boxed_food, bread, broccoli, cake, can, candle, canned_food, carrot, cereal, cheese, chips, chocolate, coffee_cup, condiment, corn, croissant, cucumber, cup, cupcake, cutting_board, donut, egg, eggplant, fish, fork, garlic, hot_dog, jam, jug, ketchup, kettle, kiwi, knife, ladle, lemon, lime, mango, milk, mug, mushroom, onion, orange, pan, peach, pear, plate, potato, rolling_pin, scissors, shaker, soap_dispenser, spatula, sponge, spoon, spray, squash, steak, sweet_potato, tangerine, teapot, tomato, tray, waffle, water_bottle, wine, yogurt]
2. Attributes of the object:
- Name: Fantasy Windmill Tower
- Description: A whimsical, hand-painted wooden windmill tower designed for low-poly environments, featuring vibrant colors and a fantasy aesthetic.
- Material: Wood
- Shape: Tower
- Primary color: Multi-colored
- Size: 30
- Other tags: tower, wooden, windmill, handpainted, low-poly, fantasy

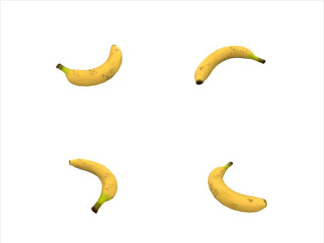
Task:
Determine whether the object should be placed in the kitchen. If it belongs to one of the specified kitchen-related object types, indicate the type. Otherwise, state it does not belong to any of the listed categories.

Answer format:
Yes, it should be placed in the kitchen. It belongs to the type [kitchen-related type].
No, it shouldn't be placed in the kitchen.
Yes, it should be placed in the kitchen. However, the object [object name] does not belong to any of the kitchen-related object types.

Figure 5. **Prompt for Filtering Kitchen-related Objects.**

C. Details of Grounded Perceiver

The grounded perceiver is composed of multiple attention layers. To illustrate its mechanism, consider a single attention layer where the input queries consist of 9 global query tokens $\mathbf{Q}_{\mathrm{g}} \in \mathbb{R}^{9\times D_{\mathrm{P}}}$, 9 target object query tokens $\mathbf{Q}_{\mathrm{o}} \in \mathbb{R}^{9\times D_{\mathrm{P}}}$, and 9 target placement query tokens $\mathbf{Q}_{\mathrm{p}} \in \mathbb{R}^{9\times D_{\mathrm{P}}}$. These queries are concatenated for parallel computation and projected to the hidden dimension of the attention layer, forming $\mathbf{Q} \in \mathbb{R}^{27\times d}$, where d represents the hidden dimension. The input 14×14 patch features $\mathbf{Z}_{\mathrm{v}}^{\mathrm{P}} \in \mathbb{R}^{196\times D_{\mathrm{v}}}$ are concatenated with the query features to construct the Key $\mathbf{K} \in \mathbb{R}^{223\times d}$ and Value $\mathbf{V} \in \mathbb{R}^{223\times d}$.



This image shows four different views of a single banana. Please describe its attributes following these rules:
1. The format should be: <COLOR> its color </COLOR> <SHAPE> its shape </SHAPE> <MATERIAL> its material </MATERIAL>.
2. For each attribute, only include the descriptive words. For example, if you describe the color as "yellow surface with brown spots", please simplify it to "yellow, brown".
3. Do not describe minor parts of the object. For example, if the picture shows a red apple with a small green stem, the color description should be "red" rather than "red, green".

Here are three example outputs for your reference:
1. <COLOR> red, yellow </COLOR> <SHAPE> round </SHAPE> <MATERIAL> organic </MATERIAL>.
2. <COLOR> silver </COLOR> <SHAPE> elongated, round </SHAPE> <MATERIAL> metal, plastic </MATERIAL>.
3. <COLOR> blue </COLOR> <SHAPE> cylindrical </SHAPE> <MATERIAL> aluminum </MATERIAL>.

Now please describe the object shown in the image.

Figure 6. **Prompt for Generating Key Attributes.**

Table 8. **Task and Data Split.** We adopt the setup of 22 atomic tasks defined in RoboCasa, categorizing them into four task types. For pick-and-place tasks, we create more complex scenes and instructions to increase the level of diversity. The quantity of training data for each task type is detailed in the table.

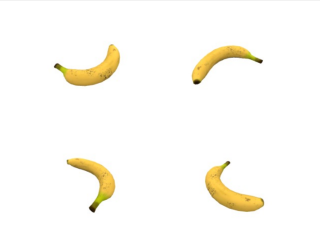
Task Type	Task Name	Train Data			
		Easy	Appea.	Spatial	Comm.
Pick and Place	PnPCounterToCab	3K	3K	5K	6K
	PnPCabToCounter	3K	3K	5K	6K
	PnPCounterToSink	3K	3K	5K	6K
	PnPSinkToCounter	3K	3K	5K	6K
	PnPCounterToMicrowave	3K	3K	5K	6K
	PnPMicrowaveToCounter	3K	3K	5K	6K
	PnPCounterToStove	3K	3K	5K	6K
	PnPStoveToCounter	3K	3K	5K	6K
Open / Close	OpenSingleDoor	3K	-	-	-
	CloseSingleDoor	3K	-	-	-
	OpenDoubleDoor	3K	-	-	-
	CloseDoubleDoor	3K	-	-	-
	OpenDrawer	3K	-	-	-
	CloseDrawer	3K	-	-	-
Press	CoffeePressButton	3K	-	-	-
	TurnOnMicrowave	3K	-	-	-
	TurnOffMicrowave	3K	-	-	-
Turn / Twist	TurnOnSinkFaucet	3K	-	-	-
	TurnOffSinkFaucet	3K	-	-	-
	TurnSinkSpout	3K	-	-	-
	TurnOnStove	3K	-	-	-
	TurnOffStove	3K	-	-	-

The attention matrix is computed as $\mathbf{A} = \frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{k}} \in \mathbb{R}^{27\times 223}$. To incorporate the masks for target objects and placement areas, the attention values corresponding to the masked regions are replaced with the highest attention value in the current matrix. Specifically, the target object masks M_o are applied to $\mathbf{A}_{[9:18,:196]}$, and the placement masks M_p are applied to $\mathbf{A}_{[18:,:196]}$. This ensures that the target object and placement query tokens focus more effectively on the relevant masked areas.

D. Limitation and Future Work

Although extensive experiments have demonstrated the effectiveness of our proposed method using grounding masks as a guide—particularly its strong generalization ability to unseen domains—there remain some limitations that warrant further exploration in future work.

For object picking, we observe a significant gap between the contact rate and the success rate, suggesting that the model struggles with reliably grasping target objects. This limitation stems primarily from the high diversity of the thousands of objects in our dataset, making it more challenging for the model to overfit compared to previous, less varied datasets. Furthermore, while grounding masks excel at providing localization guidance, they offer limited support for enhancing grasping precision. To address this, a



This image shows four different views of a single banana. Please describe its attributes following the format: <COLOR> its color </COLOR> <SHAPE> its shape </SHAPE> <MATERIAL> its material </MATERIAL>.

Here are four example outputs for your reference:
1. <COLOR> green on top and yellow on the bottom </COLOR> <SHAPE> rectangular prism </SHAPE> <MATERIAL> sponge </MATERIAL>.
2. <COLOR> white </COLOR> <SHAPE> cylindrical with a flip-top lid </SHAPE> <MATERIAL> plastic </MATERIAL>.
3. <COLOR> blue </COLOR> <SHAPE> flat with an elongated handle and slotted flipper </SHAPE> <MATERIAL> plastic or metal </MATERIAL>.
4. <COLOR> brown cap and white stem </COLOR> <SHAPE> typical mushroom shape with a rounded cap and cylindrical stem </SHAPE> <MATERIAL> organic fungus </MATERIAL>.

Now please describe the object shown in the image.

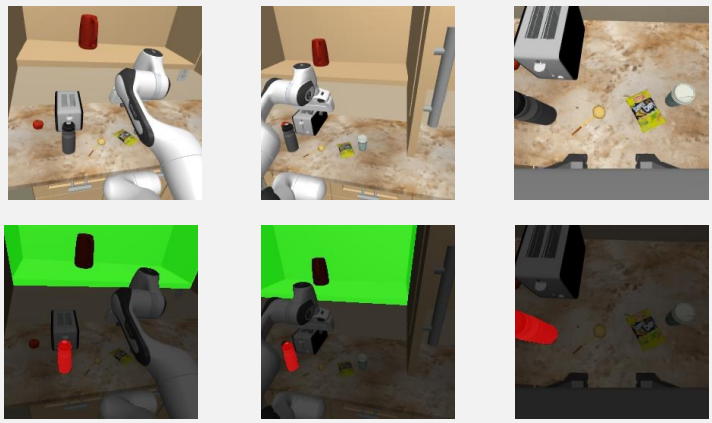
Figure 7. **Prompt for Generating Descriptive Phrases.**

promising approach could involve integrating a pre-trained grasp pose prediction network, such as AnyGrasp [13], which boasts strong generalization capabilities for novel objects. Incorporating such a network could enable the development of a more robust and generalizable policy network.

For data generation, our focus has primarily been on increasing the diversity of the target object, while largely overlooking the diversity of target placement areas. Enhancing this aspect can be achieved by generating a broader range of target placement options. To accomplish this, we need to collect additional human demonstrations for these newly generated scenes (trajectories to new placement areas) and leverage automated methods, such as MimicGen [26], to further augment the dataset.

For model architecture, we currently treat it as two distinct components: a grounded VLM for mask prediction and a policy network for action prediction. Future exploration of end-to-end architectures or slow-fast systems could be both promising and challenging, potentially enabling more robust policies and harder tasks such as long-horizon tasks.

We hope our findings inspire further research into intermediate representations that can guide low-level policies, and provide valuable insights for generating more diverse scenes and instructions in robot manipulation.



These images depict a kitchen scene where a Franka robotic arm interacts with various items. The robot is instructed to perform a pick-and-place manipulation task.
Three viewpoints are provided: left view, right view, and robot hand view. Each viewpoint includes two types of images: the raw image and one with masked targets.
The target object, a water bottle, is highlighted in red, while the target placement area, the cabinet, is marked in green. Other objects present in the scene include a water bottle, tomato, kettle, coffee cup, spoon, and chips.

Please generate five new instructions that require common-sense reasoning about the target object for me. Remember, you cannot change the target placement location. Do not mention the masked aspect.
ATTENTION: Only refer to objects that exist in the scene.

For example, the target object is kettle, the expected answer format is:
<ANSWER>
<1> I want to drink, please pick the related object to the cabinet. </1>
<2> I'm thirsty, please pick the related object to the cabinet. </2>
<3> I'm going for a hike and need to stay hydrated, please pick up the appropriate item and place it in the cabinet. </3>
<4> I need to refill something for gardening, could you put the refillable item into the cabinet? </4>
<5> I'm heading to the gym and need to fill up my container, place it in the cabinet.</5>
</ANSWER>

Figure 8. **Prompt for Generating Common-sense Instructions.**