One-Step Planner: Unified Observation and Decision-Making with Vision-Language Models

Youngjae Yoo[1,2], Jae-Woo Choi[1], Dohyung Kim[1*], and Byoung-Tak Zhang[2*]

***Abstract*— Most LLM-based embodied task planning systems adopt a modular pipeline that separates perception from planning: a vision model first produces a textual scene description, and a language model then determines the next action. This separation introduces an information bottleneck in which rich visual input is compressed into text, discarding spatial relationships and fine-grained appearance cues. Moreover, the decoupled architecture introduces redundant cross-modal processing, where visual information is encoded into text and then reinterpreted for planning, leading to increased latency and memory overhead. We present One-Step Planner, a vision-language model (VLM) that unifies perception and planning in a single forward pass. By applying Low-Rank Adaptation (LoRA), instruction tuning creates a direct perception-to-action pathway that sharpens spatial attention toward task-relevant objects. We evaluate on two partially observable benchmarks, WAH-NL++ (a modified version of WAH-NL) and ALFRED, and systematically compare the end-to-end architecture against modular pipelines equipped with four different observer types. One-Step Planner achieves up to 10% point higher task success rates while reducing GPU memory usage by up to 44.6%. Spatial-attention further shows that the model focuses on goal-critical regions.**

I. INTRODUCTION

Robust perception is a prerequisite for embodied agents operating in unstructured environments. Recent systems increasingly leverage vision-language models (VLMs) for end-to-end perception and action [1], yet many existing embodied agents, particularly those built on LLMs, still adopt a two-stage pipeline that separates perception from planning [2], [3]. In this design, a vision module first converts an egocentric image into a textual observation, which a language model then ingests to select the next action.

This modular separation introduces several limitations. First, converting rich visual input into text causes information loss, specifically discarding spatial relationships, object affordances, and fine-grained appearance cues. Second, the decoupled architecture weakens the direct alignment between visual inputs and action decisions, making it difficult to trace how specific visual evidence influences each action. Third, the decoupled design introduces redundant processing,

*This work was supported by the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. RS-2024-00336738, Development of Complex Task Planning Technologies for Autonomous Agents) and the National Research Council of Science & Technology(NST) grant by the Korea government(MSIT) (No. GTL25041-000))

*Corresponding authors.

[1]Social Robotics Lab, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea.

[2]Department of Computer Science and Engineering, Seoul National University, Seoul, Korea.

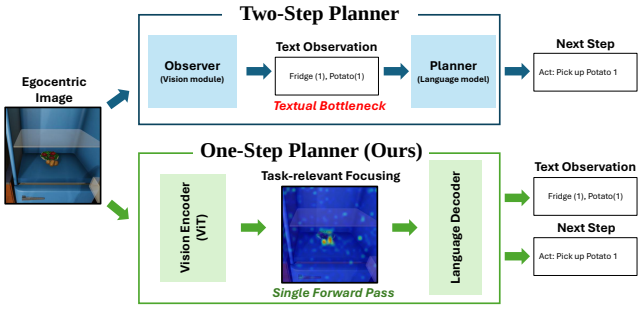

Fig. 1. **Two-step vs. One-step planning.** The two-step pipeline separates observation and planning into dedicated modules, while our One-Step Planner integrates them into a single VLM forward pass.

where visual information is first encoded into text and then reinterpreted for planning, increasing latency and memory consumption.

We propose **One-Step Planner**, a VLM-based architecture that integrates observation and action decision in a single forward pass (Fig. 1). By applying instruction tuning with LoRA [4] to both the visual encoder and the language decoder [5], we create a direct pathway from images to actions. This design mitigates the textual bottleneck, reduces inference cost, and, notably, allows us to inspect how tuning reshapes the model's spatial attention for rigorous perception.

We make the following contributions with a focus on perception robustness and interpretability:

- A **One-Step Planner**, that integrates observation and planning in a single forward pass, providing a lossless perception-to-action pathway with improved efficiency and a simplified pipeline.
- **Instruction tuning** with LoRA that concentrates the VLM's spatial attention on task-relevant objects, supported by **interpretability analysis** of spatial attention maps.
- Through experiments on **two partially observable benchmarks**, we demonstrate that the One-Step Planner achieves higher success rates with lower latency and memory cost than modular pipelines.

II. RELATED WORK

Unified perception-planning. Embodied task planning has widely relied on large language models, leveraging their strong reasoning and commonsense knowledge to decompose complex tasks into executable action sequences [2], [3]. But these text-only planners must depend on external environmental information, whether through visual grounding [3]

or simulator-provided states [6], introducing a modular perception boundary that we identify as a key fragility point. Vision-language models have been increasingly adopted to close this gap, progressing from large-scale multimodal reasoning [5] and hierarchical task decomposition [7] to generalist agents trained with supervised fine-tuning. However, these approaches typically require multi-stage pipelines, and often still decouple perception from decision-making.

Instruction tuning. Instruction tuning offers a lighter-weight alternative, adapting VLMs for temporal grounding [8], spatial reasoning [9], and visual chain-of-thought reasoning [10], yet each targets an individual capability in isolation. Our approach jointly tunes perception and planning within a single forward pass through LoRA, eliminating the modular boundary without requiring reinforcement learning or auxiliary action representations.

III. One-Step Planner

A. Preliminaries

At each time-step t, the agent receives an egocentric RGB image o_t, a natural-language task instruction I, and a belief buffer b_t that records all past observations, actions, and feedback:

$$b_t = [\hat{o}_0, a_0, f_0, \ldots, \hat{o}_{t-1}, a_{t-1}, f_{t-1}]. \tag{1}$$

Here, $\hat{o}_i$ is a text observation summarizing the image o_i at step i, a_i is the action taken, and f_i is the environment feedback (success flag, error codes, etc.). In a conventional two-step pipeline [2], an external observer E_ϕ first converts o_t into a textual observation:

$$\hat{o}_t = E_\phi(o_t), \tag{2}$$

which an LLM planner [2] then consumes to predict the next action.

B. Unified Observation-Action Pipeline

Our One-Step Planner, as illustrated in Figure 2, replaces the two-stage pipeline with a single VLM that jointly produces a textual observation $\hat{o}_t$ and a high-level action $\hat{a}_t$ in one decoding pass. At each time-step t, the multimodal prompt P_t is constructed as:

$$P_t = (P_{\text{sys}}, I, H, b_t), \tag{3}$$

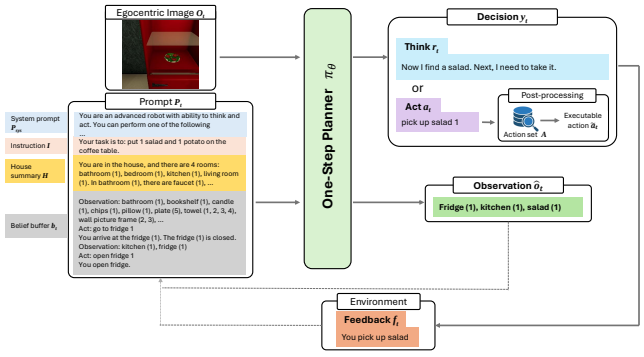

Fig. 2. Unified One-Step Planner Architecture. This fused perception-decision step removes intermediate text bridges, cutting latency and tightly coupling visual cues to action semantics.

where P_{sys} defines the ReAct-style [2] output structure with predefined atomic skills, I is the task instruction, and H contains environmental information, including the set of rooms and their associated receptacles (e.g., `coffee table`, `kitchen cabinet`). The model generates both outputs in a single forward pass:

$$(\hat{o}_t, \hat{a}_t) \sim p_{\text{VLM}}(\cdot \mid o_t, P_t), \quad \hat{a}_t \in \mathcal{L} \cup A, \tag{4}$$

where $\mathcal{L}$ denotes free-form reasoning clauses (e.g., *"Think: First I need to find a frying pan."*) and A denotes the set of environment actions (e.g., *"Act: pick up frying pan 1"*). At each timestep, the belief buffer is updated as $b_{t+1} = b_t \cup (\hat{o}_t, \hat{a}_t, f_t)$.

Post-processing for Action Matching. When $\hat{a}_t \in A$, we enumerate all valid skill–object combinations into a predefined action set $\mathcal{A}$, encode both $\hat{a}_t$ and each candidate using a sentence transformer [6], and select the candidate with highest cosine similarity: $\tilde{a}_t = \arg\max_{a \in \mathcal{A}} \text{sim}(\hat{a}_t, a)$. This ensures robustness to imprecise object names and minor formatting inconsistencies.

C. Instruction Tuning for Perception

We adapt a VLM using LoRA [4], while keeping all original weights frozen. This dual-side adaptation has two perception-critical effects. **(1) Vision-side adapters** specialize image embeddings toward task-relevant affordance cues, sharpening spatial attention to goal-critical regions. **(2) Decoder-side adapters** bias attention toward concise planner directives, ensuring that perceived visual evidence is immediately reflected in action selection. Together, these create a **lossless perception-to-action pathway** that avoids the information bottleneck of two-step methods and reduces hallucination risk.

IV. Experiments

A. Experimental Setup

Benchmarks. We evaluate on two partially observable embodied benchmarks. **WAH-NL++** [11], [12] is built on VirtualHome [13] and features 2,088 environments across 45 houses. We evaluate on 100 unseen test episodes. **ALFRED** [14] uses the AI2-THOR simulator [15] with 7 task categories. We evaluate on the 820-task validated seen split. Both benchmarks enforce egocentric partial observability.

Training data. We collect expert trajectories via in-context learning with LLaMA 3.1-70B [16], using human-collected successful trajectories as in-context demonstrations, to generate candidate action sequences, retaining only successful completions: 656 trajectories for WAH-NL++ and 695 for ALFRED. Each sample consists of an egocentric image, a textual observation, the planner prompt, and the correct next action.

LoRA Details. We finetune the VLM with LoRA (r=8, α=16), injecting adapters into the `q/k/v/o` projections and MLP layers of both visual encoder and language decoder.

Evaluation Metrics. Evaluation metrics include Success Rate (SR), Subgoal Success Rate (SSR), Action Latency (AL), and Peak VRAM. SR measures the percentage of

TABLE I

PERFORMANCE COMPARISON ON WAH-NL++ AND ALFRED.

Env.	Method	SR(%)	SSR(%)	AL(s)	VRAM(GiB)
WAH-NL++	VLM-TAMP [7]	3.0	9.0	8.53	**23.48**
	ReAct [2]	48.0	62.0	4.57	37.76
	One-Step	**58.0**	**70.5**	**4.33**	23.53
ALFRED	VLM-TAMP	6.34	–	4.41	30.81
	ReAct	18.17	–	2.21	31.26
	One-Step	**22.56**	–	**2.07**	**17.32**

successfully completed tasks, while SSR evaluates progress at the subgoal level. AL captures the average time required to predict an action, and Peak VRAM measures the maximum GPU memory usage during inference. We terminate an episode when the agent outputs *done*, or reaches the predefined maximum number($N_{\max} = 30$) of steps.

B. Main Results

Table I shows that One-Step Planner achieves the highest SR and SSR on both benchmarks. VLM-TAMP [7] shows poor SR, as it assumes full observability of the entire scene, making it ill-suited for partially observable environments where forming a complete plan is challenging. In contrast, ReAct [2] and One-Step Planner interleave perception and action at every step, incrementally updating their belief from new observations. Among these, One-Step Planner further outperforms ReAct in SR and SSR on WAH-NL++, since the unified perception-planning pathway operates directly on visual features without a textual bottleneck and enables more reliable visual grounding.

Efficiency gains are also substantial. In terms of AL, One-Step Planner is markedly faster than VLM-TAMP, which requires multiple VLM calls per step for subgoal generation and replanning. One-Step Planner is also consistently faster than ReAct, as it replaces the sequential observer–planner pair with a single VLM forward pass. One-Step uses less VRAM than ReAct because it replaces the separate observer-planner pipeline with a single unified VLM. The gap varies by benchmark because context length also affects memory. WaH-NL++ has multi-room summaries and longer histories, increasing One-Step's VRAM, while ALFRED uses shorter single-room contexts, making the relative reduction larger.

C. Perception Interpretability via Attention Analysis

Fig. 3 visualizes the spatial attention of the VLM before and after LoRA tuning. Before adaptation, attention spreads diffusely across irrelevant background regions such as floors, walls, and non-target objects. After instruction tuning, attention is *sharply concentrated on the goal-relevant objects*, such as the potato inside the refrigerator.

This provides interpretable evidence that instruction tuning actively reshapes the VLM's internal perception, rather than simply memorizing action mappings. The practical consequence is that the model translates visual attention seamlessly into task-relevant actions: when a target object enters the field of view, the One-Step Planner directly acts on it, whereas the two-step pipeline often detects the object in its observation

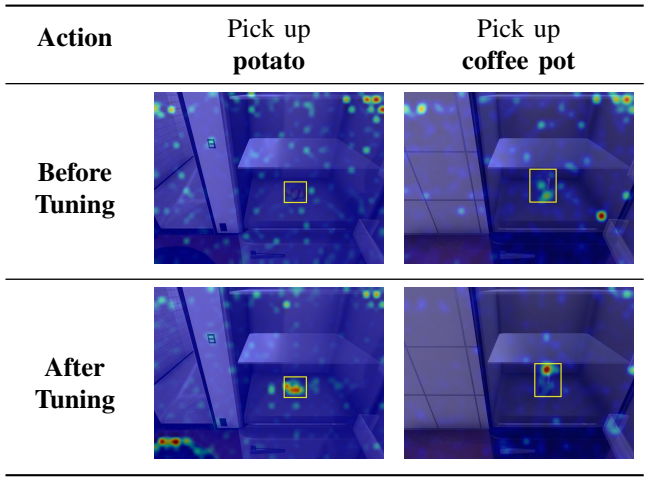

Fig. 3. Spatial attention maps before and after LoRA tuning. After tuning, attention concentrates sharply on task-relevant objects (highlighted by yellow boxes), demonstrating improved perceptual focus.

text but fails to act, because the textual representation lacks spatial salience, leading to goal drift and task failure.

D. Ablation Study on Observer Types

To isolate the effect of modular perception on downstream planning, we systematically vary the observer in the two-step ReAct pipeline across four types: simulator ground truth, VLM-based captioning, supervised object detection (YOLOv12 [17]), and open-vocabulary detection (YOLO-World [18]). Table II reveals three key findings:

(1) Instruction tuning is necessary but not sufficient. Without LoRA-tuned planners, all modular configurations fail (0–3% SR), even with ground-truth observations. This shows that perception alone cannot compensate for an untrained planner.

(2) Observer choice affects robustness. Among tuned planners, SR ranges from 41.0% (YOLO-World) to 55.0% (ground truth, YOLOv12), confirming that modular pipelines are *fragile* to the quality of the perception module. Notably, YOLOv12 matches ground-truth performance but requires expensive box annotations, creating a scalability bottleneck.

(3) End-to-end perception outperforms all modular variants. The One-Step Planner (58.0% SR) surpasses even the ground-truth-observer pipeline (55.0%). Modular base-

TABLE II

PLANNING PERFORMANCE BY OBSERVER TYPE (WAH-NL++).

Observer	LoRA	SR (%)	SSR (%)
Ground Truth	✗	3.0	9.0
	✓	55.0	67.5
VLM (Qwen2.5-VL)	✗	0.0	3.5
	✓	48.0	62.0
Object Detector (YOLOv12 [17])	✗	1.0	4.5
	✓	55.0	66.0
Open-Voca Detector (YOLO-World [18])	✗	0.0	1.5
	✓	41.0	57.5
One-Step Planner	✓	**58.0**	**70.5**

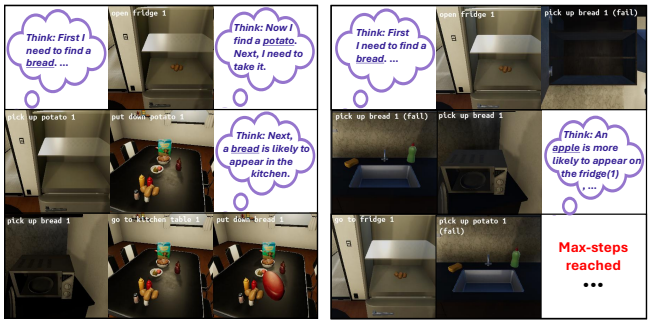

Fig. 4. Trajectory comparison: One-Step Planner vs. ReAct in WAH-NL++. The One-Step Planner immediately acts on a visually detected target, while ReAct suffers from perception loss and goal drift.

lines, even with strong observers, reduce visual input to textual object lists and lose spatial and salient visual cues. One-Step Planner preserves these cues for action selection, explaining gains beyond observer quality.

E. Qualitative Trajectory Analysis

Fig. 4 illustrates a representative episode from WAH-NL++ where both planners must place a potato and bread on the kitchen table. When the potato becomes visible in the refrigerator, the One-Step Planner immediately shifts focus and executes *pick up potato*—a direct consequence of the sharpened attention shown in Fig. 3. In contrast, the ReAct planner detects the potato in its textual observation but persistently pursues bread, leading to repeated failures and goal drift until the step limit is reached. This exposes a structural vulnerability of modular pipelines, where converting visual cues to text discards the attentional context that would prioritize goal-critical objects.

V. CONCLUSION

We presented One-Step Planner, a unified VLM architecture that integrates perception and planning in a single forward pass. The key motivation is that modular pipelines lose spatial salience when converting visual input to text, causing cascading perception errors in long-horizon tasks. Experiments on WAH-NL++ and ALFRED show that the end-to-end approach achieves higher success rates (+10% on WAH-NL++), lower resource consumption (44.6% VRAM reduction), and more interpretable perception than modular pipelines. Attention analysis confirms that instruction tuning reshapes spatial focus toward task-relevant objects, while the comparison across four observer types reveals the fragility of modular perception. These findings suggest that tighter integration of perception and decision-making, rather than more sophisticated perception modules alone, is a promising direction for rigorous robot perception. As future work, we plan to extend the VLM planner to more complex environments and manipulation tasks that require not only detecting the presence of objects but also recognizing their spatial configurations and physical states. In addition, current evaluation relies solely on success rate, which overlooks the efficiency of the action sequence taken to reach the goal. We aim to incorporate metrics that capture procedural efficiency, so that the planner learns to accomplish tasks with fewer and more efficient action steps.

REFERENCES

[1] Y. Hu, F. Lin, T. Zhang, L. Yi, and Y. Gao, "Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning," *arXiv preprint arXiv:2311.17842*, 2023.

[2] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *International Conference on Learning Representations (ICLR)*, January 2023.

[3] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar *et al.*, "Inner monologue: Embodied reasoning through planning with language models," in *6th Annual Conference on Robot Learning*, 2022. [Online]. Available: https://openreview.net/forum?id=3R3Pz5i0tye

[4] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[5] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin, "Qwen2.5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025.

[6] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," *International conference on machine learning (PMLR)*, pp. 9118–9147, 2022.

[7] Z. Yang, C. Garrett, D. Fox, T. Lozano-Pérez, and L. P. Kaelbling, "Guiding long-horizon task and motion planning with vision language models," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025, to appear. [Online]. Available: https://arxiv.org/abs/2410.02193

[8] P. Sermanet, T. Ding, J. Zhao, F. Xia, D. Dwibedi, K. Gopalakrishnan, and Y. Cao, "Robovqa: Multimodal long-horizon reasoning for robotics," in *Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2024, pp. 645–652.

[9] Y. Liu, D. Chi, S. Wu, Z. Zhang, Y. Hu, L. Zhang, and Y. Zhuang, "Spatialcot: Advancing spatial reasoning through coordinate alignment and chain-of-thought for embodied task planning," *arXiv preprint arXiv:2501.10074*, 2025.

[10] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn, A. Handa, T.-Y. Lin, G. Wetzstein, M.-Y. Liu, and D. Xiang, "Cot-vla: Visual chain-of-thought reasoning for vision-language-action models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 1702–1713.

[11] X. Puig, T. Shu, S. Li, Z. Wang, J. B. Tenenbaum, S. Fidler, and A. Torralba, "Watch-and-help: A challenge for social perception and human-ai collaboration," 2020.

[12] J.-W. Choi, Y. Yoon, H. Ong, J. Kim, and M. Jang, "Lota-bench: Benchmarking language-oriented task planners for embodied agents," in *International Conference on Learning Representations (ICLR)*, 2024.

[13] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba, "Virtualhome: Simulating household activities via programs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8494–8502.

[14] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, and D. Fox, "Alfred: A benchmark for interpreting grounded instructions for everyday tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 740–10 749.

[15] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "AI2-THOR: An Interactive 3D Environment for Visual AI," *arXiv*, 2017.

[16] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[17] Y. Tian, Q. Ye, and D. Doermann, "Yolov12: Attention-centric real-time object detectors," *arXiv preprint arXiv:2502.12524*, 2025.

[18] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2024.