C2F-Space: Coarse-to-Fine Space Grounding for Spatial Instructions using Vision-Language Models

Anonymous ACL submission

Abstract

Space grounding refers to localizing spatial references expressed through natural language instructions. Traditional methods often fail to account for complex reasoning—such as distance, geometry, and inter-object relationships—while vision-language models (VLMs), despite strong reasoning abilities, struggle to produce fine-grained outputs. To overcome these limitations, we propose C2F-Space, a novel coarse-to-fine space-grounding framework that performs coarse reasoning via propose-validate VLM prompting and refines predictions through superpixel-wise residual learning for precise local geometric reasoning. Our evaluations demonstrate that C2F-Space significantly outperforms three state-of-the-art baselines in both success rate and intersection-over-union on a new superpixel-level space-grounding benchmark.

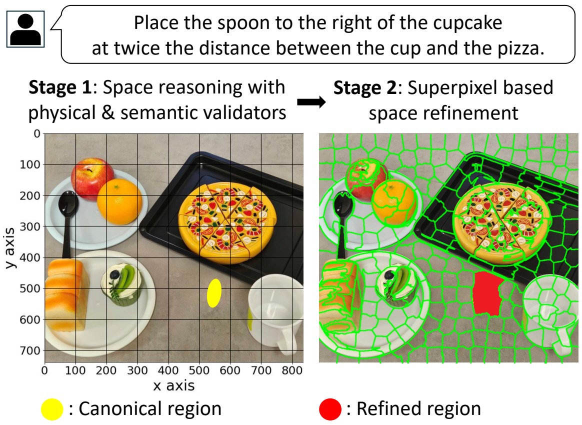

Place the spoon to the right of the cupcake
at twice the distance between the cup and the pizza.
Stage 1: Space reasoning with physical & semantic validators
Stage 2: Superpixel based space refinement
0
100
200
300
400
500
600
700
y axis
0 100 200 300 400 500 600 700 800
x axis
● : Canonical region
● : Refined region

Figure 1: Illustration of the two-stage space grounding result produced by the proposed C2F-Space. The grid-guided prompt enables the VLM to generate a coarse region proposal (e.g., an ellipsoid) through spatially multiplicative reasoning. A superpixel-based enhancement process then refines this proposal into a fine-grained spatial mask.

1 Introduction

Space grounding refers to the process of mapping linguistic expressions to spatial regions within an environment (Kim et al., 2024). The process often requires complex spatial reasoning that accounts for distance, geometry, and inter-object relationships, which have yet to be thoroughly investigated. Fig. 1 illustrates a representative example in a robotic pick-and-place scenario, where a human provides an instruction: "Place the spoon to the right of the cupcake at twice the distance between the cup and the pizza." The interpretation of this instruction requires not only estimating the distance, but also reasoning about proportional relationships to determine the target position among candidates located twice that distance to the right of the cupcake.

Early approaches link simple spatial expressions (e.g., 'near') to a limited category of segments (e.g., 'next to the stop') (Jain et al., 2023). Subsequent approaches support compositional expressions (Zhao et al., 2023; Gkanatsios et al., 2023). Recently, Kim et al. (2024) introduce a probabilistic update mechanism to resolve the ambiguity in compositional expressions. Despite these advances, their generalization remains limited due to the small scale of annotated datasets.

With advances in large language models (LLMs) and vision-language models (VLMs), researchers begin leveraging large pre-trained models for enhanced spatial reasoning. Notable examples include RoboPoint (Yuan et al., 2025), which fine-tunes a VLM to localize target regions as coarse point sets, and recent VLMs, such as Molmo (Deitke et al., 2025) and Gemini 2.5 (Comanici et al., 2025), demonstrate zero-shot 2D point grounding (Cheng et al., 2025). Nevertheless, the coarse, point-level nature of these outputs often lack the rigorous spatial precision required for downstream applications, particularly in fine-grained robotic manipulation. While visual prompting methods provide more granular guidance and complement traditional text prompts (Shtedritski et al., 2023; Cai et al., 2024), they mostly perform forward reasoning without validation.

Therefore, we propose C2F-SPACE, a novel coarse-to-fine space-grounding framework with spatio-semantic validation for complex instructions. Our method is a two-stage framework; the first stage enables a VLM to propose and validate regional candidates using a grid-inpainted image prompting, while the second stage refines the coarse outcome to precisely fit into the environment via super-pixelization.

We evaluate C2F-SPACE against baselines on a space-grounding benchmark comprising 350 problems, including instructions with (i) *single-hop space reasoning with unique references,* (ii) *single-hop space reasoning with non-unique references,* and (iii) *multi-hop space reasoning with unique/non-unique references.* Our results show that C2F-SPACE significantly improves the grounding performance of o4-mini (OpenAI, 2025), outperforming baselines such as CLIPORT (Shridhar et al., 2022), LINGO-Space (Kim et al., 2024), and ROBOPOINT (Yuan et al., 2025).

Our key contributions are as follows:

- We introduce a *propose-validate* prompting framework for a VLM that progressively performs coarse spatial reasoning to ground a natural language command.
- We provide a superpixel-based module to refine the output of the reasoning stage that allows fine, pixel-level refinements for a candidate solution, accounting for local object context.
- We introduce a space-grounding benchmark of 350 examples consisting of diverse challenging instructions, and conduct extensive comparisons with state-of-the-art baselines.

2 Related Work

Space grounding: Traditional approaches manually link each predicate in a fixed set to various representations, such as potential fields (Stopp et al., 1994) or fuzzy spatial membership functions (Bloch and Saffiotti, 2003; Tan et al., 2014). Deep learning-based methods emerge and predict pixel coordinates or pixel-level probability maps for placement (Venkatesh et al., 2021; Mees et al., 2020; Shridhar et al., 2022). Moreover, researchers explore modeling the space as probabilistic parameterizations, such as Gaussian mixture models (Zhao et al., 2023) or Boltzmann energy functions (Gkanatsios et al., 2023). Notably, LINGO-Space (Kim et al., 2024) models the space using a Bayesian update of polar distributions to understand spatiotemporal descriptions.

As VLMs prove effective on a wide range of robotic tasks (Brohan et al., 2023; Shah et al., 2023), researchers have begun applying VLMs to space grounding. These models enable direct prediction of goal points grounded by robotic instructions in images. RoboPoint (Yuan et al., 2025) predicts 2D keypoints via fine-tuning on spatial phrases, enabling the model to translate relational commands into precise points. Recent VLMs, such as Molmo (Deitke et al., 2025) and Gemini 2.5 (Comanici et al., 2025), show a zero-shot point prediction from language instructions. However, their point-based outputs remain coarse and lack fine-grained spatial precision. Our method uses learning-based superpixel refinement to precisely refine the parameterized canonical region within the target space.

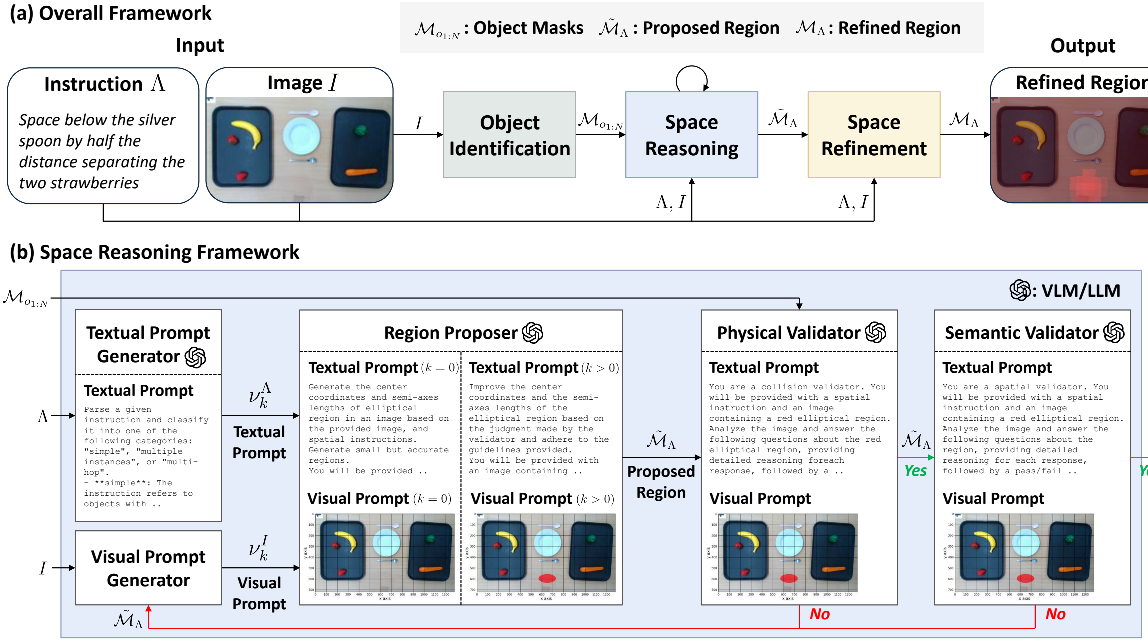
Spatial reasoning with VLMs: VLMs provide open-world multimodal understanding, making them applicable to a broad range of downstream tasks, such as image-text retrieval (Chen et al., 2023), zero-shot visual question answering (Li et al., 2023), and segmentation (Lai et al., 2024). However, early VLMs fail spatial reasoning since they behave as a bag-of-tokens, which lose positional detail (Yuksekgonul et al., 2023; Li et al., 2024; Chen et al., 2024). To overcome the limitation, recent approaches integrate depth features into VLMs to provide scale cues (Cheng et al., 2024). Furthermore, fine-tuning VLMs using extensive spatial relation annotations improves reasoning over complex object interactions in diverse scenes (Yuan et al., 2025; Song et al., 2025). Our method guides the VLM with structured visual and textual prompts, enhancing its spatial reasoning to predict the target space described by the instruction.

3 Methodology

We introduce C2F-SPACE, a hierarchical space-grounding method that combines grid-guided VLM prompting with superpixel-based refinement.

3.1 Overview

Consider an input instruction Λ and an input RGB image $I \in \mathbb{Z}^{3\times H\times W}$ containing N objects, where



Space below the silver spoon by half the distance separating the two strawberries

Parse a given instruction and classify it into one of the following categories: "simple", "multiple instances", or "multi-hop".
- **simple**: The instruction refers to objects with ..

Generate the center coordinates and semi-axes lengths of elliptical region in an image based on the provided image, and spatial instructions.
Generate small but accurate regions.
You will be provided ..

Improve the center coordinates and the semi-axes lengths of the elliptical region based on the judgment made by the validator and adhere to the guidelines provided.
You will be provided with an image containing ..

You are a collision validator. You will be provided with a spatial instruction and an image containing a red elliptical region. Analyze the image and answer the following questions about the red elliptical region, providing detailed reasoning foreach response, followed by a ..

You are a spatial validator. You will be provided with a spatial instruction and an image containing a red elliptical region. Analyze the image and answer the following questions about the region, providing detailed reasoning for each response, followed by a pass/fail ..

Figure 2: (a) Overall framework of C2F-SPACE. Given an instruction Λ and an image I, the object identification module first obtains object masks $\mathcal{M}_{o_{1:N}}$. The space-reasoning module iteratively proposes a candidate region $\tilde{\mathcal{M}}_{\Lambda}$, and the subsequent space-refinement module adjusts this proposal to produce the high-precision region $\mathcal{M}_{\Lambda}$. (b) Space-reasoning module in detail. At each iteration k, the textual prompt generator constructs an instruction-specific textual prompt ν_k^{Λ}, while the visual prompt generator creates a grid-guided visual prompt ν_k^{I}. Next, the region proposer predicts an elliptical region through the prompts. Two validators then assess the proposal: the physical validator rejects regions that collide with objects, and the semantic validator checks consistency with the spatial instruction. Validation feedback (green "Yes" / red "No") drives an iterative refinement loop; once both validators accept the region, the system forwards $\tilde{\mathcal{M}}_{\Lambda}$ to the space-refinement stage.

H and W denote the image height and width, respectively. Our goal is to predict a space mask (i.e., segment) $\mathcal{M}_{\Lambda} \in \mathbb{Z}^{H\times W}$ that corresponds to Λ. As shown in Fig. 2 (top), inference proceeds in three steps: 1) *object identification*, 2) *space reasoning*, and 3) *space refinement*. This modular design improves grounding accuracy and reliability while mitigating hallucinations in the VLM.

In detail, the *object-identification* step extracts a unified mask $\mathcal{M}_{o_{1:N}}$ covering all objects in the image. The *space-reasoning* step then infers a canonical region $\tilde{\mathcal{M}}_{\Lambda} \in \mathbb{Z}^{H\times W}$ leveraging a grid-guided visual-text prompt, and iteratively refines it until the region satisfies both physical and semantic constraints with respect to $\mathcal{M}_{o_{1:N}}$ and Λ. Finally, the *space-refinement* step locally adapts the canonical region to precisely fit the environment using superpixels.

The grid guidance helps the VLM distinguish low or texture free space lacking distinctive features while maintaining semantic consistency with the instruction. Superpixel-based refinement reduces computational cost and enhances alignment with spatio-semantic pixel distribution compared to pixel-level refinement. We describe each component in detail below.

3.2 Open-set *Object Identification*

Prior to grounding, we construct a joint object mask $\mathcal{M}_{o_{1:N}} = \mathcal{M}_{o_1} \cup ... \cup \mathcal{M}_{o_N}$, where each $\mathcal{M}_{o_i} \in \mathbb{Z}^{H\times W}$ denotes the binary mask of the i-th object. We use the constructed mask in the validation process of the *space-reasoning* step. As our focus is on space grounding without prior object knowledge, we employ Grounded-SAM (Ren et al., 2024), an open-set object-mask identifier that first detects object bounding boxes using Grounding DINO (Liu et al., 2024), and then extracts the corresponding masks using SAM (Kirillov et al., 2023), conditioned on the detected boxes.

3.3 Grid-guided *Space Reasoning*

This step guides the VLM to propose a canonical region $\tilde{\mathcal{M}}_{\Lambda}$ maximizing its reasoning capability. Fig. 2 (bottom) illustrates the iterative reasoning and validation process. At each iteration, the prompt generator creates a visual prompt ν^{I} for grid-based guidance and a text prompt ν^T to interpret the instruction Λ. Feeding these concatenated prompts into the VLM yields M ellipses $[\varepsilon_1, ..., \varepsilon_M]$, where each ε_j represents a proposed region. Note that M typically ranges from one to two depending on the VLM output. Then, we combine ellipses to form the predicted region $\tilde{\mathcal{M}}_\Lambda$.

To validate $\tilde{\mathcal{M}}_\Lambda$, we introduce two validators: a physical validator to assess feasibility for object placement, and a semantic validator to ensure consistency with the instruction Λ. We detail each component below.

1) Prompt generator: At each iteration k, our generator produces a novel grid-guided visual prompts ν_k^I by overlaying a grid $I^{\text{grid}} \in \mathbb{Z}^{3\times H\times W}$ onto the input image I, providing explicit visual cues. We draw the grid I^{grid} in black with a thickness of 1.4 pixels at 100 DPI and 100-pixel intervals, regardless of the size of the image. In the grid, we also display axis tick values and labels (e.g., "x axis") to support reasoning about direction and distance. The grid remains on the top layer throughout all iterations. We define the initial prompt as $\nu_0^I = I \oplus I^{\text{grid}}$, where $\oplus$ denotes the overlay operation. From the second iteration ($k > 0$), we additionally overlay the latest predicted region $\tilde{\mathcal{M}}_\Lambda$ as red pixels: $\nu_{k>0}^I = I \oplus I^{\text{grid}} \oplus \tilde{\mathcal{M}}_\Lambda$.

Alongside, the generator produces a text prompt ν_k^Λ using an LLM to guide the VLM in decomposing the grounding process while interpreting the visual prompt ν_k^I. The prompt includes (i) *object guidance*—identifying instruction-relevant objects and their spatial extent based on Λ, (ii) *region guidance*—prompting the VLM to output region coordinates, and (iii) *collision-free guidance*—ensuring predicted coordinates avoid object overlap. From iteration $k > 0$, the prompt also incorporates the feedback from the validators, enabling the VLM to refine proposals based on prior errors. Note that the VLM may return coordinates for multiple ellipses (see the prompt detail in Appendix B).

2) VLM-based region proposer: Upon receiving the two prompts $\nu_k^{\mathbf{I}}$ and ν_k^Λ, the VLM predicts a unified region $\tilde{\mathcal{M}}_\Lambda \in \mathbb{Z}^{H\times W}$ consisting of canonical region proposals, represented as ellipses. We parameterize each ellipse ε_j using its center coordinates, semi-axis lengths, and rotation angle, extracted directly from the VLM's structured output via a text-to-ellipse conversion. Given these parameters, we generate individual elliptical masks and combine them to form a final region $\tilde{\mathcal{M}}_\Lambda$ through logical union.

3) Physical & semantic validators: To ensure that the proposed region $\tilde{\mathcal{M}}_\Lambda$ satisfies both the physical and semantic requirements of the instruction Λ, we conduct a two-stage validation at each iteration. The first stage checks the physical validity of the proposed region mask. In the case where $\tilde{\mathcal{M}}_\Lambda$ intersects with the joint object mask $\mathcal{M}_{o_{1:N}}$, we further assess whether the intersection supports valid placements (e.g., "on the dish" or "in the basket"). Otherwise, we regard $\tilde{\mathcal{M}}_\Lambda$ is suitable for placement actions.

For the further assessment, we issue another VLM query with a validation prompt consisting of a visual prompt $\nu_k^{I,\text{phy}} = I \oplus I^{\text{grid}} \oplus \tilde{\mathcal{M}}_\Lambda$ and a text prompt $\nu_k^{\Lambda,\text{phy}}$ that asks whether placing an object at $\tilde{\mathcal{M}}_\Lambda$ is physically feasible, yielding a binary response.

For semantic validation, we reuse the visual prompt $\nu_k^{I,\text{sem}}$ $(= \nu_k^{I,\text{phy}})$ and provide a semantic text prompt $\nu_k^{\Lambda,\text{sem}}$, asking whether $\tilde{\mathcal{M}}_\Lambda$ satisfies the spatial semantics of Λ. To improve accuracy, we instruct the VLM to decompose compositional instructions and validate each sub-component individually within $\nu_k^{\Lambda,\text{sem}}$. If $\tilde{\mathcal{M}}_\Lambda$ passes both validations or if the process reaches the maximum number of iterations, we return it; otherwise, we return to the prompt generation step.

3.4 Superpixel-based *space refinement*

We locally adapt the coarse canonical region $\tilde{\mathcal{M}}_\Lambda$ to the fine-grained structure of the surrounding environment as well as free space, we predict the final manipulation region $\mathcal{M}_\Lambda$ by modeling its residual. We particularly introduce a residual learning module by decomposing the instructed region as $\mathcal{M}_\Lambda = \tilde{\mathcal{M}}_\Lambda \oplus \mathcal{M}_\Lambda^{\text{residual}}$ where $\mathcal{M}_\Lambda^{\text{residual}}$ captures local refinements over the superpixel space. To simplify learning, we approximate this decomposition in the logit space as

$$l_\Lambda = \alpha \tilde{l}_\Lambda + (1-\alpha) l_\Lambda^{\text{residual}}, \qquad (1)$$

where l_Λ, $\tilde{l}_\Lambda$, and $l_\Lambda^{\text{residual}}$ denote the superpixel-wise logits of $\mathcal{M}_\Lambda$, $\tilde{\mathcal{M}}_\Lambda$, and $\mathcal{M}_\Lambda^{\text{residual}}$, respectively; $\alpha \in [0, 1]$ is a scaling factor and $|l_\Lambda| = L$ with L denoting the number of superpixels. We describe superpixel generation and logit estimation below.

To compute $\tilde{l}_\Lambda$ for the predicted region $\tilde{\mathcal{M}}_\Lambda$, we generate superpixels from the grayscale image $I^{\text{gray}} \in \mathbb{R}^{H\times W}$ using SLIC (Achanta et al., 2019). We then assign a pseudo logit to each superpixel in

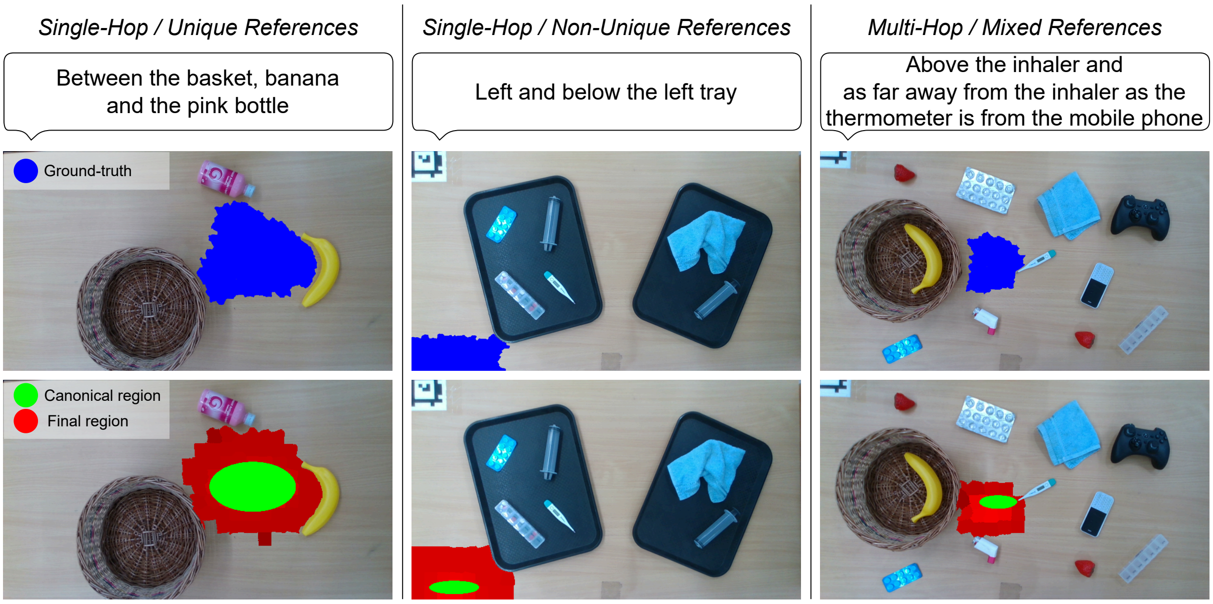

Single-Hop / Unique References
Between the basket, banana and the pink bottle
Single-Hop / Non-Unique References
Left and below the left tray
Multi-Hop / Mixed References
Above the inhaler and as far away from the inhaler as the thermometer is from the mobile phone
Ground-truth
Canonical region
Final region

Figure 3: Example data sample and prediction results from our space grounding benchmark. Starting from the left, each instruction and image pair illustrates *single-hop space reasoning with unique references*, *single-hop space reasoning with non-unique references*, and *multi-hop space reasoning with unique/non-unique references* cases. The blue regions in the top row indicate the superpixel-level ground-truth labeled by a human. In the bottom row, the green regions show the canonical region proposed by the VLM, and the red regions show the final predictions refined by the superpixel-level refinement module.

two steps:

$$\tilde{\mathcal{M}}_\Lambda \xrightarrow{smoothing} \tilde{\mathcal{M}}'_\Lambda \xrightarrow{aggregation} \tilde{l}_\Lambda, \qquad (2)$$

where $\tilde{\mathcal{M}}'_\Lambda$ represents a center-distance weighted value for each ellipse; pixels farther from the center of each ellipse have smaller values. We then compute the pseudo-logit value $\tilde{l}_\Lambda$ by averaging the pixel values within each superpixel.

To model the residual $l_\Lambda^{\text{residual}}$, we construct a superpixel graph where each node represents a superpixel and each edge connects adjacent superpixels. Node features consist of the mean, minimum, and maximum values within each superpixel in I^{gray} and $\tilde{\mathcal{M}}_\Lambda$, along with a binary indicator specifying whether the instruction Λ requires distance reasoning, identified by the LLM.

We use a graph neural network, GPS (Rampášek et al., 2022), to predict the superpixel-wise residual logit $l_\Lambda^{\text{residual}}$. To supervise it, we compute a focal loss of the predicted probabilities $\mathbf{p} = \sigma(l_\Lambda) \in \mathbb{R}^L$ given the ground-truth space labels, where σ is the sigmoid function. Finally, we project the superpixel-wise probabilities $\mathbf{p}$ back to the pixel space and binarize it, resulting in the final refined region $\mathcal{M}_\Lambda$.

4 Experimental Setup

Our experiments aim to measure performance improvements in space grounding tasks that require reasoning.

4.1 Benchmark description

We introduce a superpixel-level space grounding benchmark consisting of real-world scene images, natural language instructions in English, and human-annotated ground-truth labels. We capture tabletop scenes that may include containers holding other objects, as well as multiple identical items. The instructions cover nine types of spatial relations: 'left,' 'right,' 'above,' 'below,' 'near,' 'far,' 'inside,' 'outside,' and 'along the direction of.' We annotate the ground-truth labels at the superpixel level rather than the pixel level. We collected the data with approval from the institutional review board (IRB).

We categorize the instructions into three types based on the number of reasoning hops and the uniqueness of reference objects, as illustrated in Fig. 3: (i) *single-hop space reasoning with unique references*, (ii) *single-hop space reasoning with non-unique references*, and (iii) *multi-hop space reasoning with unique/non-unique references*

The first category, *single-hop space reasoning* *with unique references*, includes instructions involving one to four spatial expressions with clearly identifiable reference objects. These cases require only directly interpreting spatial terms and locating reference objects.

The second category, *single-hop space reasoning with non-unique references*, includes instructions that require resolving ambiguous references. These ambiguities often arise in scenes with multiple visually similar objects, as shown in the middle column of Fig. 3.

The third category, *multi-hop space reasoning with unique/non-unique references*, involves multi-hop inference. The model needs to understand the distance between two objects and apply that distance relative to another reference object, often requiring multiplicative reasoning. For example, the instruction "above the inhaler and as far away from the inhaler as the thermometer is from the mobile phone," belongs to the *multi-hop spatial reasoning* category.

The benchmark contains a total of 350 samples, which are split into 200 for training, 50 for validation, and 100 for testing. The validation and test sets maintain a balanced distribution across the three instruction categories by design.

We evaluate the performance of our method and baselines using two metrics. The first is the *success rate*, which considers a prediction successful if either the maximum-probability point or the centroid of the predicted mask lies within the ground-truth region. However, the success rate is a binary indicator and therefore cannot capture the quality of the predicted mask in finer detail. For example, even if the predicted mask covers only a small portion of the ground-truth area, the success rate remains the same as long as the maximum-probability point falls inside the ground truth. To address this limitation, we additionally measure the *intersection over union (IoU)* between the predicted and ground-truth masks in pixel space, which reflects how well the predicted region overlaps with the true target area.

4.2 Baseline Methods

We evaluate three space-grounding baselines:

- **CLIPORT** (Shridhar et al., 2022): A language-conditioned imitation learning approach for pick-and-place manipulation that predicts a pixel location based on the CLIP (Radford et al., 2021) image and language encoding. Note that we disable its rotational augmentation and extende the loss function to accept a mask as ground truth.
- **LINGO-Space** (Kim et al., 2024): A probabilistic space-grounding method that incrementally estimates spatial distributions based on composite referring expressions using configurable polar distributions. We extend the loss function to use a mask as a ground truth instead of a point.
- **ROBOPOINT** (Yuan et al., 2025): A VLM fine-tuned with synthetic instructions to predict keypoint affordances from language, including spatial instructions. ROBOPOINT predicts multiple points as an output for space grounding. Following the original paper (Yuan et al., 2025), we use the pre-trained model without modification and measure the success rate as the ratio of predicted points that fall inside the ground truth mask. We do not measure IoU for ROBOPOINT.

We apply Otsu's thresholding to convert a predicted probability into a binary mask for both our model and all baselines, except RoboPoint, since spatial regions tend to exhibit low confidence, making it difficult to select a universal threshold across different cases. We repeat the experiment twice for each methods.

5 Evaluation

We analyze the performance of our method and baselines on the superpixel-based space-grounding benchmark. As shown in Fig. 4, our method consistently outperforms all baselines in both success rate and IoU across all instruction categories.

Even in the *single-hop space reasoning with unique references* cases—where all baselines achieve their highest performance—the success rate reaches only 47.1%. Their performance drops even further in more challenging scenarios such as *non-unique references* and *multi-hop* reasoning. LINGO-Space, the strongest among the baselines, loses 21.3% and 30.5% in success rate for these categories, respectively. Since the LLM-based parser of LINGO-SPACE relies solely on the instruction without understanding the scene, it fails to resolve ambiguous references or handle multi-step reasoning. CLIPORT does not separate spatial relationships from reference objects, which leads to frequent failures even in relatively simple cases. ROBOPOINT also performs poorly on specific relations like "far" and multi-hop reasoning, as its dataset does not cover such relations.

Our method, C2F-SPACE, uses the reasoning capabilities of the VLM and achieves over 50%

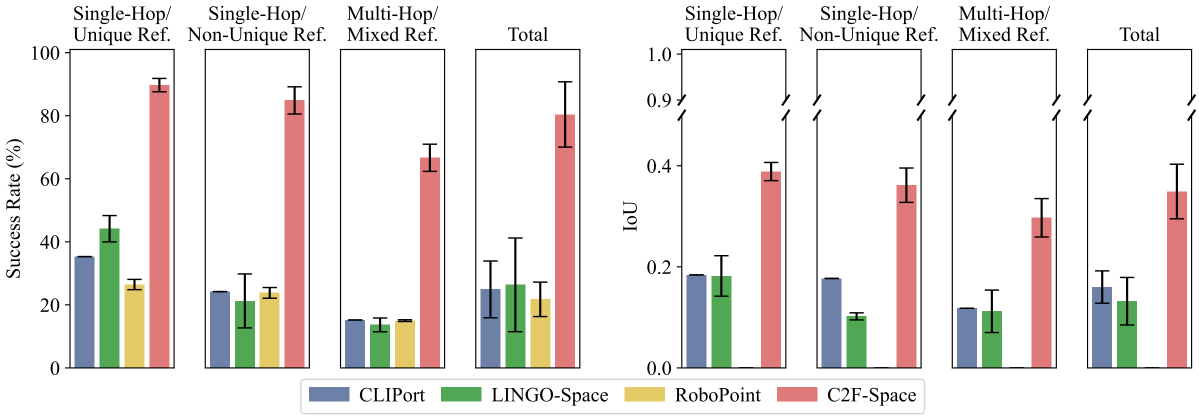


Figure 4: Performance comparison in terms of success rate [%] (left) and Intersection over Union (IoU) (right). The 'Total' denotes the average across three instruction categories. The error bar denotes standard deviation.

higher success rates than the best-performing baseline, LINGO-Space. C2F-Space accurately grounds ambiguous reference expressions, such as "a yellow block near to the leftmost blue block," and reaches a success rate of 84.9% in the *non-unique references* category. It also correctly interprets numerical relations and relative distances, achieving 66.7% success on *multi-hop spatial reasoning*.

In terms of IoU, C2F-Space significantly outperforms all baselines, achieving an average score of 0.361 across categories—more than twice that of the second-best baseline, CLIPort (0.160). This task is fundamentally different from typical object segmentation, as space grounding lacks clear object boundaries. As a result, even semantically correct predictions can yield lower IoU scores. As illustrated in the third column of Fig. 3, the IoU can remain relatively low (e.g., 0.411), even when the grounded region is reasonable. Given this inherent difficulty, C2F-Space achieves notably high IoU by refining a coarse canonical region into a fine-grained, superpixel-based space. As shown in the first and second columns of Fig. 3, the refinement module effectively captures object boundaries, contributing to the improved spatial precision.

We conduct ablation studies to evaluate the impact of the superpixel-level refinement. The success rate remains comparable with or without this refinement, as shown in Fig. 5, indicating that reasoning primarily relies on the VLM module. Including the superpixel-level refinement improves IoU by more than threefold, resulting in the highest IoU among all baselines. As shown in the bottom rows of Fig. 3, the refinement module adjust the prediction without overlapping the objects.

We further analyze the importance of each component of the grid-guided space reasoning module, focusing on the grid-based visual prompt and the validation. As shown in Fig. 6, removing the validation step and its corresponding iteration causes a slight performance drop of about 2% in success rate. However, removing the visual prompting with the grid results in a significant drop of 14.6%, and

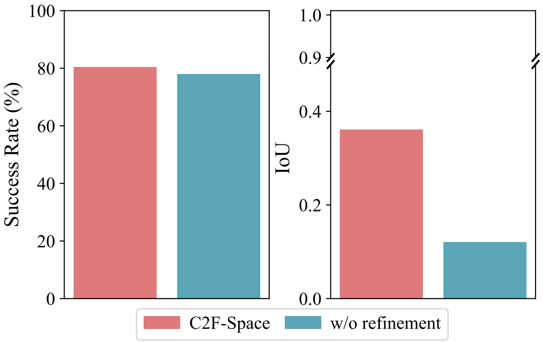


Figure 5: Success rate [%] and IoU with and without the superpixel-based space refinement module.

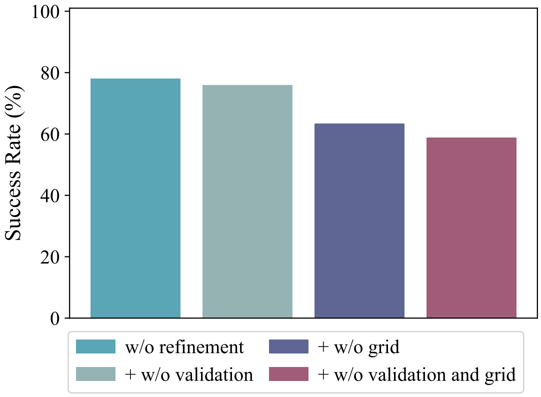


Figure 6: Ablation results on the component of the grid-guided space reasoning module.

eliminating both the visual prompting and validation leads to an additional 4.5% decrease. These results highlight the critical role of grid-guided visual prompting in applying the VLM for effective space grounding by providing essential visual guidance.

6 Conclusion

We proposed C2F-SPACE, a two-stage space-grounding framework that combines a VLM with a superpixel-level refinement module to identify regions that are both physically feasible and semantically consistent with the input instruction. Our method first uses the VLM to globally predict a coarse region based on the novel *propose-validate* prompting technique. Then, it locally refines this coarse prediction into a precise region using superpixel-level decisions. This design allows for both accurate localization and robust generalization across complex spatial expressions. Experimental results on our new benchmark show that superpixel-based refinement significantly improves IoU without reducing the success rate. In addition, the *propose-validate* prompt, combined with the grid inpainting, plays a key role in guiding the prediction process. Overall, C2F-SPACE outperforms existing baselines in grounding accuracy, demonstrating its effectiveness in resolving spatial references within complex scenes and instructions.

Limitations

While C2F-SPACE effectively refines spatial predictions from VLMs, it inherently depends on the quality of the initial coarse prediction. When the VLM produces a significantly misplaced region, the refinement stage often fails to recover the correct target, as it operates based on the initial estimate. This reliance on the VLM's spatial reasoning limits the system's ability to correct large errors. This large initial error may lead to task failure or unsafe behavior in real-world robotics. Future work could incorporate residual correction policies or multi-stage validation mechanisms that use additional context to revise poor initial predictions more robustly.

References

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, Sabine Süsstrunk, and 1 others. 2019. Slic superpixels. Technical report.

Isabelle Bloch and Alessandro Saffiotti. 2003. On the representation of fuzzy spatial relations in robot maps. In *Intelligent systems for information processing*, pages 47–57. Elsevier.

Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, and 1 others. 2023. Do as i can, not as i say: Grounding language in robotic affordances. In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 287–318. PMLR.

Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee. 2024. Vip-llava: Making large multimodal models understand arbitrary visual prompts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12914–12923.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14455–14465.

Xi Chen, Xiao Wang, Soravit Changpinyo, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, and 1 others. 2023. Pali: A jointly-scaled multilingual language-image model. In *Proceedings of the International Conference on Learning Representation (ICLR).*

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. 2024. Spatialrgpt: grounded spatial reasoning in vision-language models. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 135062–135093.

Long Cheng, Jiafei Duan, Yi Ru Wang, Haoquan Fang, Boyang Li, Yushan Huang, Elvis Wang, Ainaz Eftekhar, Jason Lee, Wentao Yuan, and 1 others. 2025. Pointarena: Probing multimodal grounding through language-guided pointing. *arXiv preprint arXiv:2505.09990.*

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261.*

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, and 1 others. 2025. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 91–104.

Nikolaos Gkanatsios, Ayush Jain, Zhou Xian, Yunchu Zhang, Christopher Atkeson, and Katerina Fragkiadaki. 2023. Energy-based models are zero-shot planners for compositional scene rearrangement. In *Proceedings of Robotics: Science and Systems (RSS)*.

Kanishk Jain, Varun Chhangani, Amogh Tiwari, K Madhava Krishna, and Vineet Gandhi. 2023. Ground then navigate: Language-guided navigation in dynamic scenes. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4113–4120. IEEE.

Dohyun Kim, Nayoung Oh, Deokmin Hwang, and Daehyung Park. 2024. Lingo-space: Language-conditioned incremental grounding for space. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, volume 38, pages 10314–10322.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, and 1 others. 2023. Segment anything. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 4015–4026.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. 2024. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9579–9589.

Chengzu Li, Caiqi Zhang, Han Zhou, Nigel Collier, Anna Korhonen, and Ivan Vulić. 2024. Topviewrs: Vision-language models as top-view spatial reasoners. pages 1786–1807.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 19730–19742. PMLR.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, and 1 others. 2024. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 38–55. Springer.

Oier Mees, Alp Emek, Johan Vertens, and Wolfram Burgard. 2020. Learning object placements for relational instructions by hallucinating scene representations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 94–100.

OpenAI. 2025. Openai o4-mini (july 15 version). https://platform.openai.com/docs/models/o4-mini.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. 2022. Recipe for a general, powerful, scalable graph transformer. *Conference on Neural Information Processing Systems (NeurIPS)*, 35:14501–14515.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, and 1 others. 2024. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*.

Dhruv Shah, Błażej Osiński, Sergey Levine, and 1 others. 2023. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 492–504. PMLR.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. 2022. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 894–906. PMLR.

Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. 2023. What does clip know about a red circle? visual prompt engineering for vlms. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 11987–11997.

Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. 2025. Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15768–15780.

Eva Stopp, Klaus-Peter Gapp, Gerd Herzog, Thomas Laengle, and Tim C Lueth. 1994. Utilizing spatial relations for natural language access to an autonomous mobile robot. In *Proceedings of the German Annual Conference on Artificial Intelligence*, pages 39–50. Springer.

Jiacheng Tan, Zhaojie Ju, and Honghai Liu. 2014. Grounding spatial relations in natural language by fuzzy representation for human-robot interaction. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1743–1750.

Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. 2014. scikit-image: image processing in Python. *PeerJ*, 2:e453.

Sagar Gubbi Venkatesh, Anirban Biswas, Raviteja Upadrashta, Vikram Srinivasan, Partha Talukdar, and Bharadwaj Amrutur. 2021. Spatial reasoning from natural language instructions for robot manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 11196–11202.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. 2025. Robopoint: A vision-language model for spatial affordance prediction in robotics. In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 4005–4020. PMLR.

M Yuksekgonul, F Bianchi, P Kalluri, D Jurafsky, J Zou, and 1 others. 2023. When and why vision-language models behave like bags-of-words, and what to do about it? In *Proceedings of the International Conference on Learning Representation (ICLR)*.

Zirui Zhao, Wee Sun Lee, and David Hsu. 2023. Differentiable parsing and visual grounding of natural language instructions for object placement. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 11546–11553.

A Implementation Details

Open-set object identification. We use Grounded-SAM with Grounding DINO-B with a visual backbone SWIN-B and a text backbone BERT-base-uncased and SAM with a visual backbone VIT-B. We prompt the Grounding DINO with texts ‘object, objects’ to extract the bounding boxes.

Grid-guided space reasoning. We use ‘o4-mini-2025-04-16’ for whole tasks. We retry execution when an error occurs on the API side, such as when the parsing result is missing.

Superpixel-based space refinement. We develop on the GPS model using the open-source code (https://github.com/vijaydwivedi75/lrgb). We use scikit-image (van der Walt et al., 2014) for generating superpixels, following the implementation of GPS. The number of total parameters is about 0.1M and takes less than an hour for data pre-processing and training on a single NVIDIA RTX 3090 GPU. The hyperparameters are as follows.

epoch	50
learning rate	1e-4
α	0.1
# of layers	2
# of heads	4
hidden dimension	64
SLIC compactness	10

Table 1: Hyperparameters for superpixel-based space refinement module

Baselines. In the case of CLIPort, we set the pick position at the center of the image, as we focus solely on the placing task during training. For LINGO-Space, we use GroundingDINO and o4-mini to ground objects in the scene and to provide additional annotations for the ground-truth reference objects. Rather than detecting objects, we ground them to better assist human annotators in identifying the reference objects. The detailed information about the baselines is as below.

- CLIPort (Apache-2.0 license): https://github.com/cliport/cliport
- LINGO-Space (MIT license): https://github.com/rirolab/LINGO-Space
- RoboPoint (Apache-2.0 license): https://github.com/wentaoyuan/RoboPoint (checkpoint: https://huggingface.co/wentao-yuan/robopoint-v1-vicuna-v1.5-13b)

B Full Prompt Examples

First, we identify the instruction categories using the prompt shown in Fig. 7. We then use the identified category as part of the node features in the superpixel-based space refinement module.

```
Parse a given instruction and classify it into one of the following categories: "simple", "multiple instances", or "multi-hop".

- **simple**: The instruction refers to objects with unambiguous and unique presence in the scene.
...

# Steps
1. Analyze the given instruction to identify all the objects it refers to.
2. Cross-reference these objects with the provided list of objects in the scene.
3. Classify the instruction based on the following criteria:
- **simple**: All referenced objects appear uniquely as singular instances.
...

# Output Format
The output should be in a structured response indicating:
- ’Instruction type’: One of "simple", "multiple Instances", or "multi-hop".
- ’Reason’: A brief justification based on object presence or relational dependency.

# Examples
**Example 1:**
Input: Instruction: "Between basket and tray and nearer to the tray"

Output:
Instruction type: simple
```

```
Reason: The instruction refers to a basket and a tray, each uniquely present in the scene.
...
```

Figure 7: Instruction category identification prompt

Based on the identified instruction category, we decide whether to add additional explanation or not. The main region-proposal text prompt for $k = 0$ is as follows.

```
Generate the center coordinates and semi-axes lengths of elliptical region in an image based on the provided image, and spatial instructions. Generate small but accurate regions.

You will be provided with an overhead image of a table containing various objects, and spatial instructions. Your task is to determine suitable center coordinates and semi-axes lengths, ensuring they satisfy the spatial instructions, maintain specified bounds, and avoid collisions with objects, except as noted.

Image: The image is an overhead image of the table with various objects on it. The image has also been overlaied with a grid to aid you in co-relating the object extent coordinates with the image. Origin (0,0) is at the top-left corner of the image, X increasing to the right and Y increasing downward.

# Spatial Instructions and Collision Avoidance
- **General Rules**:
- Values for X should be between 0 and 1280, and Y between 0 and 720.
- The elliptical region should be generated in collision-free spaces and should maintain distance from the object boundaries.

# Steps
1. **Extract Information**: Identify relevant objects and their extents from the image and spatial instructions.
2. **Generate Regions**: Use the spatial region generation guidance to determine appropriate coordinates for the regions.
3. **Ensure Collision-Free Placement**: Verify that the generated coordinates are free from collision, following collision avoidance principles.

# Output Format
The "center_coordinates" should be a list of center coordinates in the form of [[X1, Y1]]. The "semi_axes_lengths" should be the lengths of semi-major and semi-minor axes in the form of [[a, b]], a >= b. The "angle" should be the tilted angle of the ellipse in degrees.
```

Figure 8: Region proposal prompt for $k = 0$

When $k > 0$, the textual prompt is as follows.

```
Improve the center coordinates and the semi-axes lengths of the elliptical region based on the judgment made by the validator and adhere to the guidelines provided.

You will be provided with an image containing a red elliptical region, spatial instructions, previously generated center point coordinates and previously generated semi-axes lengths for that region, and judgments from a validator, ensuring they satisfy the spatial instructions, maintain specified bounds, and avoid collisions with objects, except as noted.

Image: The image is an overhead image of the table with various objects on it. The image has also been overlaied with a grid to aid you in co-relating the object extent coordinates. (0,0) at the top-left corner of the image, X increasing to the right and Y increasing downward. The red elliptical region shows previous decision.

# Spatial Instructions and Collision Avoidance
- **General Rules**:
- Values for X should be between 0 and 1280, and Y between 0 and 720.
- elliptical region should be generated in collision-free spaces and should maintain distance from the object boundaries.
- If moving the region, then make sure that the region is not overlapping with some other object after movement. If you think it will collide, then also reduce semi-axes lengths of the region.
- Predict relative movement of the prediction (e.g. move -30 pixels in X direction), and then generate the final region based on it.

# Output Format
The "center_coordinates" should be a list of list coordinates in the form of [[X1, Y1]] ensuring ‘X‘ is between 0 and 1280 and ‘Y‘ is between 0 and 720. The "semi_axes_lengths" should be the lengths of semi-major and semi-minor axes in the form of [[a, b]], a >= b. The "angle" should be the tilted angle of the ellipse in degrees.
```

Figure 9: Region proposal prompt for $k > 0$

The prompt for physical validation is as follows.

```
You are a collision validator. You will be provided with a spatial instruction and an image containing a red elliptical region. Analyze the image and answer the following questions about the red elliptical region, providing detailed reasoning for each response, followed by a pass/fail conclusion for each question.
```

# Image
- The image is an overhead image of the table with various objects on it. The image has also been overlaied with a grid. Origin (0,0) is at the top-left corner of the image, X increasing to the right and Y increasing downward.
- On the image is also the red elliptical region. The grid will help you give accurate judgments to correct the location/size of the red elliptical region if it is in collision.

# Judgment Guidelines
- For each question, judge the prediction. Then, justify your judgment.
- Do not include any suggestions for improvement in your response.

# Questions to be answered:
1. **Collision Check**: Assess the objects with which overlapping/collision is allowed out of all the objects in the list. If the objects with which the region overlaps is allowed, then answer as "pass", else, "fail".

Answer each question by referencing the image and spatial instruction. After your reasoning for each question, include a "pass" or "fail" judgment for clear assessment.

Figure 10: Physical validator prompt

The prompt for semantic validation is as shown in Fig. 11.

You are a spatial validator. You will be provided with a spatial instruction and an image containing a red elliptical region. Analyze the image and answer the following questions about the region, providing detailed reasoning for each response, followed by a pass/fail conclusion for each question. The region has already gone through a collision test and it has been verified that it is not in collision with any object that it should not collide with.

# Judgment Guidelines
- For each question, judge the prediction. Then, justify your judgment.
- Consider the distance properly. Unless a specific axis is mentioned, interpret the distance as the general square root distance between objects, considering a center coordinate or boundary.
- When projecting the distance, consider the object extents to avoid collision and to maintain appropriate spacing.
- Do not include any suggestions for improvement in your response.

# How to check if red region is satisfying the instruction:

1) **Break the instruction into multiple segments.**
- Example 1:
- Instruction: "To the right of the tray with edible items and below the spectacles."
- Segments: [right of the tray with edible items, below the spectacles]

- Example 2:
- Instruction: "left and blue cloth and far from the marker"
- Segments: [left of the blue cloth, far from the marker]

2) ** Spatial validation of the red elliptical region for each segment **: For each segment, check if the position of the red elliptical region is satifying each segment. Make use of the spatial guidelines to check validity of position of red elliptical region with each segment. Keep in mind that the segments are not individual instructions. They are dependant on the segments before and after them. Example: If the instruction is "right of the strawberry and between carrot and banana". segments will be: [right of strawberry, between carrot and banana]. Now, when you check for between carrot and banana, keep in mind the previous segment. The red elliptical region might not be directly in between the carrot and banana, as right of strawberry also has to be satisfied.

3) If the object being referred to in the instruction has multiple instances in the image, then carefully check if the red elliptical region is following the spatial instruction using the correct instance of the object.

# Questions to be answered:
1. **Spatial Compliance**: Determine whether the elliptical region is according to the spatial instruction provided. Describe your reasoning. Use the "spatial guidelines" to come to a conclusion.

Answer each question by referencing the image, spatial instruction and objects extents. After your reasoning for each question, include a "pass" or "fail" judgment for clear assessment.

Figure 11: Semantic validator prompt

C Data collection

Experienced research assistants who were familiar with the robot and the application domain performed the annotations. All annotators were male, aged between 23 and 28 years. These research assistants collected the data as part of their regular duties, and we did not recruit any external annotators. We provided the annotators with detailed instructions, including full information about the application domain and the intended use of the data (i.e., training a space grounding model). Each annotator gave informed consent before every data collection session. To avoid fatigue, no one worked for more than 90 minutes in a single sitting, and each person participated in only one sitting per day. A second annotator, possibly supported by a grammar-checking AI tool, reviewed the instructions and checked for grammatical consistency.