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

Nayoung Oh[1], Dohyun Kim[1], Junhyeong Bang[1], Rohan Paul[2*], Daehyung Park[1*]

[1]Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea.
[2]Indian Institute of Technology Delhi, Hauz Khas, India.

*Corresponding author(s). E-mail(s): rohan@cse.iitd.ac.in; daehyung@kaist.ac.kr;
Contributing authors: nyoh@kaist.ac.kr; dohyun141@kaist.ac.kr; jazzdosa@kaist.ac.kr;

Abstract

Space grounding refers to localizing a set of spatial references described in 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 a fine-grained region of outputs. To overcome these limitations, we propose C2F-Space, a novel coarse-to-fine space-grounding framework that (i) estimates an approximated yet spatially consistent region using a VLM, then (ii) refines the region to align with the local environment through superpixelization. For the coarse estimation, we design a grid-based visual-grounding prompt with a *propose-validate* strategy, maximizing VLM's spatial understanding and yielding physically and semantically valid canonical region (i.e., ellipses). For the refinement, we locally adapt the region to surrounding environment without over-relaxed to free space. We construct a new space-grounding benchmark and compare C2F-Space with five state-of-the-art baselines using success rate and intersection-over-union. Our C2F-Space significantly outperforms all baselines. Our ablation study confirms the effectiveness of each module in the two-step process and their synergistic effect of the combined framework. We finally demonstrate the applicability of C2F-Space to simulated robotic pick-and-place tasks.

Keywords: Space Grounding, Vision-Language Model, Coarse-to-fine, Spatial Reasoning

1 Introduction

Space grounding refers to the process of mapping linguistic expressions to spatial regions within an environment [19]. 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') [18]. Subsequent approaches support compositional expressions [16,

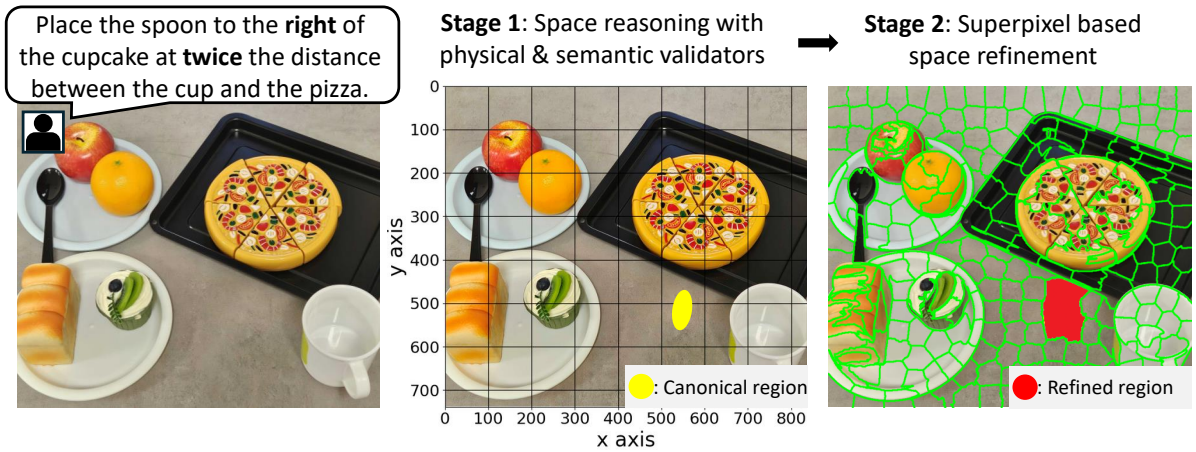

Fig. 1: Illustration of the two-stage space grounding result produced by the proposed C2F-SPACE. The grid-guided prompt enables the VLM to generate a coarse region proposal (e.g., an ellipsoid) through spatially multiplicative reasoning. A superpixel-based enhancement process then refines this proposal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

51]. Recently, Kim et al. [19] 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 to enhance spatial reasoning. Notable examples include ROBOPOINT [48], which fine-tunes a VLM to localize target regions as coarse point sets, and recent VLMs, such as Molmo [11] and Gemini 2.5 [9], demonstrate zero-shot 2D point grounding [8]. Nevertheless, the coarse, point-level nature of these outputs often lacks the rigorous spatial precision required for downstream applications, particularly in fine-grained robotic manipulation. Although visual prompting methods provide more detailed guidance and complement text instructions [4, 39], they typically rely on coarse visual markers (e.g., circles), which limit their fine-grained spatial reasoning capability.

Therefore, we propose C2F-SPACE, a novel coarse-to-fine space-grounding method that infers semantically and geometrically valid space given spatial instructions. Our method consists of two stages: the first stage enables the VLM to reason about approximate but spatially complex relationships —such as geometric relations, relational distances, and reference uniqueness—through grid-based inpainting image prompting. This then proposes a set of ellipsoidal region candidates, validating them to ensure both physical feasibility for placement actions and semantic consistency with sub-components of the spatial instruction. In the second stage, C2F-SPACE locally adapts the coarse candidates to precisely align with the surrounding environment using superpixelization, while avoiding over-relaxed regions that extend into free space.

We evaluate C2F-SPACE on a newly constructed space-grounding benchmark consisting of 350 problems. The benchmark includes spatial instructions featuring diverse relational distances and reference ambiguities, such as single- and multi-hop relations as well as unique and non-unique references—for example, "below the red bottle by half the distance between the basket and the banana." Our proposed C2F-SPACE outperforms state-of-the-art spatial grounding baselines [11, 19, 38, 48] and vision-language models [11, 33], complementing the grounding process of o4-mini [33]. Finally, we demonstrate the applicability of C2F-SPACE in simulated robotic pick-and-place tasks within PyBullet [10] environments.

Our key contributions are as follows:

- We introduce a grid-guided space-reasoning prompt, which allows a VLM to propose and validate physically feasible and semantically consistent region candidates.
- We provide a superpixel-based refinement module that adjusts regional candidates for fine-grained alignment with the surrounding environment, while preventing overextension into free space.
- We construct a new space-grounding benchmark of 350 examples consisting of diverse challenging instructions performing extensive comparisons with state-of-the-art spatial grounding baselines.

2 Related Work

Spatial Understanding: Spatial understanding is a fundamental capability for intelligent machines, enabling downstream decision-making and execution in real-world environments. Early approaches employ rule-based mapping, which directly associates spatial expressions with sensory patterns to recognize environmental layouts [40, 50]. Studies then adopt discriminative probabilistic models, such as support vector machines, to map spatial description with visual observations [21, 23]. However, these methods capture only predefined spatial relations and fail to represent the diversity necessary for comprehensive scene understanding [28].

Researchers then adopt large language models (LLMs) in combination with visual inputs, forming vision-language models (VLMs) that learn spatial semantics from diverse, large-scale multimodal datasets [29, 34, 43]. VLMs provide open-world multimodal understanding, making them applicable to a broad range of downstream tasks, such as image-text retrieval [6], zero-shot visual question answering [25], and segmentation [22]. However, early VLMs fail spatial reasoning since they behave as a bag-of-tokens, which lose positional detail [5, 24, 49]. To overcome the limitation, recent approaches integrate depth features into VLMs to provide scale cues [7]. Furthermore, fine-tuning VLMs using extensive spatial relation annotations improves reasoning over complex object interactions in diverse scenes [41, 48]. Alternatively, without direct modification of existing VLMs, our method guides a VLM with structured visual and textual prompts, enhancing its ability to understand spatial expressions and predict the target space described by the instruction.

Spatial grounding: Spatial grounding methods map linguistic spatial expressions to physical points or areas introducing various representations. Early approaches manually associate predefined predicates with structured representations, such as potential fields [42] or fuzzy spatial membership functions [2, 44]. To overcome the limitations of these hand-crafted mappings, neural network-based methods predict either pixel coordinates for target placement [46] or dense probability maps over the image [30, 38]. To capture more complex distributions, recent studies adopt parameterized probability representations, such as Gaussian mixture models [51] and Boltzmann energy functions [16]. Extending to spatiotemporal compositional reasoning, LINGO-Space models the target space via a Bayesian update of polar distributions [19].

As VLMs demonstrate generalization across diverse tasks [3, 37], researchers have begun applying them to spatial grounding [48]. These models enable direct prediction of goal points grounded in visual scenes from natural language instructions. RoboPoint [48], for example, finetunes VLMs on spatial phrases to predict 2D keypoints, translating relational commands into precise spatial points. Recent VLMs, such as Molmo [11] and Gemini 2.5 [9], further demonstrate zero-shot point prediction from language instructions. However, their point-based outputs remain coarse and lack fine-grained spatial precision. Our method enhances this limitation by introducing a coarse-to-fine inference framework that returns fine-grained regions for robotic task executions.

Space representation: Space representation plays a crucial role at the intersection of spatial grounding and robotic task execution. Early robotic studies model space using discrete grid or volumetric maps to capture the geometry of free or occupied regions [14, 45]. Researchers then introduce continuous representations, such as Euclidean or truncated signed distance fields to capture smoother geometric relationships [31, 32]. Recent studies augment these field representations with object-level semantics through

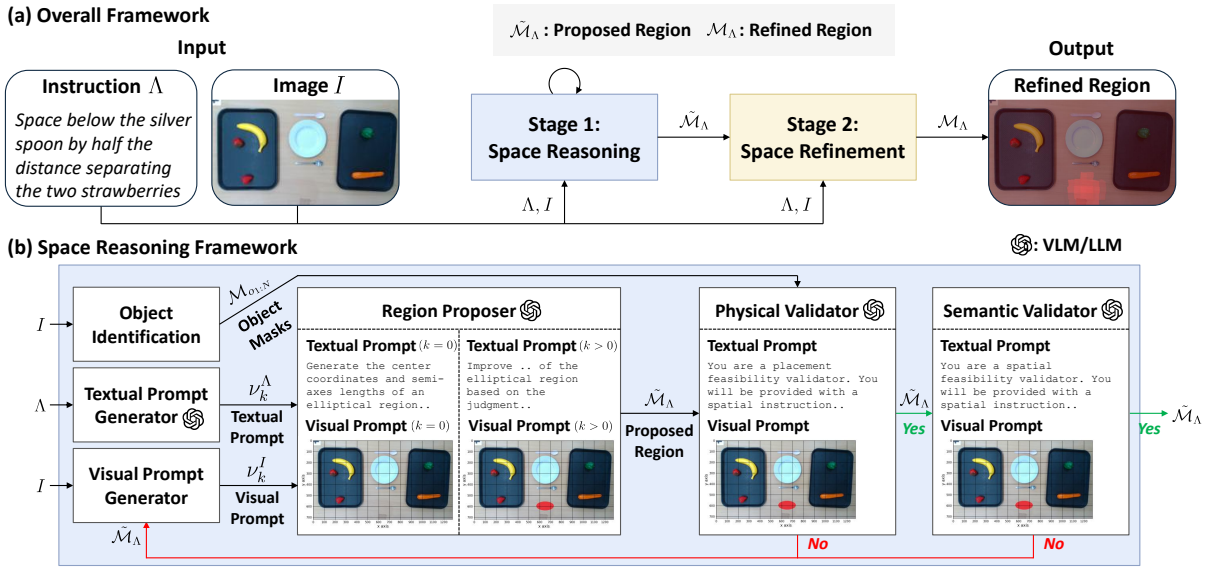

Fig. 2: (a) Overall framework of C2F-SPACE. Given an instruction Λ and an image I, the space-reasoning module proposes a candidate region $\tilde{\mathcal{M}}_\Lambda$, and the subsequent space-refinement module adapts this proposal to produce the precise region $\mathcal{M}_\Lambda$. (b) The space-reasoning module iteratively proposes and validates a canonical spatial region $\tilde{\mathcal{M}}_\Lambda$. At each iteration k, the VLM-based proper predicts an elliptical region guided by the textual prompt ν_k^Λ and the visual prompt ν_k^I, where the image I is inpainted by a spatial grid. Then, two validators subsequently ensure that the proposed region is both collision-free and semantically consistent with the instruction Λ. Note that the object identification module runs once beforehand to generate object masks used during the validation process.

semantic mapping [15, 17], although precise associations near region boundaries remain challenging. Alternatively, researchers employ superpixelization [1] to group homogeneous pixels into region-based representations that preserve sharp regional boundaries and coherent characteristics [47, 52]. Building on this idea, our method introduces learning-based superpixelization that refines VLM-generated coarse predictions in accordance with the surrounding environmental context for accurate robotic perception and mapping.

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 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 two steps: 1) *space reasoning*, and 2) *space refinement*. This modular design improves grounding accuracy and reliability while mitigating hallucinations in the VLM.

In detail, the *space-reasoning* step 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 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) 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 [36], an open-set object-mask identifier that first detects object bounding boxes using Grounding DINO [26], and then extracts the corresponding masks using SAM [20], conditioned on the detected boxes.

2) 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) *placement feasibility guidance*——ensuring that the predicted region is feasible for object placement as shown in Fig. 3. 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.

> Generate the center coordinates and semi-axis lengths of an elliptical region within an image, based on the provided overhead image and spatial instructions. The regions should be compact, accurate, and feasible to place.
> ...
> # Steps
> 1. **Extract Information**: Identify relevant objects and their extents from the image and spatial instructions.
> 2. **Generate Regions**: Determine suitable coordinates for elliptical regions following the given spatial instructions.
> 3. **Ensure Feasible Placement**: Verify that the generated coordinates are placed in valid, unobstructed locations according to the feasibility 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.
> Spatial instruction: ...

Fig. 3: A capture of the region proposal prompt for $k = 0$

3) 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.

4) 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 as shown in Fig. 4.

> You are a placement feasibility 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.
> ...
> # Judgment Guidelines
> - Describe your reasoning for each question in detail, then make a judgment.
> - Do not include any suggestions for improvement in your response.
>
> # Questions to be answered: 1. **Placement Feasibility Check**: Find out objects overlap with the predicted region. If overlap with the objects is permitted, then answer "pass"; otherwise, answer "fail".
>
> Answer each question by referencing the image and spatial instructions. After your reasoning for each question, include a "pass" or "fail" judgment for clear assessment.

Fig. 4: A capture of the physical validator prompt

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 as in Fig. 5. To improve accuracy, we instruct the VLM to decompose compositional instructions and validate each sub-component 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.

> You are a spatial feasibility 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 placement feasibility test and it has been verified that it is feasible to place and does not overlap with any object it should avoid.
> ...
> # Judgment Guidelines
> - Describe your reasoning for each question in detail, then make a judgment.
> - Do not include any suggestions for improvement in your response.
>
> # How to check if the 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]
>
> 2) **Spatial validation of the red elliptical region for each segment:** For each segment, check whether the position of the red elliptical region satisfies it. Use the spatial guidelines to validate the position of the region for each segment. ...

3) If the object mentioned in the instruction has multiple instances in the image, carefully verify that the red elliptical region follows the spatial instruction relative to the correct instance of that object.

Questions to be answered:
1. **Spatial Compliance**: Determine whether the region visually satisfies the spatial instruction provided. Describe your reasoning. Use the "spatial guidelines" and "how to check if red region is satisfying the instruction" to reach a conclusion.

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.

Fig. 5: A capture of the semantic validator prompt

3.3 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

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

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 [1]. We then assign a pseudo logit to each superpixel in two steps:

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

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 [35], 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. 6: (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*

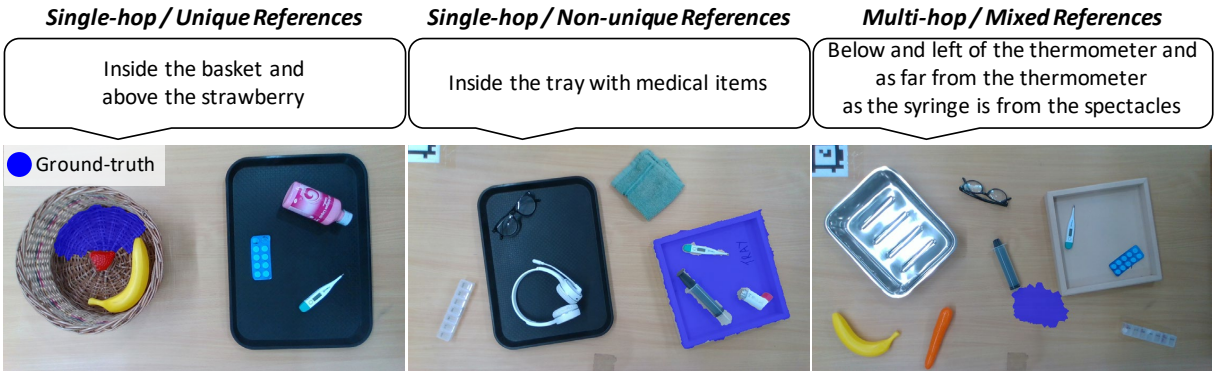

Fig. 6: Examples in the proposed space-grounding benchmark. Each example includes an instruction-image pair with the corresponding ground-truth space label (blue). For labeling consistency and efficiency, human annotators select superpixel-based regions directly on the image.

The first category, *single-hop space reasoning with unique references*, includes instructions containing one to four spatial expressions with clearly identifiable reference objects. These cases require only direct interpretation of spatial terms and localization of the reference objects, as illustrated by the example "inside the basket and above the strawberry" in the first column of Fig. 6.

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. 6.

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 by twice the distance between the inhaler and the strawberry," 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 five space-grounding baselines:

- **CLIPort** [38]: A language-conditioned imitation learning approach that predicts a pixel-level target location for pick-and-place manipulation using CLIP-based [34] image-language encoding. We disable rotational augmentation and modify the loss function to incorporate a mask as the ground-truth target.
- **LINGO-Space** [19]: A probabilistic space-grounding method that incrementally estimates the spatial distribution of target regions from composite referring expressions using configurable polar distributions. We modify the loss function to incorporate a mask as the ground-truth target instead of a single point.
- **RoboPoint** [48]: A VLM fine-tuned with synthetic instructions to predict keypoint affordances from language, including spatial descriptions. RoboPoint predicts multiple keypoints

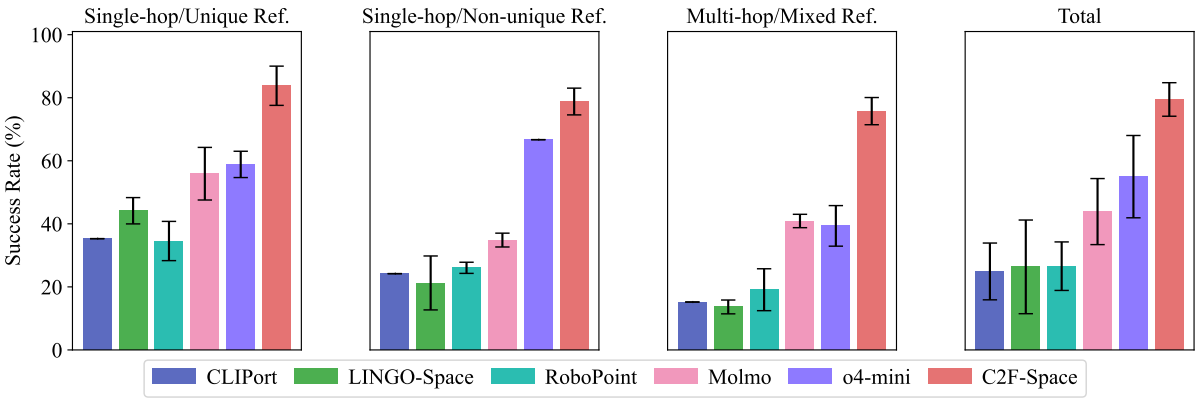

Fig. 7: Performance comparison in terms of success rate [%]. The 'Total' denotes the average across three instruction categories. The error bar denotes standard deviation.

for space grounding. We use its pre-trained model and evaluate performance by computing the success rate, defined as the ratio of predicted points within the ground-truth mask. Note that we do not measure IoU for RoboPoint.

- Molmo 72B [11]: An open-source VLM designed for multimodal reasoning, capable of 2D pointing to localize and reference specific regions within images. We adopt the point-based prompting strategy from RoboPoint and do not measure IoU as RoboPoint.
- **GPT-o4** mini [33]: A GPT-o4 mini with a point-based prediction prompt as RoboPoint and Molmo.

We apply Otsu's thresholding to convert a predicted probability into a binary mask for both our method and 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 method.

4.3 Implementation details

For object identification, we use Grounding DINO-B [26], which combines a SWIN-B [27] visual backbone with a BERT-base-uncased [12] text backbone, together with SAM employing a ViT-B [13] backbone. We prompt Grounding DINO with the query 'object, objects' to extract bounding boxes and subsequently use these boxes to prompt SAM. For grid-guided space reasoning, we employ the 'o4-mini-2025-04-16' model across all tasks. We limit the maximum number of iterations to two, as additional iterations do not yield noticeable performance improvements in our experiments. During evaluation, we enable automatic retries when API-related errors occur, such as missing parsing results.

5 Evaluation

Benchmark evaluation. We evaluate our method and baselines on the proposed space-grounding benchmark. As shown in Fig. 7- 9, our approach consistently outperforms all baselines in both success rate and IoU across all instruction categories. Although built on o4-mini, the gains over the o4-mini backbone indicate that grid-based visual prompting and superpixel-based refinement substantially strengthen grounding performance. While large-scale VLMs (i.e., Molmo and o4-mini) achieve similarly high success rates exceeding 55% on *single-hop space reasoning with unique references* task, Molmo's performance drops markedly with non-unique references. This indicates o4-mini maintains stable performance by leveraging stronger reasoning to infer implicit contextual cues that resolving referential ambiguity requires robust reasoning capabilities. These findings justify our choice of o4-mini as the foundation of the *spatial reasoning* module.

In contrast, small models exhibit poor performance due to limited reasoning capacity. The small-scale VLM, RoboPoint, achieves significantly lower success rates, not only due to its weak reasoning ability but also since it learns

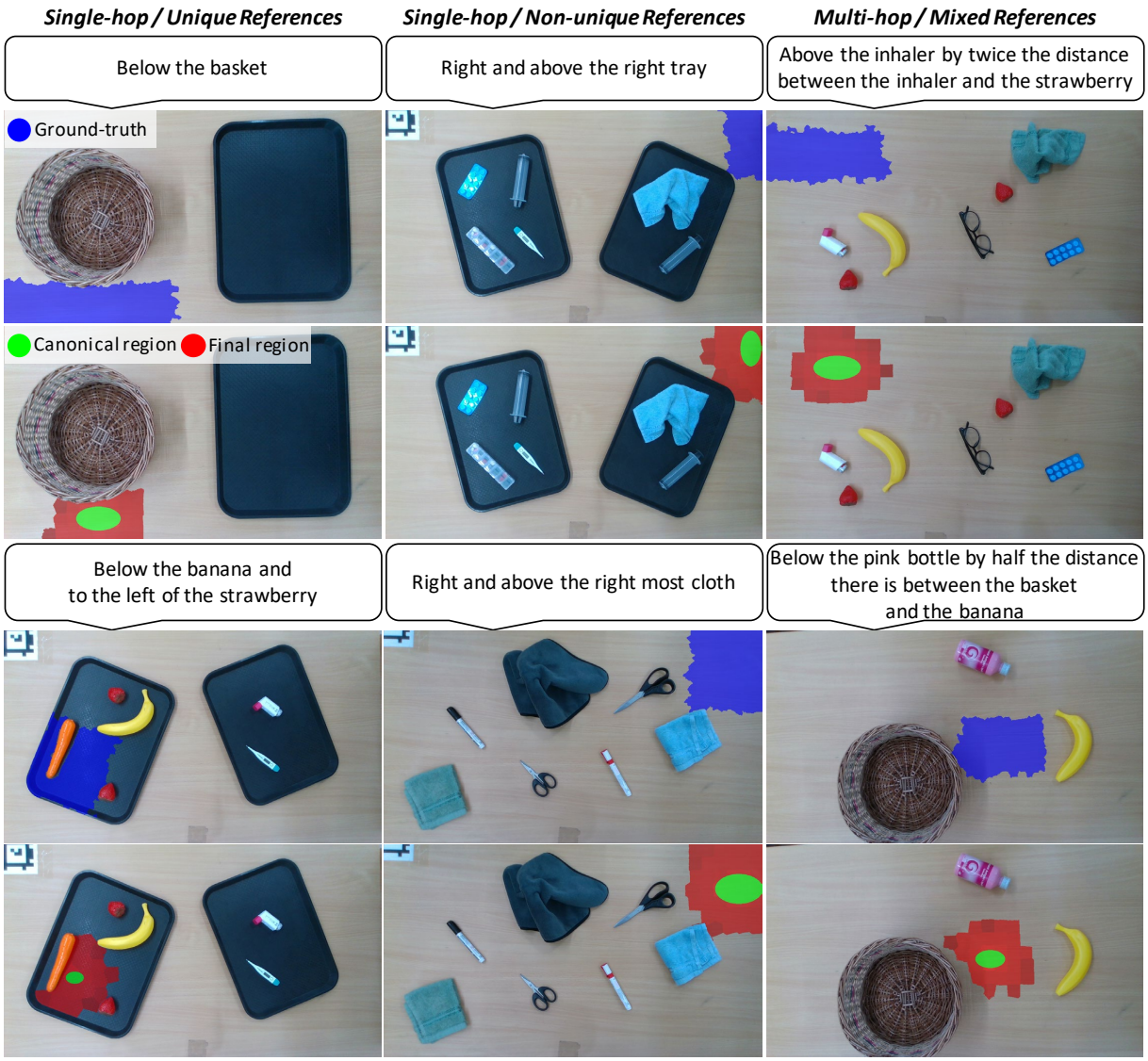

Fig. 8: Demonstrations of C2F-SPACE predictions across six space-grounding scenarios. For each linguistic instruction, the region proposer generates a canonical region (green), while the refinement module adapts it to the surrounding environment without over-relaxation using superpixelization (red). The blue regions indicate ground-truth regions. Note that we measure the distance between the centers of two referenced objects.

from a predefined set of spatial relations, failing to generalize to unseen concepts such as 'far.' Similarly, other learning-based models, CLIPORT and LINGO-Space, also show lower success rates due to limited generalization to novel relations and objects, as they rely on CLIP's limited visual–textual representations.

The results for *multi-hop space reasoning* particularly demonstrate the effectiveness of the grid-inpainting reasoning and propose-validation loop in C2F-SPACE. While C2F-SPACEachieves approximately 76% success, all baselines fall below 50%, as this task require understanding inter-object relationships and contextual dependencies. The grid overlaid on the image provides an explicit

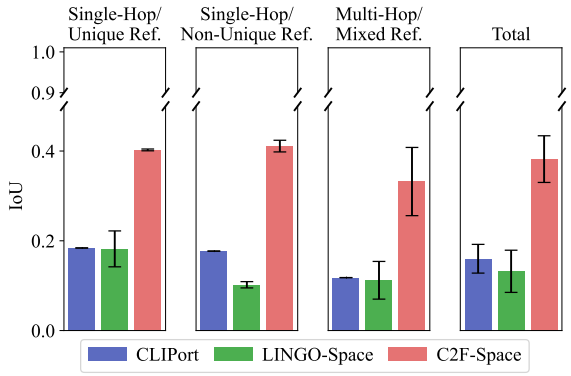

Fig. 9: Performance comparison in terms Intersection over Union (IoU). The 'Total' denotes the average across three instruction categories. The error bar denotes standard deviation.

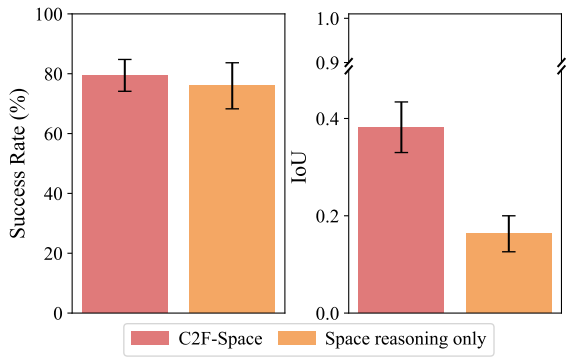

Fig. 10: Ablation study evaluating the effectiveness of the superpixel-based space refinement module in C2F-SPACE. We compare C2F-SPACE with 'space reasoning only' variant (i.e., without the refinement module) in terms of success rate [%] and IoU.

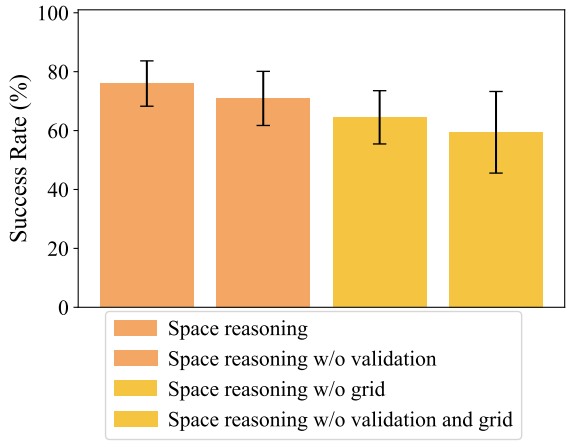

Fig. 11: Ablation study evaluating the effectiveness of the grid-guided *space reasoning* module. We assess performance degradation when removing either the validation component, the grid-based visual prompts, or both.

spatial structure, enabling o4-mini to estimate the relative distances between objects and free spaces. Moreover, the validation loop in C2F-SPACEiteratively refines prior predictions using visual feedback. Together, these modules establish a robust multi-hop spatial reasoning mechanism that enables our method to succeed in tasks where other baselines fail.

Fig. 9 shows C2F-SPACE's superior space generation performance, achieving an average IoU of 0.382 across categories—more than twice that of the second-best baseline, CLIPORT (0.160). This high IoU results not only from semantically accurate region predictions but also from fine-grained superpixelization. As shown in the first and second columns of Fig. 8, the refinement module effectively captures object boundaries, enhancing spatial precision. Moreover, as space grounding inherently lacks well-defined object boundaries, the IoU may remain relatively low (e.g., 0.493), even when the grounded region is qualitatively reasonable, as demonstrated in the first column of Fig. 8.

Ablation studies. We conduct ablation studies to evaluate the impact of superpixel-level refinement. The success rate remains comparable with or without this refinement, as shown in Fig. 10, indicating that reasoning is primarily based on the VLM module. Including the superpixel-level refinement improves IoU by more than twice, resulting in the highest IoU among all baselines. As shown in the bottom rows of Fig. 8, the refinement module adjusts the prediction without overlapping objects.

We further analyze the contribution of each component in the grid-guided space reasoning module, focusing on the grid-based visual prompt and the validation step. As shown in Fig. 11, removing the validation step and its corresponding iteration results in a slight performance drop of approximately 5% in success rate. In contrast, removing the grid-based visual prompt causes a

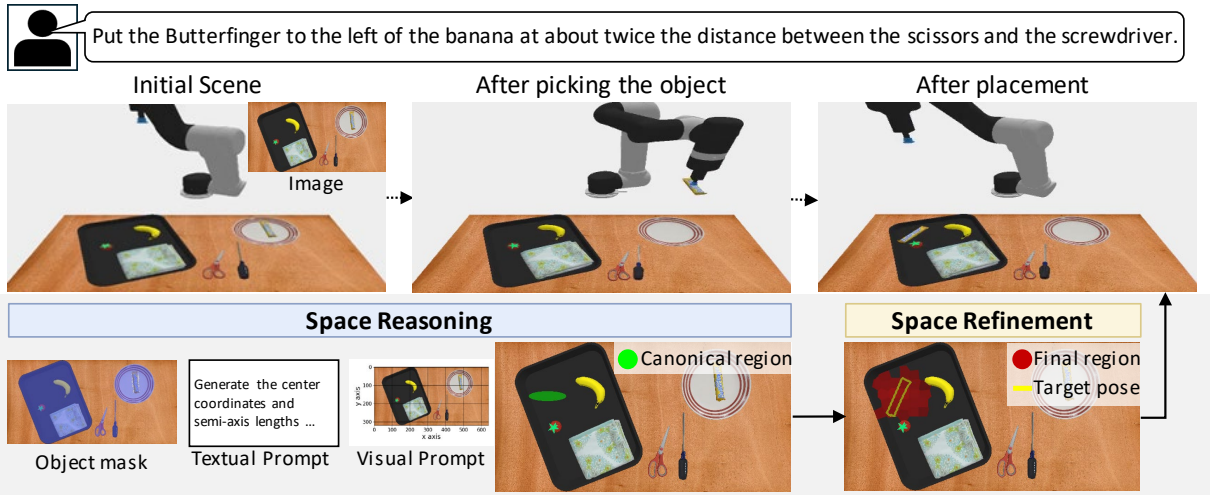

Fig. 12: A pick and place task demonstration in PyBullet simulator. Given the *multi-hop* instruction and the top-down view scene image, C2F-Space indentifies space occupied by objects, reasons space using VLMs, and then refines the generated region. Note that we measure the distance between the centers of the two referenced cloths.

substantial decrease of 11.5%, and eliminating both components leads to an additional 5.0% decline. These findings underscore the crucial role of grid-guided visual prompting in enabling effective space grounding, as it provides essential spatial structure for the VLM's reasoning process.

Simulation demonstrations. We finally demonstrate the applicability of C2F-Space to robotic manipulation tasks using the PyBullet-based simulator with UR5 manipulator from CLIPort [38]. Fig. 12 presents a demonstration sequence along with the intermediate outputs of each module, given a human instruction and a top-down image.

First, the *space-reasoning* module predicts a canonical region (green) by reasoning through o4-mini. The subsequent *space-refinement* module then produces fine-grained superpixels (red) with associated prediction scores above a certain threshold. We use the predicted superpixels as a spatial mask to sample the optimal placement position of the target object, maximizing the intersection-weighted score sum within the mask. After identifying the target object and placement position, the simulated robot grasps the object specified in the instruction and places it at the predicted location using a predefined action primitive. As shown in Fig. 12, the robot successfully positions the target object in the instructed spatial region.

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 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 *grid-inpainting* prompting technique. Then, it locally refines this coarse prediction into a precise region using superpixel-level decisions. This design allows for 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 *grid-inpainting* prompt with the propose-validate strategy 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. We finally demonstrate the practical applicability of C2F-Space in robotic manipulation tasks

through simulation, where the robot successfully places objects according to spatial instructions.

Acknowledgements. This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea Technology (MSIT) (No. RS-2022-II220311 and RS-2024-00336738), by the National Research Council of Science & Technology (NST) grant by the MSIT (No. GTL25041-000), Artificial intelligence industrial convergence cluster development project funded by the MSIT & Gwangju Metropolitan City. We thank Mr. Mohd. Nadir, Mr. Aman Tambi, Mr. Sandeep Zachariah and Mr. Ribhu Vaypei for contributing towards the Franka Emika Robot test bed setup, data collection/processing and system development/CAD modeling support during the early phase of the project. Rohan Paul acknowledges research grant support from the ICMR-National Center for Assistive Health Technology (NCAHT), Govt. of India as project: RP04267 under School of IT (SIT), IIT Delhi, Hardware support from the Equipment Matching Grant from Industrial Research & Development (IRD) Unit at IIT Delhi and administrative support from Mr. Shailendra Negi and Ms. Sunita Negi.

References

[1] Achanta R, Shaji A, Smith K, et al (2019) Slic superpixels. Tech. rep.

[2] Bloch I, Saffiotti A (2003) On the representation of fuzzy spatial relations in robot maps. In: Intelligent systems for information processing. Elsevier, p 47–57

[3] Brohan A, Chebotar Y, Finn C, et al (2023) Do as i can, not as i say: Grounding language in robotic affordances. In: Proceedings of the Conference on Robot Learning (CoRL), PMLR, pp 287–318

[4] Cai M, Liu H, Mustikovela SK, et al (2024) Vip-llava: Making large multimodal models understand arbitrary visual prompts. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12914–12923

[5] Chen B, Xu Z, Kirmani S, et al (2024) Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 14455–14465

[6] Chen X, Wang X, Changpinyo S, et al (2023) Pali: A jointly-scaled multilingual language-image model. In: Proceedings of the International Conference on Learning Representation (ICLR)

[7] Cheng AC, Yin H, Fu Y, et al (2024) Spatialrgpt: grounded spatial reasoning in vision-language models. In: Conference on Neural Information Processing Systems (NeurIPS), pp 135062–135093

[8] Cheng L, Duan J, Wang YR, et al (2025) Pointarena: Probing multimodal grounding through language-guided pointing. arXiv preprint arXiv:250509990

[9] Comanici G, Bieber E, Schaekermann M, et al (2025) Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:250706261

[10] Coumans E, Bai Y (2016) Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org

[11] Deitke M, Clark C, Lee S, et al (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, pp 91–104

[12] Devlin J, Chang MW, Lee K, et al (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), pp 4171–4186

[13] Dosovitskiy A, Beyer L, Kolesnikov A, et al (2021) An image is worth 16x16 words: Transformers for image recognition at scale. In: Proceedings of the International Conference on Learning Representation (ICLR)

[14] Elfes A (2002) Using occupancy grids for mobile robot perception and navigation. Computer 22(6):46–57

[15] Garg S, Sünderhauf N, Dayoub F, et al (2020) Semantics for robotic mapping, perception and interaction: A survey. Foundations and Trends® in Robotics 8(1–2):1–224

[16] Gkanatsios N, Jain A, Xian Z, et al (2023) Energy-based models are zero-shot planners for compositional scene rearrangement. In: Proceedings of Robotics: Science and Systems (RSS)

[17] Grinvald M, Furrer F, Novkovic T, et al (2019) Volumetric instance-aware semantic mapping and 3d object discovery. IEEE Robotics and Automation Letters (RA-L) 4(3):3037–3044

[18] Jain K, Chhangani V, Tiwari A, et al (2023) Ground then navigate: Language-guided navigation in dynamic scenes. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 4113–4120

[19] Kim D, Oh N, Hwang D, et al (2024) Lingospace: Language-conditioned incremental grounding for space. In: Proceedings of the National Conference on Artificial Intelligence (AAAI), pp 10314–10322

[20] Kirillov A, Mintun E, Ravi N, et al (2023) Segment anything. In: Proceedings of the International Conference on Computer Vision (ICCV), pp 4015–4026

[21] Kulkarni G, Premraj V, Ordonez V, et al (2013) Babytalk: Understanding and generating simple image descriptions. IEEE transactions on pattern analysis and machine intelligence(TPAMI) 35(12):2891–2903

[22] Lai X, Tian Z, Chen Y, et al (2024) Lisa: Reasoning segmentation via large language model. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9579–9589

[23] Lan T, Yang W, Wang Y, et al (2012) Image retrieval with structured object queries using latent ranking svm. In: Proceedings of the European Conference on Computer Vision (ECCV), Springer, pp 129–142

[24] Li C, Zhang C, Zhou H, et al (2024) Topviewrs: Vision-language models as top-view spatial reasoners. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp 1786–1807

[25] Li J, Li D, Savarese S, et al (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), PMLR, pp 19730–19742

[26] Liu S, Zeng Z, Ren T, et al (2024) Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In: Proceedings of the European Conference on Computer Vision (ECCV), Springer, pp 38–55

[27] Liu Z, Lin Y, Cao Y, et al (2021) Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the International Conference on Computer Vision (ICCV), pp 10012–10022

[28] Lu C, Krishna R, Bernstein M, et al (2016) Visual relationship detection with language priors. In: European conference on computer vision, Springer, pp 852–869

[29] Lu J, Batra D, Parikh D, et al (2019) Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. Conference on Neural Information Processing Systems (NeurIPS) 32

[30] Mees O, Emek A, Vertens J, et al (2020) Learning object placements for relational

instructions by hallucinating scene representations. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp 94–100

[31] Oleynikova H, Millane A, Taylor Z, et al (2016) Signed distance fields: A natural representation for both mapping and planning. In: Proceedings of Robotics: Science and Systems (RSS), University of Michigan

[32] Oleynikova H, Taylor Z, Fehr M, et al (2017) Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 1366–1373

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

[34] Radford A, Kim JW, Hallacy C, et al (2021) Learning transferable visual models from natural language supervision. In: Proceedings of the International Conference on Machine Learning (ICML), PMLR, pp 8748–8763

[35] Rampášek L, Galkin M, Dwivedi VP, et al (2022) Recipe for a general, powerful, scalable graph transformer. Conference on Neural Information Processing Systems (NeurIPS) 35:14501–14515

[36] Ren T, Liu S, Zeng A, et al (2024) Grounded sam: Assembling open-world models for diverse visual tasks. arXiv preprint arXiv:240114159

[37] Shah D, Osiński B, Levine S, et al (2023) Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In: Proceedings of the Conference on Robot Learning (CoRL), PMLR, pp 492–504

[38] Shridhar M, Manuelli L, Fox D (2022) Cliport: What and where pathways for robotic manipulation. In: Proceedings of the Conference on Robot Learning (CoRL), PMLR, pp 894–906

[39] Shtedritski A, Rupprecht C, Vedaldi A (2023) What does clip know about a red circle? visual prompt engineering for vlms. In: Proceedings of the International Conference on Computer Vision (ICCV), pp 11987–11997

[40] Skubic M, Perzanowski D, Blisard S, et al (2004) Spatial language for human-robot dialogs. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 34(2):154–167

[41] Song CH, Blukis V, Tremblay J, et al (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, pp 15768–15780

[42] Stopp E, Gapp KP, Herzog G, et al (1994) Utilizing spatial relations for natural language access to an autonomous mobile robot. In: Proceedings of the German Annual Conference on Artificial Intelligence, Springer, pp 39–50

[43] Tan H, Bansal M (2019) Lxmert: Learning cross-modality encoder representations from transformers. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp 5100–5111

[44] Tan J, Ju Z, Liu H (2014) Grounding spatial relations in natural language by fuzzy representation for human-robot interaction. In: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp 1743–1750

[45] Thrun S (2003) Learning occupancy grid maps with forward sensor models. Autonomous robots 15(2):111–127

[46] Venkatesh SG, Biswas A, Upadrashta R, et al (2021) Spatial reasoning from natural language instructions for robot manipulation. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp 11196–11202

[47] Yang M, Xiong J, Li Y (2024) Rgb-d visual odometry by constructing and matching features at superpixel level. Robotica

42(8):2619–2634

[48] Yuan W, Duan J, Blukis V, et al (2025) Robopoint: A vision-language model for spatial affordance prediction in robotics. In: Proceedings of the Conference on Robot Learning (CoRL), PMLR, pp 4005–4020

[49] Yuksekgonul M, Bianchi F, Kalluri P, et al (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)

[50] Zender H, Mozos OM, Jensfelt P, et al (2008) Conceptual spatial representations for indoor mobile robots. Robotics and Autonomous Systems 56(6):493–502

[51] Zhao Z, Lee WS, Hsu D (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), pp 11546–11553

[52] Zhu AZ, Mei J, Qiao S, et al (2023) Superpixel transformers for efficient semantic segmentation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 7651–7658