SPATIALSKILL: SELF-EVOLVING SKILLS FOR CROSS-VIEW SPATIAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-view spatial reasoning requires a model to align different viewpoints into a coherent spatial representation, yet this ability remains challenging for vision-language models despite being natural to humans. Existing methods typically improve spatial reasoning by updating model weights, which keeps the acquired knowledge implicit and tied to a specific backbone. We propose *SpatialSkill*, a weight-update-free framework that enables a frozen vision-language model to accumulate explicit natural-language reasoning skills from offline trajectories. Unlike symbolic tasks, perceptual skills cannot be reliably verified simply by executing them: a plausible spatial rule may lack visual support or require transformations that the frozen model cannot perform. SpatialSkill therefore admits candidate skills only after visual-grounding and executability checks, constrains manual evolution to prevent harmful regressions, and routes skills by spatial-reasoning category to reduce negative transfer. On CityCube, across four frozen executors, SpatialSkill yields consistent gains, and a 9B executor equipped with SpatialSkill surpasses the strongest closed-source reference in our evaluation. The skills are stored in a versioned natural-language manual, making the reasoning strategies explicit and auditable without modifying model parameters.

1 INTRODUCTION

Spatial intelligence (Hong et al., 2026; Jia et al., 2026; Majumdar et al., 2024) is central to embodied cognition. Missions such as UAV inspection and disaster response require relating objects, directions, and structures across aerial and ground-level observations. We refer to this ability as cross-view spatial reasoning (Zhou et al., 2026; Peng et al., 2026; Xiang et al., 2026; Du et al., 2024; Qiao et al., 2026). Solving these cross-view problems often requires mentally transforming and aligning viewpoints rather than relying only on visual similarity or linguistic priors, yet current vision-language models remain weak at these operations (Huang et al., 2026; Zhang et al., 2026b; 2025; Xu et al., 2026b; Lee et al., 2025). Existing approaches (Ma et al., 2025; Hong et al., 2023; Chen et al., 2024) typically improve spatial reasoning through parameter or representation updates, which encode the acquired knowledge implicitly in a particular model and make the resulting reasoning strategies difficult to inspect or revise independently of the model. We therefore ask: **can a frozen VLM improve its cross-view spatial reasoning without parameter updates, simply by accumulating explicit natural-language skills?** Fig. 1 illustrates that even a simple visually grounded rule can correct an error made by a frozen executor. The key challenge is therefore not whether explicit skills can help, but how to acquire, maintain, and deploy them reliably.

This raises three key challenges. ❶ **Reliable skill admission.** Unlike symbolic skills that can be validated through direct execution, a natural-language spatial rule lacks a direct rule-level verifier and may sound plausible even when unsupported by visual evidence (Han et al., 2026; Tong et al., 2026; Yang et al., 2025b). Moreover, a geometrically valid rule may still require reasoning operations beyond what the frozen executor can reliably perform. Thus, candidate skills must be checked for both visual grounding and executor executability before entering the skill library. ❷ **Stable skill evolution.** Even when individual candidate skills are plausible, repeated revisions may accumulate noise, interact unexpectedly, or remove useful rules during compression, gradually degrading the skill library over time (Yashwante & Yu, 2026; Sun et al., 2026). Reliable skill evolution therefore requires controlling growth and preventing harmful updates from being propagated. ❸ **Category-dependent skill effectiveness.** Different spatial-reasoning categories involve different reasoning patterns and

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.

operations, so the same manual or inference strategy may not benefit every category equally (Wang et al., 2026; Zhou et al., 2025; Wang et al., 2025b; Khalid et al., 2025). Uniformly injecting skills can therefore introduce negative transfer, motivating category-conditioned deployment.

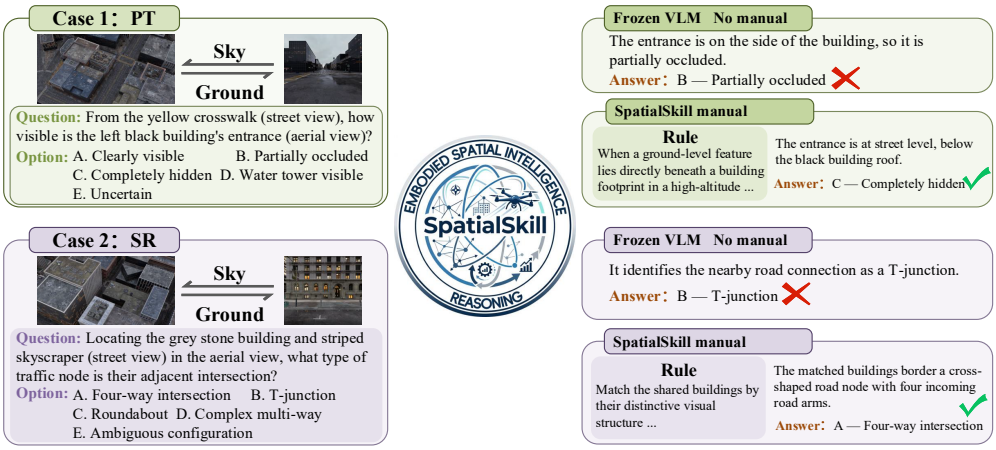


Figure 1: A perspective-taking (PT) instance and a spatial-relation (SR) instance, each pairing aerial and ground views. Frozen executor fails on both without manuals, while a visually grounded rule from the evolved SpatialSkill manual enables it to correct its prediction without any weight update.

Motivated by these challenges, we introduce ***Self-Evolving Skills for Cross-View Spatial Reasoning (SpatialSkill)***, a framework that improves a frozen VLM by accumulating explicit natural-language skills without updating its parameters. SpatialSkill consists of three core components. First, a grounded admission gate admits candidate skills only after checks for visual support and compatibility with the frozen executor's capabilities. Second, stability controls regulate skill accumulation by limiting uncontrolled growth, protecting stable skills from blind removal, and rejecting updates that cause excessive performance degradation. Third, category-conditioned routing selects how the final manual is used for each spatial reasoning category, reducing negative transfer from uniform skill deployment. Together, these components produce a versioned and auditable skill manual that can be selectively deployed while keeping the executor unchanged.

Our contributions are threefold. (i) To our knowledge, we are the first to study self-evolving explicit natural-language skills for frozen VLMs in real-world cross-view spatial reasoning, identifying three requirements for reliability: grounded admission, stable accumulation, and category-conditioned deployment. (ii) We propose SpatialSkill, a framework that addresses these requirements through grounded admission, stability-constrained evolution, and category-conditioned routing, storing acquired reasoning strategies in an explicit, versioned, and auditable natural-language manual without parameter updates. (iii) On CityCube across four frozen executors, SpatialSkill consistently yields accuracy gains of 2.65–4.92 percentage points. With SpatialSkill, the 9B executor reaches 57.58%, outperforming the strongest closed-source baseline evaluated in our experiments.

2 Related Work

Spatial Reasoning in Vision-Language Models. Spatial reasoning is a central diagnostic axis for vision-language models. Recent benchmarks examine complementary capabilities, including egocentric video (Yang et al., 2025a), multi-viewpoint localization (Li et al., 2025), multi-image intelligence (Yang et al., 2026), cognitive-psychology axes (Jia et al., 2026), unified cross-task evaluation (Huang et al., 2026), and fine-grained failure analysis (Stogiannidis et al., 2025; Liu et al., 2025). Across these settings, models remain challenged when reasoning requires viewpoint transformation and cross-view alignment rather than language priors or appearance matching. Recent methods address these limitations through several forms of model adaptation. Reinforcement learning improves grounded multi-step 3D reasoning (Batra et al., 2025), activates latent abilities (Zhan et al., 2025), and extends to multi-view transformations (Li et al., 2026c). Supervised approaches inject depth signals (Cai et al., 2025) or align implicit representations (Li et al., 2026a), while robot-scale pretraining embeds spatial structure (Qu et al., 2025). Together, these studies establish the

importance of explicit spatial structure and learned adaptation for cross-view reasoning, with most improvements encoded in model weights or representations.

Self-Evolving Agents and Skill Accumulation. Self-improvement from practice provides a complementary direction for turning experience into reusable knowledge (Gao et al., 2025). Existing approaches develop skill libraries that compile procedures into APIs (Zhao et al., 2025; Zheng et al., 2025; Prabhu et al., 2026; Tian et al., 2026), recursive augmentation with RL (Xia et al., 2026; Liu et al., 2026a), resource-to-library conversion (Shen et al., 2026; Ju et al., 2026), structured experience notes (Xu et al., 2025), and OS-style hierarchies (Kang et al., 2025). Many of these systems evolve executable artifacts or model parameters, allowing environments or training loops to provide direct evaluation signals. A close text-playbook precedent is ACE (Zhang et al., 2026a), which accumulates reusable lessons under text-task scoring. These two research directions leave a less explored intersection: evolving explicit natural-language skills for frozen VLMs in perceptual cross-view reasoning. A skill may lack visual support, exceed the executor's capabilities, or interact negatively with previously accumulated guidance. This setting therefore raises questions of grounded admission, stable accumulation, and selective deployment, which motivate SpatialSkill.

3 Methodology

3.1 Problem Definition and Framework Overview

We study cross-view spatial reasoning in a multiple-choice setting. Each example is defined as $x = (I_{1:k}, q, \mathcal{O})$, where $I_{1:k}$ represents k distinct views of a scene, q is a question, and $\mathcal{O}$ is the candidate answer set. A frozen executor predicts an answer $\hat{y} \in \mathcal{O}$. Each example belongs to one of five spatial reasoning categories: mental reconstruction (MR), perspective taking (PT), spatial relation reasoning (SR), world knowledge (WK), and comprehensive reasoning (CR), which are used for both category-wise evaluation and routing. During skill evolution, training examples additionally provide reference reasoning trajectories used as attribution references during reflection. SpatialSkill distills reusable reasoning patterns from these trajectories into natural-language skills, which are organized into a structured manual to guide the unchanged executor at inference.

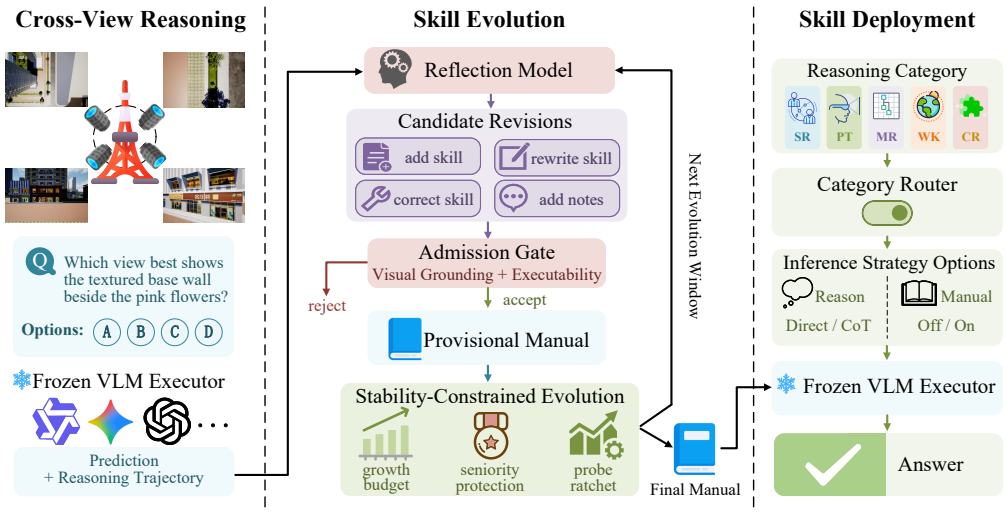


Figure 2: Overview of Self-Evolving Skills for Cross-View Spatial Reasoning (SpatialSkill)

SpatialSkill converts offline reasoning trajectories from a frozen VLM into a reusable natural-language skill manual and selectively deploys that manual at inference. Fig. 2 summarizes this process in three stages: the frozen VLM executor first produces cross-view reasoning trajectories; a skill-evolution module reflects on these trajectories and updates the manual under grounding and stability constraints; finally, a category router determines how the manual is used for each reasoning category.

3.2 SKILL REFLECTION AND GROUNDED ADMISSION

Reasoning trajectories capture useful experience from the frozen executor, but remain instance-specific. SpatialSkill therefore uses reflection to abstract reusable revisions to a natural-language skill manual. Because reflected candidates may still lack visual support or require operations beyond the executor's capabilities, they are screened for visual grounding and executability before entering the manual. We maintain a skill manual M, initialized empty and organized into four sections: planning, cross-view alignment, spatial reasoning, and failure avoidance, together with execution notes. To evolve the manual incrementally, we process the evolution stream in windows. For window w, let T_w denote the collected trajectories and M_{w-1} the current manual. A fixed reflection model Φ proposes candidate revisions:

$$\Delta_w \;=\; \Phi\big(T_w,\, M_{w-1}\big), \tag{1}$$

where Δ_w denotes the proposed revisions. Each revision performs one of four operations: adding a skill, rewriting a skill, correcting a skill, or adding an execution note. Each candidate is further associated with skill-level or execution-level attribution estimated against the reference trajectories. Execution-level revisions only update execution notes, preventing episodic noise from contaminating reusable skill rules.

Because reflected candidates are not necessarily reliable, each candidate rule r is screened by an admission gate, implemented by the fixed reflection model, that checks visual grounding and executability:

$$\mathcal{G}(r) \;=\; \mathcal{V}(r) \;\wedge\; \mathcal{E}(r), \tag{2}$$

where $\mathcal{V}(r) = 1$ requires the rule to be supported by visual evidence, while $\mathcal{E}(r) = 1$ requires the prescribed operation to remain within the frozen executor's capabilities. Only candidates satisfying $\mathcal{G}(r) = 1$ are admitted into the manual. Rejected candidates are logged with their triggering evidence, keeping the evolution process auditable.

The admitted revisions are then applied to the current manual to form a provisional manual:

$$\widetilde{M}_w \;=\; \mathcal{U}\Big(M_{w-1}, \big\{\, \rho \in \Delta_w : \mathcal{G}(r_\rho) = 1 \,\big\}\Big), \tag{3}$$

where $\mathcal{U}$ applies the admitted revisions and r_ρ denotes the rule introduced or modified by revision ρ. Although each incorporated revision is individually admissible, multiple revisions may still interact negatively after accumulation. The provisional manual is therefore further evaluated by the stability controls described next before becoming the delivered manual M_w.

3.3 STABILITY-CONSTRAINED SKILL EVOLUTION

The admission gate ensures that individual candidate rules are visually grounded and executable, but admissible revisions do not necessarily yield a better manual after accumulation. Multiple revisions may interact unexpectedly or gradually introduce noise into the skill library. We therefore evaluate each provisional manual on a held-out probe set P before delivering an update. The probe set is stratified by reasoning category and remains disjoint from the evolution stream. It is never used for reflection or revision. The probe score of a manual is defined as:

$$A(M) \;=\; \frac{100}{|P|} \sum_{x \in P} \mathbf{1}\big\{\, \hat{y}_M(x) = y(x) \,\big\}, \tag{4}$$

where $\hat{y}_M(x)$ denotes the executor prediction under manual M, $y(x)$ is the answer option, and $\mathbf{1}\{\cdot\}$ is the indicator function.

Manual-level evaluation alone, however, does not prevent instability during repeated evolution. As rules accumulate, the manual may reach its capacity and require compression. Blind compression can remove useful rules together with noisy ones, creating new failures and further revisions. We characterize this pathological grow-and-compress regime as:

$$\lim_{W \to \infty} \frac{1}{W} \sum_{w=1}^{W} \mathbf{1}\{R_w\} \;=\; 1 \qquad \text{and} \qquad \mathbb{E}\big[A(M_W)\big] \;<\; A(M_0), \tag{5}$$

where R_w denotes the event that compression is triggered in window w. To control this failure pattern, SpatialSkill uses three complementary controls: a Growth Budget, Seniority Protection,

and a Probe-Score Ratchet. They respectively limit manual expansion, protect long-standing rules during compression, and determine whether the resulting update is delivered.

First, excessive growth increases the need for later compression. We impose a net-growth budget:

$$|M_w| \;-\; |M_{w-1}| \;\le\; \beta, \tag{6}$$

where $|M|$ is the number of rules and β is the maximum net number of rules added in one window. This constraint limits excessive rule accumulation before compression.

Second, compression should not discard long-standing rules indiscriminately. We therefore protect stable rules from blind removal. A rule that survives multiple windows remains in the next manual unless an evidence-targeted revision explicitly modifies it:

$$r \in M_{w-1},\;\; \mathrm{age}(r) \ge \tau,\;\; r \notin \mathcal{T}_w \quad\Longrightarrow\quad r \in M_w, \tag{7}$$

where $\mathrm{age}(r)$ is the number of consecutive windows survived by rule r, τ is the protection window, and $\mathcal{T}_w$ is the set of rules explicitly targeted by evidence-based revisions in window w. Thus, long-standing rules are protected from blind compression, while evidence-targeted revisions remain possible.

Finally, controlling growth and compression does not guarantee that the resulting manual improves the executor as a whole. We therefore use a probe-score ratchet to determine whether a provisional update is delivered. Let $b_0 = A(M_0)$ denote the initial probe score:

$$M_w = \begin{cases} \widetilde{M}_w, & A(\widetilde{M}_w) \ge b_{w-1} - \varepsilon, \\ M_{w-1}, & \text{otherwise}, \end{cases} \qquad b_w \;=\; \max\big\{\, b_{w-1},\; A(M_w) \,\big\}, \tag{8}$$

where $\widetilde{M}_w$ denote the constrained provisional update passed to the ratchet, b_w records the best accepted probe score, and ε absorbs evaluation noise. Updates that exceed this tolerance are rejected, while the historical best score is preserved. Together, these constraints yield the following tolerance-band invariant:

$$A(M_w) \;\ge\; A^\star_w - \varepsilon, \qquad A^\star_w \;=\; \max_{t \le w} A(M_t). \tag{9}$$

where $A^\star_w$ denotes the best probe score achieved by any accepted manual up to window w. Therefore, evolution may plateau or fluctuate within the tolerance range, but large probe-score regressions are not delivered.

3.4 Category-Conditioned Skill Deployment

The final skill manual is not uniformly beneficial across reasoning categories. A strategy that helps one category may provide limited benefit for another. We therefore select the inference strategy according to the reasoning category instead of applying the manual globally.

Given the precomputed reasoning category $c \in \mathcal{C}$, the router selects a strategy $s = (i, j)$ from $\mathcal{S} = \{0, 1\}^2$, where $i = 0$ denotes direct answering, $i = 1$ denotes chain-of-thought reasoning, $j = 1$ denotes providing the final manual M_W, and $j = 0$ denotes inference without the manual. The routing policy is a mapping $\pi : \mathcal{C} \to \mathcal{S}$. If the performance of each strategy were known exactly, the optimal routing policy would maximize expected accuracy:

$$\pi^\star \;=\; \arg\max_{\pi} \sum_{c \in \mathcal{C}} p(c)\, \mathrm{Acc}\big(c, \pi(c)\big), \tag{10}$$

where $p(c)$ is the category prior and $\mathrm{Acc}(c, s)$ denotes the accuracy of strategy s on category c. In practice, these quantities are estimated from a finite probe set. To reduce sensitivity to sampling noise, we switch strategies when improvement satisfies a conservative confidence-bound criterion.

Let $\hat{a}_{c,s}$ denote the empirical accuracy of strategy s on category c. Using a Wilson confidence interval $[\ell_s(c), u_s(c)]$, the router selects a candidate strategy only when its lower bound exceeds the default strategy's upper bound by a margin δ:

$$\pi(c) \;=\; \arg\max_{s \in \mathcal{F}(c)} \hat{a}_{c,s}, \qquad \mathcal{F}(c) = \Big\{ s \in \mathcal{S} : \ell_s(c) - u_{s_0(c)}(c) \ge \delta \Big\}, \tag{11}$$

where $s_0(c)$ is the stronger of the two non-manual strategies on the probe set. If $\mathcal{F}(c)$ is empty, the router retains the default strategy. This conservative rule avoids unnecessary switching caused by probe variance.

Table 1: Main results on the CityCube dataset. Accuracy is reported in percent. Option-wise Acc. Range (pp) is the maximum difference in conditional accuracy across predicted answer options.

Method	Accuracy (%) ↑							Option-wise Acc. Range (pp) ↓
	Overall	Δ	CR	MR	PT	SR	WK	
Human	88.3	–	93.1	92.4	87.4	90.2	78.6	–
Paired executors								
Qwen3.5-4B + SpatialSkill	**52.08**	+4.92	43.24	51.22	41.77	56.10	64.06	23.52
Qwen3.5-4B	47.16	–	51.35	53.66	39.87	45.53	52.34	27.28
Qwen3.5-9B + SpatialSkill	**57.58**	+3.03	59.46	67.07	51.90	52.85	62.50	23.10
Qwen3.5-9B	54.55	–	56.76	64.63	47.47	47.97	62.50	39.21
Gemma-3-4B + SpatialSkill	**51.89**	+2.65	51.35	60.98	41.77	52.03	58.59	47.94
Gemma-3-4B	49.24	–	51.35	59.76	37.97	52.03	53.12	72.12
GPT-5.6-luna + SpatialSkill	**56.82**	+3.03	62.16	62.20	45.57	56.91	65.62	19.37
GPT-5.6-luna	53.79	–	48.65	57.32	42.41	59.35	61.72	30.43
Open-source reference models								
Qwen3-VL-4B	53.03	–	56.76	59.76	41.14	56.91	58.59	24.93
Qwen3-VL-8B	50.95	–	45.95	53.66	39.24	51.22	64.84	24.96
Qwen3-VL-32B	53.79	–	51.35	59.76	41.77	54.47	64.84	18.04
Qwen3.5-2B	46.59	–	48.65	53.66	33.54	52.85	51.56	49.77
Qwen3.5-27B	58.14	–	51.35	64.63	45.57	56.10	73.44	17.36
Qwen3.5-35B-A3B	53.22	–	56.76	54.88	41.77	55.28	63.28	31.00
InternVL3-8B	49.43	–	48.65	47.56	43.67	49.59	57.81	20.14
InternVL3-14B	51.70	–	56.76	51.22	44.30	52.03	59.38	51.97
Gemma-4-31B	54.55	–	40.54	53.66	41.77	56.10	73.44	26.52
Gemma-4-26B-A4B	52.27	–	40.54	56.10	41.14	56.91	62.50	17.78
Gemma-3-12B	50.95	–	56.76	54.88	39.87	47.15	64.06	39.70
Skywork-VL-Reward-7B	45.83	–	37.84	45.12	34.81	51.22	57.03	28.53
Open-source Spatial models								
Spatial-SSRL-7B	46.97	–	45.95	46.34	34.81	50.41	59.38	13.76
SpaceR-SFT-7B	44.13	–	54.05	47.56	36.71	33.33	58.59	41.07
Spatial-MLLM-subset-sft	39.77	–	37.84	36.59	36.71	39.02	46.88	73.94
ViLaSR	49.81	–	51.35	56.1	36.08	47.97	64.06	26.79
Closed-source reference models								
GPT-5.4	50.38	–	37.84	57.32	41.14	50.41	60.94	31.00
Gemini-3.5-Flash	48.11	–	40.54	53.66	39.87	50.41	54.69	16.95
GPT-5.6-sol	56.06	–	45.95	67.07	44.30	54.47	67.97	24.36
Qwen3.7-plus	56.06	–	48.65	62.20	46.20	56.91	65.62	12.96
Gemini-3.7-Flash	52.27	–	45.95	57.32	43.67	53.66	60.16	14.28
Claude-Sonnet-5	52.65	–	45.95	50.00	49.37	48.78	64.06	24.05
Claude-Opus-5	52.08	–	35.14	60.98	44.30	48.78	64.06	20.67

4 EXPERIMENTS

4.1 MAIN RESULTS

Table 1 reports the main results on CityCube (Xu et al., 2026a). We focus on paired comparisons between each frozen executor and its SpatialSkill-enhanced counterpart, while also reporting stronger open-source, spatially specialized, and closed-source models as reference points. Full implementation and metric details are provided in Appendix D.

SpatialSkill consistently improves frozen executors. Across all four paired executors, SpatialSkill improves overall accuracy by 2.65–4.92 percentage points, with a mean gain of 3.41 points. Qwen3.5-4B improves from 47.16% to 52.08%, Qwen3.5-9B from 54.55% to 57.58%, Gemma-3-4B from 49.24% to 51.89%, and GPT-5.6-luna from 53.79% to 56.82%. Because the executor weights remain frozen throughout, these gains are achieved through explicit skill evolution and prompt-level deployment rather than parameter updates. The resulting knowledge remains stored in a natural-language manual, making the acquired reasoning strategies explicit, inspectable, and retractable.

SpatialSkill narrows the gap to stronger reference models. Qwen3.5-9B equipped with SpatialSkill reaches 57.58%, only 0.56 percentage points below the larger Qwen3.5-27B baseline at 58.14%. It also exceeds the best-performing closed-source reference models in our evaluation,

GPT-5.6-sol and Qwen3.7-plus, both at 56.06%, by 1.52 points. Although Qwen3.5-27B remains the best-performing non-human model overall, these results show that explicit skill evolution can recover part of the performance gap to stronger models without modifying the executor backbone.

The gains are category- and executor-dependent. For Qwen3.5-4B, SpatialSkill improves SR and WK by 10.57 and 11.72 percentage points, respectively, but decreases performance on CR and MR. GPT-5.6-luna gains 13.51 points on CR but slightly declines on SR, while Qwen3.5-9B improves or matches its baseline across all five categories and Gemma-3-4B benefits mainly on MR, PT, and WK. This heterogeneity shows that skill effectiveness is not uniform across reasoning categories or executors, supporting category-conditioned routing rather than a single global deployment strategy.

A substantial gap remains between current models and human performance. For context, CityCube reports an overall human accuracy of 88.3%, whereas the best-performing non-human model in our evaluation, Qwen3.5-27B, reaches 58.14%, leaving a gap of 30.2 percentage points. The gap is particularly pronounced for perspective taking (PT), where the best non-human result is 51.9% compared with 87.4% for humans, but is much smaller for world knowledge (WK), at 73.44% versus 78.6%. This pattern suggests that current models remain particularly weak on perspective-dependent operations involving viewpoint transformation and cross-view alignment. Moreover, all four evaluated open-source spatial models remain below 50% overall accuracy, indicating that existing spatially specialized models still leave substantial room for improvement on robust cross-view reasoning.

The gains are not accompanied by larger option-dependent disparity. As a descriptive diagnostic, we report the option-wise accuracy range, defined as the maximum difference in conditional accuracy across predicted answer options. Across all four paired executors, SpatialSkill reduces this range: from 27.28 to 23.52 for Qwen3.5-4B, 39.21 to 23.10 for Qwen3.5-9B, 72.12 to 47.94 for Gemma-3-4B, and 30.43 to 19.37 for GPT-5.6-luna. Thus, the overall accuracy gains are not accompanied by greater option-dependent disparity.

4.2 Ablation Experiments

Table 2 reports results under the same frozen executor. Variant definitions are provided in Appendix E.

Table 2: Overall ablations validation accuracy (%) on Qwen3.5-4B and Gemma-3-4B.

Condition	Qwen	Gemma
Direct baseline	47.16	49.24
CoT baseline	49.62	48.67
w/o attribution-aware reflection	50.00	47.35
w/o visual-evidence criterion	50.76	50.74
w/o executability criterion	51.14	51.32
w/o joint admission gate	50.19	50.95
w/o growth budget	51.57	51.31
w/o seniority protection	51.68	51.44
w/o probe-score ratchet	51.14	50.57
SpatialSkill	52.08	51.89

Learned skills improve over no-skill baselines. The direct baseline reaches 47.16% on Qwen3.5-4B and 49.24% on Gemma-3-4B, while the CoT baseline reaches 49.62% and 48.67% without a manual. SpatialSkill reaches 52.08% and 51.89%, yielding gains of 4.92 and 2.65 percentage points over the corresponding Direct baselines. The gap between CoT and SpatialSkill shows that the improvement cannot be explained by CoT prompting alone.

Attribution-aware reflection is important for reliable skill induction. Removing attribution-aware reflection reduces accuracy to 50.00% on Qwen3.5-4B and 47.35% on Gemma-3-4B, a loss of 2.08 and 4.54 points from the complete system. The larger degradation on Gemma suggests that separating reusable and execution-specific feedback is particularly important for this executor. The result suggests that separating reusable skills from execution-specific notes is important for stable reflection.

Grounded admission benefits from both evidence and executability checks. Removing the visual-evidence criterion gives 50.76% and 50.74%, whereas removing the executability criterion gives 51.14% and 51.32% on Qwen3.5-4B and Gemma-3-4B, respectively. Removing the joint admission gate gives 50.19% and 50.95%. All three variants underperform the complete system on both executors. On Qwen3.5-4B, removing both criteria is more harmful than removing either criterion alone; on Gemma-3-4B, the effects are not strictly additive. These results support the utility of grounded admission while also showing that the relative contribution of the two criteria is executor-dependent.



(a) Refactor-event windows. (b) Qwen3.5-4B manual size. (c) Gemma-3-4B manual size.

Figure 3: Stability diagnostics over 43 evolution windows for SpatialSkill and Naive.

Stability controls provide complementary protection. Removing growth budget reduces accuracy to 51.57% and 51.31%, while removing seniority protection gives 51.68% and 51.44%. Removing probe-score ratchet produces larger drops of 0.94 points on Qwen3.5-4B and 1.32 points on Gemma-3-4B, reaching 51.14% and 50.57%. This pattern suggests that the ratchet serves as the primary safeguard against harmful revisions, whereas budget and seniority constraints regulate the trajectory prior to acceptance. Their effects are therefore complementary rather than interchangeable.

4.3 Stability of Skill Evolution

Perceptual spatial reasoning has no symbolic execution oracle, so we measure stability over 43-window trajectories from the same frozen executor and training stream. A refactor event counts when the manual is explicitly refactored or shrinks between adjacent windows. Naive denotes the ablation variant following the same skill-evolution loop but removing the probe-score ratchet, lacking the safeguard against harmful revision deployment.

Table 3: Routing summary and paired error turnover on the held-out validation split.

(a) Routing summary

Executor	Main	Probe replay	No-margin	Oracle
Qwen3.5-4B	52.08	53.03	51.33	54.17
Gemma-3-4B	51.89	50.95	51.33	53.03

(b) Paired error turnover

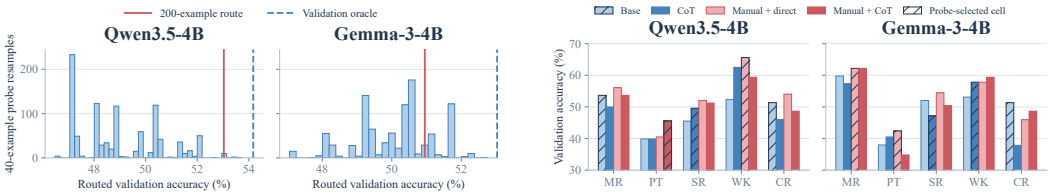
Executor	Base	Main	Fixed	Broken	Net	Persistent
Qwen3.5-4B	47.16	52.08	87	61	+26	192
Gemma-3-4B	49.24	51.89	70	56	+14	198

The ratchet blocks harmful revisions and reduces unstable refactors. SpatialSkill reduces refactor-event windows from 35 to 6 on Qwen3.5-4B and from 24 to 8 on Gemma-3-4B in Fig. 3a. The ratchet rejects 14 and 21 revisions, all of which score lower than the accepted reference, with worst drops of 9.0 and 5.5 points. In the evaluated trajectories, no accepted revision decreases the probe score. Manual-size contractions occur in 17 and 16 Naive windows, compared with 6 and 8 SpatialSkill windows shown in Figs. 3b–3c.

Stability translates into better held-out validation. SpatialSkill reaches 52.08% versus 51.14% for Naive on Qwen3.5-4B, and 51.89% versus 50.57% on Gemma-3-4B. Thus, the safeguards do not merely suppress change. The joint admission gate screens candidate rules, while the retained growth and protection controls constrain the trajectory. The ratchet prevents deploying revisions whose probe score falls beyond the tolerance band. Safe compression remains allowed, but harmful revisions are not deployed.

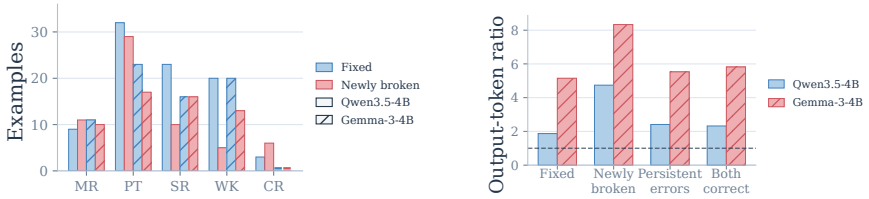
4.4 Skill-Aware Routing Analysis

Routing balances empirical performance and conservative deployment. The analysis-only probe replay reaches 53.03% for Qwen3.5-4B and 50.95% for Gemma-3-4B, compared with the Main results of 52.08% and 51.89%. The No-margin diagnostic uses the same probe-based routing rule with the switching margin set to $\delta = 0$, reaching 51.33% on both executors. The non-deployable validation oracle reaches 54.17% and 53.03% in Table 3 (a). Probe replay applies a fixed finite-probe route to the bank of validation predictions. It diagnoses sensitivity to route estimation. The lower No-margin accuracy suggests that conservative switching can reduce over-selection under finite-probe uncertainty. Moreover, routed accuracy varies widely across 1,000 resampled 40-example probes in Fig. 4a, which further motivates the default-and-margin rule: under insufficient evidence, the router retains the default instead of selecting a noise-inflated winner.



(a) Sensitivity under resampled probes. (b) Validation cells with probe selections hatched.

Figure 4: Probe-derived routing diagnostics: full-probe route and validation oracle.



(a) Error turnover by CvSI category. (b) Output-token expansion by transition.

Figure 5: Paired error-turnover diagnostics for the frozen predictions.

Optimal delivery is family- and executor-dependent. Qwen3.5-4B favors manual+direct on MR, SR, and WK, and manual+CoT on PT. Gemma-3-4B favors manual+direct on PT and SR, manual+CoT on WK, and the base configuration on CR, as shown in Fig. 4b. The hatched probe-selected cells need not equal the validation-best cells, because validation labels never enter policy construction. This variation supports the multi-referential heterogeneity premise: delivery utility depends on cognitive family and executor.

4.5 Paired Error Analysis

Gains come from targeted repairs rather than uniform correction. We froze base and SpatialSkill predictions on the same validation examples. Qwen3.5-4B repairs 87 baseline errors and introduces 61 new errors, for a net gain of 26. Gemma-3-4B repairs 70 and breaks 56, for a net gain of 14 shown in Table 3 (b). Repairs concentrate in WK and SR for Qwen and in WK and PT for Gemma, whereas 192 and 198 baseline errors persist in Fig. 5a. Thus, the evolved manual improves aggregate accuracy through targeted repairs rather than uniform correction of the baseline error set.

Longer outputs do not indicate correct answers. Conditioned on transition type, the mean SpatialSkill-to-base output-token ratio is largest for newly broken examples (4.74× for Qwen and 8.33× for Gemma), and remains elevated for fixed examples, as shown in Fig. 5b. Length expansion is therefore only a descriptive correlate of the applied reasoning process rather than a reliable indicator of correctness.

5 Conclusion

In this paper, we study the self-evolution of spatial skills in frozen vision-language models and propose *SpatialSkill*, a framework that accumulates explicit natural-language skills without parameter updates. Our analysis highlights three requirements for reliable perceptual skill evolution: grounding candidate skills, stabilizing repeated accumulation, and selectively deploying skills across reasoning categories. Experiments on CityCube show that explicit skill evolution can improve frozen executors while reducing instability across repeated evolution. We further find that the benefit of skill usage varies across spatial-reasoning categories, motivating category-aware skill routing rather than a uniform manual application strategy. The limitation is that current routing relies on precomputed category annotations, and natural-language manuals can improve reasoning strategies but do not modify the executor's underlying visual capabilities. Future work will explore the use of explicit manuals as fine-tuning priors to combine explicit and implicit knowledge updates.

AI USE STATEMENT

Generative AI models are used as explicit components of the research pipeline. A frozen vision-language executor is used to answer cross-view spatial-reasoning questions, and frozen Qwen3.5-9B, Qwen3-4B/8B models are used to generate trajectory reflections, candidate skill revisions, and manual updates. Qwen3.8-Max and DeepSeek-V4-Flash are used for offline cognitive-category labeling, while DeepSeek-V4-Pro adjudicates disagreements. These models do not receive weight updates during the runtime evolution process. We also used an LLM assistant to improve manuscript readability and organize supplementary material. We did not use generative AI to generate synthetic data, formulate the theoretical claims, provide critical ingredients for the proofs, design the methodology or experiments, interpret the experimental results, or draw the conclusions. All AI-assisted outputs were reviewed by the authors. The authors verified the annotations, experimental numbers, mathematical statements, proof assumptions, citations, figure descriptions, and final manuscript text, and take responsibility for the complete content of this submission.

ETHICS STATEMENT

We have read and will adhere to the ICLR Code of Ethics. This work uses the publicly available CityCube dataset, which consists of anonymized urban imagery. We do not collect human-subject data, infer personal identities, or intentionally analyze personally identifiable information. The derived annotations concern spatial categories and reasoning traces rather than individual persons. Improved spatial reasoning in urban scenes could nevertheless be misused for surveillance-oriented inference or other intrusive applications. We therefore emphasize that the proposed manual is intended for auditable research use, and that its deployment should account for privacy, security, and the purpose of the downstream application. Any released annotations and artifacts will be distributed in accordance with the CityCube license and the terms governing proprietary model APIs.

REPRODUCIBILITY STATEMENT

The main paper describes the problem formulation, SpatialSkill framework, admission gate, stability controls, routing policy, and experimental findings. Appendix B specifies the data splits, information-isolation protocol, and paired frozen-executor evaluation. Appendix C documents the dataset preparation, cognitive taxonomy, and trajectory annotation process. Appendix D provides the model-serving configuration, evolution hyperparameters, gate procedure, prompt interfaces, answer extraction, and evaluation metrics. Appendix E defines the ablation variants and reports supplementary experimental analyses. Appendix F gives the assumptions and complete proofs for the conditional routing implications and the stability invariant. Appendix G gives code, evolved skill manuals, routing policies, and evaluation artifacts and limitations.

REFERENCES

Hunar Batra, Haoqin Tu, Hardy Chen, Yuanze Lin, Cihang Xie, and Ronald Clark. Spatialthinker: Reinforcing 3d reasoning in multimodal llms via spatial rewards. *arXiv*, pp. 1–29, 2025.

Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 9490–9498, 2025.

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.

Haolin Du, Jingfei He, and Yuanqing Zhao. Ccr: A counterfactual causal reasoning-based method for cross-view geo-localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(11):11630–11643, 2024.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Qihan Ren, Yiran Wu, Hongru Wang, Han Xiao, Yuhang Zhou, Shaokun

Zhang, Jiayi Zhang, Jinyu Xiang, Yixiong Fang, Qiwen Zhao, Dongrui Liu, Cheng Qian, Zhenhailong Wang, Minda Hu, Huazheng Wang, Qingyun Wu, Heng Ji, and Mengdi Wang. A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *arXiv*, pp. 1–77, 2025.

Haoxuan Han, Zeyu Zhang, Yefei He, Bohan Zhuang, et al. Less detail, better answers: Degradation-driven prompting for vqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2903–2912, 2026.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 36, pp. 20482–20494, 2023.

Yining Hong, Jiageng Liu, Han Yin, Manling Li, Leonidas J. Guibas, Li Fei-Fei, Jiajun Wu, and Yejin Choi. Esi-bench: Towards embodied spatial intelligence that closes the perception-action loop. *arXiv*, pp. 1–38, 2026.

Xinmiao Huang, Qisong He, Zhenglin Huang, Boxuan Wang, Zhuoyun Li, Guangliang Cheng, Yi Dong, and Xiaowei Huang. Spatial-dise: A unified benchmark for evaluating spatial reasoning in vision-language models. In *Proceedings of the International Conference on Learning Representations*, volume 2026, pp. 135833–135865, 2026.

Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, XinQiang Yu, Jiawei He, He Wang, and Li Yi. Omnispatial: Towards comprehensive spatial reasoning benchmark for vision language models. In *International Conference on Learning Representations*, volume 2026, pp. 35634–35670, 2026.

Ruofei Ju, Xinrui Wang, Xin Ding, Yifan Yang, Hao Wu, Shiqi Jiang, Qianxi Zhang, Hao Wen, Xiangyu Li, Weijun Wang, Kun Li, Yunxin Liu, Haipeng Dai, Wei Wang, and Ting Cao. Embodiskill: Skill-aware reflection for self-evolving embodied agents. *arXiv*, pp. 1–15, 2026.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 25961–25970, 2025.

Wajahat Khalid, Bin Liu, Xulin Li, Muhammad Waqas, and Muhammad Sher Afgan. Bridging the sky and ground: Towards view-invariant feature learning for aerial-ground person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9749–9758, 2025.

Phillip Y Lee, Jihyeon Je, Chanho Park, Mikaela Angelina Uy, Leonidas Guibas, and Minhyuk Sung. Perspective-aware reasoning in vision-language models via mental imagery simulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9241–9251, 2025.

Dingming Li, Hongxing Li, Zixuan Wang, Yuchen Yan, Hang Zhang, Siqi Chen, Guiyang Hou, Shengpei Jiang, Wenqiao Zhang, Yongliang Shen, Weiming Lu, and Yueting Zhuang. Viewspatial-bench: Evaluating multi-perspective spatial localization in vision-language models. *arXiv*, pp. 1–17, 2025.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. In *Proceedings of the International Conference on Learning Representations*, volume 2026, pp. 132324–132345, 2026a.

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*, pp. 1–19, 2026b.

Zongzhao Li, Zongyang Ma, Mingze Li, Songyou Li, Yu Rong, Tingyang Xu, Ziqi Zhang, Deli Zhao, and Wenbing Huang. Star-r1: Multi-view spatial transformation reasoning by reinforcing multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12041–12051, 2026c.

Junming Liu, Yuqi Li, Yifei Sun, Maonan Wang, Piotr Koniusz, Yirong Chen, and Ding Wang. Self-evolving spatial reasoning in vision language models via geometric logic consistency. *arXiv*, pp. 1–23, 2026a.

Weichen Liu, Qiyao Xue, Haoming Wang, Xiangyu Yin, Boyuan Yang, and Wei Gao. Spatial reasoning in multimodal large language models: A survey of tasks, benchmarks and methods. *arXiv*, pp. 1–34, 2025.

Yuhong Liu, Beichen Zhang, Yuhang Zang, Yuhang Cao, Long Xing, Xiaoyi Dong, Haodong Duan, Dahua Lin, and Jiaqi Wang. Spatial-ssrl: Enhancing spatial understanding via self-supervised reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9570–9581, 2026b.

Wufei Ma, Luoxin Ye, Celso M de Melo, Alan Yuille, and Jieneng Chen. Spatialllm: A compound 3d-informed design towards spatially-intelligent large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17249–17260, 2025.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul McVay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent-Pierre Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Alexander Sax, and Aravind Rajeswaran. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16488–16498, 2024.

Kun Ouyang, Yuanxin Liu, Haoning Wu, Yi Liu, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. Spacer: Reinforcing mllms in video spatial reasoning. *arXiv*, pp. 1–16, 2025.

Kunyu Peng, Zhikun Zhou, Kailun Yang, Di Wen, Ruiping Liu, Yufan Chen, Junwei Zheng, Hao Shi, Yi Zhou, M. Saquib Sarfraz, Danda Pani Paudel, and Luc Van Gool. Seeing together: Multi-robot cooperative egocentric spatial reasoning with multimodal large language models. *arXiv*, pp. 1–20, 2026.

Viraj Prabhu, Yutong Dai, Matthew Fernandez, Krithika Ramakrishnan, Jing Gu, Yanqi Luo, silvio savarese, Caiming Xiong, Junnan Li, Zeyuan Chen, and Ran Xu. Walt: Web agents that learn tools. In *Proceedings of the International Conference on Learning Representations*, volume 2026, pp. 55589–55607, 2026.

Yuming Qiao, Liang Luo, Dan Meng, Yifan Yang, Qingyuan Wang, Juntuo Wang, Yuwei Zhang, Ru Zhen, Yanhao Zhang, Haonan Lu, et al. Aligning cross-view visual geometries in lvlms through human-like reasoning learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19345–19353, 2026.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv*, pp. 1–20, 2025.

Shuaike Shen, Wenduo Cheng, Mingqian Ma, Alistair Turcan, Martin Jinye Zhang, and Jian Ma. Skillfoundry: Building self-evolving agent skill libraries from heterogeneous scientific resources. *arXiv*, pp. 1–19, 2026.

Ilias Stogiannidis, Steven McDonagh, and Sotirios A. Tsaftaris. Mind the gap: Benchmarking spatial reasoning in vision-language models. *arXiv*, pp. 1–14, 2025.

Haotong Sun, Jianye Xie, Bocheng Xu, and Yinghui Jiang. Know thyself, know thy user: Intrinsic dual-perspective reasoning for role-playing llms. In *Proceedings of the International Conference on Machine Learning*, pp. 1–33, 2026.

Gemma Team. Gemma 4 technical report. *arXiv*, pp. 1–17, 2026a.

InternVL Team. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv*, pp. 1–27, 2025a.

Qwen Team. Qwen3-vl technical report. *arXiv*, pp. 1–42, 2025b.

Qwen Team. Qwen3.5: Towards native multimodal agents, 2026b. URL https://qwen.ai/blog?id=qwen3.5.

Shi-Yu Tian, Zhuo-Xia Wang, Xuan-Yi Zhu, Zhi Zhou, Xinwei Yang, Kun-Yang Yu, Ming Yang, Yang Chen, and Yu-Feng Li. Self-evolving neuro-symbolic skills for tool-augmented spatial reasoning. *arXiv*, pp. 1–25, 2026.

Jingqi Tong, Yurong Mou, Hangcheng Li, Mingzhe Li, Yongzhuo Yang, Ming Zhang, Qiguang Chen, Tianyi Liang, Xiaomeng Hu, Yining Zheng, et al. Thinking with video: Video generation as a promising multimodal reasoning paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41121–41129, 2026.

Hengyi Wang, Ruiqiang Zhang, Chang Liu, Guanjie Wang, Zehua Ma, Han Fang, and Weiming Zhang. Allocentric perceiver: Disentangling allocentric reasoning from egocentric visual priors via frame instantiation. *arXiv*, pp. 1–22, 2026.

Xiaokun Wang, Peiyu Wang, Jiangbo Pei, Wei Shen, Yi Peng, Yunzhuo Hao, Weijie Qiu, Ai Jian, Tianyidan Xie, Xuchen Song, Yang Liu, and Yahui Zhou. Skywork-vl reward: An effective reward model for multimodal understanding and reasoning. *arXiv*, pp. 1–13, 2025a.

Zhihu Wang, Shiwan Zhao, Yu Wang, Heyuan Huang, Sitao Xie, Yubo Zhang, Jiaxin Shi, Zhixing Wang, Hongyan Li, and Junchi Yan. Re-task: Revisiting llm tasks from capability, skill, and knowledge perspectives. In *Proceedings of the Findings of the Association for Computational Linguistics*, pp. 4925–4936, 2025b.

Junfei Wu, Jian Guan, Kaituo Feng, Qiang Liu, Shu Wu, Liang Wang, Wei Wu, and Tieniu Tan. Reinforcing spatial reasoning in vision-language models with interwoven thinking and visual drawing. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 38, pp. 143297–143330, 2025.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. In *Proceedings of the International Conference on Learning Representations Workshop on Memory for LLM-Based Agentic Systems*, pp. 1–20, 2026.

Kun Xiang, Terry Jingchen Zhang, Yinya Huang, Jixi He, Zirong Liu, Yueling Tang, Ruizhe Zhou, Lijing Luo, Youpeng Wen, Xiuwei Chen, Bingqian Lin, Jianhua Han, Hang Xu, Hanhui Li, Bin Dong, and Xiaodan Liang. Aligning perception, reasoning, modeling and interaction: A survey on physical ai. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(10):12938–12957, 2026.

Haotian Xu, Yue Hu, Zhengqiu Zhu, Chen Gao, Ziyou Wang, Junreng Rao, Wenhao Lu, Weishi Li, Quanjun Yin, and Yong Li. Citycube: Benchmarking cross-view spatial reasoning on vision-language models in urban environments. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 9190–9215, 2026a.

Weiye Xu, Jiahao Wang, Weiyun Wang, Zhe Chen, Wengang Zhou, Aijun Yang, Lewei Lu, Houqiang Li, Xiaohua Wang, Xizhou Zhu, et al. Visulogic: A benchmark for evaluating visual reasoning in multi-modal large language models. In *Proceedings of the International Conference on Learning Representations*, volume 2026, pp. 25966–26003, 2026b.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for LLM agents. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 38, pp. 17577–17604, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10632–10643, 2025a.

Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, Dahua Lin, Tai Wang, and Jiangmiao Pang. Mmsi-bench: A benchmark for multi-image spatial intelligence. In *Proceedings of the International Conference on Learning Representations*, volume 2026, pp. 157051–157088, 2026.

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, et al. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2376–2385, 2025b.

Pratham Yashwante and Rose Yu. Time series, vision, and language: Exploring the limits of alignment in contrastive representation spaces. *arXiv*, pp. 1–32, 2026.

Xiaoyu Zhan, Wenxuan Huang, Hao Sun, Xinyu Fu, Changfeng Ma, Shaosheng Cao, Bohan Jia, Shaohui Lin, Zhenfei Yin, Lei Bai, Wanli Ouyang, Yuanqi Li, Jie Guo, and Yanwen Guo. Actial: Activate spatial reasoning ability of multimodal large language models. In *Proceedings of the Advances in Neural Information Processing Systems*, 2025.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In *Proceedings of the International Conference on Learning Representations*, volume 2026, pp. 86069–86100, 2026a.

Wenyu Zhang, Wei En Ng, Lixin Ma, Yuwen Wang, Junqi Zhao, Allison Koenecke, Boyang Li, and Lu Wang. Sphere: Unveiling spatial blind spots in vision-language models through hierarchical evaluation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 11591–11609, 2025.

Yuyou Zhang, Radu Corcodel, Chiori Hori, Anoop Cherian, and Ding Zhao. Spinbench: Perspective and rotation as a lens on spatial reasoning in vlms. In *Proceedings of the International Conference on Learning Representations*, volume 2026, pp. 70072–70141, 2026b.

Ruosen Zhao, Zhikang Zhang, Jialei Xu, Jiahao Chang, Dong Chen, Lingyun Li, Weijian Sun, and Zizhuang Wei. Spacemind: Camera-guided modality fusion for spatial reasoning in vision-language models. *arXiv*, pp. 1–12, 2025.

Boyuan Zheng, Michael Y. Fatemi, Xiaolong Jin, Zora Zhiruo Wang, Apurva Gandhi, Yueqi Song, Yu Gu, Jayanth Srinivasa, Gaowen Liu, Graham Neubig, and Yu Su. Skillweaver: Web agents can self-improve by discovering and honing skills. *arXiv*, pp. 1–40, 2025.

Heng Zhou, Li Kang, Yiran Qin, Xiufeng Song, Ao Yu, Zilu Zhang, Haoming Song, Kaixin Xu, Yuchen Fan, Dongzhan Zhou, Xiaohong Liu, Ruimao Zhang, Philip Torr, Lei Bai, and Zhenfei Yin. Ego to world: Collaborative spatial reasoning in embodied systems via reinforcement learning. *arXiv*, pp. 1–28, 2026.

Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Yaxin Peng, Chaomin Shen, Feifei Feng, et al. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 5377–5395, 2025.

A Appendix Overview and Organization

This appendix provides the experimental, implementation, theoretical, and qualitative details needed to interpret and reproduce SpatialSkill. It is organized as follows:

- **Experimental Protocol and Reproducibility (Section B).** We specify the data partitions, paired frozen-executor protocol, information isolation, and known-category routing setting.
- **Dataset Preparation, Annotation, and Spatial Taxonomy (Section C).** We describe the curation pipeline, CvSI (Cross-view Spatial Intelligence) categories, trajectory primitives, and annotation quality control.
- **Models, Implementation, Prompts, and Metrics (Section D).** We document the evaluated models, serving configuration, operational admission gate, evolution hyperparameters, prompt interfaces, answer extraction, and evaluation metrics.
- **Additional Experiments and Ablations (Section E).** We define the ablation variants and report supplementary stability, routing, transfer, paired-error, and cross reflection models analyses.
- **Conditional Analysis of Routing and Stable Evolution (Section F).** We provide the Wilson-interval derivation, conditional routing implications, and the tolerance-band invariant induced by the probe-score ratchet.
- **Qualitative Case Studies, Limitations, and Broader Impact (Section G).** We analyze corrected and residual failure cases and state the limitations, LLM-use disclosure, social implications, and data-access conditions of the study.

The appendix follows the terminology and claims of the main paper.

B Experimental Protocol and Reproducibility

B.1 Evaluation Setting and Data Splits

We evaluate on the CityCube (Xu et al., 2026a) dataset using its official 4,494/528 train/validation split without modifying any instance. The training split is partitioned into an evolution stream and a probe set, while the validation split is reserved for final reporting. The resulting partitions are:

Table 4: Data partitions used by SpatialSkill. The probe set is disjoint from the evolution stream, and the validation split is accessed only for final reporting.

Split	Instances	Use
Evolution stream	4,294	answering, reflection, and revision
Probe set P	200	manual acceptance and routing selection
Validation set	528	final evaluation only

The evolution stream is processed in consecutive windows. The probe set is sampled from the training split by proportional stratification over the five CvSI categories and is never used to generate trajectories, reflections, or candidate revisions. The validation set is not accessed during evolution or route construction.

B.2 Paired Frozen-Executor Protocol

SpatialSkill is evaluated as a prompt-level intervention on frozen executors. Each treated executor is compared with its own no-skill counterpart using the same examples, answer choices, decoding configuration, and answer-extraction procedure. The treated executors are Qwen3.5-4B, Qwen3.5-9B, Gemma-3-4B, and GPT-5.6-luna. The executor parameters are never updated, and the reflection model is also frozen. The final artifact consists of the evolved natural-language manual together with its category-conditioned delivery policy.

B.3 Information Isolation

The three data roles are separated throughout the pipeline. The evolution stream supplies trajectories for reflection and revision. The probe set supplies the evidence used by the acceptance ratchet and the offline routing rule. The validation split is used only after the manual and routing policy have been fixed. Consequently, validation correctness labels do not influence skill induction, manual acceptance, or route construction.

B.4 Known-Category Routing Protocol

Each CityCube record carries a pre-annotated CvSI category. At inference time, the router receives this category and applies the corresponding strategy selected offline by Eq. (11). This is a known-category protocol: it evaluates category-conditioned delivery rather than the separate task of predicting a cognitive category from images and text. We report accuracy within each category and ordinary micro accuracy over all 528 validation examples; the overall score is therefore not an unweighted average of the five category scores.

C Dataset Preparation, Annotation, and Spatial Taxonomy

C.1 Curation Pipeline

Our annotations are derived from CityCube through a three-layer pipeline. Each record contains reference reasoning trajectories that are decomposed into atomic steps for attribution during reflection; these annotations are not supplied as an additional inference-time input to the executor.

L1: Structural parsing. A deterministic, model-free parser extracts the question, option set, answer, reference reasoning trajectories, image list, and view metadata from the raw multi-image VQA records. The parser accepts two to six answer options and uses the corresponding option letters A–F. An exhaustive consistency audit found no option-segmentation errors in the processed records, and every extracted answer letter appears in its associated option set.

L2: Cognitive labeling. Each instance is assigned one CvSI cognitive category and a spatial-relation expression. Qwen3.8-Max and DeepSeek-V4-Flash independently assign the category label. Agreement receives the shared label directly; disagreement is sent to DeepSeek-V4-Pro for adjudication. Each label records whether it came from agreement or adjudication. Two computer-science PhD students independently audit the labeled set for category consistency, spatial-relation correctness, and provenance integrity. Instances flagged by either auditor are returned to the adjudication channel until both auditors confirm the corrected label.

L3: Trajectory decomposition. The reference reasoning trajectories are decomposed into an ordered sequence of atomic steps. Each step receives one operation primitive. These trajectories are used only as an attribution reference during offline reflection and do not alter the evaluation labels or the executor weights.

C.2 CvSI Cognitive Categories

Table 5 summarizes the five cognitive categories used for stratification and routing. The categories describe the dominant reasoning demand of an instance; they are not predicted by the executor at inference time.

The validation set further stratifies by viewpoint dynamics: rotation (211), high–low (105), scale–translation (122), and hover (90). It also stratifies by spatial-relation expression: camera–object (350), camera–camera (124), and object–object (54). Each instance contains between two and ten images. In the training stream, answer letters occur as A: 1,618, B: 1,196, C: 973, D: 567, and E: 140; the corresponding option-count distribution is 2: 83, 4: 2,871, 5: 1,527, and 6: 13.

Table 5: CvSI cognitive categories and their validation-set sizes.

Category	Definition	Val n
CR (Comprehensive Reasoning)	Multi-step spatial reasoning and hypothesis verification, such as hypothesizing a navigation relation and checking it against the target scene.	37
MR (Mental Reconstruction)	Cross-view spatial transformation and mental simulation of movement, such as inferring motion direction across views.	82
PT (Perspective Taking)	Adopting the viewpoint of an object or camera, such as determining what is visible from a specified location and orientation.	158
SR (Spatial Relation)	Estimating distances, directions, and topological relations between objects or between a camera and an object.	123
WK (World Knowledge)	Urban commonsense, object geometry, function, visibility, cross-view counting, and scene description.	128

C.3 Annotation Quality and Provenance

The annotation pipeline records the source and adjudication status of each category label. The audit is performed after automatic labeling and before the train/evolution/probe/validation protocol is applied. The resulting labels define the known categories used by the routing analysis; they are not treated as a learned category-prediction result. The reference trajectories and their primitive tags are similarly used for attribution only. No validation answer is used to construct a skill, accept a revision, or select a routing strategy.

D Models, Implementation, Prompts, and Metrics

D.1 Models and Serving

The treated executors are Qwen3.5-4B, Qwen3.5-9B, Gemma-3-4B, and GPT-5.6-luna. The open-source reference models in the main table are Qwen3-VL-4B/8B/32B (Team, 2025b), Qwen3.5-2B/27B/35B-A3B (Team, 2026b), InternVL3-8B/14B (Team, 2025a), Gemma-4-31B/26B-A4B, Gemma-3-12B (Team, 2026a), and Skywork-VL-Reward-7B (Wang et al., 2025a). We also evaluate the open-source spatial models Spatial-SSRL-7B (Liu et al., 2026b), SpaceR-SFT-7B (Ouyang et al., 2025), Spatial-MLLM-subset-sft (Li et al., 2026b), and ViLaSR (Wu et al., 2025). The cross-model transfer analysis additionally uses Qwen3.5-2B as a target executor. The proprietary reference baselines are GPT-5.4, GPT-5.6-sol, Gemini-3.5-Flash, Gemini-3.7-Flash, Claude-Opus-5, and Claude-Sonnet-5; GPT-5.6-luna is also evaluated as a treated executor.

The executor and reflection model are frozen throughout. The reflection model is Qwen3.5-9B and is served locally for candidate generation, admission checks, and manual revision. Open-source executors are served with vLLM on a machine with four NVIDIA RTX 5090 GPUs. Smaller models use independent single-GPU replicas; 12B–14B models use tensor parallelism of two; and models of 26B or larger use a four-GPU tensor-parallel server. GPU memory utilization is set to 0.95, each request accepts up to ten images, and each server processes one concurrent multimodal sequence. Local runs use greedy decoding, while API baselines use their fixed native serving configuration and the same answer-extraction rule.

D.2 Operational Admission Gate

The admission gate operationalizes Eq. (2). It is a structured screening procedure performed by the frozen reflection model, not an oracle verifier. The gate receives the candidate rule, the current manual, the executor trajectory that motivated the candidate, and the relevant images.

The procedure has four stages. First, the reflection output is normalized into a candidate operation with a target section, rule content, attribution type, and evidence fields. Skill-level candidates may insert, rewrite, or correct a reusable rule; execution-level candidates may add or revise an execution

note but may not alter the reusable skill body. Second, the visual-evidence check evaluates each assertion in the candidate. An assertion passes only when it refers to an observable object, attribute, or relation in the provided views. Answer-specific facts, dataset-specific landmark names, and unsupported explanations receive $\mathcal{V}(r) = 0$. Third, the executability check evaluates whether the rule asks the frozen executor to perform a directly inspectable operation, such as matching landmarks, tracking a visual anchor, comparing relative sizes, or checking co-visibility. Rules that require an unsupported internal transformation, coordinate arithmetic, symbolic left/right algebra, or other operations beyond the demonstrated prompt-level capability receive $\mathcal{E}(r) = 0$. Finally, a candidate enters the provisional manual only when both predicates are positive:

$$\mathcal{G}(r) = \mathcal{V}(r) \wedge \mathcal{E}(r). \tag{12}$$

Rejected candidates are retained in the evolution log together with the failed predicate and its supporting explanation. The gate therefore removes visually unsupported or operationally unusable candidates before manual update, while the probe-score ratchet provides the external acceptance signal for the resulting provisional manual.

D.3 Evolution Configuration

The evolution stream is processed in windows of 100 instances. At the end of each window, the reflection model proposes at most one candidate per trajectory. Admitted candidates are applied to form a provisional manual, after which the growth budget and seniority protection constrain the update before the probe-score ratchet decides whether the provisional manual is delivered. The complete configuration is:

Table 6: Implementation configuration used by SpatialSkill.

Component	Parameter	Value	Description
Evolution	window size B	100 instances	reflection interval
	candidates / trajectory	≤ 1	candidate-generation limit
	revision token budget	8,192	prevents truncation
Stability	net-growth budget β	3 rules/window	limits growth
	protection window τ	3 windows	protects senior rules
	probe size $\lvert P \rvert$	200	held-out training examples
	tolerance ε	1.5 points	acceptance band
	baseline refresh	3 rejections	re-measures current baseline
	compression threshold/target	36/28 rules	bounded manual size
Routing	strategy space	$\{0,1\}^2$	reasoning mode $\times$ injection
	switching margin δ	0.015	probability scale
Inference	maximum generation	3,072 tokens	prevents truncation
	frequency penalty	0	preserves directional vocabulary
	maximum images	10	dataset maximum
	context length	32,768 tokens	multimodal context

The manual is initialized empty. The four fixed manual sections are Planning, Cross-View Alignment, Spatial Reasoning, and Failure Avoidance, followed by execution notes. Each routing strategy is represented as $(i, j) \in \{0, 1\}^2$, where $i = 0$ denotes direct answering, $i = 1$ denotes chain-of-thought prompting, $j = 1$ injects the final manual, and $j = 0$ withholds it. The final accepted manual is the last delivered manual after the evolution stream has been processed; the routing policy is then selected offline and paired with that manual.

D.4 Prompt Interfaces and Answer Extraction

The prompt interfaces are illustrated in Figs. 6–11. The executor prompt delivers the selected manual guidance and requests a parseable answer letter. The success-reflection prompt extracts a reusable image-agnostic strategy from a correct trajectory, while the failure-reflection prompt distinguishes a missing or misleading skill from a failure to follow an existing skill. The revision prompts update

the four manual sections or execution notes under the attribution and targeting constraints described above. The parser first searches for the last explicit '`ANSWER: <option>`' line, then falls back to the final non-empty line when it contains a valid option letter. Outputs without a valid option are counted as incorrect, and no manual correction is applied.

Executor core answer prompt

You are answering a cross-view urban spatial-reasoning multiple-choice question. Multiple images are given as different views of the same scene.\n\n[Cross-View Urban Spatial-Reasoning Skill Manual] The following are reusable, image-agnostic reasoning strategies distilled from past experience. Apply the relevant ones; they are general procedures, never answers:\n{skill_text}\n\nQuestion: {question}\n\nChoices:\n {letter}. {choice_text}\n ...\n\nThink step by step briefly, then end with a final line EXACTLY like:\nANSWER: <letter>\n\nAnswer FIRST with one line EXACTLY like:\nANSWER: <letter>\nThen add 1-2 short sentences naming the key visual evidence you used.\n\nRespond with EXACTLY one line:\nANSWER: <letter>

Skill manual delivery prompt

- [{section_title}] {rule}\n- [Execution Note for {section_title}: {target_skill}] {note}\n- [Execution Note] {note}

Figure 6: Executor delivery prompt. The prompt provides the selected manual guidance, the cross-view question, and the answer format used by the extraction procedure. Direct: respond with exactly one line ANSWER letter. CoT: think step by step briefly, then end with a final line exactly ANSWER letter.

D.5 Evaluation Metrics

For a validation set Q of 528 questions, let y_i denote the ground-truth option and $\hat{y}_i$ the extracted prediction. Overall accuracy is:

$$\mathrm{Acc} = \frac{1}{|Q|} \sum_{i \in Q} \mathbb{I}(\hat{y}_i = y_i). \tag{13}$$

Per-category accuracy is computed by restricting the same expression to the corresponding CvSI category. All reported scores are deterministic under the fixed decoding and parsing protocol; no across-run variance estimate is used in the main comparisons.

Reasoning efficiency is the mean number of generated tokens per question, counted with the executor's tokenizer:

$$\text{Reasoning Efficiency} = \frac{1}{|Q|} \sum_{i \in Q} T_i, \tag{14}$$

where T_i includes every generated token, including reasoning traces when they are present. Lower values indicate fewer generated tokens.

For option o, the option-conditioned accuracy is:

$$\mathrm{Acc}_{\mathrm{opt}}(o) = \frac{\sum_{i \in Q} \mathbb{I}(\hat{y}_i = o \wedge y_i = o)}{\sum_{i \in Q} \mathbb{I}(\hat{y}_i = o)}, \tag{15}$$

with value zero when the denominator is zero. The option-wise accuracy range is:

$$\text{Option-wise Acc. Range} = \max_o \mathrm{Acc}_{\mathrm{opt}}(o) - \min_o \mathrm{Acc}_{\mathrm{opt}}(o). \tag{16}$$

We use this range as a descriptive diagnostic of option-dependent behavior, not as evidence of complete causal debiasing.

E Additional Experiments and Ablations

E.1 Component Ablation Variants

All ablation settings use the same frozen executor, decoding parameters, validation split, and answer-extraction procedure as the complete system. Evolution-based variants restart from the empty manual and produce a terminal artifact that is fixed before validation.

The Direct baseline answers without a manual or chain-of-thought prompting. The CoT baseline uses chain-of-thought prompting without a manual. SpatialSkill uses the final manual with the category-conditioned policy.

The component variants remove one named mechanism at a time:

- *w/o attribution-aware reflection* removes the separation between reusable skill-level feedback and execution-level notes;
- *w/o visual-evidence criterion* removes $\mathcal{V}$ while retaining $\mathcal{E}$;
- *w/o executability criterion* removes $\mathcal{E}$ while retaining $\mathcal{V}$;
- *w/o joint admission gate* removes both predicates from $\mathcal{G}(r)$;
- *w/o growth budget* removes the per-window net-growth constraint β;
- *w/o seniority protection* removes the protection threshold τ; and
- *w/o probe-score ratchet* removes the probe-based acceptance rule and is referred to as *Naive* in the stability analysis.

E.2 Stability Diagnostics

We measure stability over the 43 evolution windows generated from the same frozen executor and evolution stream. A refactor-event window is a window in which the manual is explicitly refactored or its size decreases relative to the preceding delivered manual. The Naive condition follows the same reflection and revision loop but removes the probe-score ratchet. This comparison isolates the role of acceptance control while retaining the other evolution components.

The stability figures in the main paper report refactor-event windows and manual-size trajectories. We also count revisions rejected by the ratchet and compare their probe scores with the accepted reference. These diagnostics test whether the safeguards suppress harmful revisions rather than merely suppressing all manual growth.

E.3 Routing Diagnostics

The routing analysis compares the deployed probe-selected policy with a probe replay condition and a validation-only oracle. The oracle is used only as an analysis reference and is not available to the deployed system. The probe replay applies a fixed route estimated from the probe to frozen validation predictions; it is therefore a diagnostic of route sensitivity rather than a second end-to-end training run. The resampled-probe analysis further shows how route choices vary under small changes in the probe, motivating the default-and-margin rule used by the deployed router.

E.4 Cross-Model Transfer with Source-Routed Delivery

We evaluate whether a manual and its category-level delivery policy can transfer to a different executor without target-specific adaptation. For each pair, the target executor, validation split, decoding configuration, and metric remain fixed; the source manual and source-derived routing policy are applied directly to the target. Each pair uses a pre-specified terminal source artifact, and no transfer round is selected using held-out validation performance.

All eight completed cross-model transfers improve over their target bases, with gains ranging from 0.57 to 6.82 percentage points and a mean gain of 2.72 points. Gemma-source gives the largest gain on Qwen3.5-4B, improving from 47.16% to 53.98%, while Qwen-source is strongest on Qwen3.5-2B, improving from 46.59% to 50.19%. GPT-5.6-luna reaches 55.87% with Qwen-source and

56.82% with Gemma-source. Gemini-3.7-Flash reaches 54.73% and 54.36% with the two source manuals, respectively. The results indicate that source-routed skills can transfer across executor families, but neither source manual is uniformly optimal.

E.5 Skill Manual Transfer Experiment

Our external transfer set originates from MMSI-Bench (Yang et al., 2026) as the second dataset. After data cleaning, we use the 1,000 instances for transfer evaluation. We evaluate the frozen executor-specific manual and routing policy without adapting the model or manual.

Table 7: Own-base and frozen manual-routed MMSI-Bench accuracy and option-wise accuracy range. Each executor's frozen manual and source-derived routing policy is applied to the 1,000-example transfer set .

Method	Overall Acc. (%) ↑	Option-wise Acc. Range (pp) ↓
Qwen3.5-4B Base	29.30	10.45
Qwen3.5-4B Manual + route	**31.30**	**8.01**
GPT-5.6-luna Base	34.30	10.62
GPT-5.6-luna Manual + route	**38.40**	**3.95**

This experiment tests transfer across datasets and scene domains rather than transfer between executors. We retain overall accuracy and the option-wise accuracy range used in Table 7.

Both frozen manual policy pairs improve over their bases. Qwen3.5-4B improves by 2.00 points, from 29.30% to 31.30%, while its option-wise accuracy range decreases from 10.45 to 8.01 points. GPT-5.6-luna improves by 4.10 points, from 34.30% to 38.40%, while its option-wise accuracy range decreases from 10.62 to 3.95 points. These single-run external results support the transfer of the frozen delivery artifacts. They further suggest that the learned manual has some transferability to similar spatial-reasoning tasks.

E.6 Paired Error Analysis

For Qwen3.5-4B and Gemma-3-4B, base and SpatialSkill predictions are paired on the same validation examples. We classify each example as repaired, broken, fixed, or persistent according to the transition from the base prediction to the SpatialSkill prediction. Qwen3.5-4B repairs 87 baseline errors and introduces 61 new errors, for a net gain of 26. Gemma-3-4B repairs 70 baseline errors and introduces 56 new errors, for a net gain of 14. The remaining baseline errors persist in the majority of cases, showing that the manual produces targeted repairs rather than uniform correction.

We also compare generated-token ratios across transition types. The largest SpatialSkill-to-base expansion occurs on newly broken examples, with ratios of 4.74 for Qwen3.5-4B and 8.33 for Gemma-3-4B. Output length is therefore a descriptive correlate of skill-conditioned reasoning, not evidence of correctness or a causal mechanism.

E.7 Generality Across Reflection Models

The main experiments use Qwen3.5-9B as the fixed reflection model for candidate skill generation, admission checking, and manual revision. To examine whether SpatialSkill depends on this particular reflector, we keep the executor fixed as Qwen3.5-4B and repeat the complete skill-evolution pipeline with two alternative reflection models: Qwen3-4B and Qwen3-8B. All other settings, including the evolution stream, probe split, window size, admission criteria, stability controls, routing procedure, and executor decoding configuration, are kept unchanged. Thus, the reflection model is the only experimental variable.

Table 8 reports the results. With the default Qwen3.5-9B reflector, SpatialSkill improves the frozen Qwen3.5-4B executor from 47.16% to 52.08%. Replacing the reflector with Qwen3-4B and Qwen3-8B yields accuracies of 50.57% and 49.05%, corresponding to gains of +3.41 and +1.89 percentage points over the same frozen executor baseline. The consistent gains across these reflectors suggest that SpatialSkill is not tied to the default reflection model, although performance remains sensitive to reflector choice.

Table 8: Reflection-model sensitivity with Qwen3.5-4B as the frozen executor. Only the reflection model is changed; all other evolution and inference settings are fixed.

Reflection Model	Executor	Accuracy (%) ↑	Δ (pp) ↑
None	Qwen3.5-4B	47.16	–
Qwen3-4B	Qwen3.5-4B	50.57	+3.41
Qwen3-8B	Qwen3.5-4B	49.05	+1.89
Qwen3.5-9B	Qwen3.5-4B	52.08	+4.92

F Conditional Analysis of Routing and Stable Evolution

This section formalizes the two safeguards deferred from the main text. The routing results are conditional on simultaneous interval coverage; the stability result is deterministic with respect to the reused probe set. Neither result is a distribution-shift guarantee.

F.1 Notation and Assumptions

Let $\mathcal{C}$ denote the set of CvSI categories and $\mathcal{S}$ the strategy space. For category $c \in \mathcal{C}$ and strategy $s \in \mathcal{S}$, let $p_{c,s}$ be the population accuracy, n_c the number of probe examples, $k_{c,s}$ the number of correct answers, and $\hat{a}_{c,s} = k_{c,s}/n_c$ the empirical accuracy. The default strategy $s_0(c)$ is the stronger of the two non-injected strategies on the probe set. The deployed strategy is $\pi(c)$.

For each fixed c and s, we assume that correctness indicators across probe examples are i.i.d. Bernoulli variables with success probability $p_{c,s}$. For each strategy, let $[\ell_s(c), u_s(c)]$ be the nominal Wilson interval at confidence level $1-\alpha$. The router forms:

$$\mathcal{F}(c) = \left\{ s \in \mathcal{S} : \ell_s(c) - u_{s_0(c)}(c) \geq \delta \right\}, \tag{17}$$

and selects the empirical best member of $\mathcal{F}(c)$ when the set is non-empty; otherwise it retains $s_0(c)$. Define the simultaneous coverage event:

$$\mathcal{E}_c = \{ p_{c,s} \in [\ell_s(c), u_s(c)] \text{ for all } s \in \mathcal{S} \}. \tag{18}$$

The deployed implementation uses nominal, unadjusted 95% Wilson intervals with $z = 1.96$. Because strategies and the default are selected from the same probe observations, marginal interval coverage alone does not establish simultaneous finite-sample coverage for the data-dependent router. The propositions below are therefore logical implications conditional on $\mathcal{E}_c$, rather than unconditional 95% performance guarantees.

F.2 Wilson Score Interval

For a fixed category and strategy, the score statistic for testing $H_0 : p = p_0$ is:

$$Z(p_0) = \frac{\hat{a}_{c,s} - p_0}{\sqrt{p_0(1-p_0)/n_c}}.$$

Inverting the two-sided score test with critical value $z = \Phi^{-1}(1-\alpha/2)$ yields:

$$[\ell_s(c), u_s(c)] = \frac{\hat{a}_{c,s} + \dfrac{z^2}{2n_c} \mp z\sqrt{\dfrac{\hat{a}_{c,s}(1-\hat{a}_{c,s})}{n_c} + \dfrac{z^2}{4n_c^2}}}{1 + \dfrac{z^2}{n_c}}. \tag{19}$$

The lower endpoint uses the minus branch and the upper endpoint uses the plus branch. For a fixed strategy, this construction has the usual nominal asymptotic marginal coverage. It does not,

without multiplicity correction or an independent routing probe, provide simultaneous coverage for all data-dependent comparisons used by the router.

F.3 Conditional Routing Implications

Proposition 1 (Conditional switch implication)**.** *Fix a category c. On $\mathcal{E}_c$, every route selected by a switch improves over the default by at least the margin:*

$$\pi(c) \neq s_0(c) \quad \Longrightarrow \quad p_{c,\pi(c)} \geq p_{c,s_0(c)} + \delta. \tag{20}$$

Proof. If $\pi(c) \neq s_0(c)$, then the margin test gives

$$\ell_{\pi(c)}(c) \geq u_{s_0(c)}(c) + \delta.$$

On $\mathcal{E}_c$, both selected intervals cover their population accuracies. Therefore,

$$p_{c,\pi(c)} \geq \ell_{\pi(c)}(c) \geq u_{s_0(c)}(c) + \delta \geq p_{c,s_0(c)} + \delta.$$

□

Proposition 2 (Conditional no-deterioration)**.** *Fix a category c. On $\mathcal{E}_c$, the deployed route is no worse than the default:*

$$p_{c,\pi(c)} \geq p_{c,s_0(c)}. \tag{21}$$

Proof. If no switch occurs, then $\pi(c) = s_0(c)$ and equality holds. If a switch occurs, Proposition 1 gives the stronger inequality $p_{c,\pi(c)} \geq p_{c,s_0(c)} + \delta$. Thus Eq. (21) holds in both cases. □

Remark 1. The propositions are conditional implications of the margin rule. They should not be read as unconditional repeated-sampling guarantees for the deployed router, because the reported implementation uses unadjusted intervals and selects strategies from the same probe observations used to construct those intervals.

F.4 Simultaneous Conditional Implication

Corollary 1 (Simultaneous conditional no-deterioration)**.** *On $\bigcap_{c\in\mathcal{C}} \mathcal{E}_c$, no deployed route is worse than its category default:*

$$\forall c \in \mathcal{C}: \qquad p_{c,\pi(c)} \geq p_{c,s_0(c)}.$$

Proof. The intersection event implies $\mathcal{E}_c$ for every category. Applying Proposition 2 category by category yields the result. □

F.5 Tolerance-Bounded Evolution

Proposition 3 (Tolerance-bounded evolution under ratchet acceptance)**.** *Let m_t be the delivered manual after window t, let m'_t be the provisional candidate manual, and let*

$$b_t = \max_{0\leq j\leq t} \mathrm{Score}(m_j)$$

be the best probe score attained by a delivered manual through window t. If the acceptance rule is

$$m_{t+1} = \begin{cases} m'_t & \text{if } \mathrm{Score}(m'_t) \geq b_t - \varepsilon, \\ m_t & \text{otherwise,} \end{cases}$$

then, for every $t \geq 0$,

$$\mathrm{Score}(m_t) \geq b_t - \varepsilon.$$

When $\varepsilon = 0$, the delivered probe score is monotonically non-decreasing.

Proof. We proceed by induction. At an accepted step, the candidate satisfies $\mathrm{Score}(m'_t) \geq b_t - \varepsilon$. If it becomes the new best, then $b_{t+1} = \mathrm{Score}(m'_t)$ and the claim holds with equality relative to the new best. Otherwise $b_{t+1} = b_t$, and the acceptance condition gives the required bound. At a rejected step, both the delivered manual and the best score remain unchanged. Therefore the bound holds for every window. □

Remark 2. With $\varepsilon = 1.5$ percentage points, the ratchet permits a revision to fall within the prescribed tolerance band but does not deliver a larger probe-score degradation. The growth budget and seniority protection constrain the candidate before this acceptance test; they are stability controls, not statistical guarantees.

G Qualitative Case Studies, Limitations, and Broader Impact

G.1 Case Study Protocol

We present three corrected cases and three residual failure cases. Each case records the spatial question, the relevant executor response, the correct answer, and the visual reasoning issue exposed by the transition. Corrected cases illustrate targeted repairs supplied by the manual and its route. Failure cases document remaining limitations and are not interpreted as evidence that every spatial error can be repaired by a natural-language skill.

G.2 Corrected Cases

Case C1: Cross-view regional alignment. The question asks where a green sign, observed beside the PIER 23 entrance in a street view, lies in the aerial view. The baseline selects the north-west region because it maps the sign to the wrong side of the diagonally oriented building. The SpatialSkill response aligns the street-facing side with the aerial building footprint and uses the water-facing side as a visual anchor, yielding the south-west region. The correction illustrates how landmark matching and explicit cross-view anchoring repair a perspective-dependent spatial relation.

Case C2: Viewpoint-dependent visibility. The question asks whether a person standing in a plaza can directly see a signboard on the building. The baseline selects "No, blocked." The routed chain-of-thought response aligns the plaza view with the building facade and checks whether the sign lies on the facade facing the open plaza, concluding that it is visible straight ahead. The skill contributes a field-of-view check rather than an assumption that depth or viewpoint change implies occlusion.

Case C3: Immediate spatial adjacency. The question asks which object is immediately left of a light-blue building. The baseline selects a foreground gazebo that is visually nearby but separated by open space. The SpatialSkill response distinguishes the foreground distractor from the tall residential building that is immediately adjacent in the relevant view. This case illustrates a failure-avoidance rule that checks structural proximity and cross-view identity rather than relying on the first salient object.

G.3 Residual Failure Cases

Case F1: Cross-view position mapping. The crosswalk is visible in a street view and must be located in the aerial view. The model selects the south-west corner instead of the north-east corner. The error indicates a residual failure to preserve the crosswalk's relation to the intersection when changing viewpoint, even though the target marking is visually distinctive.

Case F2: Camera motion versus object relation. The question asks how the relation between a bus-stop shelter and a multi-story building changes across a scale sequence. The model interprets forward camera motion as evidence that the two scene objects become closer, although their mutual spatial relation remains unchanged. This exposes a remaining distinction between image-plane change caused by camera motion and a true change in object-to-object geometry.

Case F3: Viewpoint orientation. The question asks whether the scene in one view is visible from a billboard location. The model correctly identifies that the scene is visible but incorrectly describes it as being to the side rather than straight ahead. The failure reflects incomplete viewpoint-orientation reasoning: the model detects visibility but does not correctly recover the relative viewing direction.

G.4 Manual Cases

Figure 19 presents the completed evolution manual paired with the Qwen3.5-4B executor after the evolution stream and probe-based acceptance procedure have terminated. The final artifact contains the four fixed sections Planning, Cross-View Alignment, Spatial Reasoning, and Failure Avoidance together with execution notes. It records the reusable spatial regularities that survived visual evidence screening, executability checks, bounded revision, and the probe score ratchet, and is delivered to the frozen executor through the category conditioned routing policy.

The manual provides an auditable and reusable natural-language interface for organizing spatial reasoning in a frozen executor. The corrected cases above illustrate how its guidance promotes landmark alignment, viewpoint-aware interpretation, and explicit checks against common spatial confounds. Together with the quantitative improvements, the completed manual demonstrates that evolution can consolidate recurring observations into a compact set of transferable execution strategies for cross-view spatial reasoning.

G.5 Limitations and Broader Impact

Limitations. Our study is scoped to cross-view urban spatial reasoning. Transfer to indoor embodied scenes, medical images, or other perceptual domains remains untested. The framework does not update executor weights and therefore cannot overcome the executor's perceptual ceiling. Skill induction relies on offline annotated trajectories and a training-split probe, so "training-free" refers specifically to the absence of executor fine-tuning or weight updates. The probe-driven ratchet controls the repeatedly measured probe trajectory but does not establish protection under distribution shift. Finally, the router receives a pre-annotated cognitive category at test time, and the quality of the natural-language manual remains bounded by the reflection model's ability to verbalize reusable spatial regularities.

LLM use and disclosure. Large language models participate in the research pipeline as frozen executors, as the frozen Qwen3.5-9B reflection model, and as offline assistants for cognitive-category labeling and adjudication. An LLM assistant was also used for manuscript language polishing. The runtime evolution loop does not update model weights; it operates through the frozen executor, the frozen reflection model, the annotated evolution stream, and the probe-based acceptance procedure.

Social impact. An auditable natural-language manual can reduce the computational cost of adapting a frozen vision-language model and makes accumulated guidance inspectable and retractable. The same spatial capabilities could also be misused for surveillance-oriented inference. CityCube contains anonymized urban imagery and no personally identifiable information to the best of our knowledge, but deployment decisions should still account for the context and purpose of spatial inference.

Data access and licensing. The experiments use the publicly available CityCube dataset under its research-use terms. Any derivative annotations and released artifacts will be distributed only in accordance with the dataset license and the terms governing the proprietary model APIs. Code is available on https://anonymous.4open.science/r/SpatialSkill-2DC8. The evolved manuals, routing policies, and evaluation artifacts are intended for release upon publication, subject to those conditions.

Success reflection prompt

You are an expert at maintaining a high-level skill manual for a cross-view spatial-reasoning answerer.

You maintain a high-level skill manual that helps a vision-language model answer cross-view urban spatial-reasoning multiple-choice questions. Each question gives several images that are different views (different camera positions/ orientations, hover, rotation, scale/translation, or high vs low altitude) of the same urban scene, and asks about spatial relations, directions, distances, camera motion, perspective transforms, counting, or world knowledge. The manual is a set of reusable, image-agnostic reasoning strategies. It must never contain answer letters, dataset-specific landmark names, or per-question facts.

Rule-quality HARD constraints (the answering model is a SMALL vision-language model that answers DIRECTLY, without chain-of-thought; rules act as visual priors it glances at, never as multi-step procedures for it to execute):
- Every rule MUST describe a concrete, directly eye-verifiable visual check, e.g. "compare the shadow direction of the same building across views", "track a distinctively colored/shaped rooftop to re-anchor after a viewpoint change", "check whether object sizes grew or shrank to tell zoom-in vs zoom-out".
- STRICTLY FORBIDDEN: abstract mental transforms of any kind - forward vectors, coordinate frames/axes, rotating the frame, 90/180/270-degree CW/CCW reasoning, bearing or azimuth arithmetic, symbolic left/right algebra. An automatic filter deletes any rule containing such terms, so writing them wastes the update.
- Prefer rules about WHAT TO LOOK AT (anchors, shadows, road markings, building faces, relative sizes, co-visible objects) over rules about HOW TO COMPUTE.

You will be given: the current skill manual, one question that was answered CORRECTLY, and the model's answer with the brief visual evidence it reported.

Your job is to decide whether this case contains a meaningful, broadly reusable reflection about the manual. It is acceptable and expected that many cases produce no reflection.

Rules:
- The reflection must be high level and generally applicable across many questions.
- If the case suggests multiple possible reflections, choose only the single most critical and broadly useful one.
- Do not mention specific option letters, specific landmark names, image ids, or per-question facts.
- Use concise, precise language. Keep `content` brief and information-dense.
- Put analysis, diagnosis, and justification only in `thinking`.
- `content` must contain only the exact reusable strategy text to add or rewrite. Do not include narrative or phrases like "the model answered correctly because" in `content`.
- `target_skill` must be copied verbatim from the current input manual when you are referring to an existing rule. Never invent, paraphrase, or normalize a rule text. If no existing rule is being discussed, output an empty string.
- Only produce a non-empty reflection when the case reveals a broadly reusable manual update.
- If the case is redundant, too question-specific, already covered by the manual, or does not reveal a meaningful update, return an empty reflection. Empty reflections will be discarded.
- `reflection_type` must be exactly one of:
  - S_NEW: the case reveals a new useful strategy missing from the manual. For `S_NEW`, do not point to an existing rule. `target_skill` must be an empty string, and `target_section` is required: it must name the section where the new rule should be added.
  - S_BETTER: the manual already contains a related rule, but the case reveals a better high-level strategy. For `S_BETTER`, both `target_skill` and `target_section` are required. `target_skill` must exactly match one existing manual rule from the input, and `target_section` must be the section that contains that exact rule. If either target is missing or cannot be matched exactly, use `S_NEW` instead of `S_BETTER` and leave both target fields empty.
- Section titles must be exactly one of: `Planning`, `Cross-View Alignment`, `Spatial Reasoning`, `Failure Avoidance`.

Output requirements:
- Return valid JSON only.
- If there is no meaningful reflection, return only:

```
{
  "is_empty": true,
  "reason": "short explanation"
}
```

- Otherwise use this schema exactly:

```
{
  "is_empty": false,
  "reflection_type": "S_NEW | S_BETTER",
  "target_section": "short section name or empty string",
  "target_skill": "the existing skill rule being discussed, or empty string",
  "thinking": "brief analysis explaining why this reflection type and target are appropriate",
  "content": "only the exact reusable strategy text to add or rewrite"
}
```

Figure 7: Success-reflection prompt. The reflection model extracts one concise, reusable visual strategy from a correct trajectory when the case provides a non-redundant lesson.

Failure reflection prompt

You are an expert at maintaining a high-level skill manual for a cross-view spatial-reasoning answerer.

You maintain a high-level skill manual that helps a vision-language model answer cross-view urban spatial-reasoning multiple-choice questions. Each question gives several images that are different views (different camera positions/orientations, hover, rotation, scale/translation, or high vs low altitude) of the same urban scene, and asks about spatial relations, directions, distances, camera motion, perspective transforms, counting, or world knowledge. The manual is a set of reusable, image-agnostic reasoning strategies. It must never contain answer letters, dataset-specific landmark names, or per-question facts.

Rule-quality HARD constraints (the answering model is a SMALL vision-language model that answers DIRECTLY, without chain-of-thought; rules act as visual priors it glances at, never as multi-step procedures for it to execute):
- Every rule MUST describe a concrete, directly eye-verifiable visual check, e.g. "compare the shadow direction of the same building across views", "track a distinctively colored/shaped rooftop to re-anchor after a viewpoint change", "check whether object sizes grew or shrank to tell zoom-in vs zoom-out".
- STRICTLY FORBIDDEN: abstract mental transforms of any kind - forward vectors, coordinate frames/axes, rotating the frame, 90/180/270-degree CW/CCW reasoning, bearing or azimuth arithmetic, symbolic left/right algebra. An automatic filter deletes any rule containing such terms, so writing them wastes the update.
- Prefer rules about WHAT TO LOOK AT (anchors, shadows, road markings, building faces, relative sizes, co-visible objects) over rules about HOW TO COMPUTE.

You will be given: the current skill manual, one question that was answered INCORRECTLY (the correct option is provided), and the model's answer with the brief visual evidence it reported. Your job is to decide whether this case contains a meaningful, broadly reusable reflection about the manual. It is acceptable and expected that many cases produce no reflection.

Rules:
- The reflection must be high level and generally applicable across many questions.
- If the case suggests multiple possible reflections, choose only the single most critical and broadly useful one.
- Do not mention specific option letters, specific landmark names, image ids, or per-question facts.
- Use concise, precise language. Keep `content` brief and information-dense.
- Put analysis, diagnosis, and justification only in `thinking`.
- `content` must contain only the exact reusable strategy text or execution reminder to add or rewrite. Do not include causal narrative or phrases like "the model failed because" in `content`.
- `target_skill` must be copied verbatim from the current input manual when you are referring to an existing rule. Never invent, paraphrase, or normalize a rule text. If no existing rule is being discussed, output an empty string.
- Only produce a non-empty reflection when the case reveals a broadly reusable manual update or execution reminder.
- If the case is redundant, too question-specific, already covered by the manual, or does not reveal a meaningful update, return an empty reflection. Empty reflections will be discarded.
- `reflection_type` must be exactly one of:
  - FAIL_SKILL: the manual itself is missing, misleading, or incomplete. When correcting an existing bad or incomplete rule, `target_skill` is required and must exactly match one existing manual rule from the input, and `target_section` must be the section that contains that rule. If the failure instead reveals a completely missing rule, `target_skill` must be empty and `target_section` is required.
  - FAIL_EXECUTION: the manual rule is acceptable, but the model failed to actually follow a specific existing manual rule (e.g. skipped cross-view matching it was told to do). For `FAIL_EXECUTION`, both `target_skill` and `target_section` are required. `target_skill` must exactly match the existing manual rule that was not followed, and `target_section` must be the section containing it. If either target is missing or cannot be matched exactly, use `FAIL_SKILL` instead and leave both target fields empty.
- FAIL_EXECUTION reflections must not rewrite the manual body. They produce an execution reminder instead.
- Section titles must be exactly one of: `Planning`, `Cross-View Alignment`, `Spatial Reasoning`, `Failure Avoidance`.

Output requirements:
- Return valid JSON only.
- If there is no meaningful reflection, return only:

```
{
  "is_empty": true,
  "reason": "short explanation"
}
```

- Otherwise use this schema exactly:

```
{
  "is_empty": false,
  "reflection_type": "FAIL_SKILL | FAIL_EXECUTION",
  "target_section": "short section name or empty string",
  "target_skill": "the existing skill rule being discussed, or empty string",
  "thinking": "brief analysis explaining whether the failure came from manual quality or execution",
  "content": "only the exact reusable strategy text or execution reminder to add or rewrite"
}
```

Figure 8: Failure-reflection prompt. The reflection model distinguishes a missing or misleading skill from an execution failure and emits the corresponding structured update.

Mannual section revision prompt

You maintain the main sections of a versioned high-level skill manual for a cross-view spatial-reasoning answerer.

You maintain a high-level skill manual that helps a vision-language model answer cross-view urban spatial-reasoning multiple-choice questions. Each question gives several images that are different views (different camera positions/orientations, hover, rotation, scale/translation, or high vs low altitude) of the same urban scene, and asks about spatial relations, directions, distances, camera motion, perspective transforms, counting, or world knowledge. The manual is a set of reusable, image-agnostic reasoning strategies. It must never contain answer letters, dataset-specific landmark names, or per-question facts.

Rule-quality HARD constraints (the answering model is a SMALL vision-language model that answers DIRECTLY, without chain-of-thought; rules act as visual priors it glances at, never as multi-step procedures for it to execute):
- Every rule MUST describe a concrete, directly eye-verifiable visual check, e.g. "compare the shadow direction of the same building across views", "track a distinctively colored/shaped rooftop to re-anchor after a viewpoint change", "check whether object sizes grew or shrank to tell zoom-in vs zoom-out".
- STRICTLY FORBIDDEN: abstract mental transforms of any kind - forward vectors, coordinate frames/axes, rotating the frame, 90/180/270-degree CW/CCW reasoning, bearing or azimuth arithmetic, symbolic left/right algebra. An automatic filter deletes any rule containing such terms, so writing them wastes the update.
- Prefer rules about WHAT TO LOOK AT (anchors, shadows, road markings, building faces, relative sizes, co-visible objects) over rules about HOW TO COMPUTE.

You will be given: the current manual sections, a batch of reflections of type `S_NEW`, `S_BETTER`, or `FAIL_SKILL`.

Your job is to revise the manual sections only.

Revision rules:
- Preserve concise, high-level, broadly applicable guidance.
- Use S_NEW reflections to add missing general strategies.
- Use S_BETTER reflections to improve existing rules.
- Use FAIL_SKILL reflections to correct incomplete or misleading rules, or to add a missing rule revealed by a failure.
- Treat each reflection as a scoped edit request. The target item is the anchor for the edit.
- Preserve untargeted existing manual items by default, but you may modify, merge, or delete an untargeted item when it directly conflicts with or is substantially redundant with a targeted edit or an S_NEW addition.
- For S_NEW reflections, add new content only to the specified `target_section`; do not modify any existing item.
- For S_BETTER reflections, only modify or delete the exact existing item named by `target_skill` in `target_section`. If the target pair is missing or does not exactly match the manual, ignore that reflection.
- For FAIL_SKILL reflections with a valid `target_skill`, modify or delete the exact existing item named by `target_skill` in `target_section`.
- For FAIL_SKILL reflections with an empty `target_skill`, add the missing rule to the specified `target_section` when the content is broadly reusable. If `target_section` is missing or invalid, ignore that reflection.
- Group candidate additions in the same section, deduplicate and conflict-check them, and merge compatible ones into one compact rule. Discard conflicting candidates that cannot be reconciled.
- Do not add multiple new rules that differ only by wording. Prefer one compact rule capturing the shared lesson.
- STRICT ADDITION BUDGET: net growth of the manual in one revision must not exceed 3 items. Rank candidate additions by how many reflections support them and add only the top ones; discard the rest. Rewriting an existing rule in place does not count against the budget. Excess additions are programmatically dropped, so exceeding the budget wastes the update.
- Use concise wording. Prefer short, information-dense rules over long explanations.
- Do not produce execution notes here.
- Section titles must be exactly these four and appear only once each: `Planning`, `Cross-View Alignment`, `Spatial Reasoning`, `Failure Avoidance`. Do not invent, rename, or duplicate sections.

Output requirements:
- Return valid JSON only.
- Use this schema exactly:

```
{
 "sections": [
   {
    "title": "Planning" | "Cross-View Alignment" | "Spatial Reasoning" | "Failure Avoidance",
    "items": [
      "one concise high-level rule for this section",
      "another concise high-level rule for this section"
    ]}]}
```

Figure 9: Manual-section revision prompt. Candidate reflections are consolidated into the four fixed manual sections while untargeted guidance is preserved.

Execution notes revision prompt

You maintain the execution-notes appendix of a versioned skill manual for a cross-view spatial-reasoning answerer.

You maintain a high-level skill manual that helps a vision-language model answer cross-view urban spatial-reasoning multiple-choice questions. Each question gives several images that are different views (different camera positions/orientations, hover, rotation, scale/translation, or high vs low altitude) of the same urban scene, and asks about spatial relations, directions, distances, camera motion, perspective transforms, counting, or world knowledge. The manual is a set of reusable, image-agnostic reasoning strategies. It must never contain answer letters, dataset-specific landmark names, or per-question facts.

You will be given:
- the manual sections after this round's section revision,
- the current execution notes before this round's execution-note revision,
- a batch of reflections of type `FAIL_EXECUTION`.

Your job is to revise the execution notes only.

Revision rules:
- Convert the reflections into concise execution reminders (e.g. "actually perform cross-view matching before deciding").
- Use the revised manual sections as context so notes do not contradict the main manual.
- Bind every note to the current revised manual sections: output `target_skill` must exactly match an item that exists in the revised sections, and `target_section` must be the section containing it.
- If a reflection's original target still exists, keep it; if it was rewritten/merged, retarget to the current item; if it was deleted with no clear replacement, discard that reflection.
- Maintain at most one execution note per exact (`target_section`, `target_skill`) pair; modify the existing note instead of adding a duplicate.
- Group notes by `target_section`, merge or rewrite overlapping/conflicting notes, and deduplicate.
- Preserve unrelated existing notes. Keep notes short, actionable, non-redundant.
- Do not rewrite the main manual sections here.

Output requirements:
- Return valid JSON only.
- Use this schema exactly:

```
{
 "execution_notes": [
   {
    "target_section": "section containing the emphasized manual item",
    "target_skill": "exact existing manual item being emphasized",
    "note": "concise execution reminder"
   }
 ]
}
```

Figure 10: Execution-note revision prompt. Execution failures are converted into concise reminders bound to existing manual rules without rewriting the manual body.

Mannual refactor prompt

You maintain the main sections of a versioned high-level skill manual for a cross-view spatial-reasoning answerer.

You maintain a high-level skill manual that helps a vision-language model answer cross-view urban spatial-reasoning multiple-choice questions. Each question gives several images that are different views (different camera positions/orientations, hover, rotation, scale/translation, or high vs low altitude) of the same urban scene, and asks about spatial relations, directions, distances, camera motion, perspective transforms, counting, or world knowledge. The manual is a set of reusable, image-agnostic reasoning strategies. It must never contain answer letters, dataset-specific landmark names, or per-question facts.

Rule-quality HARD constraints (the answering model is a SMALL vision-language model that answers DIRECTLY, without chain-of-thought; rules act as visual priors it glances at, never as multi-step procedures for it to execute):
- Every rule MUST describe a concrete, directly eye-verifiable visual check, e.g. "compare the shadow direction of the same building across views", "track a distinctively colored/shaped rooftop to re-anchor after a viewpoint change", "check whether object sizes grew or shrank to tell zoom-in vs zoom-out".
- STRICTLY FORBIDDEN: abstract mental transforms of any kind - forward vectors, coordinate frames/axes, rotating the frame, 90/180/270-degree CW/CCW reasoning, bearing or azimuth arithmetic, symbolic left/right algebra. An automatic filter deletes any rule containing such terms, so writing them wastes the update.
- Prefer rules about WHAT TO LOOK AT (anchors, shadows, road markings, building faces, relative sizes, co-visible objects) over rules about HOW TO COMPUTE.

You will be given the current manual sections only. Refactor them because they have become too long.

Refactor rules:
- HARD LIMIT: After refactoring, the TOTAL number of items across ALL 4 sections MUST be at most 28. Merge aggressively.
- PROTECTED RULES: the user prompt lists protected rules that survived many evolution windows and are empirically validated. Copy every protected rule VERBATIM into its original section. NEVER modify, merge, reword, or delete a protected rule; compress only the unprotected ones. Any missing protected rule is programmatically restored, so touching them wastes the compression.
- Keep only high-value, broadly reusable guidance.
- Merge similar rules into one concise rule. Prefer fewer, more general rules over many specific ones.
- Eliminate conflicting rules; if two conflict and cannot be reconciled, discard both.
- Remove duplicate or near-duplicate rules and low-value wording.
- Simplify phrasing. Prefer short, information-dense rules (max 30 words each).
- Do not produce execution notes here.
- Section titles must be exactly these four and appear only once each: `Planning`, `Cross-View Alignment`, `Spatial Reasoning`, `Failure Avoidance`.

Output requirements:
- Return valid JSON only.
- Use this schema exactly:

```
{
 "sections": [
   {
    "title": "Planning" | "Cross-View Alignment" | "Spatial Reasoning" | "Failure Avoidance",
    "items": ["one concise high-level rule", "another concise high-level rule"]
   }
 ]
}
```

Figure 11: Manual-refactor prompt. Oversized sections are compressed by merging redundancy and removing conflicts while protected rules are preserved.

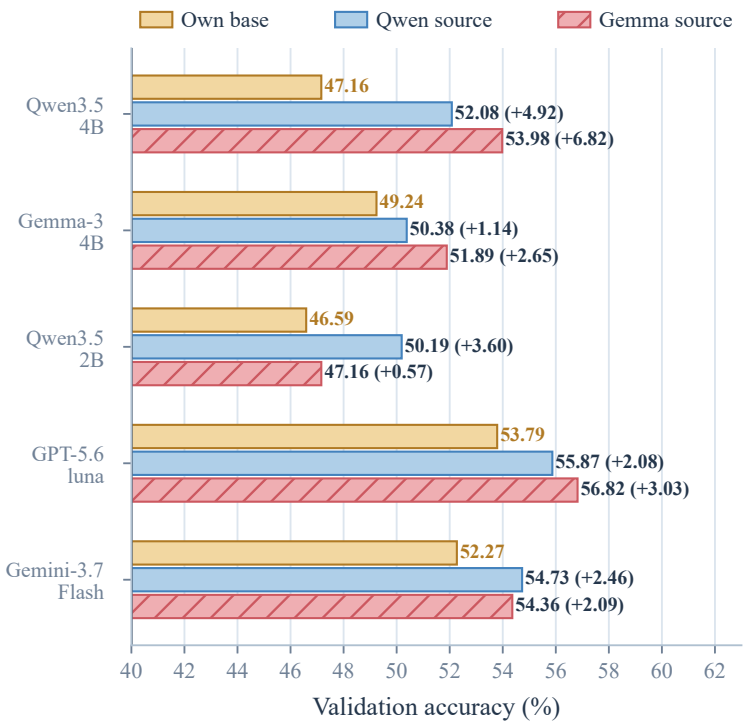


Figure 12: Own-base and cross-model source-routed validation accuracy. Each transferred manual and its source-derived routing policy is applied to the target executor without target-specific adaptation; N/A denotes an incomplete pairing.

Case 1

Input images




Question and answer

Question: In street view, a green sign hangs from a lamppost to the left of the "PIER 23" building entrance. In aerial view, in which region is this green sign located?

Choices: A. North-east region B. South-east region C. North-west Region D. South-west region E. Cannot determine precisely.

Answer: D. South-west region

Cognitive category: SR

Before and after answers

Before: The sign is to the left of the entrance in the street view. After locating the diagonally oriented PIER 23 building in the aerial image, the response maps this position to the top-left of the building footprint and concludes that it is in the north-west region. **Answer: C**

After: The response first aligns the street-facing side of PIER 23 with the pier in the aerial view. It then uses the water-facing side and the diagonal building axis to map the sign-bearing lamppost to the bottom-left of the aerial image,i.e., the south-west region. **Answer: D**

Route selection: skill_direct

Manual content used:

1. Failure Avoidance: When verifying if a landmark appears in multiple views, explicitly match the target object's specific structural details(e.g., window patterns, roof features) and its precise relative position to other co-visible landmarks.

2. Failure Avoidance: When determining camera movement direction, strictly derive the answer from visual evidence such as the lateral shift of shared landmarks or changes in background context.

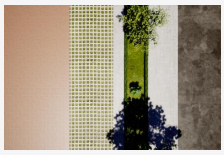
Figure 13: Corrected SR case: cross-view regional alignment of a street-view sign and an aerial building footprint. The baseline selects the north-west region, whereas the routed SpatialSkill response uses the building orientation and shared landmarks to select the south-west region.

Case 2

Input images







Question and answer

Question: If a person is on the plaza area depicted in view2 (Building Scale), looking towards the building, can they directly see the `名庄荟` signboard visible on the building in view3 (Detail Scale)?

Choices: A. Yes, straight ahead B. Yes, to the side C. No, blocked D. Uncertain.

Answer: A. Yes, straight ahead

Cognitive category: PT

Before and after answers

Before: The baseline gives no supporting explanation and selects “No, blocked.” **Answer: C**

After: View 2 shows the plaza and the building complex, while view 3 resolves the `名庄荟` signboard on the ground-floor facade. Matching the plaza and the facade across views shows that the sign is on the side of the building facing the open plaza. There is an unobstructed direct line of sight. **Answer: A**

Route selection: skill_cot

Manual content used:

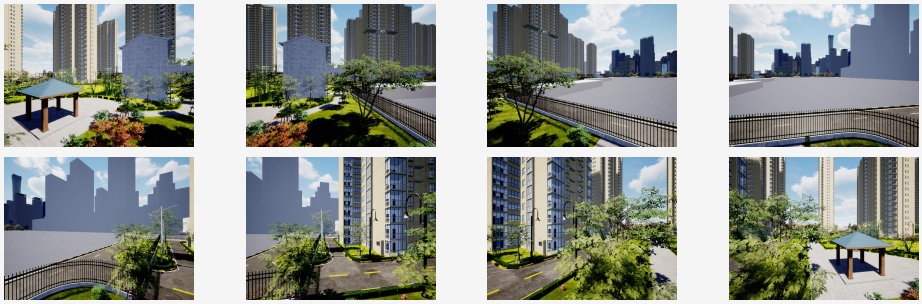
1. Cross-View Alignment: When determining if a target object is visible from a specific viewpoint with a different camera angle, first establish the relative orientation between the two views; then verify if the target object lies within the new field of view rather than assuming occlusion or blind spots based solely on depth or proximity.

2. Cross-View Alignment: When determining if two specific objects can be seen simultaneously from a single viewpoint, inspect the candidate view to verify that both objects are physically present within the frame boundaries at the same time.

Figure 14: Corrected PT case: cross-view visibility from a plaza to a building facade. The baseline predicts blockage, while the routed response verifies the facade orientation and the unobstructed line of sight.

Case 3

Input images



Question and answer

Question: What object is located to the immediate left of the light blue building observed in view_0?

Choices: A. The blue-roofed gazebo B. A tall residential building in the Background C. A black metal fence D. Not visible.

Answer: B. A tall residential building in the background

Cognitive category: SR

Before and after answers

Before: View 0 shows a light blue, boxy building on the right side of the frame. To its immediate left is a paved area with a blue-roofed gazebo. Other views confirm the gazebo's location next to the light blue building. **Answer: A**

After: In view 0, the light blue building is the large greyish-blue structure on the right. Immediately to its left in the background is a tall residential building with a beige facade and many windows. The gazebo is in the foreground and is separated by open space, so it is a distractor rather than the immediately adjacent object. **Answer: B**

Route selection: skill_direct

Manual content used:

1. Failure Avoidance: When verifying if a landmark appears in multiple views, explicitly match the target object's specific structural details (e.g., window patterns, roof features) and its precise relative position to other co-visible landmarks.

Figure 15: Corrected SR case: distinguishing immediate adjacency from foreground proximity. The baseline selects the gazebo, whereas the SpatialSkill response identifies the tall residential building as the immediate left neighbor.

Case 1

Input images



Question and answer

Question: A crosswalk marking is visible in the street view. How is its position marked in the aerial view?

Choices: A. Northeast corner of intersection B. Southwest corner of Intersection C. North side of main road D. South side of main road.

Answer: A. Northeast corner of intersection

Cognitive category: SR

Model output

The crosswalk is at the southern end of the road and corresponds to the southwest corner of the intersection.

Answer: B

Figure 16: Residual SR failure: the model maps a street-view crosswalk to the wrong corner of the aerial intersection, selecting the south-west corner instead of the correct north-east corner.

Case 2

Input images



Question and answer

Question: During the scale change from view1 (Regional Scale) to view4 (Scale 4), how does the spatial relationship between the bus stop shelter and the multi-story building on the right with red signs change?

Choices: A. Closer B. Further apart C. Remains unchanged D. Cannot determine.

Answer: C. Remains unchanged

Cognitive category: SR

Model output

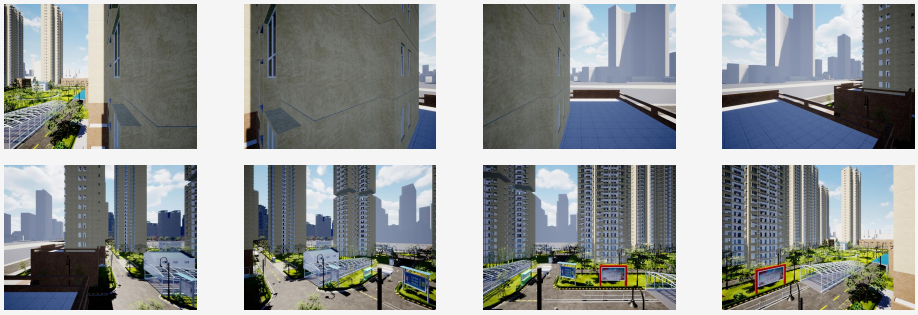
As the camera moves from view1 to view4, it is advancing down the street. This forward movement brings the bus stop shelter and the multi-story building on the right into closer proximity relative to the camera's position.

Answer: A

Figure 17: Residual SR failure: the model confuses camera advancement with a change in the object-to-object relation and predicts "closer" instead of "remains unchanged."

Case 3

Input images



Question and answer

Question: If you are a pedestrian/driver at the large red billboard, can you see the scene shown in view_0 from your perspective?

Choices: A. Yes, straight ahead B. Yes, to the side C. Yes, behind D. Cannot see.

Answer: A. Yes, straight ahead

Cognitive category: PT

Model output

The scene in view_0 is visible from the billboard, but the person would need to turn their head and look to the right.

Answer: B

Figure 18: Residual PT failure: the model predicts side visibility for a scene that is visible straight ahead from the billboard viewpoint.

Planning
1. To determine the hierarchical scale structure, explicitly check for specific visual markers corresponding to each scale level: skyline or street network for the urban region, the entire multi-story structure for the building complex, the specific facade or signage for the individual building, and surface textures or small objects for local details; then order the views from the level where the broadest markers are visible to the level where the finest markers are visible.
2. When zooming from a broad view to a specific view to locate a target, establish a sequential lock-on path by progressively narrowing the scope: first identify the broad regional context, then lock onto the specific directional location or side of the building complex containing the target (e.g., left-side, right-side, or center), and finally isolate the specific detail or signboard on that structure.
3. When counting objects across multiple views to determine a total, identify distinct spatial regions (e.g., foreground street, background complex, specific courtyard) in the wide-context views, count the objects in each region separately, and sum the counts from all regions to verify the total.

Cross-View Alignment
1. When determining if a ground-level feature is visible from a high-altitude or rooftop view, check if the view displays rooftops or building tops; if so, any ground-level feature located directly beneath the building's footprint is completely hidden and cannot be seen from above.
2. When determining if a ground-level feature (e.g., signs, facade elements, light poles, street furniture) is visible from any viewpoint, check the aerial or high-altitude view to see if the feature lies beneath the building's footprint; if so, the building's own mass completely blocks the line of sight, and the feature cannot be seen from the ground level.
3. When determining what is directly in front of the camera in a street view by referencing an aerial view, locate the bottom edge of the aerial frame; the feature directly in front in the street view corresponds to the road junction or intersection visible at that bottom boundary. Check if the aerial view shows a cross-street or junction at the bottom edge to identify the feature directly ahead.
4. When determining if a camera motion involves both vertical descent and optical zoom, verify that the primary subject grows larger in the frame while the surrounding background context simultaneously shrinks and loses detail; this simultaneous change in subject scale and field of view width indicates a combined descent and zoom pattern rather than a single isolated motion.
5. When determining if a feature on a building facade (e.g., signs, floor numbers, windows) is visible from a rooftop or high-altitude view, check if the feature is located on a surface physically below the observer's position; if the feature is on the underside of the building mass relative to the camera, it is completely hidden and cannot be seen.
6. When determining if two specific objects can be seen simultaneously from a single viewpoint, inspect the candidate view to verify that both objects are physically present within the frame boundaries at the same time; if either object is missing or only one is visible, that viewpoint does not allow simultaneous observation of both.
7. When determining if two opposite sides of the same building or object are visible simultaneously from a specific viewpoint, check if the camera is positioned at an angle between those two sides; if the view captures both faces in the same frame, the viewpoint is located laterally between them, allowing simultaneous observation of both sides.
8. When determining if a target object is visible from a specific viewpoint defined by a foreground object (e.g., the church), inspect the background elements visible in that viewpoint; if the target object is located in the direction opposite to the camera's facing direction (i.e., in the background behind the observer), the object is visible only if the observer turns around, but from the current straight-ahead perspective, it is behind and not visible in the forward field of view.
9. When determining if a vehicle seen in a front view can see a vehicle seen in a rear view, infer the driving direction and lane alignment from the front view; if the rear-view vehicle is positioned in the same lane or directly adjacent to the path of travel relative to the front-view vehicle's orientation, check for occlusion by the front-view vehicle's own structure (e.g., hood, A-pillars) or blind spots, which may block visibility even if the object is theoretically in the line of sight.
10. When determining if an object visible in a reference view is visible from a new viewpoint with a different camera angle, first establish the relative orientation between the two views; then verify if the target object lies within the new field of view defined by that orientation, rather than assuming occlusion or blind spots based solely on depth or proximity.

Spatial Reasoning
1. When predicting the predominant feature after a turn, identify the prominent landmark in the center of the reference view and locate the landmark that appears in the center of the target view.
2. When determining how spatial relationships change after moving towards a target, verify the target's position relative to the path of travel; landmarks directly on the path appear to get closer, while those off to the side or behind the camera appear to recede or move away from the center of view.
3. When determining how proximity to multiple landmarks changes after moving towards a specific target, verify the initial relative positions of all landmarks: landmarks located on the direct path of travel will appear to get closer, while landmarks located off to the side or behind the camera's initial position will appear to recede or move away from the center of view, resulting in different proximity changes for different objects.

Failure Avoidance
1. When determining camera movement direction, strictly derive the answer from visual evidence such as the lateral shift of shared landmarks or changes in background context.
2. When verifying if a landmark appears in multiple views, explicitly match the target object's specific structural details (e.g., window patterns, roof features) and its precise relative position to other co-visible landmarks.

Figure 19: Completed evolution manual paired with the Qwen3.5-4B executor. The figure shows the terminal natural-language artifact after candidate screening, bounded revision, and probe-based acceptance; the manual is delivered through the category-conditioned routing policy without updating executor weights.