THE SPATIAL HARNESS: FROM FALLIBLE VISUAL TOOLS TO RELIABLE AGENTIC SPATIAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

While vision-language models (VLMs) excel at semantic understanding, they struggle to locate evidence from videos or multi-view images, and express it as precise spatial measurements, such as object sizes and distances. Visual tools can supply these locations and measurements, yet their outputs remain fallible and may introduce plausible but incorrect observations into downstream reasoning. Reliable tool-assisted reasoning therefore requires verifying these observations before accepting them as evidence and seeking alternatives when they are rejected or insufficient. Building on this principle, we introduce SPATIAL HARNESS, which models visual tools as stochastic observation channels and formalizes evidence acquisition as an output-feedback control process. Implemented as a planner–executor–verifier loop, the planner is the acquisition controller, proposing primary and backup routes for the executor to instantiate, while verification and memory updates jointly provide observer-like interpretation of tool outputs. Updated memory guides replanning for evidence still missing, while routes retaining no evidence trigger backup execution. To train the roles for distinct responsibilities while reducing reward conflicts, we first optimize specialized teachers on offline role-local contexts containing successful and failed tool outputs. We consolidate their expertise into one role-conditioned agent through multi-teacher on-policy distillation, combining teacher feedback on student rollouts for capability acquisition with task-reward reinforcement learning for refinement. On ReVSI and Ego3D-Bench, the harness improves the accuracy of open-weight models and closed-source APIs, outperforming spatially specialized models. Post-training yields further gains by improving collaborations between agents and imperfect tools.

1 INTRODUCTION

Spatial reasoning over images and videos of physical scenes requires understanding object locations and spatial relationships, which support embodied tasks such as navigation and manipulation. Vision-language models (VLMs) combine visual understanding and world knowledge with reasoning to comprehend tasks and seek relevant evidence. However, long videos and multiple viewpoints expand the input context and scatter task-relevant observations across frames, making exhaustive inspection difficult. Even when relevant observations are found, standard VLM responses do not reliably provide precise object locations, distances, or cross-view correspondences. These limitations motivate explicit spatial evidence that an agent can inspect and use in reasoning.

Existing spatial reasoning methods differ in how they acquire spatial evidence and for reasoning. One line of work internalize geometry into VLMs through spatial information, 3D-aware representations, memory, or test-time adaptation (Chen et al., 2024; Ranasinghe et al., 2024; Chen et al., 2026d; Liu et al., 2026; Li et al., 2026a). However, these methods map the spatial distribution of training dataset into the model's spatial knowledge, limiting direct inspection of the underlying evidence and flexibility in out-of-distribution scenarios (Zhang et al., 2026a). To address these limitations, recent approaches expose intermediate spatial observations through structured spatial codes (Chen et al., 2026b), inference-time visual tools, and executable programs (Marsili et al., 2025; Zhang et al., 2026b; Chen et al., 2026c; Cho et al., 2026; Dai et al., 2026). These interfaces make object-level observations and geometric estimates available for composition, and reuse. Yet exposing observations alone does not ensure reliable and sufficient evidence for reasoning.

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.

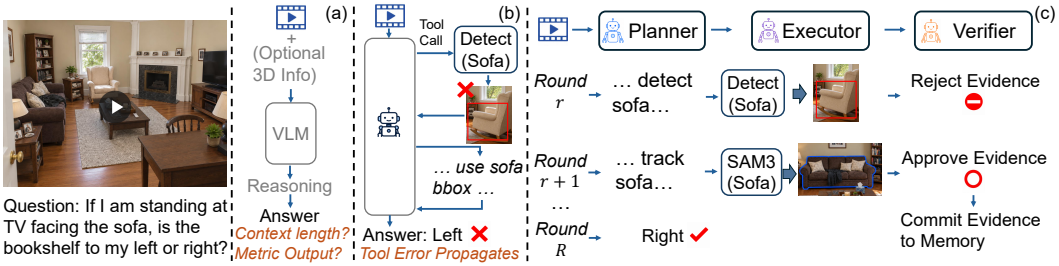

Figure 1: **Comparison of spatial reasoning paradigms.** (a) VLMs reason directly from visual inputs without an explicit tool-evidence interface. (b) Traditional spatial tool use pipelines leaves the fallible tool outputs *(i.e. the rocking chair for a sofa detection)* unchecked, harming the downstream reasoning. (c) SPATIAL HARNESS rejects faulty tool outputs, use alternative tool calls to find the real evidence, and commits the accepted observation to memory for final reasoning.

Learned visual tools, such as detectors, segmentors and depth estimators, infer scene properties from partial observations. However, they may fail to return usable evidence or produce plausible but incorrect outputs. Missing results leave evidence requirements unresolved, while plausible errors can pass unnoticed into subsequent reasoning. In Figure 1(b), the detector selects a rocking chair when asked to locate the sofa. The returned bounding box is accepted as evidence of the sofa's location without checking whether it encloses the requested object. Consequently, the resulting spatial judgment is based on the wrong object, leading to the incorrect answer shown in Figure 1(b).

To address these weaknesses, we propose SPATIAL HARNESS, which makes fallible visual tools usable through agent-side evidence control. We model tool returns through stochastic observation channels within an output-feedback process. On the tool side, functional tool sets provide alternatives that preserve the evidence goal while varying tools or evidence specifications. On the agent side, the planner organizes these alternatives into primary and backup routes, and the executor converts the selected route into an executable program. The verifier applies semantic and spatial checks before observations enter memory, as illustrated in Figure 1(c). Memory retains committed evidence and unresolved candidates, both of which inform the planner about evidence already acquired. A route yielding no retained evidence activates the next backup from the existing list. The first route retaining evidence ends this traversal, and remaining evidence gaps trigger replanning from the updated memory. Final reasoning integrates both memory partitions with the visual input and question to produce an answer.

Training SPATIAL HARNESS requires optimizing its planner, executor, and verifier for distinct responsibilities. Directly optimizing a shared model across heterogeneous tool demands can introduce conflicting rewards, so we first train role-specialized teachers using DAPO with responsibility-aligned rewards. We then freeze these teachers and consolidate their expertise into a unified role-conditioned student through Multi-Teacher On-Policy Distillation (MOPD) (Ma et al., 2026; Lian et al., 2026), which provides role-specific teacher feedback on the student's own rollouts. We combine this guidance for capability acquisition with DAPO task-reward updates for refinement. Modeling visual tools as stochastic observation channels supports offline role-local training on diverse successful and failed outputs. This reduces reliance on costly live evidence collection and eliminates waits for visual tools during optimization. Experiments show that SPATIAL HARNESS improves pretrained open-weight models and closed-source APIs on multi-view and video spatial reasoning benchmarks. The inference-only harness outperforms spatially specialized and tool-assisted baselines, while training further improves the student's adaptation to the harness and visual context.

Our contributions are summarized as follows.

- We identify two fundamental reliability issues in tool-assisted spatial reasoning — unrecoverable failures of fixed tool routes and silent propagation of unverified observations, then formulate evidence acquisition through stochastic observation channels and output feedback.
- We introduce Spatial Harness, which couples primary and backup routes over functional tool sets with verification-controlled memory. Semantic and spatial checks govern evidence retention, and memory feedback guides recovery through backup execution and replanning.
- We train role-specialized teachers on offline contexts with diverse tool outcomes and role-specific rewards. MOPD and DAPO consolidate them into a unified role-conditioned agent, adapting supervision to student competence for capability acquisition and refinement.

2 Related Work

Spatial Reasoning. Spatial reasoning has expanded from relational VQA (Johnson et al., 2017; Hudson & Manning, 2019) to metric grounding and cross-view understanding. Spatial supervision, geometry-aware representations, and test-time memory strengthen VLMs' spatial knowledge (Chen et al., 2024; Ranasinghe et al., 2024; Chen et al., 2026d; Liu et al., 2026; Li et al., 2026a). Other approaches expose evidence through spatial codes, perception tools, and executable programs (Chen et al., 2026b; Marsili et al., 2025; Chen et al., 2026c; Zhang et al., 2026b). Further, SpatialClaw and S-Agent support iterative inspection, computation, and evidence accumulation over tool interfaces(Cho et al., 2026; Dai et al., 2026). Motivated by tool fallibility, we make semantic and spatial evidence control explicit rather than relying on backbone-driven reflection. Recovery therefore seeks usable substitutes for rejected observations, rather than merely filling gaps left by missing outputs.

Tool Use and Agent Training. Reasoning–action prompting and learned API invocation extend LLMs beyond parametric knowledge (Yao et al., 2022; Schick et al., 2023; Qin et al., 2024; Patil et al., 2024). Multimodal agents integrate visual models and interactive tools (Shen et al., 2023; Wu et al., 2023; Yang et al., 2023; He et al., 2024). Subsequent work uses reinforcement learning and preference optimization to improve multi-step tool use (Wu et al., 2026; Li et al., 2025; Kumar et al., 2025). AgentFlow optimizes its planner with live execution and verification feedback while keeping other modules fixed (Li et al., 2026b). Our fallibility assumption makes examining semantic and spatial validity a central learning objective for the verifier. We therefore train planning, execution, and verification on role-local contexts containing both correct and faulty tool observations, with role-specific supervision.

Tool Reliability and Verification. Agent reliability research examines prompt injection, tool-feedback attacks, and retrieval poisoning (Greshake et al., 2023; Debenedetti et al., 2024; Zheng et al., 2026; Zou et al., 2025). Another line uses critique and self-correction to revise reasoning and answers through feedback (Shinn et al., 2023; Madaan et al., 2023; Gou et al., 2024). Beyond adversarial corruption, *Tools Fail* investigates silent errors arising within tools, including natural visual-tool failures, and methods for detecting them (Sun et al., 2024). Controlled perturbations further support robustness evaluation and training under noisy tool feedback (Wang et al., 2026; Chen et al., 2026e;a). We model naturally fallible learned visual tools as stochastic observation channels. Since detecting an error does not itself recover the required evidence, our formulation links verification with alternative acquisition through output feedback.

3 Method

Given visual input V and a spatial question Q, Spatial Harness produces an answer y by acquiring spatial evidence from fallible visual tools. Tool outputs are treated as observations to be checked rather than automatically accepted as evidence. For a tool t and an executable call c, we model the returned observation o through a stochastic observation channel:

$$o \sim \mathcal{O}_t(\cdot \mid V, c), \tag{1}$$

where $\mathcal{O}_t(\cdot \mid V, c)$ characterizes possible tool outputs. Deterministic tools as point-mass cases here.

To obtain usable evidence from these channels, Spatial Harness coordinates planning, execution, and verification in an output-feedback loop, as illustrated in Figure 2. The planner proposes primary and backup routes, the executor converts the selected route into an executable program, and the verifier checks the returned observations to update evidence memory M. The updated memory guides subsequent planning and final reasoning.

This section introduces functional tool alternatives in Section 3.1, formalizes output-feedback inference with verification and recovery in Section 3.2, and describes role-level optimization and multi-teacher on-policy distillation in Section 3.3.

3.1 Functional Tool Sets for Fallible Visual Tools

Multiple tools can provide alternative ways to obtain the same required evidence. As illustrated by the silent errors in Figure 1 where a rocking chair is misrecognized as a sofa, a tool can return plausible but unusable evidence without an explicit failure signal. Spatial Harness therefore groups tools by spatial functionality, with $\mathcal{T}_f$ denoting the tools that provide functionality f. Tools in the

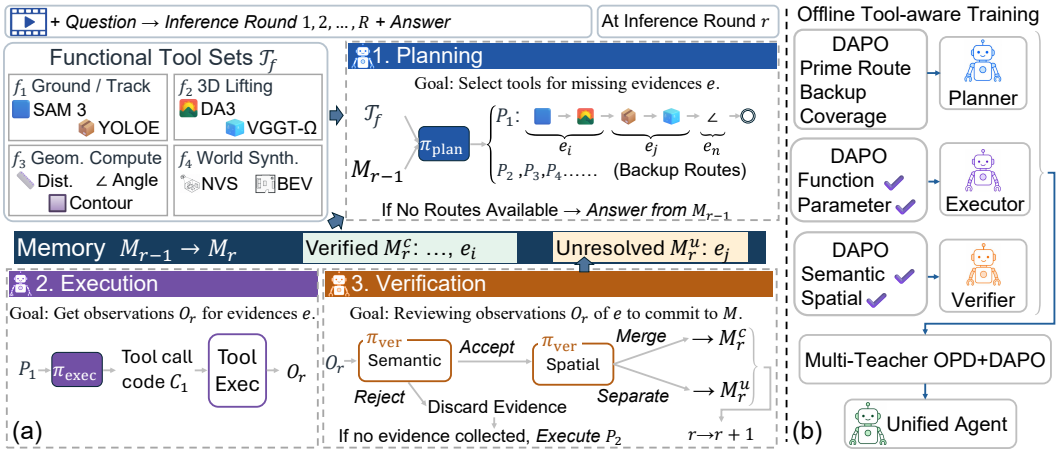

Figure 2: **Overview of Spatial Harness.** (a) At each inference round, the planner constructs memory-conditioned primary and backup routes over functional tool sets. The executor generates tool calls. The verifier uses semantic and spatial checks to update committed evidence and unresolved candidates in memory. Routes retaining no evidence trigger backups. Otherwise, replanning for next round begins. (b) Offline role-local DAPO trains specialized teachers. MOPD distills their expertise into a unified role-conditioned student alongside DAPO task-reward updates.

same functional set share an intended evidence function while differing in their implementation, interface, or underlying assumptions. As illustrated in Figure 2, these functionalities include f_1 grounding and tracking, f_2 3D lifting, f_3 geometric computation, and f_4 world synthesis.

Within each functionality f, an evidence specification s describes the task-specific evidence to be obtained. For a certain evidence goal, we denote by $\mathcal{A}_f$ a collection of alternatives (t, s), where $t \in \mathcal{T}_f$. These alternatives pursue the same goal but may differ in their tools or specifications. The evidence goal determines which members of a functional set can serve as alternatives for the current reasoning step. A tool-use route P orders these choices to obtain the required evidence, while the executor instantiates it with concrete tool-specific arguments.

Functional alternatives provide a simple reliability benefit. For alternatives targeting the same evidence goal, recovery remains possible whenever at least one alternative provides usable evidence:

$$\Pr[\text{no usable result from } \mathcal{A}_f] \leq \min_{(t,s)\in\mathcal{A}_f} \Pr[\text{no usable result from } (t, s)]\,. \tag{2}$$

Here, each alternative (t, s) produces observations through the corresponding channel $\mathcal{O}_t(\cdot \mid V, c)$, while usability is evaluated against the shared evidence goal specified by s. The bound above requires no independence assumption. When the failure events of the alternatives are independent, their joint failure probability factorizes and can be substantially lower than this bound. Spatial Harness realizes this functional redundancy through primary and backup routes, whose selection and recovery are described in Section 3.2.

3.2 Output-Feedback Inference with Verification and Recovery

To prevent tool-level error from affecting spatial reasoning logic, Spatial Harness is designed as an output-feedback loop centered around evidence memory M which stores verified observations and iterates through evidence seeking planner–executor–verifier loops.

Evidence Memory. The evidence memory $M_r = (M_r^c, M_r^u)$ stores evidence retained through round r. The committed partition M_r^c contains evidence accepted as resolved by the verifier, while the unresolved partition M_r^u retains plausible but ambiguous candidates, for example, different items of a certain concept which is unsure which one to use in reasoning.

Planning. The planner π_θ^{plan} reads the visual input V, question Q and the entire memory M to identify missing evidences. It constructs an ordered route list $\mathcal{P}^r = (P_1^r, P_2^r, \ldots)$. The first entry P_1^r is the primary route, while the remaining entries $P_2^r, P_3^r, \ldots$ are backup routes. These routes provide alternative tool and evidence-specification choices, and includes the specifications and context needed by the executor.

Execution. For the i-th attempted route P_i^r, the executor π_θ^{exec} generates a complete executable program $C_{r,i}$. Running the program yields observations $O_{r,i}$ from all tools used in that route. Their joint observation channel $\mathcal{O}(O_{r,i} \mid V, C_{r,i})$ is induced by the tool-specific channels in Equation 1 and captures dependencies along the program.

Verification. The verifier π_θ^{ver} assesses the supplied candidates and cues using the route's evidence specifications, producing local judgments $v_{r,i}$. Semantic checks assess whether candidates match the requested concepts. For candidates requiring spatial association, spatial checks use metric 3D lifting, reprojection, and common-view cues to assess whether they correspond to the same physical object. Resolved evidence is added to committed memory M_r^c, while plausible but ambiguous candidates are retained in M_r^u. Rejected candidates are discarded.

Recovery. When a route contributes no qualified evidence, the next backup receives its own execution and verification pass within round r. The first route contributing evidence to either M_r^c or M_r^u ends round r. After evidence is retained, any required component still absent from the memory M_r triggers planning round $r+1$ from M_r. When all requirements are covered, or available routes are exhausted without retained evidence, the agent produces y by reasoning over the entire memory M.

Together, these stages produce the final answer y and an inference trajectory τ. The trajectory records the route lists $\mathcal{P}$, programs C, observations O, verification judgments v, and memory states M. Let K_r denote the number of routes attempted in round r. Under the route-selection and stopping rules above, the joint distribution factorizes as

$$p_\theta(y, \tau \mid V, Q) = p_\theta(y \mid V, Q, M_R) \prod_{r=1}^{R} \Bigg[\pi_\theta^{\text{plan}}(\mathcal{P}^r \mid V, Q, M_{r-1}) \times \prod_{i=1}^{K_r} \Big[\pi_\theta^{\text{exec}}(C_{r,i} \mid P_i^r)\, \mathcal{O}(O_{r,i} \mid V, C_{r,i}) \times \pi_\theta^{\text{ver}}(v_{r,i} \mid P_i^r, O_{r,i}) \Big] \Bigg]. \tag{3}$$

Memory accumulation retains previous evidence and incorporates the local judgments to supply M_r to subsequent planning and final reasoning. More information about the convergence analysis at the inference time optimization is at Appendix C.2.

3.3 Role-Level Optimization and Multi-Teacher OPD

Fallible tool outputs create varied intermediate evidence conditions hard to cover through complete trajectories. Under our stochastic observation channel $\mathcal{O}$, role-level training targets appropriate responses to imperfect evidence, which not necessarily exist in live inputs from online agentic training methods (Li et al., 2026b). We therefore construct role-local contexts offline from successful and failed outputs and controlled perturbations. Role-wise supervision aligns rewards with distinct decision responsibilities and simplifies credit assignment. Precomputed visual evidence and local contexts shorten inputs and remove waiting for GPU-hosted visual tools during optimization.

For each role $\omega \in \Omega = \{\text{plan}, \text{exec}, \text{ver}\}$, we prepare a training set $\mathcal{D}_\omega$ and optimize a separate teacher $\pi_{\phi_\omega}^{\omega}$ using DAPO (Yu et al., 2026) with role-specific rewards:

$$\begin{aligned} R_{\text{plan}} &= R_{\text{fmt}} + R_{\text{cov}} + \mathbf{1}_{\text{init}} R_{\text{rank}}, \\ R_{\text{exec}} &= R_{\text{fmt}} + R_{\text{func}} + R_{\text{param}}, \\ R_{\text{ver}} &= R_{\text{fmt}} + \mathbf{1}_{\text{sem}} R_{\text{sem}} + \mathbf{1}_{\text{spat}} R_{\text{spat}}. \end{aligned} \tag{4}$$

Here, R_{fmt} evaluates output format, while the remaining terms measure evidence coverage and primary-route ranking, functional and parameter correctness, and semantic and spatial verification correctness, respectively. The indicators activate ranking for initial planning and the applicable verification reward.

After independent optimization, we freeze the teachers and consolidate their specialized behaviors into a unified student π_θ through multi-teacher on-policy distillation (MOPD) (Ma et al., 2026). For each prepared context $x \in \mathcal{D}_\omega$, the rollout student $\pi_{\theta_{\text{old}}}(\cdot \mid x, \omega)$ samples a group of responses. The matched teacher scores the student-generated tokens conditioned on their generated prefixes.

We allocate supervision according to each rollout group's mean binary success $\bar{a}_g$. All-success groups are discarded, and sampling continues until the retained batch is filled. DAPO uses the

Table 1: **ReVSI results comparison.** We report average and per-subcategory results. Best results are in bold while second best is underlined.

Model / Method	Frames	Avg.	Numerical Answer				Multiple-Choice Answer		
			Obj. Count	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan
Proprietary Models									
GPT-5.2	64	50.9	56.2	41.5	73.9	63.0	48.4	34.9	38.2
GPT-5.6-Sol	64	59.0	58.0	53.7	76.2	<u>65.9</u>	69.4	43.4	38.9
Gemini 3 Flash	1 FPS	57.6	<u>65.7</u>	53.1	77.6	52.8	64.6	47.9	41.8
Gemini 3 Pro	1 FPS	60.9	60.1	54.7	<u>79.3</u>	51.9	68.1	56.0	56.4
Open-weight General VLMs									
Qwen3-VL-8B-Instruct	64	49.1	40.4	52.3	69.0	45.1	57.1	39.5	40.5
Qwen2.5-VL-7B-Instruct	4 FPS	32.6	36.9	15.0	49.7	29.0	31.5	29.5	36.7
InternVL3.5-8B	64	47.9	43.3	54.6	64.2	47.6	45.0	36.3	44.4
LLaVA-Video-7B-Qwen2	64	30.3	31.3	1.4	52.5	16.7	38.3	33.3	38.4
Spatially Supervised Specialists									
Cambrian-S-7B	128	49.1	48.4	60.5	65.5	46.7	37.1	48.5	37.0
VST-7B-SFT	4 FPS	46.4	35.4	52.6	67.9	47.2	49.2	36.9	35.4
Cambrian-P	128	52.0	41.4	68.3	66.7	45.9	40.2	48.7	53.1
GS-Reasoner	64	30.5	48.9	26.4	36.2	8.2	31.7	31.1	31.2
Spatial-TTT	64	44.4	41.6	55.6	38.1	44.0	60.8	30.8	39.8
GeoThinker	64	54.8	37.6	73.5	62.2	48.6	66.7	54.9	40.1
Prior Spatial Tool Agents									
SpatialClaw	64	52.1	52.6	56.4	59.2	36.6	52.5	47.6	40.1
S-Agent	64	58.8	54.0	45.6	62.6	53.4	63.6	66.4	<u>66.1</u>
Ours									
Spatial Harness (Qwen3-VL)	64	63.7	56.7	75.7	59.7	54.0	67.5	**72.4**	59.9
Spatial Harness (GPT-5.2)	64	<u>67.6</u>	61.5	<u>76.7</u>	71.0	55.7	<u>70.2</u>	<u>72.3</u>	64.5
Spatial Harness (Trained)	64	**72.7**	**66.1**	**76.9**	**80.0**	**66.9**	**70.7**	<u>72.3</u>	**68.5**

role rewards on all retained groups, while the distillation mask $\mu_g = \mathbf{1}[\bar{a}_g = 0]$ activates teacher guidance for all-failure groups.

Task rewards and teacher feedback provide complementary policy-gradient signals, with the latter expressed as token-level teacher–student log-probability advantages. For response y_i from group $g(i)$, let $h_{it} = (x_i, \omega_i, y_{i,<t})$ denote the token context. We define the clipped teacher advantage and minimize the combined objective:

$$A_{it}^{\mathrm{T}} = \mathrm{sg}\left[\mathrm{clip}\left(\log \frac{\pi_{\phi_{\omega_i}}^{\omega_i}(y_{it} \mid h_{it})}{\pi_{\theta_{\mathrm{old}}}(y_{it} \mid h_{it})}, -c, c\right)\right],$$
$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{DAPO}}(\theta) - \frac{\lambda}{B}\sum_{i=1}^{B}\frac{\mu_{g(i)}}{T_i}\sum_{t\in\mathcal{V}_i} A_{it}^{\mathrm{T}} \log \pi_\theta(y_{it} \mid h_{it}). \tag{5}$$

Here, sg denotes stop-gradient, $c = 5$, $\lambda = 0.1$, and $\mathcal{V}_i$ contains the T_i valid response-token positions. The DAPO term uses standardized task advantages and policy-ratio clipping. The teacher loss averages over tokens within each response and then over all B selected responses, including zero contributions from inactive masks. Training convergence analysis is in Appendix C.3 and C.4.

By discarding all-success groups, training concentrates on capabilities not yet mastered. Within-group task advantages refine partially mastered behaviors in mixed groups, while teacher feedback supplements task optimization in all-failure groups, supporting capability acquisition without requiring successful responses or within-group reward variation. This mixed form of distillation and rewards span training signals to the student's observed competence across roles, contexts and mastery. Data construction, reward computation, and optimization details are in supplementary material.

4 Experiment

We evaluate Spatial Harness on multi-view and video-based spatial reasoning benchmarks. Our experiments cover two regimes: *out-of-the-box agents*, where pretrained VLMs or closed source APIs are used directly as planner, executor, and verifier without task-specific tuning, and *trained agents*, where an open-weight VLM is adapted with our role-level training recipe. We study whether reflective tool use improves strong proprietary and open-weight VLMs, and whether tool-aware fine-tuning further transfers this behavior into a compact deployable agent.

Table 2: **Ego3D-Bench results comparison.** We follow the original Ego3D-Bench layout and report accuracy (%) and RMSE. Higher accuracy and lower RMSE are better.

Model / Method	Accuracy (%) ↑								RMSE ↓	
	Ego Dist.	Obj Dist.	Loc.	Ego Mot.	Obj Mot.	Travel Time	Ego Rel.	Obj Rel.	Ego Dist.	Obj Dist.
Proprietary Models										
GPT-5.2	65.9	50.1	54.6	88.3	87.6	45.4	66.7	75.5	7.23	11.16
GPT-5.6-Sol	73.2	59.7	67.8	89.6	90.1	62.3	73.1	84.7	7.24	10.77
Gemini 3 Flash	72.5	60.1	67.3	86.7	88.9	62.4	72.7	79.8	7.11	11.07
Open-weight General VLMs										
Qwen2.5-VL-Instruct	39.0	32.9	25.1	63.9	57.5	40.4	60.8	73.3	15.53	46.97
Qwen3-VL-Instruct	44.8	31.8	41.3	49.8	55.7	44.8	60.0	79.8	9.61	10.13
Qwen3-VL-Think	43.1	31.2	51.2	58.4	85.8	46.5	60.3	76.5	12.31	25.64
Spatially Supervised Specialists										
Think with 3D	54.0	52.3	36.5	52.7	56.6	38.2	66.0	83.1	–	–
Ego3D (from Qwen2.5-VL)	59.4	54.5	33.1	62.3	58.2	49.4	50.5	66.9	8.10	10.90
Prior Spatial Tool-Call Agents										
SpatialClaw	72.1	56.9	52.7	88.2	78.2	63.3	70.0	84.7	12.3	16.1
Ours										
Spatial Harness (Qwen3-VL)	77.7	62.1	97.1	91.3	84.1	63.9	76.0	87.4	7.03	9.40
Spatial Harness (GPT-5.2)	87.3	85.1	**97.8**	**94.8**	95.2	74.0	84.6	92.7	6.95	7.05
Spatial Harness (Trained)	**89.6**	**88.3**	97.5	94.7	**96.4**	**78.3**	**86.4**	**92.9**	**6.83**	**6.37**

4.1 Experimental Setup

Evaluation setup. We evaluate on two complementary benchmarks. Ego3D-Bench (Gholami et al., 2025) for egocentric multi-view reasoning, requiring integration across views to infer object spatial relationships; and ReVSI (Zhang et al., 2026a), a manually corrected VSI-Bench (Yang et al., 2025a) variant for video spatial reasoning where objects move out and re-occur in videos. Together, they test whether agents can acquire, verify, and aggregate spatial evidence from long visual contexts.

Baselines. We compare against four groups of baselines. Closed-source APIs include GPT-5.2, GPT-5.6-Sol, Gemini 3 Pro, and Gemini 3 Flash. Open-weight general VLMs include Qwen3-VL-8B-Instruct (Bai et al., 2025a), Qwen2.5-VL-7B-Instruct (Bai et al., 2025b), InternVL-3.5-8B (Wang et al., 2025), and LLaVA-Video-7B-Qwen2 (Zhang et al., 2024). 3D feature-guided specialists include Cambrian-S (Yang et al., 2025c), Cambrian-P (Yang et al., 2026), VST (Yang et al., 2025b), GS-Reasoner (Chen et al., 2026d), GeoThinker (Li et al., 2026a), and Spatial-TTT-nano (Liu et al., 2026). We also compare with prior spatial tool-use agents, including SpatialClaw (Cho et al., 2026) and S-Agent (Dai et al., 2026). Qwen-3.5-Plus (Alibaba Cloud, 2026) API hosts SpatialClaw.

Out-of-the-box VLM agents. To isolate the effect of the framework itself, we instantiate Spatial Harness with pretrained VLMs as role-conditioned agents. For proprietary agents, we use GPT-5.2; for open-weight agents, we use Qwen3-VL-8B-Instruct. The same backbone is prompted to act as planner, executor, or verifier according to the role context. This set of experiments showcases improvements from functional tool sets and the memory conditioned feedback control loop.

Trained agents. For the trained-agent regime, we fine-tune Qwen3-VL-8B-Instruct with the role-level rewards in Section 3.3. Training data are generated from a blend of SpaceVista (Sun et al., 2025) and VSI-590K (Yang et al., 2025c) with role-specific rules, where the planner's routes are decided by the tool success rates, the executor's rewards are judged by the accuracy of tool calls and parameters, and the verifier decisions are evaluated by camera-based visibility and VLM semantic checks of the underlying scene. The role-specific teachers are first trained separately for 100 steps, and the MOPD (Ma et al., 2026) takes 300 steps. Dataset construction and training details are provided in the supplementary material.

4.2 Quantitative Results.

Results of the Harness. Table 1 reports results on ReVSI. In the inference-only setting, Spatial Harness improves both Qwen3-VL and GPT-5.2 backbones, reaching 63.7 and 67.6 average accuracy, respectively. Without task-specific training, the GPT-5.2 harness outperforms the strongest prior spatial tool-use agent by 8.8 points.

Table 2 reports results on Ego3D-Bench. Spatial Harness improves over the corresponding VLM backbones and prior spatial tool-use baselines across nearly all metrics. The GPT-5.2 harness

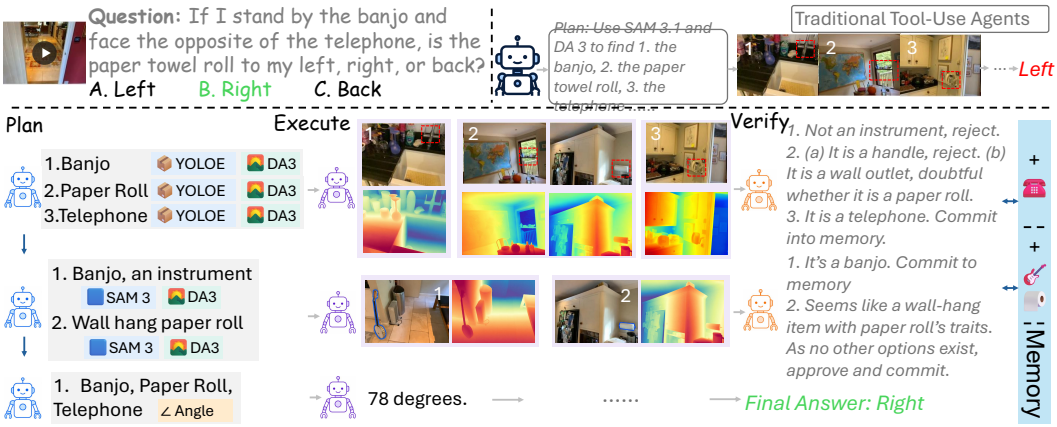

Figure 3: **Qualitative results of SPATIAL HARNESS.** Traditional tool-use agents gives a wrong answer from erroneous tool outputs, while SPATIAL HARNESS succeeds through evidence control loops. Early verification blocks false output from memory such as mislocalized banjo parts. Later, the planner uses committed evidence and global task context to refine the ambiguous object's description; the executor revises tool prompts accordingly, resolving the verifier's initial doubt about the wall hang paper roll. Corrected evidence enters memory and supports the final reasoning.

improves every accuracy metric and reduces RMSE for both distance tasks relative to its backbone. The Qwen3-VL harness reaches 97.1 localization accuracy and also improves performance across object, ego-centric, and relational subtasks.

Table 3 also compares input and output token consumption and API cost per sample in the inference-only setting. SpatialClaw uses Qwen-3.5-Plus and Spatial Harness uses Qwen3-VL-8B and report the translated API pricing. Compared with SpatialClaw, SPATIAL HARNESS uses 78.8% fewer input tokens and 69.0% fewer output tokens. Its API cost is \$0.05 per sample, compared with \$0.40 for SpatialClaw and \$0.22 for GPT-5.6-Sol, showcasing sueperior cost efficiency of our harness.

Model	Input	Output	Cost
GPT-5.6-Sol	24.1k	6.1k	\$0.22
SpatialClaw	666.3k	25.3k	\$0.40
Spatial Harness	141.3k	7.9k	\$0.05

Table 3: **Inference cost comparison.**

Results of the Trained Agents. The last rows of Tables 1 and 2 show that role-level training further improves the Qwen3-VL-8B harness on both benchmarks. On ReVSI, the trained agent reaches 72.7 average accuracy, outperforming untrained Qwen3-VL harness by 9.0 points and GPT-5.2 harness by 5.1 points. On Ego3D-Bench, it achieves the best results on six of the eight accuracy metrics and the lowest RMSE for both distance tasks. These gains show that training on role-local contexts improves agent-tool collaboration, benefiting end-to-end spatial reasoning in the full harness.

4.3 QUALITATIVE RESULTS.

Figure 3 illustrates how SPATIAL HARNESS prevents both tool errors and accidental role-level VLM mistakes from propagating into the final answer. A conventional tool-use pipeline directly chains erroneous detections and depth outputs, leading to a wrong spatial conclusion. In contrast, SPATIAL HARNESS iterates through plan–execute–verify loops, where false tool results are blocked before memory update, while later planning can revise prompts and recover from doubtful verifier decisions using committed evidence. This shows that the harness acts as a safety net not only for stochastic tools, but also for accidentally faulty planner or verifier judgments.

4.4 ANALYSIS AND FINDINGS.

Finding 1. *Explicit tool evidence is necessary.* The results show that spatial reasoning benefits from explicit metric and directional evidence. Raw VLMs must infer object level geometric relationship implicitly from long visual inputs, while tools expose them more directly. This explains why tool-augmented systems, especially SPATIAL HARNESS, improve over open-weight general VLMs and many spatially supervised specialists. The ReVSI results further suggest that specialists trained on fixed spatial annotations can be sensitive to annotation noise or dataset-specific geometric priors, whereas inference-time tools can recover the missing scene evidences.

Finding 2. *Evidence control complements emergent reflection.* Our gains over prior spatial tool-use agents show that strong tool access and iterative reasoning are not always sufficient. Spatial-

Claw (Cho et al., 2026) already lets agents to inspect intermediate results and revise computations through a persistent code interface. However, this reflection remains largely backbone-emergent rather than explicit policy for silent tool errors. SPATIAL HARNESS makes evidence control explicit: the verifier blocks distractors before memory update, uncertain evidence is retained without forcing a decision, and the planner can re-query with revised routes when partial evidence or global context suggests an incomplete earlier judgment. In Figure 3, later replanning corrects an initially doubtful verifier judgment.

Finding 3. *Harness design should consider backend stability and category coverage.* Robust spatial tool use should be evaluated not only by peak accuracy, but also by stability across VLM backends and task categories. Our logs suggest that API-hosted tool-use baselines can exhibit noticeable quality variation across cases, which affects systems whose reflection is strongly tied to the backbone's moment-to-moment agentic capability. In contrast, SPATIAL HARNESS maintains strong performance under both GPT-5.2 and Qwen3-VL backbones. This suggests that functional tool sets and the memory conditioned feedback control loop provide a more stable safety net across backends and categories.

ID	Variant	ReVSI	Ego3D
A	Full Harness	**67.6**	**85.1**
B	w/o Memory	62.7	79.9
C	w/o Verifier	56.8	73.9
D	w/o Tools	50.9	69.2
E	Direct Answer	50.7	60.5

Table 4: **Harness Ablation.** Average accuracy on ReVSI and Ego3D-Bench.

4.5 ABLATION STUDIES

Harness components. Table 4 ablates the core components of SPATIAL HARNESS under GPT-5.2. The full system (A) performs best on both benchmarks. Removing evidence-based memory (B) reduces performance by 4.9 and 5.2 points, showing that retained partial evidence is important for later replanning. Removing the verifier (C) causes a larger drop, indicating that unchecked tool outputs introduces silent errors to downstream reasoning. Removing tool use (D) or reasoning (E) further degrades the performance. Without tools, the agent must infer metric and directional relations from raw visual context, leading to substantial drops on both benchmarks. Direct-answering worsens Ego3D-Bench performance, suggesting that reasoning on evidences is particularly important for cross-view spatial composition.

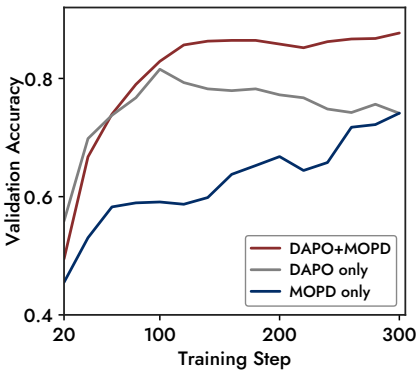

Figure 4: **Training ablation** on role-averaged validation accuracy over steps.

Training strategies. Figure 4 compares the average validation accuracy of DAPO+MOPD, DAPO (Yu et al., 2026) only, and MOPD (Ma et al., 2026) only. DAPO alone improves quickly at first, but it suffers from reward conflicts and validation accuracy later declines. MOPD alone improves more gradually and finishes below the combined strategy, showcasing that the environment reward and group level policy gradient is a strong compliment to teacher signals. In contrast, DAPO+MOPD reaches the highest validation accuracy, showing that task rewards and teacher feedback assist each other during training.

5 CONCLUSION

We presented SPATIAL HARNESS, an output-feedback evidence-control framework for spatial reasoning with imperfect visual tools. By organizing stochastic neural network tools such as detectors, segmentors, depth estimators, and reconstruction models into functional tool sets, and by reasoning through a memory-conditioned plan–execute–verify loop, Spatial Harness acquires explicit metric evidence while preventing unreliable observations from being blindly committed. Experiments on ReVSI and Ego3D-Bench show that this design improves both proprietary and open-weight VLM backbones. The quantitative and qualitative results further indicate that the harness provides a safety net not only for stochastic tool outputs, but also for accidental failures of the VLM roles themselves. Future work may extend evidence control from passive verification to estimating numerical error bars for tool outputs, and from fixed tool sets to agents that evolve to discover, compose, or construct tools during reasoning.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for paper content polishing, assist with translation and clean and reformat dataset. We have not used generative AI tools for generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, interpret results. and support qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for figure color suggestions and creating software code. We have reviewed all AI-assisted work, including the polished content, the figure color under the suggestion, the implemented code and the procedure of the dataset parsing. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

See the ICLR 2027 AI Policy for Authors for more details. This statement should not be more than 1 page.

REFERENCES

Alibaba Cloud. qwen3.5-plus. Alibaba Cloud Model Studio documentation, 2026. URL https://www.alibabacloud.com/help/en/model-studio/qwen3-5-plus. Accessed: 2026-09-25.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a. URL https://arxiv.org/abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b. URL https://arxiv.org/abs/2502.13923.

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

Chaoran Chen, Vy Nguyen, Ziji Zhang, Abhinav Gullapalli, Ziyi Wang, Yuxuan Lu, Dakuo Wang, Jing Huang, Zhou Yu, and Jin Lai. Retry, switch, or abstain? learning strategy-aware tool-use policies via controlled error injection. *arXiv preprint arXiv:2608.11977*, 2026a.

Jieneng Chen, Wenxin Ma, Ruisheng Yuan, Yunzhi Zhang, Jiajun Wu, and Alan Yuille. Thinking with spatial code for physical-world video reasoning. *arXiv preprint arXiv:2603.05591*, 2026b.

Siyi Chen, Mikaela Angelina Uy, Chan Hee Song, Faisal Ladhak, Adithyavairavan Murali, Qing Qu, Stan Birchfield, Valts Blukis, and Jonathan Tremblay. Spacetools: Tool-augmented spatial reasoning via double interactive rl. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37109–37120, 2026c.

Yiming Chen, Zekun Qi, Wenyao Zhang, Xin Jin, Li Zhang, and Peidong Liu. Reasoning in space via grounding in the world. In *International Conference on Learning Representations*, volume 2026, pp. 65549–65570, 2026d.

Yuxin Chen, Xiaodong Cai, Junfeng Fang, Zhuowen Han, Yu Wang, Yaorui Shi, Yi Zhang, Qi Gu, Xunliang Cai, Xiang Wang, et al. Learning to act under noise: Enhancing agent robustness via noisy environments. *arXiv preprint arXiv:2605.27209*, 2026e.

Seokju Cho, Ryo Hachiuma, Abhishek Badki, Hang Su, Byung-Kwan Lee, Chan Hee Song, Sifei Liu, Subhashree Radhakrishnan, Seungryong Kim, Yu-Chiang Frank Wang, et al. Spatialclaw: Rethinking action interface for agentic spatial reasoning. *arXiv preprint arXiv:2606.13673*, 2026.

Yalun Dai, Hao Li, Shulin Tian, Runmao Yao, Yuhao Dong, Fangzhou Hong, Zhaoxi Chen, Fangfu Liu, Baoliang Tian, Dingwen Zhang, et al. S-agent: Spatial tool-use elicits reasoning for spatial intelligence. *arXiv preprint arXiv:2606.20515*, 2026.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents. *Advances in neural information processing systems*, 37:82895–82920, 2024.

Mohsen Gholami, Ahmad Rezaei, Zhou Weimin, Sitong Mao, Shunbo Zhou, Yong Zhang, and Mohammad Akbari. Spatial reasoning with vision-language models in ego-centric multi-view scenes, 2025. URL https://arxiv.org/abs/2509.06266.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, 2024.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM workshop on artificial intelligence and security*, pp. 79–90, 2023.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6864–6890, 2024.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6693–6702. IEEE, 2019.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

Sunil Kumar, Bowen Zhao, Leo Dirac, and Paulina Varshavskaya. Reinforcing vlms to use tools for detailed visual reasoning under resource constraints. *arXiv preprint arXiv:2506.14821*, 2025.

Haoyuan Li, Qihang Cao, Tao Tang, Kun Xiang, Zihan Guo, Jianhua Han, JiaWang Bian, Hang Xu, and Xiaodan Liang. Thinking with geometry: Active geometry integration for spatial reasoning. *arXiv preprint arXiv:2602.06037*, 2026a.

Pengxiang Li, Zhi Gao, Bofei Zhang, Yapeng Mi, Xiaojian Shawn Ma, Chenrui Shi, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, and Qing Li. Iterative tool usage exploration for multimodal agents via step-wise preference tuning. *Advances in Neural Information Processing Systems*, 38, 2025.

Zhuofeng Li, Haoxiang Zhang, Seungju Han, Sheng Liu, Jianwen Xie, Yu Zhang, Yejin Choi, James Y Zou, and Pan Lu. In-the-flow agentic system optimization for effective planning and tool use. In *International Conference on Learning Representations*, volume 2026, pp. 50524–50570, 2026b.

Niu Lian, Alan Chen, Zhehao Yu, Chengzhen Duan, Fazhan Liu, Hui Liu, Pei Fu, Jian Luan, Yaowei Wang, Shu-Tao Xia, et al. Ui-mopd: Multi-platform on-policy distillation for continual gui agent learning. *arXiv preprint arXiv:2607.04425*, 2026.

Fangfu Liu, Diankun Wu, Jiawei Chi, Yimo Cai, Yi-Hsin Hung, Xumin Yu, Hao Li, Han Hu, Yongming Rao, and Yueqi Duan. Spatial-ttt: Streaming visual-based spatial intelligence with test-time training. In *European Conference on Computer Vision*, pp. 339–357. Springer, 2026.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, et al. Mopd: Multi-teacher on-policy distillation for capability integration in llm post-training. *arXiv preprint arXiv:2606.30406*, 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.

Damiano Marsili, Rohun Agrawal, Yisong Yue, and Georgia Gkioxari. Visual agentic ai for spatial reasoning with a dynamic api. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19446–19455. IEEE, 2025.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37:126544–126565, 2024.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, volume 2024, pp. 9695–9717, 2024.

Kanchana Ranasinghe, Satya Narayan Shukla, Omid Poursaeed, Michael S Ryoo, and Tsung-Yu Lin. Learning to localize objects improves spatial reasoning in visual-llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12977–12987. IEEE, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Jimin Sun, So Yeon Min, Yingshan Chang, and Yonatan Bisk. Tools fail: Detecting silent errors in faulty tools. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 14272–14289, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.790. URL https://aclanthology.org/2024.emnlp-main.790/.

Peiwen Sun, Shiqiang Lang, Dongming Wu, Yi Ding, Kaituo Feng, Huadai Liu, Zhen Ye, Rui Liu, Yun-Hui Liu, Jianan Wang, and Xiangyu Yue. SpaceVista: All-scale visual spatial reasoning from mm to km. *arXiv preprint arXiv:2510.09606*, 2025. URL https://arxiv.org/abs/2510.09606.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12, pp. 1057–1063, 1999.

Ruipeng Wang, Yuxin Chen, Yukai Wang, Chang Wu, Junfeng Fang, Xiaodong Cai, Qi Gu, Hui Su, An Zhang, Xiang Wang, et al. Agentnoisebench: Benchmarking robustness of tool-using llm agents under noisy condition. *arXiv preprint arXiv:2602.11348*, 2026.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025. URL https://arxiv.org/abs/2508.18265.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023.

Mingyuan Wu, Jingcheng Yang, Jize Jiang, Meitang Li, Kaizhuo Yan, Hanchao Yu, Minjia Zhang, Chengxiang Zhai, and Klara Nahrstedt. Vtool-r1: Vlms learn to think with images via reinforcement learning on multimodal tool use. In *International Conference on Learning Representations*, volume 2026, pp. 78298–78319, 2026.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643, June 2025a. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_Thinking_in_Space_How_Multimodal_Large_Language_Models_See_Remember_CVPR_2025_paper.html.

Jihan Yang, Zifan Zhao, Xichen Pan, Shusheng Yang, Junyi Zhang, Bingyi Kang, Hu Xu, Shang-Wen Li, and Saining Xie. Cambrian-P: Pose-grounded video understanding, 2026. URL https://arxiv.org/abs/2605.22819.

Rui Yang, Ziyu Zhu, Yanwei Li, Jingjia Huang, Shen Yan, Siyuan Zhou, Zhe Liu, Xiangtai Li, Shuangye Li, Wenqian Wang, Yi Lin, and Hengshuang Zhao. Visual spatial tuning. *arXiv preprint arXiv:2511.05491*, 2025b. URL https://arxiv.org/abs/2511.05491.

Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis Brown, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, Daohan Lu, Rob Fergus, Yann LeCun, Li Fei-Fei, and Saining Xie. Cambrian-S: Towards spatial supersensing in video. *arXiv preprint arXiv:2511.04670*, 2025c. URL https://arxiv.org/abs/2511.04670.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Yiming Zhang, Jiacheng Chen, Jiaqi Tan, Yongsen Mao, Wenhu Chen, and Angel X. Chang. Revsi: Rebuilding visual spatial intelligence evaluation for accurate assessment of vlm 3d reasoning. *arXiv preprint arXiv:2604.24300*, 2026a.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data, 2024. URL https://arxiv.org/abs/2410.02713.

Zaibin Zhang, Yuhan Wu, Lianjie Jia, Yifan Wang, Zhongbo Zhang, Yijiang Li, Binghao Ran, Fuxi Zhang, Zhuohan Sun, Peng Yizhuang, Zhenfei Yin, Lijun Wang, and Huchuan Lu. Think3d: Thinking with space for spatial reasoning. *arXiv preprint arXiv:2601.13029*, 2026b.

Jingnan Zheng, Yanzhen Luo, Jingjun Xu, Bingnan Liu, Yuxin Chen, Chenhang Cui, Gelei Deng, Chaochao Lu, Xiang Wang, An Zhang, et al. Risky-bench: Probing agentic safety risks under real-world deployment. *arXiv preprint arXiv:2602.03100*, 2026.

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. {PoisonedRAG}: Knowledge corruption attacks to {Retrieval-Augmented} generation of large language models. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 3827–3844, 2025.

A APPENDIX

This supplementary material provides additional details and analyses of Spatial Harness. Section B presents the inference algorithm, including memory updates, backup execution, and stopping rules. Section C analyzes feedback-guided inference and establishes conditions under which role-local training improves the end-to-end system objective. Section D reports additional ablations on recovery policies, unresolved memory, functional tool alternatives, and training strategies. Finally, Section E describes training data construction, role-specific supervision, and the optimization settings for teacher specialization and student consolidation.

B INFERENCE ALGORITHM OF SPATIAL HARNESS

Algorithm 1 summarizes the inference procedure in Section 3.2. Each round constructs primary and backup routes for evidence absent from both memory partitions. A route that retains no evidence activates the next backup within the same round. The first route retaining evidence ends the round, and any remaining evidence gaps trigger replanning from the updated memory.

Algorithm 1 Output-Feedback Inference of SPATIAL HARNESS

Require: Visual input V, question Q, functional tool sets $\{\mathcal{T}_f\}$, planner π_θ^{plan}, executor π_θ^{exec}, verifier π_θ^{ver}, and final-answer distribution p_θ
Ensure: Final answer y

Initialize $r \leftarrow 0$, $M_0 = (M_0^c, M_0^u) \leftarrow (\varnothing, \varnothing)$
repeat
  $r \leftarrow r + 1$, $M_r \leftarrow M_{r-1}$
  1. Memory-conditioned Planning
  $\mathcal{P}^r = (P_1^r, P_2^r, \ldots) \sim \pi_\theta^{\text{plan}}(\cdot \mid V, Q, M_{r-1})$
  for $i = 1, \ldots, |\mathcal{P}^r|$ **do** {*Primary route first, then backups*}
    2. Route Execution
    $C_{r,i} \sim \pi_\theta^{\text{exec}}(\cdot \mid P_i^r)$ {*Generate a complete program*}
    $O_{r,i} \sim \mathcal{O}(\cdot \mid V, C_{r,i})$ {*Run the program; resolve dependencies*}
    3. Verification and Memory Update
    $v_{r,i} \sim \pi_\theta^{\text{ver}}(\cdot \mid P_i^r, O_{r,i})$ {*Semantic and applicable spatial checks*}
    $M_r \leftarrow f_{\text{mem}}(M_{r-1}, P_i^r, O_{r,i}, v_{r,i})$
    4. Recovery
    if $M_r = M_{r-1}$ **then**
      continue {*No retained evidence; try the next backup*}
    else
      break {*End this round once evidence is retained*}
    end if
  end for
until $M_r = M_{r-1}$ **or** all evidence required by Q is covered by M_r
$R \leftarrow r$
$y \sim p_\theta(\cdot \mid V, Q, M_R)$ {*Reason over both memory partitions*}
return y

Memory update and stopping. The update function f_{mem} preserves existing evidence, adds resolved evidence to M_r^c, retains plausible but ambiguous candidates in M_r^u, and discards rejected candidates. Evidence coverage is assessed over both partitions; unresolved candidates remain distinct from committed evidence during final reasoning. If a route adds nothing to either partition, $M_r = M_{r-1}$ and the next backup is attempted. If the route list is empty or all routes fail to retain evidence, the loop stops and the agent answers from the available memory. Otherwise, the next round targets only evidence still absent from both partitions. R denotes the number of rounds executed.

C THEORETICAL ANALYSIS OF SPATIAL HARNESS

We analyze SPATIAL HARNESS at two complementary levels: feedback-guided decision optimization during inference and role-local policy optimization during training. Both are directed toward the end-to-end objective induced by Equation (3). We first characterize the value of conditioning inference decisions on acquired evidence. We then show that the system policy gradient admits a role-local representation and establish sufficient conditions under which updates learned from prepared local contexts improve the system objective.

C.1 SYSTEM OBJECTIVE AND PRELIMINARIES

Let $\xi = (V, Q, y^\star) \sim \mathcal{D}$ denote a task instance, and let $U_\xi(y) \in [0, 1]$ evaluate the final answer. For binary correctness, the system objective is its expected accuracy:

$$J(\theta) = \mathbb{E}_{\xi\sim\mathcal{D}}\mathbb{E}_{(y,\tau)\sim p_\theta(\cdot,\cdot|V,Q)}[U_\xi(y)]. \tag{6}$$

Equation (3) defines the distribution underlying this expectation; optimization concerns expected task utility rather than the likelihood of an arbitrary trajectory. The reference answer $y^\star$ is used to define the objective.

For a role $\omega \in \Omega = \{\text{plan}, \text{exec}, \text{ver}\}$, write x^ω for its supplied context and a^ω for its generated response. These responses are route lists, executable programs, and verification judgments, respectively. A role response is treated as a complete token sequence. Let $\mathcal{I}_\omega(\tau)$ index the invocations of role ω in a trajectory.

We analyze a bounded-budget process with finite bounds on role invocations and response lengths. The observation channels and the memory-update and stopping rules are fixed conditional on their inputs. Policy probabilities are differentiable with locally fixed support, and expectations and derivatives may be interchanged, without the demand of tool channels being differentiable, unbiased, or mutually independent. Throughout, U abbreviates $U_\xi(y)$.

C.2 FEEDBACK-GUIDED INFERENCE-TIME OPTIMIZATION

At inference, model parameters remain fixed while role decisions adapt to acquired observations. The agent controls which routes and programs to select and which observations to retain. The relevant decision problem is therefore to choose useful continuations after observing feedback instead of prescribing an entire successful trajectory in advance.

Proposition C.1 (Value of feedback-conditioned decisions)**.** *Consider a fixed inference prefix and its resulting feedback Z. Let $\mathcal{A}$ be a common finite menu of feasible continuation decisions under the same remaining budget. Let $W(z, a)$ be the expected terminal utility when decision a is selected after feedback z, with subsequent policies held fixed. Then*

$$\mathbb{E}_Z\left[\max_{a\in\mathcal{A}} W(Z, a)\right] \geq \max_{a\in\mathcal{A}} \mathbb{E}_Z[W(Z, a)]. \tag{7}$$

The left side allows the decision to depend on feedback; the right side selects the same decision before observing that feedback.

Proof. For every fixed $a \in \mathcal{A}$, $\max_{a'} W(z, a') \geq W(z, a)$ for every z. Taking expectations and then maximizing over a proves the result. A randomized feedback-independent decision is a mixture of fixed decisions and cannot exceed their maximum. □

The comparison uses the same acquired feedback and remaining budget, so it isolates the value of adapting the continuation. Strict improvement is possible when different feedback outcomes favor different decisions. This is an achievable decision-level benefit; the quality of the role policies determines how effectively the deployed harness realizes it.

Functional alternatives and verification. The harness supplies both alternative decisions and evidence for selecting among them. For alternatives targeting the same evidence requirement, let F_i

denote the failure of alternative i. Event inclusion gives

$$\Pr\left(\bigcap_{i=1}^{k} F_i\right) \leq \min_i \Pr(F_i), \tag{8}$$

without independence. The additional recovery opportunity provided by alternative $k+1$ is $\Pr(F_1 \cap \cdots \cap F_k \cap F_{k+1}^c)$.

Verification governs whether a returned candidate becomes committed evidence. Within a fixed acquisition context, let p be the probability that a candidate is valid, and let α and β be its commitment probabilities conditional on validity and invalidity, respectively. When commitment has positive probability, Bayes' rule yields

$$\Pr(\text{invalid} \mid \text{committed}) = \frac{(1-p)\beta}{p\alpha + (1-p)\beta}. \tag{9}$$

For $0 < p < 1$, this is lower than the unfiltered error rate $1-p$ whenever $\alpha > \beta$. Meanwhile, the valid-commitment probability is $p\alpha$, showing why selective commitment must be accompanied by recovery rather than rejection alone.

Under Section 3.2, a no-retention outcome activates a backup; retention in either M^c or M^u ends the round and enables planning for the remaining requirements. Thus, alternatives provide recovery opportunities, verification determines retention, and memory feedback selects the subsequent continuation. Unresolved retention counts toward coverage but is not a certificate of correctness. The availability bound does not assume that every alternative is actually executed.

C.3 Role-Local Representation of System Optimization

We next connect the decision policies in Equation (3) to the system objective. The derivation applies the score-function policy-gradient identity (Sutton et al., 1999) to the factorized agentic process.

Fix a reference parameter θ_0, and evaluate the following quantities at θ_0. Let $m_\omega = \mathbb{E}[|\mathcal{I}_\omega(\tau)|]$ be the expected invocation count. For $m_\omega > 0$, define the role-context distribution d_ω by

$$\mathbb{E}_{x\sim d_\omega}[f(x)] = \frac{1}{m_\omega}\mathbb{E}\left[\sum_{k\in\mathcal{I}_\omega(\tau)} f(x_k^\omega)\right]. \tag{10}$$

This weights contexts by their expected number of visits. Roles with zero expected invocations contribute no term.

Under this invocation-weighted distribution, define

$$\begin{aligned} W_\omega(x,a) &= \mathbb{E}[U \mid x^\omega = x, a^\omega = a], \\ A_\omega(x,a) &= W_\omega(x,a) - \mathbb{E}_{a'\sim\pi_{\theta_0}^\omega(\cdot|x)}[W_\omega(x,a')], \\ \psi_\omega(x,a) &= \nabla_\theta \log \pi_\theta^\omega(a \mid x)|_{\theta=\theta_0}, \\ g_\omega(x) &= \mathbb{E}_{a\sim\pi_{\theta_0}^\omega(\cdot|x)}[A_\omega(x,a)\psi_\omega(x,a)]. \end{aligned} \tag{11}$$

The continuation value averages over histories compatible with the supplied context and over subsequent system execution. It does not require that the context be a sufficient physical-state representation.

Theorem C.2 (Role-wise decomposition of the system gradient)**.** *Under the assumptions of Section C.1,*

$$\nabla J(\theta_0) = g_{\text{ans}} + \sum_{\omega\in\Omega} m_\omega \mathbb{E}_{x\sim d_\omega}[g_\omega(x)], \tag{12}$$

where the direct contribution from final-answer generation is

$$g_{\text{ans}} = \mathbb{E}\big[U\ \nabla_\theta \log p_\theta(y \mid V, Q, M_R)|_{\theta=\theta_0}\big]. \tag{13}$$

Proof. For a realized trajectory, the log of Equation (3) is the sum of the role log-probabilities, the final-answer log-probability, and parameter-independent observation terms. The fixed memory and stopping rules add no direct score terms. Therefore, the score-function identity gives

$$\nabla J(\theta_0) = g_{\text{ans}} + \mathbb{E}\left[U \sum_{\omega\in\Omega} \sum_{k\in\mathcal{I}_\omega(\tau)} \psi_\omega(x_k^\omega, a_k^\omega)\right]. \tag{14}$$

Grouping invocations using Equation equation 10 and conditioning on each (x, a) replaces U by $W_\omega(x, a)$. Finally, $\mathbb{E}_{a\sim\pi_{\theta_0}^\omega}[\psi_\omega(x, a)] = 0$, so subtracting the action-independent baseline leaves the expectation unchanged. This gives Equation equation 12. □

First-order equivalence with local objectives. Hold d_ω, m_ω, and A_ω fixed at the reference policy and define

$$\mathcal{L}_{\text{local}}(\theta; \theta_0) = \sum_{\omega\in\Omega} m_\omega \mathbb{E}_{x\sim d_\omega} \mathbb{E}_{a\sim\pi_\theta^\omega(\cdot|x)}[A_\omega(x, a)]. \tag{15}$$

Its gradient at θ_0 equals $\nabla J(\theta_0) - g_{\text{ans}}$. Consequently, role-local decisions, weighted by their continuation advantages and deployment visitation, recover the role contributions to end-to-end optimization. With a separately fixed answer policy, they recover the full system gradient.

This is a first-order identity at the reference policy, not an equality of objective values for arbitrary parameters. It shows that a complete trajectory need not be supplied as a role's training input: its downstream influence can be summarized by a local learning signal. Obtaining an appropriate signal and representative contexts remains essential. The decomposition also holds with shared parameters; role contributions need not be independent or mutually aligned, and the direct answer term must then be retained.

C.4 Role-Local Training toward System Improvement

Section 3.3 uses prepared contexts and responsibility-specific rewards rather than exact continuation advantages. Planning rewards target coverage and route ranking, execution rewards target functional and parameter correctness, and verification rewards target semantic and spatial judgments. These signals are tractable surrogates for decisions that influence the terminal objective. We now quantify how their local updates relate to the system gradient.

Let $\widetilde{d}_\omega$ denote the effective training-context distribution, and let $\widehat{g}_\omega(x)$ denote its conditional expected negative loss gradient at context x. This definition accommodates token-level feedback, group selection, normalization, and clipping. Let $w_\omega \geq 0$ be the effective role weights, expressed on the same overall scale as m_ω; a common positive rescaling can be absorbed into the step size. Define

$$g_{\text{local}} = \sum_\omega w_\omega \mathbb{E}_{x\sim\widetilde{d}_\omega}[\widehat{g}_\omega(x)], \qquad G_{\text{role}} = \sum_\omega m_\omega \mathbb{E}_{x\sim d_\omega}[g_\omega(x)]. \tag{16}$$

All quantities describe population expectations at the reference policy. For contexts outside deployment support, extend A_ω within $[-1, 1]$ and define g_ω by Equation equation 11; their training mass is accounted for by distribution mismatch.

Theorem C.3 (Local-to-global update alignment)**.** *Suppose $\|\psi_\omega(x, a)\| \leq B_\omega$ on the deployment and training supports. Let*

$$\delta_\omega = \text{TV}(\widetilde{d}_\omega, d_\omega), \qquad \epsilon_\omega = \mathbb{E}_{x\sim\widetilde{d}_\omega}\left[\|\widehat{g}_\omega(x) - g_\omega(x)\|\right]. \tag{17}$$

Then

$$\|g_{\text{local}} - G_{\text{role}}\| \leq \mathcal{E} := \sum_\omega \left[w_\omega\epsilon_\omega + 2m_\omega B_\omega\delta_\omega + B_\omega|w_\omega - m_\omega|\right]. \tag{18}$$

Moreover,

$$\begin{aligned}\langle\nabla J(\theta_0), g_{\text{local}}\rangle \geq \|g_{\text{local}}\|^2 - \mathcal{E}\|g_{\text{local}}\| \\ + \langle g_{\text{ans}}, g_{\text{local}}\rangle.\end{aligned} \tag{19}$$

Whenever the right-hand side is positive, g_{local} is an ascent direction for the system objective, and $J(\theta_0 + \eta g_{\text{local}}) > J(\theta_0)$ for all sufficiently small $\eta > 0$.

Proof. Since $U \in [0,1]$, $|A_\omega| \le 1$, giving $\|g_\omega(x)\| \le B_\omega$ on deployment support. For each role, add and subtract $w_\omega \mathbb{E}_{\widetilde{d}_\omega} g_\omega$ and $m_\omega \mathbb{E}_{\widetilde{d}_\omega} g_\omega$. The triangle inequality bounds the conditional-update error by $w_\omega \epsilon_\omega$, the weight error by $B_\omega |w_\omega - m_\omega|$, and the distribution error by $2 m_\omega B_\omega \delta_\omega$. Summing proves Equation equation 18. Writing $\nabla J = g_{\mathrm{ans}} + G_{\mathrm{role}}$ and applying Cauchy–Schwarz proves Equation equation 19. Differentiability then gives $J(\theta_0 + \eta g_{\mathrm{local}}) = J(\theta_0) + \eta \langle \nabla J, g_{\mathrm{local}} \rangle + o(\eta)$, which establishes the final statement. □

The bound separates supervision error, context mismatch, and role-weight mismatch. For a response-level surrogate with on-policy responses and $\widehat{g}_\omega(x) = \mathbb{E}_a[\widehat{A}_\omega(x,a)\psi_\omega(x,a)]$, its conditional-update error satisfies

$$\epsilon_\omega \le B_\omega \, \mathbb{E}_{x \sim \widetilde{d}_\omega,\, a \sim \pi^\omega_{\theta_0}} \left[|\widehat{A}_\omega(x,a) - A_\omega(x,a)| \right], \tag{20}$$

Thus, local supervision contributes to system optimization through its alignment with downstream decision value, not merely through an increase in the local reward.

Implications for the training design. Diverse successful, failed, and ambiguous contexts provide exposure to the conditions encountered by the inference loop. Responsibility-aligned rewards and role-specialized teachers target the quality of the corresponding local decisions. MOPD consolidates these policies through teacher feedback on student responses, while DAPO supplies task-reward updates. In Equation (5), identical task rewards within a group eliminate centered task advantages, but do not force token-level teacher feedback to vanish. This provides a complementary source of supervision, whose downstream alignment is represented by ϵ_ω rather than assumed to be exact.

The answer-gradient term accounts for changes to final reasoning when parameters are shared. In particular, $\|g_{\mathrm{local}}\| > \mathcal{E}$ and $\langle g_{\mathrm{ans}}, g_{\mathrm{local}} \rangle \ge 0$ are sufficient for local improvement. The result characterizes population-level ascent under aligned updates; it does not require or imply that every finite-batch DAPO or MOPD step improves terminal accuracy.

Together, the results establish two complementary optimization interfaces for Equation (3): feedback-conditioned decisions can improve inference within the available budget, and role-local training can improve the policies making those decisions when its context distribution and supervision are sufficiently aligned with deployment. Inference adapts decisions to the evidence of the current task; training improves those decisions across tasks.

D ADDITIONAL ABLATION STUDIES

We extend the main-paper ablations to examine recovery policies, unresolved memory, and functional tool alternatives on ReVSI. We also evaluate training strategies through the complete harness, complementing the role-level validation curves in the main paper.

D.1 EVIDENCE-CONTROL MECHANISMS

We study the inference-only harness with GPT-5.2. Table 5 reports overall and per-category performance. In the full system, routes yielding no retained evidence trigger backups, whereas partial progress followed by remaining evidence gaps triggers replanning from updated memory.

Table 5: **Additional harness ablations on ReVSI.** We report overall and per-category scores (%) for the GPT-5.2 harness. Bold indicates the best result in each column.

Variant	Overall	Numerical Answer				Multiple-Choice Answer		
		Obj. Count	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan
Full SPATIAL HARNESS	**67.6**	**61.5**	**76.7**	**71.0**	**55.7**	**70.2**	**72.3**	**64.5**
Primary route once	56.6	57.2	68.4	69.2	55.5	69.9	30.8	26.7
Replanning only	52.3	58.2	58.6	69.8	56.0	50.6	34.1	28.6
Backup only	42.5	38.1	38.2	70.1	55.6	39.5	31.4	21.5
Verification + original-route retry	39.8	28.5	40.8	69.0	51.0	37.1	28.7	19.9
w/o Unresolved Memory	54.5	11.5	76.7	69.2	55.3	69.2	46.6	35.0
One tool per function	62.1	56.9	60.2	66.0	55.0	67.1	64.7	46.3

Recovery policies. *Primary route once* executes the initial primary route without backups or replanning. *Replanning only* removes backup routes, whereas *backup only* retains the initial alternatives without generating new memory-conditioned routes. *Verification + original-route retry* keeps verification but retries the original route instead of using alternative routes or replanning. The full harness reaches 67.6 overall, compared with 56.6 for a single primary route, 52.3 for replanning only, 42.5 for backup only, and 39.8 for original-route retry. Relative direction and route planning are particularly affected by single-pass execution, falling from 72.3 to 30.8 and from 64.5 to 26.7, respectively.

Notably, replanning alone, backups alone, and original-route retry all perform worse than single-pass execution. Failure cases include remaining stuck on a failing tool without backups and repeatedly inserting already collected observations when fixed routes are reused. These results show that allowing repeated attempts does not by itself ensure better answers. Effective recovery needs both alternative acquisition routes and updated targets based on the evidence already retained.

Unresolved memory. This variant retains committed evidence but discards unresolved candidates. The overall score drops from 67.6 to 54.5, a decrease of 13.1 points. The largest loss occurs in object counting, which falls from 61.5 to 11.5; relative direction and route planning also decline to 46.6 and 35.0. Failure cases include discarding observations of repeated objects and mistaking empty memory for object absence. Keeping plausible but ambiguous candidates available for planning and final reasoning avoids forcing an early rejection without treating them as resolved facts.

Functional tool alternatives. Restricting each functionality to one tool reduces the overall score from 67.6 to 62.1. Absolute distance falls from 76.7 to 60.2, and route planning falls from 64.6 to 46.3. These results support providing alternative tools for the same evidence goal rather than depending on a single tool whose failure can leave that goal unmet.

D.2 TRAINING STRATEGIES AND TEACHER CONSOLIDATION

We evaluate Qwen3-VL-8B-Instruct under different training and deployment strategies within the full harness. *Multi-target DAPO only* optimizes a shared role-conditioned model with role-level task rewards but no distillation. *MOPD only* uses teacher feedback without DAPO task-reward updates. *Separate role teachers* deploys the corresponding specialized teacher for each role instead

Table 6: **End-to-end training ablations on ReVSI.** All variants use Qwen3-VL-8B-Instruct within the full harness. The untrained and DAPO+MOPD reference rows are reproduced from the main-paper results. Scores are reported in percent; bold indicates the best result in each column.

Variant	Overall	Numerical Answer				Multiple-Choice Answer		
		Obj. Count	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan
Untrained Harness	63.7	56.7	75.7	59.7	54.0	67.5	**72.4**	59.9
Multi-target DAPO Only	53.6	55.6	64.9	64.6	53.6	65.1	28.6	24.4
MOPD Only	54.9	57.8	67.1	70.2	54.3	63.8	27.0	26.0
Separate Role Teachers	64.3	57.3	59.1	66.3	53.3	**71.1**	70.4	55.9
DAPO+MOPD (Ours)	**72.7**	**66.1**	**76.9**	**80.0**	**66.9**	70.7	72.3	**68.5**

of the unified student. Table 6 compares these variants with the untrained harness and the full DAPO+MOPD recipe.

Complementary training signals. DAPO+MOPD reaches 72.7 overall, exceeding DAPO only and MOPD only by 19.1 and 17.8 points, respectively. Both single-objective variants fall below the untrained harness's 63.7 overall score. Their losses are particularly large in relative direction and route planning: DAPO only obtains 28.6 and 24.4, while MOPD only obtains 27.0 and 26.0, compared with 72.3 and 68.5 for the combined recipe. The two signals serve different purposes in our training procedure: teacher feedback supplies supervision for all-failure groups, while task-reward updates refine behavior using role-level rewards. The end-to-end results support combining these signals rather than relying on either alone.

Teacher consolidation. Directly deploying the separate role teachers reaches 64.3 overall, while the unified student trained with DAPO+MOPD reaches 72.7, an improvement of 8.4 points. The gains include absolute distance, from 59.1 to 76.9, and room size, from 53.3 to 66.9. The separate teachers retain a slightly higher relative-distance score (71.1 versus 70.7), but their overall performance is lower. This is because different roles share the common knowledge of the underlying scene, and the joint training of different roles to some extent reveals the scene knowledge from different sides, assisting the unified understanding of the unified agent. Our combined student-training recipe therefore improves the complete harness over directly deploying separate specialists, while retaining a single role-conditioned model.

E Training Data Construction and Training Details

E.1 Training Data Construction

We construct role-local training data from SpaceVista (Sun et al., 2025) and VSI-590K (Yang et al., 2025c). The construction pipeline first establishes scene-grounded references, then collects tool observations and success statistics, and finally derives supervision for the planner, executor, and verifier. Visual evidence is prepared offline, while policy responses are sampled during optimization.

Scene filtering and annotation recovery. We retain examples that can be traced back to their upstream 3D scenes and provide object-level 3D bounding boxes in world coordinates. Each retained example links the question and its frames to the corresponding scene and annotated objects. Examples without the required scene or object annotations are excluded. The recovered annotations provide a common spatial reference for checking object visibility and associating observations across frames.

Frame-level visibility and semantic checks. For each target object and each frame associated with a question, we first perform a camera-based visibility test using the upstream 3D annotations. We then apply a VLM-based semantic check to determine whether the visible content matches the target concept specified in the question. These checks establish frame-level references rather than assuming that an object mentioned in a question is visible throughout the video. Together, they connect the question's semantic targets to the observations in individual frames.

Tool evaluation and evidence collection. We sample object concepts from the retained questions and apply the tools relevant to their evidence requirements. Tool outputs are evaluated against the scene-grounded and semantically checked references to obtain empirical success statistics for the corresponding acquisition stages. A success rate measures the fraction of evaluated calls that satisfy the associated evidence requirement. We retain both successful and failed outputs. The aggregated statistics provide supervision for route selection, while the individual outputs supply evidence contexts for verifier training. We additionally introduce human-designed perturbations to broaden the range of evidence conditions represented in these contexts.

E.2 Role-level Examples and Supervision

The constructed references are used to evaluate sampled responses through the role-specific rewards in Equation (4). They specify the required decisions and evidence conditions rather than prescribing a single reference response string.

Planner. For each question, we identify its evidence requirements and use the precomputed tool success statistics to assign a preferred primary route and order the available backup routes. Initial-planning examples concern the full set of required evidence. Memory-conditioned replanning examples additionally account for evidence already retained, so their route targets concern the remaining evidence gaps. The planner is evaluated on output format, evidence coverage, and primary-route ranking through R_{fmt}, R_{cov}, and R_{rank}, respectively. Consistent with Equation (4), the explicit ranking reward is activated only for initial planning.

Executor. Executor contexts specify the selected route, its evidence specifications, and the relevant tool interfaces. The executor generates the executable program that instantiates this route. The reference checks assess whether the output follows the required format, invokes functions consistent with the planned operations, and supplies arguments satisfying the tool-specific requirements. These checks determine R_{fmt}, R_{func}, and R_{param}. This supervision targets correct implementation of the route, while the reliability of the resulting visual observations is assessed by the verifier.

Verifier: semantic validity. We construct verifier contexts from the collected tool outputs and controlled perturbations. Each context includes the requested concept and the candidate observation to be assessed. Semantic reference decisions are derived from the frame-level visibility and VLM checks, allowing supervision over valid observations, missed targets, distractors, and noisy evidence. The semantic reward R_{sem} evaluates whether the verifier correctly judges the supplied observation

Setting	Value
Backbone	Qwen3-VL-8B-Instruct
Teacher optimization	DAPO, independently for each role
Teacher training steps	100 per role
Unified student optimization	MOPD+DAPO
Unified student training steps	300
Rollout batch size	384 prompt groups
Update batch size	128 prompt groups
Maximum responses per prompt	10 (`rollout.n`)
Teacher advantage clipping	$[-5, 5]$
Teacher-loss coefficient	$\lambda = 0.1$

Table 7: **Training settings.** The rollout and update batch sizes apply to both DAPO and MOPD+DAPO and count prompt groups rather than individual responses. Teacher-feedback settings apply to the combined objective.

against the requested concept. A tool call returning an output is therefore distinct from that output providing valid evidence.

Verifier: spatial association. Spatial verification additionally checks whether candidate observations correspond to the same physical 3D object. We use the upstream visibility checks for the relevant frames and the candidates' associations with annotated 3D objects to construct these labels. A positive association requires correspondence to a single physical instance; observations belonging to different instances are nonmatches even when they share the same semantic category. These labels supervise R_{spat} and support the association decisions used during memory updates. Semantic validity and spatial identity are thus separate targets: recognizing the requested category does not establish that two observations depict the same object.

Role-local contexts. The three datasets retain the information needed for their respective decisions: questions and memory conditions for planning, route specifications for execution, and candidate observations with verification cues for verification. Successful outputs, failed outputs, and controlled perturbations expose the roles to different intermediate evidence conditions without requiring a complete live trajectory for every training example. Precomputing visual evidence also avoids repeated waits for GPU-hosted visual tools during policy optimization.

E.3 Training Details

Teacher specialization and student consolidation. We use Qwen3-VL-8B-Instruct as the trained-agent backbone. We first optimize the planner, executor, and verifier teachers separately with DAPO (Yu et al., 2026), using their corresponding role-local datasets and rewards. Each teacher is trained for 100 steps. We then freeze the teachers and train a unified role-conditioned student for 300 steps by combining MOPD (Ma et al., 2026) teacher feedback with DAPO task-reward updates. The teacher matched to the current role evaluates the student's sampled responses. DAPO-only training uses the task-reward objective without this teacher-feedback term.

Rollout and update batches. Both DAPO and MOPD+DAPO use a rollout batch size of 384 prompt groups and an update batch size of 128 prompt groups. We sample at most 10 responses for each prompt, corresponding to the maximum setting of `rollout.n`. These batch sizes count input contexts rather than individual responses. An update therefore contains up to 1,280 responses; the response count, rather than the number of prompt groups, corresponds to B in Equation (5).

Outcome-dependent supervision. For combined MOPD+DAPO training, we classify each rollout group by its mean binary success. All-success groups are discarded, and further sampling fills the retained update batch. All retained groups are evaluated by the DAPO task-reward objective. Teacher feedback is activated only for all-failure groups, while mixed groups receive task-reward updates without the teacher term. The binary success used for this allocation is distinct from the composite role reward in Equation (4). Consequently, an all-failure group can still provide a task-reward learning signal when its component rewards vary across responses.

Task and teacher updates. The DAPO objective uses standardized within-group task advantages and policy-ratio clipping. For teacher feedback, the frozen role-matched teacher scores each student-generated token conditioned on the student's own prefix. Following Equation (5), we compute the teacher advantage as the teacher log probability minus the rollout student's log probability, clip it to $[-5, 5]$, and stop gradients through this value. This advantage weights a separate token log-probability loss with coefficient $\lambda = 0.1$; it is not added to the task advantage before policy-ratio clipping. The teacher advantages are not group-standardized, and the teacher term does not use a separate importance-ratio surrogate. We average the teacher contribution over valid response tokens and then over all selected responses, with zero contribution from groups whose teacher mask is inactive.