SPATIAL-JARVIS: A META-LEVEL SPATIAL COGNITIVE REGULATOR FOR INTERACTIVE SPATIAL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) have made significant progress in passive spatial reasoning, yet they still face substantial challenges in long-horizon active spatial interaction tasks. During such interactions, inaccurate beliefs about the environment and task progress can lead to erroneous decisions and persist across subsequent steps, ultimately resulting in task failure. We attribute these failures to insufficient spatial metacognition, i.e., the ability to monitor and regulate evolving cognitive states and decision-making processes. To address this limitation, we propose **Spatial-JARVIS**, a Meta-Level Regulator decoupled from the policy model. At each interaction step, Spatial-JARVIS monitors the agent's inferred cognitive state and evaluates the proposed decision before execution, generating corrective feedback to guide the policy model in revising its mistaken beliefs and adjusting its subsequent behavior. To learn this capability, we construct the **MetaSpatial-35K** dataset and adopt a two-stage training strategy consisting of SFT and MC-GRPO. We further introduce **MetaSpatial-Bench**, a static benchmark designed to directly evaluate models' monitoring and correction capabilities in spatial interaction. Experimental results show that Spatial-JARVIS outperforms the strongest compared MLLMs by 14.6% and 13.3% in monitoring and control accuracy on MetaSpatial-Bench, respectively. Moreover, Spatial-JARVIS improves the task success rates of six different policy models by an average of 7.7% on SpatialWorld, while reducing ineffective exploration and erroneous operations.

1 INTRODUCTION

Multimodal large language models (MLLMs) have made substantial progress in spatial understanding(Yang et al., 2025a; Wu et al., 2025; Chen et al., 2024). However, the ultimate goal of spatial intelligence extends beyond passive spatial question answering to acting and interacting in space to accomplish practical tasks. Such tasks require long-horizon decision-making, where models must generate high-level actions based on evolving environmental states to progressively achieve task goals. This high-level decision-making capability is complementary to the low-level action execution capabilities of embodied foundation models (Physical Intelligence et al., 2025). Nevertheless, long-horizon spatial interaction remains challenging for current MLLMs (Yang et al., 2025b; Hong et al., 2026; Gao et al., 2026; Li et al., 2026).

Unlike passive spatial question answering, long-horizon spatial interaction requires agents to continuously update their cognitive states about the environment and task at every step. Existing approaches primarily focus on improving the interaction policy itself (Lei et al., 2025; Chen et al., 2025). However, even strong policies may develop erroneous cognition that lead to unreliable decisions. As shown in Figure 1, an agent may repeatedly revisit explored regions due to faulty spatial beliefs or mistakenly believe that a subtask has been completed when it has not. Without timely correction, such cognitive errors may propagate through subsequent decisions and lead to task failure.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

These failures highlight a fundamental requirement for reliable spatial interaction: *Reliable interaction requires agents not only to make effective decisions but also to continuously monitor and regulate the cognitive processes underlying them.* This motivates us to revisit spatial interaction from the perspective of metacognition (Nelson, 1990). Metacognitive theory distinguishes between an *Object Level*, responsible for task-directed cognitive activities, and a *Meta Level*, which monitors ongoing Object-Level activities and regulates their subsequent course. The two levels form a bidirectional loop through *metacognitive monitoring* and *metacognitive control*. From this perspective, existing approaches primarily focus on strengthening the Object-Level policy, while Meta-Level regulation of spatial cognition remains underexplored.

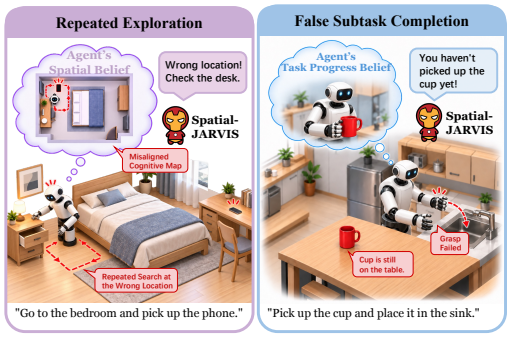


Figure 1: Left: erroneous spatial beliefs cause the agent to repeatedly explore redundantly. Right: the agent mistakenly believes it has picked up the cup, making subsequent operations ineffective.

This naturally raises the question: *how can such Meta-Level capability be concretely modeled?* We propose **Spatial-JARVIS**, which models the Meta Level as an independent Regulator model. Spatial-JARVIS continuously monitors the agent's cognitive state and the action proposed by the policy model at each interaction step, assessing their validity (*metacognitive monitoring*). When a potentially unreasonable decision is identified, Spatial-JARVIS further generates corrective feedback to guide the policy model in revising its mistaken beliefs and adjusting its current decision, while appending the feedback to the interaction memory to prevent recurring cognitive errors in subsequent decisions (*metacognitive control*). The proposed method is decoupled from the policy model.

Specifically, we design a scalable automated data construction pipeline to derive Meta-Level supervision signals for decision monitoring and correction from spatial interaction trajectories. Based on this pipeline, we construct **MetaSpatial-35K**, a dataset for learning to monitor and regulate agent decisions and cognition during spatial interaction. We train Spatial-JARVIS through SFT followed by **MC-GRPO**, further strengthening its ability to identify problematic decisions and provide effective corrective feedback. To evaluate these capabilities, we introduce **MetaSpatial-Bench**, a static benchmark for assessing decision monitoring and correction in spatial interaction. Spatial-JARVIS significantly outperforms existing MLLMs on MetaSpatial-Bench. Further online experiments demonstrate that integrating Spatial-JARVIS with different policy models consistently improves task success rates across multiple interaction benchmarks while reducing redundant exploration and ineffective interactions. Our contributions are summarized as follows:

1. We revisit spatial interaction from a metacognitive perspective, introducing a two-level framework to address the lack of explicit mechanisms for monitoring and regulating evolving spatial cognitive states in existing policy models.
2. We design a scalable automated data construction pipeline and construct **MetaSpatial-35K**, a dataset for learning metacognitive capabilities in spatial interaction. We further introduce **MetaSpatial-Bench**, a static benchmark specifically designed to evaluate the ability to monitor and correct cognitive states and decisions during spatial interaction.
3. We propose **Spatial-JARVIS**, an independent Meta-Level Regulator trained with SFT and **MC-GRPO** for decision monitoring and corrective intervention. On SpatialWorld, Spatial-JARVIS improves task success rates by **7.7% on average** across different policy models, while also improving interaction efficiency.

2 Related Work

2.1 Spatial Interaction with MLLMs

Multimodal large language models have achieved significant progress in passive spatial tasks such as spatial relation understanding and depth perception, yet transferring these capabilities to spatial

interaction scenarios requiring continuous perception and action remains challenging. Recent work has begun to employ MLLMs as high-level decision-making models for embodied agents and systematically evaluate their capabilities in active exploration, navigation and interaction(Yang et al., 2025b; Hong et al., 2026; Gao et al., 2026; Li et al., 2026). Research in this area is still at an early stage. Existing approaches enhance embodied agents through memory-augmented planning (Lei et al., 2025; Ma et al., 2026) or reinforcement learning for embodied decision-making (Chen et al., 2025; Kim et al., 2026). These approaches primarily focus on enhancing the decision-making capability of the policy itself. In contrast, we focus on the complementary Meta Level, which monitors and regulates decisions made by existing policies.

2.2 METACOGNITION IN LLM

Recent studies have introduced metacognitive mechanisms into LLMs to enhance their ability to monitor and regulate reasoning processes. For example, MeCo leverages internal cognitive signals to adaptively determine tool invocation (Li et al., 2025), while Meta-R1 employs metacognitive regulation for active planning and adaptive reasoning control (Dong et al., 2025). Other studies explore metacognition for knowledge augmentation and test-time reasoning control (Chen et al., 2026; Cao et al., 2026). In contrast, we focus on metacognition for spatial agents, whose cognitive states continuously evolve through multi-step interactions. Unlike in static reasoning tasks, erroneous spatial cognition may persist across interaction steps and propagate through subsequent decisions, requiring continuous monitoring and regulation of both cognition and decision-making.

2.3 LLM REFLECTION AND CRITIC

Another related line of work explicitly provides feedback and corrects errors in model reasoning or decision-making. Self-Refine and Reflexion rely on models to generate their own feedback and iteratively improve subsequent outputs or behaviors (Madaan et al., 2023; Shinn et al., 2023). However, such intrinsic self-correction is constrained by the model's own ability to identify errors (Huang et al., 2024). To address this limitation, subsequent studies introduce independent critics to evaluate and provide feedback on language outputs, multimodal reasoning processes, or GUI operations (McAleese et al., 2024; Xiong et al., 2025; Zhang et al., 2025; Wanyan et al., 2025). However, these methods primarily focus on evaluating and correcting individual reasoning outputs or proposed operations. In contrast, spatial interaction involves interdependent subtasks, where agents' spatial cognition accumulates through continuous observations and exploration (rather than at isolated steps), which in turn shape subsequent decisions. Moreover, only correcting an individual decision may leave erroneous beliefs unresolved, allowing errors to propagate across subsequent subtasks. Therefore, we focus on continuously monitoring evolving cognitive states and decisions, correcting not only erroneous decisions but also the underlying beliefs that influence subsequent behavior.

3 METHOD

In this section, we first formulate the task of interactive spatial agents. We then introduce our data construction pipeline and training strategy for developing Spatial-JARVIS. Finally, we describe how Spatial-JARVIS is integrated with different policy models to regulate their cognition and decisions.

3.1 PROBLEM DEFINITION

At each step, the environment has a state s_t, recording object positions and attributes. The agent cannot access s_t and instead receives an observation o_t containing the instruction, current-view image, and interaction history. Based on the history $h_t = (o_1, a_1, \ldots, o_t)$, the agent selects a high-level action a_t. The environment then transitions and returns a new observation until the agent determines that the task is complete or reaches the maximum step limit. Each task has completion conditions $C = \{c_1, \ldots, c_m\}$ and is completed if and only if all conditions are satisfied in s_t.

3.2 DATA CONSTRUCTION

We design a scalable data synthesis pipeline for training a Meta-Level Regulator and construct the **MetaSpatial-35K** dataset, comprising 3175 task trajectories across 1410 scenes and 35K step-

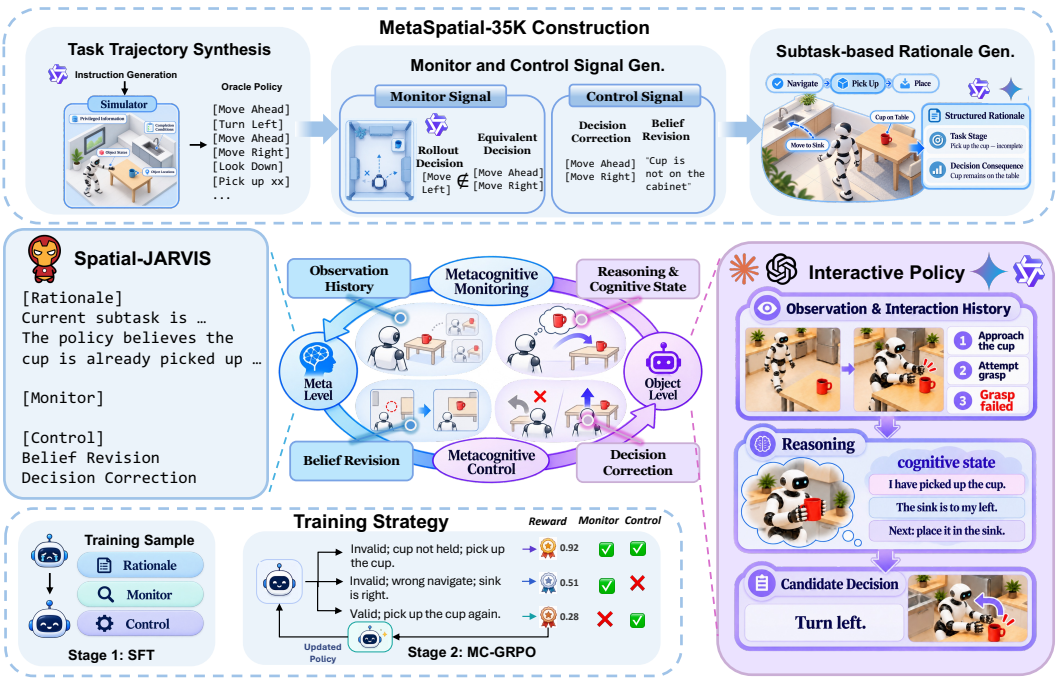


Figure 2: Overview of **Spatial-JARVIS**. The central panel illustrates the interaction between Spatial-JARVIS and the policy model. At each interaction step, Spatial-JARVIS monitors the policy's evolving cognitive state and candidate decision, providing corrective feedback through decision and belief revision when necessary. The upper panel presents the automated construction of **MetaSpatial-35K**, which provides metacognitive supervision for monitoring and control. The lower panel illustrates the two-stage training strategy, comprising SFT and MC-GRPO.

level samples. Since the correctness of cognitive states is difficult to quantify directly, we leverage decision-level supervision to learn two complementary capabilities: (1) monitoring, which assesses the reasonableness of policy decisions to identify potential cognitive and decision errors; and (2) control, which provides corrective feedback to guide the policy model in revising its erroneous beliefs and adjusting subsequent decisions. Details of the data pipeline are provided in Appendix E.

Task Trajectory Synthesis. We synthesize data in two simulation environments, AI2-THOR and ProcTHOR (Kolve et al., 2017; Deitke et al., 2022). A task is denoted as $\tau = (S, C, I)$, where S is a randomly initialized scene, $C = \{c_1, \ldots, c_m\}$ is a set of randomly generated completion conditions, and the instruction I is generated by Qwen3.5-27B (Qwen Team, 2026a) conditioned on S and C. Reference trajectories are generated by an oracle policy π^* with access to the global state. At each step, π^* determines the unfinished goals based on the current true state and plans the corresponding action trajectory using the actual positions and interaction states of objects in the environment until all conditions in C are satisfied.

Monitor and Control Signal Construction. For monitoring signals, using only reference-trajectory decisions may lead to rigid cognitive and decision-making patterns. To address this, we construct an equivalent-decision set through simulation, where equivalent decisions are defined as feasible alternatives that achieve the same subgoal progress as the reference decision. On the other hand, randomly generated erroneous decisions are often too simple to reflect the cognitive biases arising during actual interaction. We therefore use Qwen3.5-27B to perform interaction rollouts from each state along the reference trajectory and construct binary monitoring signals based on whether the resulting decisions belong to the equivalent-decision set.

For control signals, we consider both *decision correction* and *belief revision*. For decision correction, we randomly select a reasonable decision from the equivalent-decision set as the correction target. For belief revision (e.g., "You have not picked up the apple yet" or "The phone is actually to the

right of the bed"), we identify erroneous beliefs by comparing the policy's cognitive states during interaction with the ground-truth simulator states, and use Qwen3.5-Plus to generate corresponding corrective feedback. Together, these signals guide the policy model in revising its erroneous beliefs and adjusting its current and subsequent decisions.

Subtask-based Rationale Generation. To equip Spatial-JARVIS with structured cognitive reasoning rather than merely learning from supervision labels, we construct rationales as training supervision. Since long-horizon spatial interaction involves multiple interdependent subtasks, effective decision reasoning requires understanding the current subtask and its progress. We therefore adopt subtask-based reasoning to provide a clearer cognitive reference for monitoring and regulation.

Specifically, we use Qwen3.5-Plus to segment reference trajectories into subgoals and obtain task-stage supervision g_t verified by ground-truth environment states. We further simulate candidate decisions to obtain state transitions for decision-consequence reasoning. Both are used only for data construction. Based on these signals, we use Qwen3.5-Plus to generate structured rationales that identify the current subtask and its progress from interaction history, infer the consequences of the proposed decision, and assess whether it advances the current subtask.

Finally, we use Gemini 3.5 Flash (Vinyals & Shazeer, 2026) to filter samples that directly restate supervision labels, contradict environmental transitions, or contain low-quality reasoning. The resulting training set is $\mathcal{D} = \{(h_t, a_t, \rho_t, l_t, u_t)\}$, where ρ_t denotes the structured cognitive rationale, and l_t and u_t denote the monitoring and control signals, respectively.

3.3 SPATIAL-JARVIS MODEL

To equip Spatial-JARVIS with effective metacognitive monitoring and control capabilities, we adopt a two-stage training strategy based on SFT and **Monitoring-Control Group Relative Policy Optimization (MC-GRPO)**. Specifically, starting from Qwen3.5-9B (Qwen Team, 2026a), we first perform SFT to establish structured cognitive reasoning patterns, followed by MC-GRPO to enhance monitoring accuracy and control effectiveness.

Supervised Fine-Tuning. We first perform supervised fine-tuning (SFT) on 20k samples with structured rationales, enabling Spatial-JARVIS to learn task-stage identification, decision-consequence inference, and the corresponding monitoring–control output patterns. Through this process, the model learns to assess decision reasonableness based on the current interaction context and generate corrective feedback for erroneous beliefs and decisions. SFT equips Spatial-JARVIS with structured metacognitive reasoning capabilities.

MC-GRPO. We further perform reinforcement learning using 15k samples to enhance Spatial-JARVIS's metacognitive monitoring and control capabilities. For each input x, the model samples a group of candidate outputs $\{o_i\}_{i=1}^{G}$ and is optimized using the objective:

$$\mathcal{L}_{\mathrm{GRPO}} = -\frac{1}{|G|}\sum_{i=1}^{G}\min\left(\frac{\pi_\theta(o_i \mid x)}{\pi_{\theta_{\mathrm{old}}}(o_i \mid x)}A_i, \mathrm{clip}\left(\frac{\pi_\theta(o_i \mid x)}{\pi_{\theta_{\mathrm{old}}}(o_i \mid x)}, 1-\epsilon, 1+\epsilon\right)A_i\right) + \beta D_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{ref}}) \tag{1}$$

where A_i denotes the advantage estimate obtained from the relative reward within the group, and π_{ref} denotes the reference policy. To align reinforcement learning with metacognitive monitoring and control, we design a hierarchical reward consisting of format, monitoring, and control rewards:

$$r_{\mathrm{total}} = \lambda_f r_f + \lambda_{mc}\mathbb{I}[\hat{l} = l]\left(\alpha + (1-\alpha)\mathbb{I}[\hat{d} \in \mathcal{A}^{\mathrm{eq}}(s_t)]\,\mathrm{Judge}(\hat{b}, b^*)\right) \tag{2}$$

Here, r_f rewards adherence to the structured reasoning format, while $\hat{l}$ and l denote the predicted and ground-truth monitoring labels. The control reward r_c jointly evaluates *decision correction* and *belief revision*. Specifically, $\hat{d}$ denotes the generated corrective decision, which is evaluated against the equivalent-decision set $\mathcal{A}^{\mathrm{eq}}(s_t)$. Meanwhile, $\hat{b}$ and b^* denote the generated and reference belief corrections, respectively, with an LLM judge assessing their semantic consistency. The coefficients λ_f and λ_{mc} weight the format and monitoring-control rewards, while α limits the reward attainable through correct monitoring alone.

This hierarchical design makes accurate monitoring a prerequisite for effective control. A correct monitoring judgment alone receives only a partial reward, while the full monitoring-control re-

Table 1: Static evaluation on MetaSpatial-Bench. M and C denote monitoring and control accuracy, respectively. Best results are bolded and second-best results are underlined.

Model	Search		Navigate		Interact		Overall	
	M	C	M	C	M	C	M	C
Closed-source								
Claude-Sonnet-5	<u>59.87</u>	25.30	56.67	<u>24.11</u>	66.27	25.81	59.87	24.44
Qwen3.5-Plus	55.47	<u>25.53</u>	44.53	20.55	<u>72.56</u>	42.42	57.52	31.77
Gemini 3.5 Flash-Lite	49.70	5.36	<u>56.71</u>	23.53	58.09	**63.33**	54.83	35.51
GPT-5.6-Luna	58.17	16.67	51.27	22.97	71.96	<u>57.60</u>	<u>60.47</u>	<u>38.74</u>
Open-source								
Qwen3.5-9B	49.33	16.67	43.33	16.47	60.42	37.61	51.02	24.32
Qwen3.5-27B	52.92	12.77	34.48	13.51	**76.76**	55.26	54.72	33.12
Qwen3.6-35B-A3B	56.67	18.60	47.38	16.22	62.67	24.42	55.57	19.92
Spatial-JARVIS-9B (ours)	**83.51**	**43.86**	**78.86**	**60.67**	62.82	52.89	**75.06**	**52.04**

ward requires both accurate decision correction and belief revision. This discourages the model from optimizing monitoring accuracy alone while neglecting corrective feedback. Thus, MC-GRPO organizes reinforcement learning as "**first monitor correctly, then regulate effectively**", jointly enhancing metacognitive monitoring and control capabilities.

3.4 Meta-Level Regulation Loop

We regard the original policy as the **Object Level**, responsible for environmental perception and decision-making, and Spatial-JARVIS as an independent **Meta-Level Regulator**. The two form a closed loop through bidirectional information flow: during monitoring, the Meta Level receives the Object Level's current observation, interaction history, current reasoning, and candidate decision, and assesses the candidate decision in this context; during control, the Meta Level provides decision correction and belief revision feedback based on this assessment, guiding the current decision and updating the cognitive states that inform subsequent decisions.

At the beginning of each task, the policy decomposes the task instruction into ordered subplans based on the initial observation, providing stage-level goals for subsequent interaction. At each step, the policy generates a candidate decision conditioned on the interaction history h_t and the current subplan p_{k_t}:

$$a_t \sim \pi_{\text{agent}}\left(\cdot \mid h_t, p_{k_t}\right). \tag{3}$$

Before execution, Spatial-JARVIS receives the interaction history, current subplan, and candidate decision, and generates monitoring and control signals:

$$(l_t, u_t) = \pi_{\text{regulator}}\left(h_t, p_{k_t}, a_t\right), \tag{4}$$

where $l_t \in \{0, 1\}$ indicates whether the candidate decision is reasonable, and u_t denotes the corrective feedback. If $l_t = 1$, the candidate decision is executed directly. Otherwise, Spatial-JARVIS provides decision correction and belief revision feedback to the Object Level, prompting it to update its understanding of the environment or task progress and generate a revised decision. The feedback is retained in the subsequent interaction context, allowing corrected beliefs to guide future decisions. This design enables Spatial-JARVIS to regulate different spatial interaction policies without replacing their original decision-making processes.

4 Experiments

4.1 Static Regulation Evaluation

Existing benchmarks provide limited evaluation of meta-level decision monitoring and correction during spatial interaction. To address this gap, we construct **MetaSpatial-Bench**, which independently evaluates models' ability to assess the validity of candidate decisions and correct erroneous decisions. The benchmark contains 527 scenes and 2k decision samples, covering three typical interaction stages: Explore, Navigate, and Interact. We use monitoring accuracy and control accuracy

Table 2: Online task success rates on SpatialWorld. Values in parentheses indicate absolute changes after introducing Spatial-JARVIS.

Model	Daily	Work	Entertain	Travel	Social	Overall
Qwen3.5-Plus	12.7	25.4	13.6	15.4	30.4	15.9
+ Spatial-JARVIS	**18.6** (+5.9)	**32.2** (+6.8)	**18.2** (+4.6)	**30.8** (+15.4)	28.2 (-2.2)	**21.3** (+5.4)
Qwen3.5-27B	4.1	6.8	3.0	0.0	19.6	5.6
+ Spatial-JARVIS	**18.0** (+13.9)	**28.8** (+22.0)	**10.6** (+7.6)	**30.8** (+30.8)	**21.7** (+2.1)	**19.0** (+13.4)
Qwen3.6-35B-A3B	5.6	3.4	3.0	7.7	10.9	5.6
+ Spatial-JARVIS	8.3 (+2.7)	**11.9** (+8.5)	**13.6** (+10.6)	**30.8** (+23.1)	**13.0** (+2.1)	**10.3** (+4.7)
GPT-5.6-Luna	10.9	15.3	10.6	15.4	30.4	13.2
+ Spatial-JARVIS	**23.1** (+12.2)	**30.5** (+15.2)	**27.3** (+16.7)	**46.2** (+30.8)	26.1 (-4.3)	**25.3** (+12.1)
Gemini 3.5 Flash-Lite	12.7	13.6	10.6	15.4	2.2	11.7
+ Spatial-JARVIS	**14.5** (+1.8)	**16.9** (+3.3)	6.1 (-4.5)	15.4 (+0.0)	**4.3** (+2.1)	**12.8** (+1.1)
Claude-Sonnet-5	14.8	28.6	11.1	12.5	25.0	17.0
+ Spatial-JARVIS	**27.4** (+12.6)	**32.1** (+3.5)	**19.4** (+8.3)	**25.0** (+12.5)	25.0 (+0.0)	**26.4** (+9.4)

to measure decision validity assessment and erroneous decision correction, respectively. To avoid scene and task leakage, all evaluation scenes and interaction tasks are excluded from the training data. Detailed benchmark construction and evaluation protocols are provided in Appendix D.

As shown in Table 1, Spatial-JARVIS-9B achieves 75.06% monitoring accuracy and 52.04% control accuracy on the overall benchmark, improving over the base model Qwen3.5-9B by 24.0 and 27.7 points, respectively, and outperforming the latest closed-source models GPT-5.6-Luna and Claude-Sonnet-5. These results demonstrate that, after targeted Meta-Level training, Spatial-JARVIS can effectively identify unreasonable decisions in spatial interaction and further generate beneficial regulatory signals.

4.2 Online Spatial Interaction Evaluation

The ultimate value of Spatial-JARVIS lies in whether it can improve online long-horizon spatial interaction. We therefore conduct online evaluation on SpatialWorld (Gao et al., 2026), which covers diverse spatial tasks with different interaction requirements. To evaluate the generality of Meta-Level regulation across policy models, we compare the performance of multiple representative policies before and after integrating Spatial-JARVIS. Table 2 reports the task success rates. Introducing Spatial-JARVIS improves task success rates by an average of 7.7% across the evaluated policies, demonstrating its ability to enhance spatial interaction without retraining the original policies. Detailed evaluation settings, outcome transitions, and capability boundaries under which Spatial-JARVIS improves performance are provided in Appendix G.

To further evaluate generalization under different interaction settings, we conduct experiments on EmbodiedBench-Nav (Yang et al., 2025b), which uses a different action space. This benchmark primarily focuses on navigation decisions based on visual inputs and therefore provides a targeted test for cross-environment validation of Spatial-JARVIS. As shown in Table 3, introducing Spatial-JARVIS improves task performance by 5.3% and 9.0% on Qwen3.5-9B and Qwen3.5-27B, respectively, further demonstrating that Spatial-JARVIS does not depend on a specific spatial interaction environment or action space.

4.3 Comparison with Direct Policy Training

Can the benefits of meta-level regulation be achieved by directly optimizing the interaction policy? To investigate this, we train Qwen3.5-9B as an interaction policy using the same source task data as MetaSpatial-35K, which are used to generate the reference trajectories in Section 3.2. We compare

Table 3: Online task success rates on EmbodiedBench-Nav. Values in parentheses indicate absolute changes after introducing Spatial-JARVIS.

Model	Base	Common	Complex	Visual	Long	Overall
Qwen3.5-27B	53.3	55.0	50.0	36.7	31.7	45.3
+ Spatial-JARVIS	**65.0** (+11.7)	**66.7** (+11.7)	**51.7** (+1.7)	**48.3** (+11.6)	**40.0** (+8.3)	**54.3** (+9.0)
Qwen3.5-9B	65.0	38.3	50.0	38.3	41.7	46.7
+ Spatial-JARVIS	58.3 (-6.7)	**51.7** (+13.4)	**60.0** (+10.0)	**48.3** (+10.0)	41.7 (+0.0)	**52.0** (+5.3)

direct policy optimization with Spatial-JARVIS through online interaction experiments. Table 4 reports the results, and we reveal two key findings as follows.

(1) Optimizing the regulator yields greater gains than directly optimizing the policy. Direct policy SFT improves task performance by **4.6%**, whereas introducing Spatial-JARVIS yields an improvement of **9.5%**. With training data from the same source and scale, Spatial-JARVIS achieves a larger gain. This may stem from their different learning objectives: direct policy training must learn decision generation, whereas Spatial-JARVIS focuses on decision assessment and belief revision. The former involves a more challenging learning objective and may require more training data to achieve comparable gains.

Table 4: Comparison of Policy SFT (PSFT) and Spatial-JARVIS (SJ).

PSFT	SJ	Daily	Work	Entertain	Travel	Social	Overall
✗	✗	2.1	5.1	0.0	0.0	13.0	3.1
✗	✓	11.8	15.3	7.6	23.1	19.6	12.6
✓	✗	6.5	11.9	6.1	15.4	10.9	7.7
✓	✓	12.1	18.6	12.1	23.1	15.2	13.4

(2) Meta-Level learning is complementary to direct policy optimization. When Spatial-JARVIS is further introduced after the policy model has already undergone direct policy training, it still brings an additional performance improvement of **5.7%**. This demonstrates that introducing Spatial-JARVIS at the Meta Level does not conflict with direct optimization of the interaction policy; instead, the two are complementary and can jointly improve performance.

4.4 Efficiency Analysis

We further analyze the impact of Spatial-JARVIS on task efficiency. We use the average number of interaction steps for successful tasks (SS) and SPL as evaluation metrics. SPL is computed as $\text{SPL} = \text{TSR} \times \frac{1}{N}\sum_{i=1}^{N} \frac{L_i^{\text{ref}}}{\max(L_i, L_i^{\text{ref}})}$.

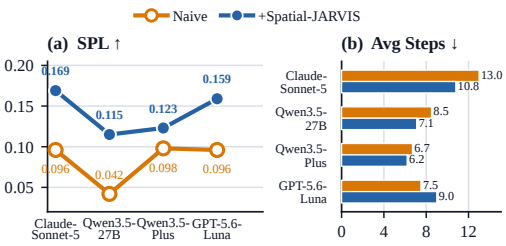


Figure 3: Efficiency on SpatialWorld: SPL and average interaction steps for successful tasks, with and without Spatial-JARVIS across four policy models.

As shown in Fig. 3, introducing Spatial-JARVIS reduces the average number of interaction steps for multiple policy models while substantially improving SPL. This result indicates that Spatial-JARVIS not only corrects erroneous decisions that would otherwise lead to task failure, but also intervenes before erroneous behavior further accumulates, thereby reducing redundant exploration and ineffective interactions caused by local errors. Compared with approaches that perform reflection after action execution, Spatial-JARVIS performs decision checking before action execution and therefore does not require additional interaction steps to perform the correction process of "observing the outcome, reflecting, and acting again." Detailed analysis of inference time and computational cost is provided in Appendix G.

4.5 Ablation Studies

Training Strategy. We first conduct ablations on the two-stage training strategy and the key components of MC-GRPO, with results reported in Table 5.

SFT alone enables the model to learn subtask-based structured reasoning patterns, but it is difficult to further establish stable Meta-Level monitoring and control capabilities. Direct RL training, meanwhile, also fails to effectively establish this capability. In contrast, SFT + MC-GRPO substantially improves both metrics. Further removing either the monitoring reward or the control reward leads to a corresponding performance degradation, verifying the effectiveness of the two rewards.

Table 5: Ablation of training strategy.

SFT	MC-GRPO			MetaSpatial-Bench	
	r_f	r_m	r_c	Monitor	Control
✓	✗	✗	✗	72.66	50.30
✗	✓	✓	✓	68.62	40.13
✓	✓	✗	✓	67.72	51.65
✓	✓	✓	✗	67.47	37.34
✓	✓	✓	✓	**75.06**	**52.04**

Data Construction.

We further ablate the key components of our data construction pipeline, using the SFT model trained on the complete dataset as the baseline. As shown in Table 6, removing subtask-based rationales degrades performance, highlighting the importance of understanding task progress and decision reasoning. Removing equivalent-decision samples further reduces control accuracy, suggesting that learning multiple valid decisions helps the regulator assess and correct decisions beyond imitating reference trajectories. Replacing negative examples from actual policy rollouts with randomly generated decisions also reduces control accuracy, suggesting that erroneous decisions from actual policies better reflect the mistaken spatial beliefs that arise during interaction.

Table 6: Ablation of data pipeline (EA denotes equivalent decisions).

	MetaSpatial-Bench	
Data recipe	Monitor	Control
Full pipeline	**72.66**	**50.30**
w/o rationale	71.71	48.98
w/o EA	68.87	41.72
w/o policy rollout	69.82	39.35

Analysis of Meta-Level Regulation. We use Qwen3.5-27B as the fixed policy to analyze different regulation mechanisms and compare general-purpose Regulators on SpatialWorld.

(1) Regulation Mechanisms. We further compare with several potentially related methods discussed in the related work to validate the effectiveness of our proposed mechanism in spatial interaction scenarios. As shown in Table 7, all alternative mechanisms underperform Spatial-JARVIS. Self-reflection asks the policy to reassess its own decisions, but remains limited by its reasoning capabilities, adds an extra decision-making step, and cannot undo irreversible outcomes such as breaking a bowl or prematurely terminating a task. When Spatial-JARVIS serves only as a critic (w/o Control Signal), the policy must still revise its decisions based on its own potentially mistaken beliefs. Without belief revision, correcting individual decisions may leave underlying erroneous beliefs unresolved, allowing similar errors to recur.

Table 7: Mechanisms.

Mechanism	SR (%)
None	5.6
Self-Reflection	8.2
Critic	4.6
w/o Belief Revision	16.1
Spatial-JARVIS	**19.0**

Table 8: Regulators.

Regulator	SR (%)
None	5.6
Qwen3.5-27B	7.7
Qwen3.6-35B-A3B	8.2
Qwen3.5-Plus	12.0
Gemini 3.5 Flash	10.1
Spatial-JARVIS	**19.0**

(2) General-Purpose Regulators. As shown in Table 8, general-purpose Regulators improve the original policy's success rate from 5.6% to 7.7–12.0%, while Spatial-JARVIS achieves 19.0%. This highlights the benefits of dedicated training for meta-level monitoring and control.

5 Conclusion

We revisit the failures of MLLMs in spatial interaction through the lens of metacognition, linking repeated exploration and task-state misjudgment to insufficient monitoring and regulation of their evolving cognitive states. Building on this perspective, we propose Spatial-JARVIS, an independent Meta-Level Regulator that monitors candidate decisions and provides both decision correction and belief revision feedback. To support its training, we develop a scalable, subtask-based data construction pipeline and construct MetaSpatial-35K, followed by a two-stage training procedure combining SFT and MC-GRPO. On our proposed MetaSpatial-Bench, Spatial-JARVIS improves monitoring and control accuracy by 14.6% and 13.3%, respectively. It also consistently improves online task success rates across different policy models on SpatialWorld, demonstrating the value of meta-level regulation for long-horizon spatial interaction.

REFERENCES

Anthropic. Introducing claude sonnet 5. https://www.anthropic.com/news/claude-sonnet-5, 2026. Official model announcement.

Qi Cao, Yufan Wang, Peijia Qin, Shuhao Zhang, and Pengtao Xie. Llms know when they know, but do not act on it: A metacognitive harness for test-time scaling. *arXiv preprint arXiv:2605.14186*, 2026.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://arxiv.org/abs/2401.12168.

Hanyang Chen, Mark Zhao, Rui Yang, Qinwei Ma, Ke Yang, Jiarui Yao, Kangrui Wang, Hao Bai, Zhenhailong Wang, Rui Pan, Mengchao Zhang, Jose Barreiros, Aykut Onol, ChengXiang Zhai, Heng Ji, Manling Li, Huan Zhang, and Tong Zhang. Era: Transforming vlms into embodied agents via embodied prior learning and online reinforcement learning. *arXiv preprint arXiv:2510.12693*, 2025.

Hao Chen, Ye He, Yuchun Fan, Yukun Yan, Zhenghao Liu, Qingfu Zhu, Maosong Sun, and Wanxiang Che. Know more, know clearer: A meta-cognitive framework for knowledge augmentation in large language models. *arXiv preprint arXiv:2602.12996*, 2026.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Haonan Dong, Haoran Ye, Wenhao Zhu, Kehan Jiang, and Guojie Song. Meta-r1: Empowering large reasoning models with metacognition. *arXiv preprint arXiv:2508.17291*, 2025.

Hongcheng Gao, Hailong Qu, Jingyi Tang, Jiahao Wang, Zihao Huang, Hengkang Qiao, Shihong Huang, Junming Yang, Yi Li, Hongyixuan Yuan, Wenjie Li, Bohan Zeng, Wenbo Li, Bo Wang, Jianhui Liu, Olive Huang, Haoyang Huang, Wentao Zhang, Guoqing Huang, Nan Duan, and Yinpeng Dong. Spatialworld: Benchmarking interactive spatial reasoning of multimodal agents in real-world tasks. *arXiv preprint arXiv:2606.09669*, 2026.

Yining Hong, Jiageng Liu, Han Yin, Manling Li, Leonidas Guibas, Li Fei-Fei, Jiajun Wu, and Yejin Choi. Esi-bench: Towards embodied spatial intelligence that closes the perception-action loop. *arXiv preprint arXiv:2605.18746*, 2026.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations (ICLR)*, 2024.

Dongyoung Kim, Sumin Park, Huiwon Jang, Jinwoo Shin, Jaehyung Kim, and Younggyo Seo. Robot-r1: Reinforcement learning for enhanced embodied reasoning in robotics. *Advances in Neural Information Processing Systems*, 38:161472–161507, 2026.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. URL https://arxiv.org/abs/1712.05474.

Mingcong Lei, Honghao Cai, Yuyuan Yang, Yimou Wu, Jinke Ren, Zezhou Cui, Liangchen Tan, Junkun Hong, Gehan Hu, Shuangyu Zhu, Shaohan Jiang, Ge Wang, Junyuan Tan, Zhenglin Wan, Zheng Li, Zhen Li, Shuguang Cui, Yiming Zhao, and Yatong Han. Robomemory: A brain-inspired multi-memory agentic framework for interactive environmental learning in physical embodied systems. *arXiv preprint arXiv:2508.01415*, 2025.

Siyao Li, Jiawei Gu, Shuai Liu, Kairui Hu, Zekun Li, Linjie Li, Chengcheng Tang, Po-Chen Wu, Ivan Shugurov, Lingni Ma, Michael Zollhoefer, Sizhe An, Abhay Mittal, Amy Zhao, Ranjay Krishna, Manling Li, Ziwei Liu, and Chuan Guo. Humanclaw: Can vision-language models act through a body?, 2026. URL https://arxiv.org/abs/2607.27180.

Wenjun Li, Dexun Li, Kuicai Dong, Cong Zhang, Hao Zhang, Weiwen Liu, Yasheng Wang, Ruiming Tang, and Yong Liu. Adaptive tool use in large language models with meta-cognition trigger. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 13346–13370, 2025.

Xiaoyu Ma, Lianyu Hu, Wenbing Tang, Zixuan Hu, Zeqin Liao, Zhizhen Wu, and Yang Liu. Brainmem: Brain-inspired evolving memory for embodied agent task planning. *arXiv preprint arXiv:2604.16331*, 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Nat McAleese, Rai Michael Pokorny, Juan Felipe Cerón Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. Llm critics help catch llm bugs. *arXiv preprint arXiv:2407.00215*, 2024.

Thomas O Nelson. Metamemory: A theoretical framework and new findings. In *Psychology of learning and motivation*, volume 26, pp. 125–173. Elsevier, 1990.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition. https://openai.com/index/gpt-5-6/, 2026. Official model announcement.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, 2026a. Official model announcement.

Qwen Team. Qwen3.6-35b-a3b: Agentic coding power, now open to all. https://qwen.ai/blog?id=qwen3.6-35b-a3b, 2026b. Official model announcement.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. *URL https://arxiv. org/abs/2303.11366*, 1:5, 2023.

Oriol Vinyals and Noam Shazeer. Gemini 3.5: Frontier intelligence with action. https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-5/, 2026. Official model announcement.

Yuyang Wanyan, Xi Zhang, Haiyang Xu, Haowei Liu, Junyang Wang, Jiabo Ye, Yutong Kou, Ming Yan, Fei Huang, Xiaoshan Yang, Weiming Dong, and Changsheng Xu. Look before you leap: A gui-critic-r1 model for pre-operative error diagnosis in gui automation. *arXiv preprint arXiv:2506.04614*, 2025.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *arXiv preprint arXiv:2505.23747*, 2025.

Tianyi Xiong, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. Llava-critic: Learning to evaluate multimodal models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. In *International Conference on Machine Learning (ICML)*, 2025b.

Di Zhang, Junxian Li, Jingdi Lei, Xunzhi Wang, Yujie Liu, Zonglin Yang, Jiatong Li, Weida Wang, Suorong Yang, Jianbo Wu, Peng Ye, Wanli Ouyang, and Dongzhan Zhou. Critic-v: Vlm critics help catch vlm errors in multimodal reasoning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

APPENDIX CONTENTS

A Use of large language models 14
B Reproducibility Statement 14
C Limitations 14
D Details of MetaSpatial-Bench 14
D.1 Benchmark Construction 14
D.2 Dataset Statistics 15
E Details of MetaSpatial-35K 15
E.1 Data Construction Pipeline 15
E.2 Dataset Statistics 17
F Training Details 18
G Additional Experimental Details and Results 19
G.1 MetaSpatial-Bench Evaluation 19
G.1.1 Experimental Setup 19
G.1.2 Additional Evaluation Metrics 20
G.2 Online Spatial Interaction Evaluation 21
G.2.1 Experimental Setup 21
G.2.2 Performance under Different Interaction Budgets 21
G.2.3 Outcome Transitions under Meta-Level Regulation 22
G.2.4 Capability Boundaries of Meta-Level Regulation 23
G.2.5 Inference Time and Computational Overhead 23
G.3 Effectiveness across Regulator Model Sizes 24
H Case Study 24
H.1 Visualization on MetaSpatial-Bench 24
H.2 Visualization on SpatialWorld 25

A USE OF LARGE LANGUAGE MODELS

Large language models (LLMs) were used during the preparation of this manuscript to assist with writing refinement, literature discovery, and research discussion. Specifically, LLMs were used to improve the clarity and organization of written sections, assist in identifying related works, support research ideation and discussion, and provide suggestions for drafting parts of the manuscript. All generated suggestions were carefully reviewed, verified, and revised by the authors. The authors take full responsibility for the final content, methodology, experimental results, and conclusions presented in this paper.

B REPRODUCIBILITY STATEMENT

We will open-source the complete data construction pipeline, MetaSpatial-35K, and MetaSpatial-Bench to facilitate reproducible research and future exploration of metacognitive spatial interaction. We will also release the training configurations and model-development details required to reproduce Spatial-JARVIS, as described in Appendix F. In addition, we will provide the complete evaluation framework for online spatial interaction benchmarks, including the integration interface with Spatial-JARVIS.

C LIMITATIONS

The erroneous decisions used to construct monitoring and control supervision are primarily collected from interaction rollouts of Qwen3.5-27B(Qwen Team, 2026a). Although these policy-generated errors reflect realistic cognitive failures during spatial interaction, some model-specific failure patterns may remain underrepresented. Nevertheless, Spatial-JARVIS consistently improves the performance of diverse policy models, demonstrating the transferability of the learned metacognitive regulation capabilities. Future work could explore more generalizable data collection pipelines to broaden the coverage of cognitive failures and further enhance the potential of meta-level regulation.

D DETAILS OF METASPATIAL-BENCH

D.1 BENCHMARK CONSTRUCTION

Existing spatial intelligence benchmarks primarily evaluate spatial understanding or task execution, without specifically assessing models' metacognitive monitoring and control capabilities during spatial interaction. To fill this gap, we introduce **MetaSpatial-Bench**, a static benchmark designed to independently evaluate these two capabilities.

Since underlying spatial cognition is difficult to quantify directly and objectively, we assess metacognitive capabilities through decision-level monitoring and correction, which provide more objectively verifiable evaluation targets. The broader effects of metacognitive regulation on long-horizon spatial interaction are further examined through online interaction benchmarks, such as SpatialWorld.

We manually select evaluation scenes that are completely disjoint from those used in MetaSpatial-35K and construct non-overlapping tasks with diverse spatial interaction requirements.

Following the Task Trajectory Synthesis procedure described in Section 3.2, we employ an oracle policy with access to privileged simulator states to generate reference trajectories until the task completion conditions are satisfied.

Decision-Level Sample Construction. Based on the reference trajectories, we select decision points at different stages of spatial interaction and manually construct erroneous candidate decisions in the simulator, alongside reasonable decisions, to represent potential failures during interaction.

Each evaluation sample contains the interaction context, a candidate decision, and the corresponding supervision for decision monitoring and correction.

For monitoring, the model is required to determine whether the candidate decision is reasonable. For control, the model is required to propose an appropriate corrected decision when the candidate decision is unreasonable.

D.2 Dataset Statistics

Scene and Task Selection. MetaSpatial-Bench contains 2,000 decision-level samples collected from 527 evaluation scenes. To examine monitoring and correction capabilities under different interaction requirements, we organize the benchmark into three categories: *Search*, *Navigate*, and *Interact*.

The Search category focuses on decisions related to environmental exploration and target discovery. The Navigate category evaluates navigation decisions toward target locations after the target has been identified, while the Interact category covers high-level decisions involving object manipulation.

The benchmark contains 1,200 reasonable and 800 unreasonable candidate decisions, with each stage contributing approximately one-third of the samples. This composition enables the evaluation of decision validity assessment and erroneous decision correction across distinct interaction requirements. Figure 4 presents the label and stage distributions.

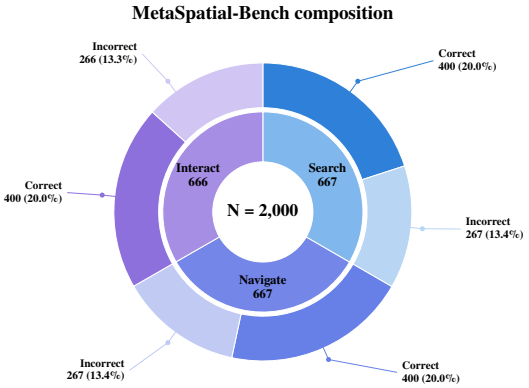


Figure 4: Statistics of MetaSpatial-Bench, showing the distribution of reasonable and unreasonable candidate decisions and the distribution of samples across the Search, Navigate, and Interact stages.

E Details of MetaSpatial-35K

In this section, we provide additional details on the construction of MetaSpatial-35K, including the data construction pipeline and dataset statistics.

E.1 Data Construction Pipeline

Task Trajectory Synthesis. We construct MetaSpatial-35K using two simulated environments, AI2-THOR and ProcTHOR (Kolve et al., 2017; Deitke et al., 2022), which provide diverse indoor scenes and support navigation and object interaction.

For each task, we initialize a scene, sample a set of task completion conditions, and employ Qwen3.5-27B (Qwen Team, 2026a) to generate a corresponding natural-language task instruction. The prompt used for instruction generation is shown in Fig. 5.

Following the formulation in Section 3.2, a task is represented as

$$\tau = (S, C, I), \tag{5}$$

where S denotes the initialized scene, $C = \{c_1, \ldots, c_m\}$ denotes the set of task completion conditions, and I denotes the corresponding task instruction.

Given a task, an oracle policy π^* generates a reference trajectory with privileged access to the ground-truth simulator state. Unlike the interaction policy, which makes decisions based on partial observations and interaction history, the oracle policy can access the actual positions, attributes, and interaction states of objects in the environment.

At each interaction step, the oracle policy identifies the remaining task objectives according to the current environment state s_t and generates corresponding high-level decisions, aiming to satisfy the task completion conditions in as few interaction steps as possible. Each decision is executed in the simulator, and the environment state is updated accordingly. This process continues until

all task completion conditions are satisfied or the trajectory terminates. The resulting reference trajectories provide task-directed interaction sequences and ground-truth environment information for subsequent supervision construction.

Monitoring Signal Construction. A straightforward approach to constructing monitoring supervision is to treat decisions from reference trajectories as positive examples and other decisions as negative examples. However, this approach may incorrectly penalize alternative decisions that differ from the reference trajectory but remain reasonable under the current interaction state.

To avoid learning rigid decision patterns, we construct an equivalent-decision set for each reference state. An equivalent decision is defined as a feasible alternative that achieves the same subgoal progress as the reference decision. Let s_t denote the ground-truth environment state and a_t^* denote the reference decision. We use the simulator to evaluate alternative decisions and construct

$$\mathcal{A}^{\text{eq}}(s_t) = \{a \mid a \text{ is feasible and achieves equivalent subgoal progress to } a_t^*\}. \tag{6}$$

For example, when an agent navigates toward an object located in its front-left direction, both moving left and moving ahead may be reasonable decisions if they achieve equivalent progress toward the current navigation subgoal.

To obtain negative examples that reflect realistic policy failures, we employ Qwen3.5-27B (Qwen Team, 2026a) to generate interaction rollouts from states along the reference trajectories. Compared with randomly sampled erroneous decisions, policy-generated deviations better reflect the spatial misconceptions, inaccurate task-progress judgments, and decision patterns that arise during actual interaction. Given a rollout decision a_t, we construct the binary monitoring label as

$$l_t = \begin{cases} 1, & a_t \in \mathcal{A}^{\text{eq}}(s_t), \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

This construction provides decision-level supervision that accounts for multiple reasonable alternatives, rather than requiring exact imitation of oracle trajectories.

```
You are a task instruction generator for an embodied AI agent operating in a 3D
indoor environment (AI2-THOR / ProcTHOR).

Given task completion conditions and privileged simulator information, generate
ONE natural-language task instruction that a human might naturally give to a
household robot.

The instruction should describe the desired task objectives rather than the
underlying simulator states or execution steps.

Requirements:
1. Cover all specified task completion conditions.
2. Use the exact object names provided in the conditions.
3. Generate natural, concise, and everyday household instructions.
4. Use privileged simulator information only to ensure that the instruction is
consistent with the environment and task.
5. Do NOT reveal privileged information, such as initial object locations,
hidden object states, simulator metadata, or the robot's internal state, unless
explicitly required by the task.
6. Do NOT describe navigation paths, intermediate actions, or step-by-step
execution plans.
7. Do NOT introduce additional task objectives or constraints.
8. Avoid technical terms, simulator-specific condition names, or unnatural
descriptions of state changes.

Examples of natural household instructions:

- "Please put the Cup in the Sink."

- "Please clean the Plate and put the Plate in the Cabinet."

Now generate an instruction for the following task.

[Task Completion Conditions]
{TASK_CONDITIONS}

[Object States and Locations]
{OBJECT_STATES_AND_LOCATIONS}

Output only ONE natural-language task instruction in English.
Do not include explanations, simulator information, or additional text.
```

Figure 5: Task instruction prompt.

Control Signal Construction. For rollout decisions identified as unreasonable, we further construct control supervision consisting of decision correction and belief revision. Decision correction provides a feasible alternative that can appropriately advance the current subtask. Instead of always using the original reference decision, we randomly select a correction target from the equivalent-decision set:

$$d_t^* \sim \text{Uniform}\left(\mathcal{A}^{\text{eq}}(s_t)\right). \tag{8}$$

This construction exposes the regulator to diverse valid corrections and avoids treating a particular reference decision as the only acceptable intervention. Belief revision addresses the cognitive errors that may underlie unreasonable decisions. During offline data construction, we compare the policy's expressed or inferred cognitive state with the corresponding ground-truth simulator state to identify discrepancies in spatial beliefs and task-progress beliefs.

For example, a policy may proceed to place a cup in the sink after an unsuccessful pickup attempt because it incorrectly believes that the cup is already in its hand. The simulator state reveals that the cup remains on the table. In this case, decision correction guides the policy to pick up the cup, while belief revision explicitly corrects the mistaken task-progress belief. Similarly, a policy may repeatedly search an incorrect location because its spatial belief about the target object's position is

inconsistent with the environment state. Belief revision provides feedback that corrects this discrepancy rather than merely redirecting the next decision.

These two components serve complementary purposes: decision correction addresses the current inappropriate decision, while belief revision helps prevent the underlying cognitive error from influencing subsequent interaction steps.

Subtask-based Rationale Generation. In addition to monitoring and control labels, we construct structured cognitive rationales to provide explicit supervision for task-progress understanding and decision-consequence reasoning.

Long-horizon spatial interaction typically requires the agent to complete multiple interdependent subtasks. The appropriateness of a decision therefore depends not only on the immediate environment observation, but also on the current task stage and the progress of preceding interactions. Understanding this task context further helps the model assess the current interaction policy's cognitive state.

We use Qwen3.5-Plus (Qwen Team, 2026a) to segment reference trajectories into meaningful subgoals and obtain task-stage supervision g_t verified by the ground-truth environment states. For each interaction state, the task-stage annotation identifies the current subtask and its progress. Based on these signals, Qwen3.5-Plus generates a two-stage structured rationale:

```
1. Task Stage Identification:
Recover the current task progress from the interaction history and
identify the subtask being executed.
2. Decision Consequence Inference:
Infer the potential state transition caused by the proposed
decision and assess whether it advances the current subtask.
```

The two components provide complementary reasoning supervision. Task Stage Identification establishes the agent's current objective, while Decision Consequence Inference evaluates the proposed decision with respect to that objective. For example, consider a task requiring the agent to pick up a cup and place it in a sink. If a pickup attempt fails, the cup remains on the table and the pickup subtask is still incomplete. A subsequent decision to proceed toward the sink can therefore be analyzed in terms of both the unfinished task stage and the consequence of leaving the cup behind.

This structured rationale differs from a generic explanation of a binary monitoring label. Rather than merely stating that a decision is unreasonable, it provides a task-grounded account of the current interaction stage and the consequences of the proposed decision.

Finally, we use Gemini 3.5 Flash (Vinyals & Shazeer, 2026) to filter samples containing reasoning that directly restates supervision labels, exhibits clear logical inconsistencies, or lacks sufficient task-relevant reasoning. The resulting dataset is represented as

$$\mathcal{D} = \{(h_t, a_t, \rho_t, l_t, u_t)\}, \tag{9}$$

where h_t denotes the interaction history, a_t denotes the proposed decision, ρ_t denotes the structured rationale, l_t denotes the monitoring label, and u_t denotes the corresponding control supervision.

E.2 Dataset Statistics

MetaSpatial-35K contains approximately 35K step-level samples constructed from 3,175 task trajectories across 1,410 scenes. As shown in Figure 6, the SFT subset is balanced between correct and incorrect decisions, while the MC-GRPO subset uses a 55:45 ratio. The reference trajectories have a median length of 23 interaction steps and cover diverse interaction types, objects, and rooms. For each sample, we retain up to 10 historical observation frames and 20 interaction steps, preserving sufficient interaction history to identify discrepancies between the spatial beliefs supported by accumulated observations and the agent's current cognitive state, while keeping multimodal input lengths manageable.

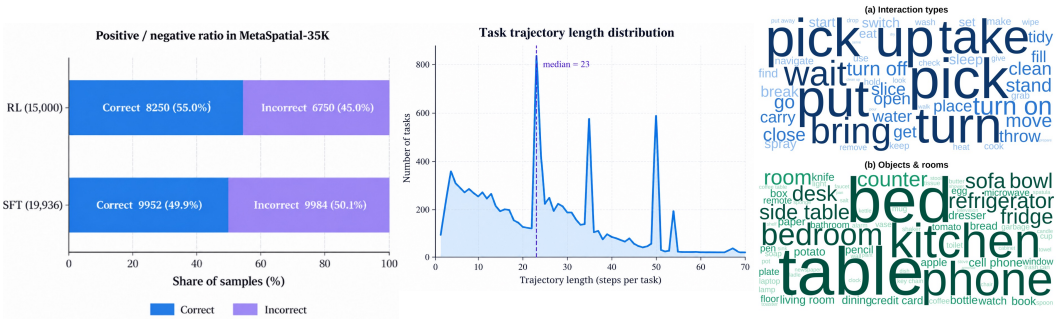


Figure 6: Statistics of MetaSpatial-35K. Left: distributions of correct and incorrect samples in the SFT and MC-GRPO subsets. Center: distribution of reference-trajectory lengths. Right: vocabularies of interaction types, objects, and rooms represented by word clouds.

F Training Details

All training experiments were conducted using 8 NVIDIA H200 GPUs, each with 141 GB of high-bandwidth memory. Spatial-JARVIS was initialized from Qwen3.5-9B (Qwen Team, 2026a) and trained in two stages. We first perform supervised fine-tuning (SFT) to establish structured metacognitive reasoning and monitoring-control output patterns. We subsequently apply MC-GRPO to further enhance the model's metacognitive monitoring and control capabilities through reinforcement learning.

Supervised Fine-Tuning (SFT). During SFT, we update all model parameters, including the language-model backbone, vision encoder, and vision-language aligner. Given the SFT dataset $\mathcal{D}_{\mathrm{SFT}} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i denotes the input context and y_i denotes the target monitoring or control response, we optimize the standard autoregressive negative log-likelihood objective:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{SFT}}}\left[\sum_{t=1}^{|y|}\log P_\theta(y_t \mid x, y_{<t})\right]. \tag{10}$$

We train the model for one epoch. We use the Adam optimizer with a peak learning rate of 5×10^{-6}, followed by cosine decay to 1×10^{-6}. The warmup ratio is set to 5%, with a weight decay of 0.1 and a maximum gradient norm of 1.0. Training is implemented using ms-swift with Megatron-LM. The main SFT hyperparameters are summarized in Table 9.

Monitoring-Control Group Relative Policy Optimization (MC-GRPO). We initialize MC-GRPO from the SFT checkpoint and use the initial SFT model as the KL reference policy. For each input x, the regulator samples a group of eight candidate outputs $\{o_i\}_{i=1}^{G}$. Group-relative advantages are then used to optimize the regulator. The training hyperparameters and reward coefficients are summarized in Table 10.

Table 9: SFT hyperparameters.

Configuration	Value
Training epochs	1
Maximum sequence length	12,288
Global batch size	64
Peak learning rate	5×10^{-6}
Final learning rate	1×10^{-6}
Learning rate schedule	Cosine
Warmup ratio	0.05
Optimizer	Adam
Adam (β_1, β_2)	$(0.9, 0.95)$
Weight decay	0.1
Gradient clipping	1.0

Table 10: MC-GRPO hyperparameters.

Configuration	Value
Samples per prompt	8
Prompt batch size	32
Maximum prompt length	8,192
Maximum response length	8,192
Maximum context length	16,384
Learning rate	1×10^{-6}
Learning rate schedule	Constant
Optimizer	Adam
Adam β_1	0.9
Adam β_2	0.98
Weight decay	0.1
KL coefficient	0.10
λ_f	0.1
λ_{mc}	0.9
α	0.3

Hierarchical Reward Implementation. The hierarchical reward consists of format, monitoring, and control rewards, following the formulation in the main paper:

$$r_{\text{total}} = \lambda_f r_f + \lambda_{mc} \mathbb{I}[\hat{l} = l] \left(\alpha + (1 - \alpha) \mathbb{I}[\hat{d} \in \mathcal{A}^{\text{eq}}(s_t)] \, \text{Judge}(\hat{b}, b^*) \right). \tag{11}$$

Specifically, r_f is determined by whether the generated response follows the required structured output format, consisting of Rationale, Monitor, and Control. The monitoring judgment is evaluated by checking whether $\hat{l}$ matches the ground-truth label l. For control, the generated corrective decision $\hat{d}$ is checked against the equivalent-decision set $\mathcal{A}^{\text{eq}}(s_t)$, while Qwen3.5-Plus serves as the LLM judge to evaluate the semantic consistency between the generated belief correction $\hat{b}$ and the reference belief correction b^*. The reward coefficients are provided in Table 10. The training reward curves are shown in Figure 7.

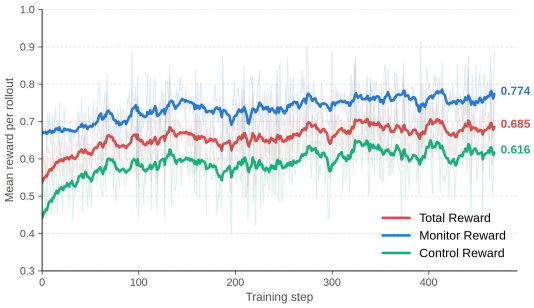


Figure 7: MC-GRPO reward curves during training.

G Details of Experiment Results

This section provides additional experiment results and analyses for Spatial-JARVIS, including static regulation evaluation, online spatial interaction, inference overhead, and deeper analysis.

G.1 MetaSpatial-Bench Evaluation

G.1.1 Experimental Setup

We evaluate Spatial-JARVIS on MetaSpatial-Bench to assess its metacognitive monitoring and control capabilities at each decision step of spatial interaction. Specifically, the evaluation measures whether a model can assess the reasonableness of a candidate decision based on the agent's current spatial cognitive state and provide appropriate corrective feedback when necessary.

We compare Spatial-JARVIS with representative open-source and closed-source multimodal language models. The closed-source models include Claude-Sonnet-5(Anthropic, 2026), Qwen3.5-Plus(Qwen Team, 2026a), Gemini 3.5 Flash-Lite(Vinyals & Shazeer, 2026), and GPT-5.6-Luna(OpenAI, 2026). The open-source models include Qwen3.5-9B(Qwen Team, 2026a), Qwen3.5-27B(Qwen Team, 2026a), and Qwen3.6-35B-A3B(Qwen Team, 2026b). All models are evaluated under the same evaluation protocol. We report monitoring and control accuracies in the main-text evaluation table, calculated as follows:

Table 11: Additional monitoring evaluation on MetaSpatial-Bench. P, R, and F1 are computed for invalid decisions; FP and FN denote their proportions among all samples.

Model	Acc	P	R	F1	FP	FN
Closed-source						
Claude-Sonnet-5	59.9	49.9	45.0	47.3	18.1	22.0
Qwen3.5-Plus	57.5	46.1	37.2	41.2	17.4	25.1
Gemini 3.5 Flash-Lite	54.9	40.4	26.9	32.3	15.8	29.1
GPT-5.6-Luna	60.5	51.0	31.6	39.0	12.2	27.4
Open-source						
Qwen3.5-9B	51.0	38.4	37.0	37.7	23.8	25.2
Qwen3.5-27B	54.7	43.0	40.0	41.4	21.3	24.0
Qwen3.6-35B-A3B	55.6	42.4	30.8	35.7	16.7	27.7
Spatial-JARVIS	**75.1**	**75.9**	**55.2**	**64.0**	**7.0**	**17.9**

Monitoring Accuracy. Monitoring accuracy measures whether the model correctly identifies the validity of a candidate decision under the given interaction context. Let N denote the total number of evaluation samples, l_i the ground-truth monitoring label, and $\hat{l}_i$ the predicted label. Monitoring accuracy is computed as

$$\mathrm{Acc}_M = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[\hat{l}_i = l_i]. \tag{12}$$

Control Accuracy. Control accuracy measures whether the model provides an appropriate corrected decision for erroneous candidate decisions. Let $\mathcal{N}$ denote the set of samples requiring correction. We define

$$\mathrm{Acc}_C = \frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}} \mathbb{I}[\hat{d}_i \in \mathcal{D}_i^{\mathrm{valid}}], \tag{13}$$

where $\mathcal{D}_i^{\mathrm{valid}}$ denotes the set of acceptable corrected decisions for sample i.

The benchmark directly evaluates decision monitoring and decision correction. Belief revision is not treated as a separate accuracy metric in the static benchmark; its contribution is further examined through online interaction and regulation ablations.

G.1.2 Additional Evaluation Metrics

While monitoring accuracy provides an overall measure of decision assessment, it does not fully characterize a regulator's ability to identify erroneous decisions without unnecessarily intervening in reasonable ones. To provide a more comprehensive analysis of metacognitive monitoring, we introduce additional metrics for the binary monitoring task, treating invalid decisions as the positive class. Specifically, we report precision, recall, and F1 score, calculated as follows:

$$\mathrm{Precision}_{\mathrm{invalid}} = \frac{TP}{TP+FP}, \tag{14}$$

$$\mathrm{Recall}_{\mathrm{invalid}} = \frac{TP}{TP+FN}, \tag{15}$$

$$\mathrm{F1}_{\mathrm{invalid}} = \frac{2TP}{2TP+FP+FN}, \tag{16}$$

where TP denotes invalid decisions correctly identified as invalid, FP denotes valid decisions incorrectly classified as invalid, and FN denotes invalid decisions incorrectly classified as valid. Precision measures the reliability of error identification, while recall reflects the ability to detect erroneous decisions. F1 provides a balanced measure of these two aspects. We additionally report the proportions of FP and FN among all evaluation samples, reflecting unnecessary interventions and missed erroneous decisions, respectively. The results are presented in Table 11.

As shown in Table 11, Spatial-JARVIS achieves 75.9% precision, 55.2% recall, and 64.0% F1, outperforming the compared models across all three metrics. Notably, it reduces the proportion of false positives to 7.0%, compared with 12.2–23.8% for the other models, while also achieving the lowest false-negative proportion of 17.9%. These results demonstrate that Spatial-JARVIS can more reliably identify erroneous decisions while reducing unnecessary interventions on reasonable decisions, highlighting its effectiveness in metacognitive monitoring.

G.2 Online Spatial Interaction Evaluation

G.2.1 Experimental Setup

We conduct online spatial interaction experiments to evaluate whether Spatial-JARVIS can improve task execution when integrated with different policy models. Our primary evaluation is conducted on SpatialWorld(Gao et al., 2026), which covers diverse task categories, including Daily, Work, Entertain, Travel, and Social, involving different combinations of environmental exploration, navigation, and object interaction. We focus on its Physical subset, as the Digital subset falls outside our task definition of spatial interaction. We evaluate six policy models: Qwen3.5-Plus(Qwen Team, 2026a), Qwen3.5-27B(Qwen Team, 2026a), Qwen3.6-35B-A3B(Qwen Team, 2026b), GPT-5.6-Luna(OpenAI, 2026), Gemini 3.5 Flash-Lite(Vinyals & Shazeer, 2026), and Claude-Sonnet-5(Anthropic, 2026). We additionally evaluate on EmbodiedBench-Nav(Yang et al., 2025b), selecting its navigation subset because the other task categories primarily emphasize planning rather than the high-level interaction considered in this work.

For each policy model, we compare its original task performance with that obtained after integrating Spatial-JARVIS. The original policy remains responsible for generating task-directed decisions, while Spatial-JARVIS serves as an independent Meta-Level Regulator. We use task success rate (TSR) as the primary evaluation metric:

$$\mathrm{TSR} = \frac{N_{\mathrm{success}}}{N_{\mathrm{total}}}, \tag{17}$$

where N_{success} and N_{total} denote the numbers of successfully completed tasks and total evaluation tasks, respectively. Task completion is determined by the benchmark's environment-level completion conditions rather than the policy's self-reported judgment. We further analyze interaction efficiency using SPL and the average number of interaction steps for successfully completed tasks.

G.2.2 Performance under Different Interaction Budgets

To further investigate how Meta-Level regulation affects long-horizon spatial interaction, we analyze task success rates under different maximum interaction step budgets on SpatialWorld. This evaluation examines whether Spatial-JARVIS improves task completion within limited interaction budgets and whether its benefits persist as more interaction steps become available.

As shown in Figure 8, the original policies can complete a subset of tasks within relatively few interaction steps, but their success rates tend to plateau as the step budget increases. This suggests that simply allowing more interaction steps is insufficient to resolve many remaining task failures. In long-horizon spatial interaction, an agent may gradually develop inaccurate beliefs about the environment or task progress, leading to repeated exploration, ineffective operations, or decisions based on falsely completed subtasks. Once such errors persist in the interaction history, additional steps may reinforce the same ineffective behavior rather than

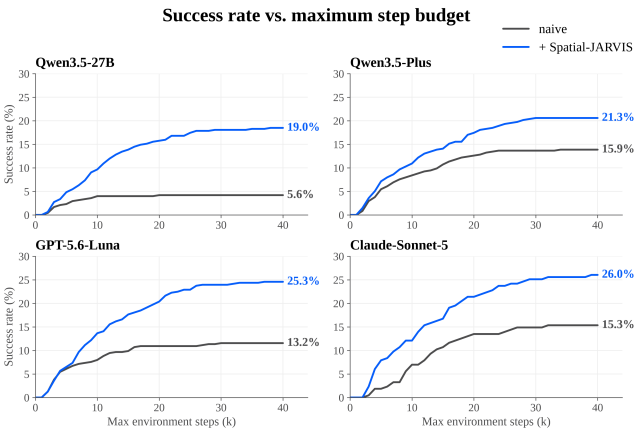


Figure 8: Task success rates under different maximum interaction budgets, with and without Spatial-JARVIS.

bring the agent closer to task completion.

In contrast, Spatial-JARVIS consistently achieves higher success rates across different interaction budgets, with particularly pronounced improvements as longer trajectories are permitted. For example, Qwen3.5-27B reaches approximately 5% success within the first 10 steps and shows little further improvement, whereas its Spatial-JARVIS-enhanced counterpart continues to improve, reaching 19.0% at 40 steps. Similar trends are observed for GPT-5.6-Luna and Claude-Sonnet-5, whose success rates increase from 13.2% to 25.3% and from 17.0% to 26.4%, respectively, after introducing Spatial-JARVIS. These findings suggest that the effectiveness of Meta-Level regulation extends beyond reducing the number of interaction steps: by identifying erroneous decisions and revising the underlying cognitive states, Spatial-JARVIS helps prevent agents from remaining trapped in ineffective interaction patterns, enabling them to make better use of additional interaction opportunities and progress through longer task trajectories.

G.2.3 Outcome Transitions under Meta-Level Regulation

To further understand how Meta-Level regulation changes task outcomes beyond aggregate success rates, we compare the outcomes of the same evaluation tasks before and after integrating Spatial-JARVIS on SpatialWorld. We categorize these outcome transitions into four cases: *Preserved* (Success → Success), *Regressed* (Success → Failure), *Recovered* (Failure → Success), and *Unresolved* (Failure → Failure). The proportions of these cases are calculated relative to the total number of evaluated tasks for each policy model. This analysis allows us to distinguish improvements obtained by recovering originally failed tasks from potential regressions on tasks that the original policy could already complete.

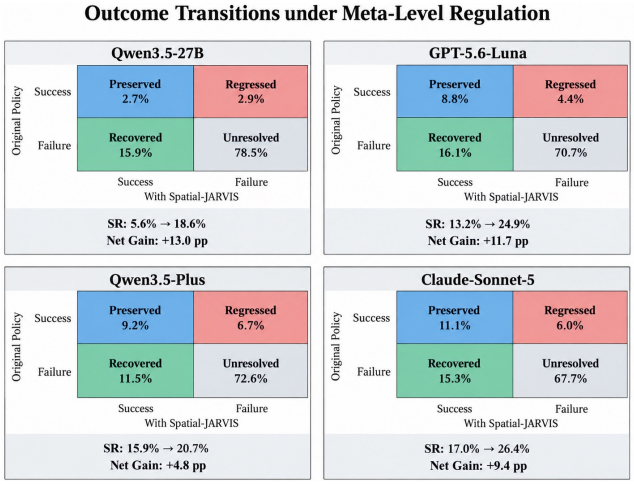


Figure 9: Outcome transitions after integrating Spatial-JARVIS. SR denotes task success rate.

As shown in Figure 9, Meta-Level regulation can influence task outcomes in both directions, as corrective feedback may affect not only the current decision but also the cognitive states underlying subsequent interactions. As reflected by the static evaluation results on MetaSpatial-Bench, the regulator does not always provide accurate monitoring and control signals. An inaccurate monitoring judgment or belief revision may occasionally introduce erroneous beliefs into an otherwise successful trajectory, potentially leading to downstream failures. However, such regressions account for only 2.9–6.7% of the evaluated tasks across the four policy models. In contrast, Spatial-JARVIS recovers 11.5–16.1% of the evaluated tasks that originally failed, demonstrating that the benefits of corrective regulation substantially outweigh these occasional regressions. For example, Qwen3.5-27B achieves a recovery proportion of 15.9% with only 2.9% regression, yielding a net recovery margin of 13.0 percentage points (recovery minus regression). These results suggest that many interaction failures can be alleviated by identifying and revising erroneous spatial beliefs or misjudged task progress, allowing the original policy to better utilize its existing capabilities without direct policy optimization. We further investigate the capability boundaries underlying these outcome changes in Appendix G.2.4.

G.2.4 Capability Boundaries of Meta-Level Regulation

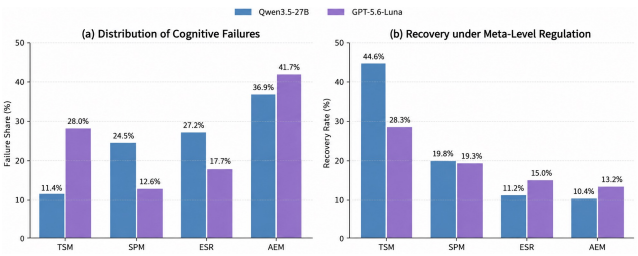


Figure 10: Capability boundaries of Meta-Level regulation on SpatialWorld.

To investigate which cognitive failures can be effectively addressed by Meta-Level regulation, we categorize the original policy failures into four types: *Task-State Misjudgment*, where the agent incorrectly assesses task progress or prematurely abandons a task; *Subtask-Plan Misalignment*, where the agent fails to organize the remaining subtasks coherently; *Erroneous Spatial Representation*, where the agent maintains inaccurate or incomplete spatial beliefs; and *Action-Effect Miscalibration*, where the agent incorrectly predicts or assesses the consequences of its actions. Figure 10 presents the distribution and recovery rates of these failures for Qwen3.5-27B and GPT-5.6-Luna. Such cognitive errors may persist throughout the interaction without being recognized or corrected by the original policy, further highlighting the importance of introducing independent Meta-Level monitoring and regulation.

As shown in Figure 10, Task-State Misjudgment exhibits the highest recovery rates of 44.6% and 28.3% for the two policies, respectively, followed by Subtask-Plan Misalignment at 19.8% and 19.3%. These results suggest that learning to identify and revise task-progress and subtask-plan inconsistencies is relatively easier, as they can often be inferred from explicit interaction history. In contrast, Erroneous Spatial Representation and Action-Effect Miscalibration exhibit lower recovery rates. This suggests that learning to reconstruct the environment from partial interaction history and proactively maintain accurate spatial beliefs is relatively more challenging, highlighting a promising direction for further enhancing this ability in Meta-Level regulation.

G.2.5 Inference Time and Computational Overhead

To assess the computational overhead introduced by Spatial-JARVIS, we compare the inference time and token consumption of the original policies and their Spatial-JARVIS-enhanced counterparts. Since the timing measurements for different models were collected at different times, variations in API latency, service throughput, and hardware availability may affect the observed wall-clock time. Therefore, these results are intended to provide an approximate characterization of the additional inference overhead rather than a strictly controlled comparison of inference speed across models. We focus on successfully completed episodes, as original policies may prematurely terminate failed trajectories by abandoning the task or mistakenly believing that the task has already been completed, making comparisons over all episodes potentially misleading. We report the average wall-clock time per interaction step, the total time per successful episode, and the total token consumption per episode. For the regulated setting, token consumption includes all model calls from both the original policy and Spatial-JARVIS throughout the interaction process.

As shown in Table 12, integrating Spatial-JARVIS introduces additional regulator inference, but its impact on overall task execution time remains relatively moderate. The average episode time increases by 0.1 and 0.3 minutes for Qwen3.5-Plus and GPT-5.6-Luna, respectively, while Qwen3.5-27B exhibits a slight reduction from 0.8 to 0.7 minutes. This observation highlights an important characteristic of Meta-Level regulation: although additional inference is required to monitor and correct candidate decisions, timely belief revision and decision correction can reduce redundant exploration and ineffective operations, partially offsetting the additional computational cost at the episode level. Consequently, the introduction of an independent regulator does not necessarily lead to a proportional increase in end-to-end interaction time. More importantly, these modest changes in execution time are accompanied by substantial improvements in task success rates, demonstrating that Spatial-JARVIS can enhance the reliability of long-horizon spatial interaction while maintaining practical inference efficiency.

Table 12: Inference time and computational overhead on SpatialWorld. Results are reported for successfully completed episodes. Each entry compares the original policy with its Spatial-JARVIS-enhanced counterpart (Naive → + Spatial-JARVIS), with the absolute change shown in parentheses. Token consumption includes all model calls within an episode.

Policy Model	Step Time (s)	Episode Time (min)	Tokens / Episode
Qwen3.5-Plus	$8 \rightarrow 8\ (\pm 0)$	$0.9 \rightarrow 1.0\ (+0.1)$	$25\text{k} \rightarrow 31\text{k}$
GPT-5.6-Luna	$6 \rightarrow 9\ (+3)$	$0.6 \rightarrow 0.9\ (+0.3)$	$24\text{k} \rightarrow 42\text{k}$
Qwen3.5-27B	$4 \rightarrow 4\ (\pm 0)$	$0.8 \rightarrow 0.7\ (-0.1)$	$16\text{k} \rightarrow 15\text{k}$

Table 13: Static evaluation of general-purpose models and Spatial-JARVIS regulators on MetaSpatial-Bench. Values are monitoring and control accuracies.

Model	Monitor (%)	Control (%)
Qwen3.5-27B	54.72	33.12
GPT-5.6-Luna	60.47	38.74
Spatial-JARVIS-2B (ours)	63.85	39.24
Spatial-JARVIS-9B (ours)	**75.06**	**52.04**

Table 14: Online task success rates on SpatialWorld with Qwen3.5-27B as the fixed interaction policy and different regulators.

Regulator	TSR (%)
None	5.6
Qwen3.5-27B	7.7
Qwen3.5-Plus	12.0
Spatial-JARVIS-2B (ours)	12.8
Spatial-JARVIS-9B (ours)	**19.0**

G.3 Effectiveness across Regulator Model Sizes

To explore the potential of Spatial-JARVIS for more efficient deployment, we further train a lightweight 2B regulator using the same two-stage training framework. We evaluate its monitoring and control capabilities on MetaSpatial-Bench and its online regulation effectiveness on SpatialWorld, using Qwen3.5-27B as the fixed Object-Level policy.

As shown in Tables 13 and 14, Spatial-JARVIS-2B outperforms the compared general-purpose models in both static metacognitive evaluation and online spatial interaction, despite its substantially smaller model size. Although its performance remains below that of Spatial-JARVIS-9B, these results demonstrate that the proposed training framework can establish effective monitoring and control capabilities even in a lightweight regulator. Notably, the 2B regulator can improve the spatial interaction performance of a substantially larger 27B policy without modifying its parameters. This further highlights the potential of dedicated Meta-Level training to enhance spatial interaction through a lightweight, independent regulator, while increased regulator capacity can further strengthen its monitoring and control capabilities.

H Case Study

H.1 Visualization on MetaSpatial-Bench

We present two qualitative examples from MetaSpatial-Bench to illustrate the metacognitive capabilities of Spatial-JARVIS. As shown in Figure 11, both compared models consider moving forward a reasonable decision, overlooking that the current direction provides limited opportunities to discover the target. Spatial-JARVIS instead reasons about the expected consequence of the candidate action, identifies the uninformative exploration behavior, and redirects the policy toward an unexplored direction.

As shown in Figure 12, the compared models incorrectly reject the `DONE` decision by interpreting the current observation without considering the accumulated interaction history. By integrating previ-

ous pickup and cleaning actions with visual changes across observations, Spatial-JARVIS correctly infers that the pot has been cleaned and is currently held by the agent, validating task completion.

These examples demonstrate that effective spatial interaction requires reasoning over action consequences and evolving cognitive states rather than evaluating decisions from isolated observations.

H.2 VISUALIZATION ON SPATIALWORLD

Figures 13 and 14 illustrate how Spatial-JARVIS regulates erroneous cognitive states during online spatial interaction.

In Figure 13, the policy has already observed the trash can location in previous interactions, but fails to incorporate this accumulated spatial information and selects an unnecessary exploratory action. Spatial-JARVIS identifies that the current subtask is navigation toward the previously located trash can, revises the agent's spatial belief, and redirects the policy toward the correct region. This example demonstrates how Meta-Level regulation prevents ineffective exploration caused by inconsistent spatial beliefs.

In Figure 14, the policy correctly identifies the phone but incorrectly estimates that it is immediately interactable. Spatial-JARVIS recognizes that the current distance is insufficient for interaction and corrects the underlying action-effect belief by encouraging further movement before attempting pickup. As a result, the agent successfully completes the interaction after reaching an appropriate distance.

These examples show that Spatial-JARVIS does not merely replace individual actions, but regulates the cognitive states underlying decision making, including where the agent believes the target is located and whether the current state supports successful interaction.



Task Name: I need it running, so please turn on the laptop and wait until you can tell it's powered on.

History Decision: **None**

Interactive Policy Decision: **Move Ahead**

Spatial-JARVIS ✓

[Rationale]

The current frame shows a bedroom with a sofa, bed, and window, but no laptop is visible on any surface. Since the target hasn't been spotted yet, the agent remains in ***"Find the Laptop"*** stage.
Moving forward into the open space is premature; the agent should first rotate to scan the room's other sides, particularly near the door or unseen corners, before navigating.

[Monitor]

Decision is not reasonable.

[Control]

Belief Revision: Still searching for the laptop, which isn't in view.

Decision Correction: Rotate Left

Claude-Sonnet-5 ✗

[Rationale]

The current sub-goal is to find and navigate to the laptop, but the current observation shows no laptop in view—only a door, couch, bed, window, and plant. Moving ahead into the room is a reasonable exploratory action since no laptop is visible yet, and there's no obstruction blocking forward movement.

[Monitor]

Decision is reasonable.

[Control]

There is no need to control. Continue moving forward to explore the room.

Qwen3.5-27B ✗

[Rationale]

The agent is in a bedroom and the task is to turn on a laptop. The current view shows a door, a couch, a bed, and a window, but no laptop is visible. The planned action Move Ahead will move the agent forward into the room, which is a reasonable exploratory step to locate the laptop.

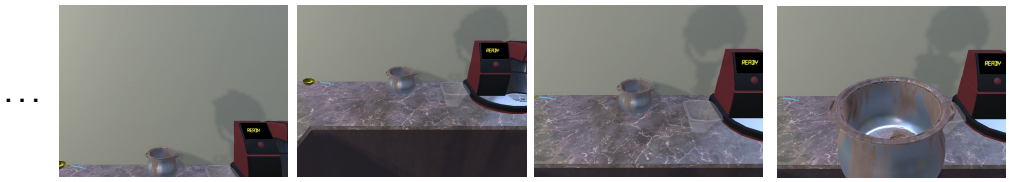
[Monitor]

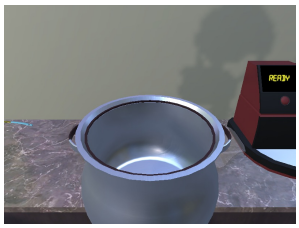
Decision is reasonable.

[Control]

No Control Signal

Figure 11: Correcting an uninformative exploration decision on MetaSpatial-Bench. Spatial-JARVIS identifies that moving ahead is unlikely to reveal the target laptop and redirects the policy to inspect an unexplored direction.





Task Name: Please clean the dirty pot and pick it up.

History Decision: **LookDown – Move Ahead – Pick up Pot – Clean Pot**

Interactive Policy Decision: **Done**

Spatial-JARVIS

[Rationale]

The pot floats centrally in the frame, confirming it is held, and its interior is now clean and shiny compared to the dirty state in earlier frames. This proves the pickup and cleaning actions succeeded. Since the instruction requires cleaning and picking up the pot, and both are visually verified as complete, declaring DONE is the correct final step.

[Monitor]

Decision is reasonable.

[Control]

None

Claude-Sonnet-5

[Rationale]

The pot is now clean. However, ... Looking at the current observation, the pot is still resting on the counter surface, not floating in the agent's hand. The pickup step was attempted earlier but the frame at that time still showed the pot on the counter, and the current observation confirms the pot is still on the counter, not held. Since the task requires holding the pot after cleaning, and it is not currently held, calling DONE now would leave the task incomplete.

[Monitor]

Decision is not reasonable.

[Control]

Pick up the pot.

Qwen3.5-27B

[Rationale]

The current observation shows a clean pot on the counter, indicating that the agent is not holding it. Although the pot has been cleaned, the task also requires picking it up afterward. Therefore, calling DONE would be premature; the agent should execute “pick up the pot” to complete the task.

[Monitor]

Decision is not reasonable.

[Control]

Pick up the pot.

Figure 12: Verifying task completion through historical and visual evidence on MetaSpatial-Bench. Spatial-JARVIS integrates action history and current observations to recognize that the cleaned pot is held by the agent and validates the DONE decision.



Figure 13: **Correcting erroneous spatial beliefs during online interaction.** Spatial-JARVIS identifies that the target location has already been observed in previous steps and redirects Qwen3.5-27B from unnecessary exploration toward the known trash-can location.



Figure 14: **Correcting Action-effect belief failure during online interaction.** Spatial-JARVIS recognizes that the phone is outside the current interaction range and replaces a premature pickup attempt with a movement action that enables successful interaction.