Perspective-Aware Reasoning in Vision-Language Models via Mental Imagery Simulation — Appendix

Phillip Y. Lee[1] Jihyeon Je[2] Chanho Park[1] Mikaela Angelina Uy[3]
Leonidas Guibas[2] Minhyuk Sung[1]
[1]KAIST [2]Stanford University [3]NVIDIA

In this appendix, we first discuss the limitations of our work and potential directions for future work (Sec. A). We then analyze the failure cases of two dense reconstruction-based baselines—SpatialPIN* [12] and ViewCrafter [18]—in allocentric reasoning scenarios (Sec. B). We describe the implementation details of our APC framework (Sec. C) and provide details on the evaluation setups (Sec. D). Finally, we provide the text prompts used in each stage of our method (Sec. E).

A. Limitations and Future Work

Our APC framework empowers VLMs with perspective-aware spatial reasoning, but its use of multiple vision foundation models [7, 10, 16] introduces additional memory usage compared with running the VLM alone. In our experiments, we ran inference on two NVIDIA RTX 3090 GPUs each with 24GB VRAM.

Moreover, the robustness of the scene abstraction stage depends on the accuracy its detection, segmentation, and orientation modules. To quantify each module's impact, we conducted an ablation study by substituting its predictions with the oracle ground truth. The left-hand table in Fig. A.1 shows the ablated accuracy on the *left/right* split of COMFORT++, while the right-hand figures depict representative failure cases. Injecting oracle information consistently improves performance, indicating that APC's accuracy is indeed influenced by each module's robustness, yet does not work as a critical bottleneck.

	Accuracy
APC-Vis	89.67
+ det./seg./depth	91.67
+ ori.	96.67
+ det./seg./depth/ori.	97.33

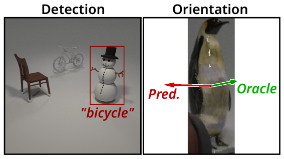

Figure A.1. **Ablation.** Accuracy of APC with oracle values (left), and failure cases of the vision modules (right).

While in this work we introduced a minimal yet effective form of 3D abstraction for perspective change in VLMs, exploring richer scene abstractions from images could offer an promising direction for future research—such as the use of 3D bounding boxes [1, 14, 17] and coarse, semantic 3D scene reconstructions [2, 3, 15].

B. Analysis on Dense Reconstruction Baselines

In this section, we further discuss the dense reconstruction-based baselines introduced in Sec. 4.1. In contrast to APC's abstraction-based approach, another intuitive approach for perspective-aware spatial reasoning is to perform a dense 3D reconstruction of the scene and then render a novel view from the target perspective. This new view can then be provided to the VLM instead of the visual prompt used in Sec. 3.3. We explore two such approaches that involve dense 3D reconstruction process: (1) a modified version of SpatialPIN [12], which directly lifts objects from the image into meshes and renders them from the target view, and (2) ViewCrafter [18], which synthesizes novel views by using an intermediate point cloud reconstruction. As the original SpatialPIN [12] does not include a rendering phase for novel target perspectives, we refer to our extended pipeline as SpatialPIN*. For the inference of SpatialPIN*, we used One-2-3-45 [8] in contrast to One-2-3-45++ [9] in the original paper due to the limited access of the API.

Method	SpatialPIN* [12]	ViewCrafter [18]	APC (Ours)
Time (s)	336.21	260.57	17.47

Table B.1. **Inference Time Comparison.** Both dense reconstruction-based baselines [12, 18] require over 14 times the inference time of our APC to answer a single question.

While a dense reconstruction-based approach may appear to be an obvious alternative to our abstraction-based framework, our experiments show that constructing an accurate and descriptive view of the target perspective is challenging and expensive. As illustrated in Fig. A.2, the synthesized novel views from both SpatialPIN* (row 1) and ViewCrafter (row 2) are often excessively noisy and fail to preserve the

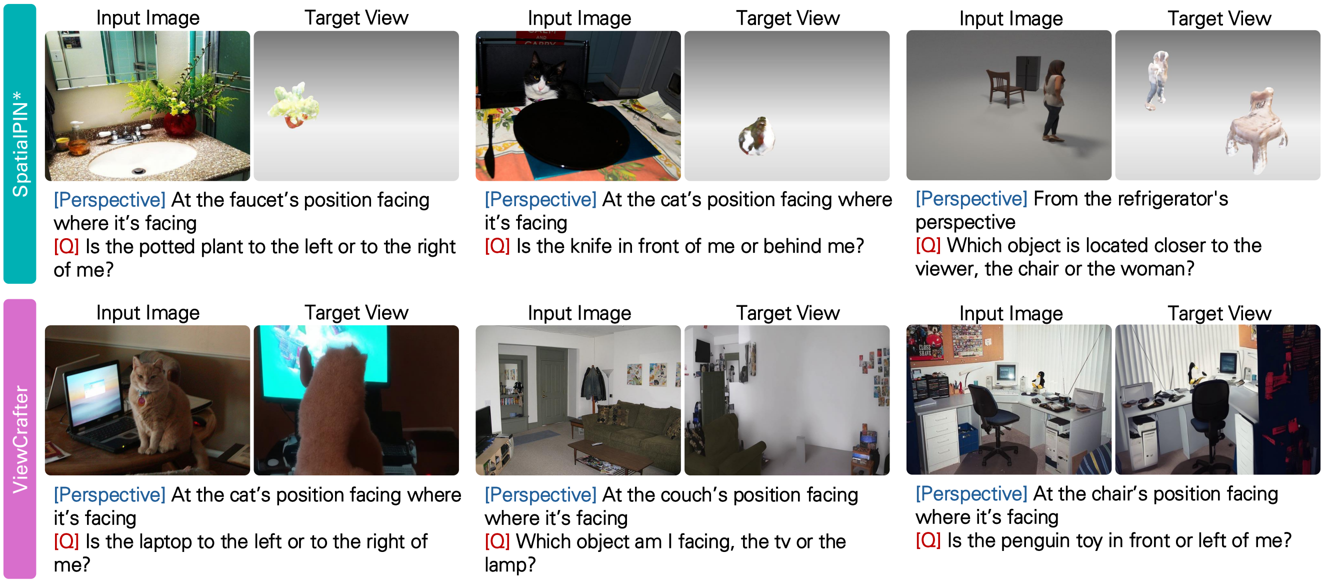
SpatialPIN*
ViewCrafter
Input Image
Target View
[Perspective] At the faucet's position facing where it's facing
[Q] Is the potted plant to the left or to the right of me?
[Perspective] At the cat's position facing where it's facing
[Q] Is the knife in front of me or behind me?
[Perspective] From the refrigerator's perspective
[Q] Which object is located closer to the viewer, the chair or the woman?
[Perspective] At the cat's position facing where it's facing
[Q] Is the laptop to the left or to the right of me?
[Perspective] At the couch's position facing where it's facing
[Q] Which object am I facing, the tv or the lamp?
[Perspective] At the chair's position facing where it's facing
[Q] Is the penguin toy in front or left of me?

Figure A.2. **Dense Reconstruction Baseline Examples.** Novel views synthesized by SpatialPIN* [12] and ViewCrafter [18] both display noisy and inaccurate objects and scene structures lacking the original context of the input image, thereby leading to low accuracy when VLMs are fed the images as a visual input for spatial reasoning.

context of the input image. Consequently, providing these reconstructed views to the VLM for spatial reasoning results in lower accuracy as previously shown in Tab. 1. In addition, both methods incur notably longer inference times due to the dense 3D reconstruction steps, as shown in Tab. B.1. In contrast, as in our APC, constructing an minimal abstraction of the scene with precise mappings between the original objects and their abstractions not only yields more accurate reasoning but also substantially reduces inference time.

C. Implementation Details

In this section, we provide the implementation details of our APC framework in Sec. 3. As the backbone VLM, we used Qwen2.5-VL-7B-Instruct[1].

C.1. Scene Abstraction

Detection Refinement with VLM. While GroundingDINO [10] excels in object detection, it often struggles when the input text prompt is complex. We add a simple refinement stage utilizing the VLM for improved detection accuracy. For each object description t_i we keep GroundingDINO's predicted candidates whose confidence exceeds a threshold s, then select the top k candidates. The corresponding image crops are laid out in a grid, and we query the VLM to select the crop that best matched t_i. We set $s = 0.15$ and $k = 5$. Fig. C.1 illustrates a case in which the initial GroundingDINO output is incorrect but is corrected by this refinement step.

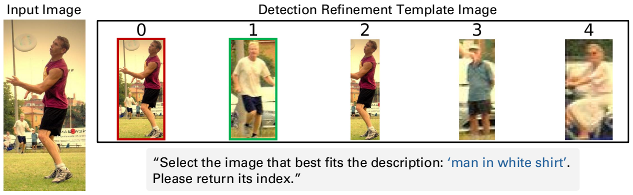
Input Image
Detection Refinement Template Image
0
1
2
3
4
"Select the image that best fits the description: 'man in white shirt'. Please return its index."

Figure C.1. **Detection Refinement Example.** Starting with candidate detections from GroundingDINO [10], we select the top k predictions and present them as a grid of cropped images (right). We then query the VLM to return the index that best aligns with the input text prompt. Red indicates GroundingDINO's initial choice and green indicates the refined choice.

Filtering Outliers. To obtain the 3D position of each object abstraction $O_i \in S_E$, we unproject the segmented pixels using the predicted depth map. To handle outliers caused by background pixels being included in the segmentation masks, we filter out the points whose depth values fall outside the range $[0.9d_i, 1.1d_i]$, where d_i is the mode depth within the mask. We then assign the coordinate-wise median of the remaining points in the remaining points as the 3D position c_i of object O_i.

C.2. Egocentric Rephrasing

Recall that our APC converts an *allocentric* question Q—originally stated with respect to a reference viewpoint A—into an *egocentric* one posed from A itself. To ensure compatibility with the perspective prompts introduced in Sec. 3.3, we remove the explicit perspective descriptions from Q. In practice, we query the VLM to rewrite Q, ex-

[1] https://huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct

cluding the phrases that mention a reference perspective. In turn we obtain a perspective-agnostic reformulation of the task, which is then used in each type of perspective prompt.

C.3. Visual Prompt Rendering.

To render a *visual prompt* from the transformed scene abstraction $S_A = \{O'_i\}_{i=1}^n$ as shown in Fig. 5, we use the Trimesh renderer [4]. Note that S_A is defined in the coordinate system of the reference perspective A. Each object O'_i is converted to an equal-sized cube with distinct colors, and the visual prompt is obtained by rendering the scene accordingly. Given the camera in S_A faces in the positive z-direction, only the objects with $z > 0$ appear in the visual prompt. Objects with $z \leq 0$ are considered to be out of view (*i.e.* not visible) from perspective A.

Normalization. To prevent cubes from appearing too small or large in the visual prompt, we normalize the coordinates of S_A, ensuring z values lie within a predefined range $[z_{\text{min}}, z_{\text{max}}]$. Likewise, we scale the x, y coordinates into a fixed range $[-d^*, d^*]$ to keep objects within the view frustum.

Camera Translation. By default, we place the camera at reference viewer's position—the origin of S_A. As an exception, for the *left/right* task in 3DSRBench [13], we shift the camera backward along the z-axis to ensure all objects in the scene appear in the visual prompt. This adjustment is applied to match the benchmark's setup, where an object that lies *behind* and *to the right* of a reference viewer is still treated as being on the right side from that viewer's perspective.

D. Evaluation Details

In this section, we provide further details on our evaluation setup in Sec. 4. Each VLM response is scored with a two-step process that combines *exact matching* and *LLM-assisted* evaluation. We first perform exact matching: if the response consists solely of the correct option index or the exact answer phrase, we label it as correct. Otherwise, we pass the entire response to an LLM along with the answer to determine its correctness. For this, we used the judgment prompt template from VLMEvalKit [6].

Following 3DSRBench [13], we employ CircularEval [11], which takes into account VLM's response consistency by permuting the answer options for each image-question pair. The VLM is considered to be correct for a question Q only if it selects the correct across all permutations. CircularEval is applied for both COMFORT++ [19] and 3DSRBench [13].

To construct the COMFORT++ benchmark for each task, we first collected 7 object meshes from the original implementation [19] and additional 6 meshes from Objaverse-XL [5]. For the *left/right* and *closer* tasks, we arranged three objects in a predefined layout, designating one as the reference viewer, and added random perturbations to the objects' x, y coordinates to diversify the layouts. We prepared 60 scenes and rendered each from 20 evenly spaced azimuth viewpoints. Then, we randomly sampled five views per scene, resulting in a total of 300 images for each task. For the *visibility* task, we created 160 scenes, each containing a reference viewer and single target object positioned so that the object is either visible or invisible from the viewer's perspective. We rendered each scene two opposite viewpoints, yielding 320 images. Finally, for the *facing* task, we arranged three objects in a linear configuration, setting the central object as the reference viewer, and oriented it to face either one of the two remaining objects. Each scene is rendered once, resulting in 300 images in total.

For 3DSRBench [13], we used the original *left/right* and *facing* criteria. We recasted the *front/behind* task as a *visibility* judgment for two reasons: (i) the provided task can be more naturally interpreted as deciding whether an object is visible from the reference object's viewpoint, and (ii) VLMs struggle to infer that an object is *behind* it when the object is not present in the image itself. This adjustment better serves our goal of measuring the egocentric and allocentric reasoning capabilities of VLMs.

E. Details on Text Prompts

In this section, we present the text prompts used at each stage of our `APC` pipeline. To guide the VLM towards the desired response format, we include examplar question-answer pairs for in-context learning. For the text prompt fed along with the visual prompt, we add simple prompt engineering to help suppress hallucinations: we (i) define the the relation *"facing towards"* and (ii) explicitly that the larger object is considered as being closer to the viewer—an assumption that holds since our abstraction assigns equal size to every object.

(1) Scene Abstraction (Sec. 3.1) — Extracting Objects of Interest.

```
# Situation Description
Given an image and a spatial-reasoning
question, identify all entities mentioned
in the question.

# Example
[Question] You are standing at the
airplane's position, facing where it
is facing.  Is the person on your left or
right?
[Detect] [airplane, person]

# Your Task
```

```
Now, given the question below, list the
entities that appear in the question.

[Question] {Question}
[Detect]
```

(2) Perspective Change (Sec. 3.2) — Setting a Reference Perspective

```
Given a question about spatial reasoning,
we want to extract the perspective of the
question.  If the question is from the
camera's perspective, return ++camera++.

# Example
[Question] From the woman's perspective,
is the tree on the left or right?
[Perspective] ++woman++

# Your Task
Given the question below, please specify
the perspective from which the question
is asked.
You must return in the format:
[Perspective] ++object_name++

[Question] {Question}
[Options] obj1, obj2, ..., camera
[Perspective]
```

(3) Egocentric Rephrasing (Sec. C.2)

```
From a sentence with a perspective
description, we need to remove the
perspective description.

# Example
[Question] From the car's perspective,
which is on the right side:  the person
or the tree?
[Output] Which is on the right side:  the
person or the tree?

# Your Task
Given the question below, please remove
the perspective description.

[Question] {Question}
[Output]
```

(4) Perspective Prompting (Sec. 3.3) — Visual Prompt.

```
This is an image of a 3D scene.

- The viewer is facing towards the object
that is closest to the center.
- A larger object is closer to the viewer
compared to a smaller object.

# Task
Based on the image, please answer the
following question.

{Question}

Please only return the answer.
```

(5) Perspective Prompting (Sec. 3.3) — Numerical Prompt.

```
Imagine that you are at the {src_obj}'s
position and facing where it is facing.
We have the coordinates of different
objects in {src_obj}'s coordinate system.

# Coordinate System
- The origin is at the {src_obj}'s
position.
- The {src_obj}'s facing direction is [0,
0, 1], which is aligned with the z-axis.
- The x-axis is to the right, the y-axis
is up, and the z-axis is forward.

# Object Coordinates
[...]

# Task
Given the above {src_obj}'s coordinate
system and the object coordinates, please
answer the following question:

[Question] {Question}
```

References

[1] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3d: A large benchmark and model for 3d object detection in the wild. In *CVPR*, 2023. 1

[2] Anh-Quan Cao and Raoul De Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, 2022. 1

[3] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3d scene reconstruction from a single rgb image. In *NeurIPS*, 2021. 1

[4] Dawson-Haggerty et al. trimesh. 3

[5] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. In *NeurIPS*, 2023. 3

[6] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *ACM MM*, 2024. 3

[7] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 1

[8] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. In *NeurIPS*, 2023. 1

[9] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *CVPR*, 2024. 1

[10] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, 2024. 1, 2

[11] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *ECCV*, 2024. 3

[12] Chenyang Ma, Kai Lu, Ta-Ying Cheng, Niki Trigoni, and Andrew Markham. Spatialpin: Enhancing spatial reasoning capabilities of vision-language models through prompting and interacting 3d priors. In *NeurIPS*, 2024. 1, 2

[13] Wufei Ma, Haoyu Chen, Guofeng Zhang, Celso M de Melo, Alan Yuille, and Jieneng Chen. 3dsrbench: A comprehensive 3d spatial reasoning benchmark. *arXiv preprint arXiv:2412.07825*, 2024. 3

[14] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *CVPR*, 2020. 1

[15] Yinyu Nie, Angela Dai, Xiaoguang Han, and Matthias Nießner. Learning 3d scene priors with 2d supervision. In *CVPR*, 2023. 1

[16] Zehan Wang, Ziang Zhang, Tianyu Pang, Chao Du, Hengshuang Zhao, and Zhou Zhao. Orient anything: Learning robust object orientation estimation from rendering 3d models. *arXiv*, 2024. 1

[17] Jin Yao, Hao Gu, Xuweiyi Chen, Jiayun Wang, and Zezhou Cheng. Open vocabulary monocular 3d object detection. *arXiv preprint arXiv:2411.16833*, 2024. 1

[18] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024. 1, 2

[19] Zheyuan Zhang, Fengyuan Hu, Jayjun Lee, Freda Shi, Parisa Kordjamshidi, Joyce Chai, and Ziqiao Ma. Do vision-language models represent space and how? evaluating spatial frame of reference under ambiguities. In *ICLR*, 2025. 3