BEYOND ANSWER CORRECTNESS: EXECUTABLE DERIVATION FAITHFULNESS FOR URBAN SPATIAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Urban spatial reasoning typically relies on external tools to acquire spatial evidence and on multi-step derivations to reach final decisions. However, answer correctness does not imply derivation faithfulness: models may produce correct answers despite misusing spatial evidence or making errors in intermediate derivations. Outcome-only evaluation and training cannot distinguish these trajectories from faithful ones that yield the same answer. To address this issue, we propose UrbanSDF (**Urban** **S**patial **D**erivation **F**aithfulness), a framework for derivation faithfulness in urban spatial reasoning. UrbanSDF explicitly models how entity bindings determine tool arguments, how returned observations support spatial computations, and how these computations determine final decisions. To verify these dependencies, we construct executable derivation certificates that assess derivation faithfulness separately from outcome correctness. The resulting feedback guides a progressive training strategy: after initial tool-protocol supervision, we address detected errors through targeted process supervision with faithful demonstrations, improving the model's ability to generate faithful trajectories under limited sampling. Building on this supervision, we employ Conditional-Dual GRPO optimization during online interaction, using separate outcome and process feedback to favor faithful derivations among answer-correct trajectories. Experiments on four types of spatial reasoning tasks show that UrbanSDF maintains task accuracy while substantially increasing the proportion of faithful trajectories among those yielding correct answers, with an average improvement of 45.6% over the strongest baseline. Our code and data are available at https://anonymous.4open.science/r/UrbanSDF-5A02.

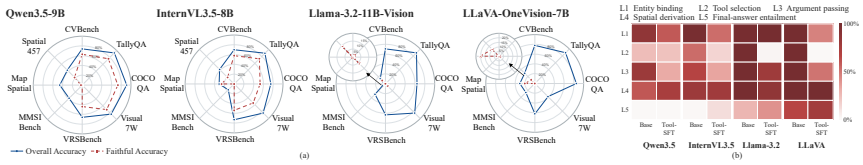


Figure 1: Are VLMs' spatial reasoning results trustworthy? (a) Answer accuracy and faithful accuracy across four VLMs and eight benchmarks. (b) Layer-wise failure rates within correct-but-unfaithful trajectories, indicating where the execution chain breaks.

1 INTRODUCTION

Urban spatial reasoning aims to model visual scenes, map structures, and spatial semantics to infer directional, topological, and other spatial relations, supporting complex spatial decisions such as route planning (Feng et al., 2025). Models typically query external GIS tools to obtain geographic facts that reasoning models and limited visual inputs alone cannot reliably supply.

However, successfully invoking tools does not imply that a model can faithfully use spatial evidence to derive an answer. In practice, moving from observations to final decisions typically involves multiple steps of complex spatial derivation, with later steps relying on observations and intermediate results from earlier steps. Breaks in these data dependencies can lead to incorrect entity-coordinate bindings, misuse of spatial reference frames, or even derivations that contradict the actual road net-

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.

work. More subtly, when the answer space is small, a model may still produce the correct answer despite errors in evidence acquisition or spatial computation. For example, it may miscalculate the length of a route yet still select the shortest one if the ranking among candidate routes remains unchanged. Figure 1(a) shows an average gap of 35.4 percentage points between answer accuracy and faithful accuracy across four VLMs on multiple spatial benchmarks (Tong et al., 2024; Acharya et al., 2019; Ren et al., 2015; Zhu et al., 2016; Li et al., 2024; Yang et al., 2025; MapSpatial, 2026; Wang et al., 2025). Further layer-wise analysis of correct-but-unfaithful trajectories reveals frequent errors in stages such as argument passing (L3) and spatial derivation (L4) (Figure 1(b)), indicating that evaluation based solely on final answers overlooks critical derivation errors and cannot fully reflect a model's true spatial reasoning ability.

To mitigate execution errors, existing research primarily adopts workflow constraints and process supervision (Wei et al., 2026): the former regulates tool calls through predefined pipelines, while the latter provides step-level supervision through process fine-tuning or reinforcement learning. However, compliant workflows or semantically plausible steps do not guarantee that spatial conclusions are faithfully derived from actual tool observations. Ultimately, these methods lack explicit checks on the full spatial data flow: mapping visual entities to query parameters, transferring tool outputs across steps, and geometric and topological derivations supporting final decisions.

This gap not only makes execution faithfulness difficult to verify, but also poses two key challenges to faithful optimization. First, the reward signal is ambiguous: rewards based solely on final answers cannot distinguish faithful from unfaithful correct trajectories, making it difficult to establish a clear preference for faithfulness among answer-correct trajectories. Second, faithful positive trajectories are scarce: complex spatial tasks require multiple data dependencies to hold simultaneously, making fully faithful trajectories difficult to generate under limited sampling and leaving insufficient positive examples to support online optimization.

Building on these considerations, we propose UrbanSDF (**Urban** **S**patial **D**erivation **F**aithfulness), a framework for faithful urban spatial reasoning. It models complex reasoning as a complete execution chain from factual perception to spatial derivation and constructs hierarchical derivation certificates to distinguish faithful from unfaithful execution among answer-correct trajectories. Certificate-identified failures guide targeted process supervision with faithful demonstrations, increasing faithful trajectory generation under limited sampling. We then apply Conditional-Dual GRPO for online optimization. Rather than merging local process scores into a global reward, we model task outcomes and local derivation credit separately, shifting policy probability from unfaithful correct trajectories toward faithful ones with complete data dependencies.

Our main contributions are summarized as follows:

- We develop a fine-grained faithfulness evaluation mechanism for urban spatial reasoning that decouples answer correctness from derivation faithfulness, providing a quantitative and verifiable basis for identifying unfaithful derivations behind correct answers.
- We propose UrbanSDF, which uses hierarchical derivation certificates to guide targeted process supervision that alleviates the scarcity of faithful trajectories under limited sampling, and conditional dual-track optimization that strengthens faithfulness preferences among answer-correct trajectories, reducing erroneous reinforcement of unfaithful derivations by outcome rewards.
- Experiments across five cities and four reasoning tasks show that UrbanSDF substantially increases faithful trajectories among correct answers while maintaining task accuracy, with an average gain of 45.6% over the strongest baseline.

2 Related Work

Urban Spatial Reasoning and Geospatial Agents. Existing studies have mainly explored the urban spatial understanding and reasoning capabilities of multimodal models. UrbanLLaVA (Feng et al., 2025) integrates urban imagery, geographic information, and trajectory data, while UrbanFeel (He et al., 2026) broadens the scope of urban spatial evaluation. However, for specific road networks, precise distances, and complex spatial relations, relying solely on model parameters and limited visual inputs makes it difficult to ensure accurate metric and topological judgments. To address this limitation, geospatial agents incorporate external tools, such as map queries and spatial computation, to obtain geographic observations. This raises a key question for urban spatial reasoning: how to ensure that these observations are faithfully incorporated into subsequent derivations.

Approaches to Faithful Urban Spatial Reasoning. Existing methods mainly improve the faithfulness of urban spatial reasoning through three directions: workflow constraints, evidence consistency, and process feedback (Wei et al., 2026). Spatial-Agent (Bao et al., 2026) regulates tool use through structured workflows, but executable workflows do not necessarily ensure that returned values are correctly used in subsequent derivations. CodeV and DeFacto (Hou et al., 2026; Xu et al., 2026) constrain the consistency between evidence and answers through tool-related rewards and counterfactual training, respectively, while C-GRPO (Zhang et al., 2026) uses process rewards to guide multi-step decision making. However, these methods primarily constrain tool-use behavior, local consistency, or process feedback, and still lack joint verification of cross-step data dependencies. In contrast, we focus on the full spatial data flow, examining how external observations are propagated and used across steps to support model decisions.

3 PRELIMINARIES

Given urban visual evidence ξ and a natural-language instruction q, the model uses an interaction policy π_θ to call the predefined GIS tool set $\mathcal{T}$, progressively acquire spatial observations, and generate an explicit derivation r and final answer y, where θ denotes the model parameters.

Tool interaction and execution trajectories. At decision step t, the model generates an action α_t conditioned on the input $x = (\xi, q)$, tool specifications, and interaction history. The action is either a tool call $a_t = (g_t, p_t)$, where $g_t \in \mathcal{T}$ and p_t are the selected tool and its arguments, or a terminal submission $\mathrm{Fin}(r, y)$. The environment $\mathcal{E}$ executes each call and returns an observation $o_t = \mathcal{E}(a_t)$, which is added to the interaction history. Let K_τ denote the number of tool calls, and $A_{1:K_\tau}$ and $O_{1:K_\tau}$ the corresponding call and observation sequences. Formally:

$$\alpha_t \sim \pi_\theta(\cdot \mid x, \mathcal{T}, h_t), \quad \tau = (A_{1:K_\tau}, O_{1:K_\tau}, r, y) \sim \mathbb{P}_{\pi_\theta, \mathcal{E}}(\cdot \mid x, \mathcal{T}), \tag{1}$$

here, h_t contains prior tool calls $A_{<t}$, observations $O_{<t}$, and retained planning text; the environment $\mathcal{E}$ provides the task-specific geographic region and GIS data.

Trajectory validity, correctness, and faithfulness. $V_x(\tau) \in \{0, 1\}$ indicates whether a trajectory is valid for answer extraction and faithfulness assessment. For valid trajectories, $C_x(\tau), H_x(\tau) \in \{0, 1\}$ denote answer correctness and derivation faithfulness, where faithfulness requires the derivation to be grounded in the input evidence and actual interaction records and to entail the final answer. We define $\mathcal{F}_x = \{\tau : C_x(\tau) = H_x(\tau) = 1\}$ as the correct-and-faithful set and $\mathcal{U}_x = \{\tau : C_x(\tau) = 1, H_x(\tau) = 0\}$ as the correct-but-unfaithful set.

Our objective is to increase the probability that the model generates correct and faithful trajectories through autonomous tool interaction over the task distribution $x \sim \mathcal{D}$:

$$\theta^\star \in \arg\max_\theta \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{P}_{\pi_\theta, \mathcal{E}} \left(\tau \in \mathcal{F}_x \mid x, \mathcal{T} \right) \right]. \tag{2}$$

4 PROPOSED METHOD

As shown in Figure 2, UrbanSDF combines five-layer executable certificates with three progressive training stages: (i) Tool Interaction Supervision, (ii) Certificate-Guided Process Supervision, and (iii) Conditional-Dual GRPO, to improve correct and faithful trajectory generation.

4.1 DERIVATION FAITHFULNESS AND EXECUTABLE CERTIFICATES

To characterize spatial data flow, we construct a five-layer execution chain and verify each valid trajectory against the task specification Ω_x. Let $\mathcal{K}_\ell(x, \tau; \Omega_x)$ denote the applicable checks at layer ℓ, with $c_{\ell,k}(\tau) \in \{0, 1, \perp\}$ indicating failure, pass, or unresolved status. Once resolved, their conjunction defines the layer verdict:

$$c_\ell(\tau) = \prod_{k \in \mathcal{K}_\ell} c_{\ell,k}(\tau), \qquad \ell = 1, \ldots, 5. \tag{3}$$

Omitted required checks are retained and fail when assessable; inapplicable checks are excluded, and empty sets score 1. See Appendix B for unresolved cases.

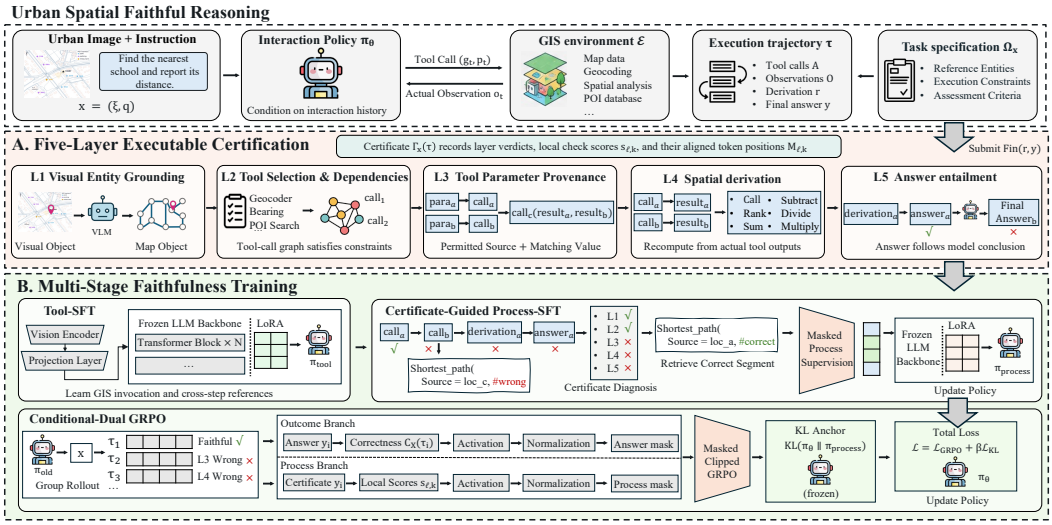


Figure 2: Overview of UrbanSDF. Five-layer executable certificates verify GIS interaction trajectories and guide targeted Process-SFT and Conditional-Dual GRPO for faithful spatial reasoning.

L1: Visual entity grounding. We check whether the model binds the visual objects specified by the instruction to the correct map entities: for role j, $\widehat{e}_{\tau,j}$ is the entity bound by trajectory τ and $e^{\star}_{x,j}$ is the task-specified reference, so $c_{1,j}(\tau) = \mathbf{1}[\widehat{e}_{\tau,j} = e^{\star}_{x,j}]$ for $j \in \mathcal{K}_1$.

L2: Tool selection and dependencies. We construct a dependency graph $\mathcal{G}_\tau$, with tool calls as nodes and result dependencies as directed edges, and check whether it satisfies the task-specified constraints $\phi_{x,k}$: $c_{2,k}(\tau) = \mathbf{1}[\mathcal{G}_\tau \models \phi_{x,k}]$, $k \in \mathcal{K}_2$.

L3: Tool parameter provenance. For each call, we further check whether parameter sources satisfy the task constraints and whether supplied values match the actual referenced content:

$$c_{3,t}(\tau) = 1 \quad \Longleftrightarrow \quad \forall j \in \mathcal{J}_t,\ \left[\sigma_{\tau,t,j} \in \mathcal{S}_{x,t,j}(\tau)\ \wedge\ p^{\mathrm{val}}_{\tau,t,j} = p^{\mathrm{src}}_{\tau,t,j}\right], \quad t \in \mathcal{K}_3, \tag{4}$$

here, $t \in \mathcal{K}_3$ denotes a call position in trajectory τ, and $\mathcal{J}_t$ its required arguments; $\sigma_{\tau,t,j}$ and $\mathcal{S}_{x,t,j}(\tau)$ are the actual and permitted sources, while $p^{\mathrm{val}}_{\tau,t,j}$ and $p^{\mathrm{src}}_{\tau,t,j}$ are the supplied value and the value read from the cited source, compared by type. Checks across trajectories are aligned by a shared semantic key k.

L4: Spatial derivation. After tool execution, the model combines tool outputs with prior derivations to form spatial conclusions. The verifier recomputes the derived results D^{rep} and spatial decisions Z^{rep} from the actual outputs and compares them with the model-reported $\widehat{D}$ and $\widehat{Z}$:

$$c_{4,k}(\tau) = 1 \quad \Longleftrightarrow \quad \Delta_{x,k}\left((\widehat{D}, \widehat{Z}), (D^{\mathrm{rep}}, Z^{\mathrm{rep}})\right) \le \delta_{x,k}, \qquad k \in \mathcal{K}_4, \tag{5}$$

here, $\Delta_{x,k}$ and $\delta_{x,k}$ denote the discrepancy measure and tolerance for check k, respectively; see Appendix B for comparison rules.

L5: Final-answer entailment. We check whether the submitted answer y agrees with $R_x(\widehat{Z})$, the answer entailed by the model's conclusion: $c_{5,1}(\tau) = \mathbf{1}[y = R_x(\widehat{Z})]$. Unlike answer correctness, this check does not compare y with the reference answer.

Certificate construction and faithfulness assessment. For checks with a definite pass/fail verdict, we record a local score $s_{\ell,k}(\tau) \in [0, 1]$ and corresponding output positions $\mathcal{M}_{\ell,k}(\tau)$. Unresolved checks are recorded as $(\perp, \perp, \varnothing)$, with no local score or token mask. These records and verdicts form the certificate $\Gamma_x(\tau)$. Once all layer verdicts are determined, overall faithfulness is the conjunction of the five layers:

$$\Gamma_x(\tau) = \left(\{c_\ell(\tau)\}_{\ell=1}^{5}, \left\{\left(c_{\ell,k}(\tau), s_{\ell,k}(\tau), \mathcal{M}_{\ell,k}(\tau)\right)\right\}_{\ell,k}\right), \qquad H_x(\tau) = \prod_{\ell=1}^{5} c_\ell(\tau). \tag{6}$$

Detailed verification and scoring rules are provided in Appendix B.

4.2 Multi-Stage Faithfulness Training

To improve correct and faithful trajectory generation during autonomous interaction, we further propose a three-stage training framework.

Tool Interaction Supervision. We first use Tool-SFT to supervise tool calls and arguments in demonstrations $\tau^\star$. Let $u_n^\star$, $h_n^\star$, and m_n denote the target token, context, and binary mask, and $\mathcal{D}_{\text{sup}}$ the training distribution. Minimizing the following loss yields π_{tool}:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(x,\tau^\star,m)\sim\mathcal{D}_{\text{sup}}}\left[\frac{\sum_n m_n \log \pi_\theta(u_n^\star \mid h_n^\star)}{\sum_n m_n}\right]. \tag{7}$$

Certificate-Guided Process Supervision. After tool training, the model may still make errors in result referencing and derivation computation, making faithful trajectories difficult to sample consistently. Process-SFT applies certificate checks to trajectories τ generated by π_{tool} and selects the corresponding correct segments for failed checks from a faithful demonstration $\tau^\star \in \mathcal{F}_x$. Let $\mathcal{P}_x(\tau, \tau^\star)$ denote the token positions of these segments, and define $m_n = \mathbf{1}[n \in \mathcal{P}_x(\tau, \tau^\star)]$; using this mask to continue optimizing $\mathcal{L}_{\text{SFT}}$ yields π_{process}.

Conditional Dual-GRPO. To handle autonomous interaction scenarios beyond those covered by supervision, we initialize the online policy from π_{process} and use Conditional Dual-GRPO, based on GRPO (Shao et al., 2024), to compare answer correctness and local process scores separately.

For each input x, the current policy π_{old} interacts with the environment $\mathcal{E}$ to sample N trajectories, $G_x = \{\tau_i\}_{i=1}^N$. Among valid trajectories $\mathcal{V} = \{i : V_x(\tau_i) = 1\}$, the Outcome branch compares answer correctness $C_x(\tau_i)$, while the Process branch aligns checks by entity role, parameter purpose, and computation target and compares local scores $s_{\ell,k}(\tau_i)$. A branch participates in the update only when it has at least two comparable samples and its within-group score standard deviation exceeds η. The corresponding Outcome and Process advantages, A_i^{o} and $A_{i,\ell,k}^{\text{p}}$, are defined as:

$$A_i^{\text{o}} = \frac{C_x(\tau_i) - \mu_C}{\sigma_C}, \qquad A_{i,\ell,k}^{\text{p}} = \frac{s_{\ell,k}(\tau_i) - \mu_{\ell,k}}{\sigma_{\ell,k}}, \qquad i \in \mathcal{V}, \tag{8}$$

here, μ and σ are the mean and population standard deviation over each branch's comparison samples. Token masks localize the advantages: A_i^{o} to final-answer tokens and $A_{i,\ell,k}^{\text{p}}$ to certificate positions $\mathcal{M}_{\ell,k}(\tau_i)$. Let $m_{i,n}^b$ denote the mask, with $N_{i,b} = \sum_n m_{i,n}^b$ covered tokens. Branches with all-positive or all-negative nonzero advantages are excluded; the rest are recentered as $\widetilde{A}_i^b$. For token $u_{i,n}$, define $\rho_{i,n}(\theta) = \pi_\theta(u_{i,n} \mid h_{i,n})/\pi_{\text{old}}(u_{i,n} \mid h_{i,n})$, where $h_{i,n}$ is its context and π_{old} is fixed within each sampling group.

To limit policy drift, we use GRPO clipping with a sampled penalty to frozen π_{process}:

$$\mathcal{L}_{\text{KL}}(\theta; G_x) = \mathbb{E}_i\left[\frac{1}{T_i}\sum_{n\in\mathcal{Q}_i}\left(e^{d_{i,n}} - d_{i,n} - 1\right)\right], d_{i,n} = \log\frac{\pi_{\text{process}}(u_{i,n} \mid h_{i,n})}{\pi_\theta(u_{i,n} \mid h_{i,n})}. \tag{9}$$

here, $\mathcal{Q}_i$ contains the retained model-generated positions across stages, $T_i = |\mathcal{Q}_i|$, and $\mathbb{E}_i$ averages over valid trajectories used for the update. We normalize each branch loss by its covered tokens, combine the branches, and add the regularizer to obtain the Dual-GRPO objective:

$$\mathcal{L}_{\text{Dual}}(\theta; G_x) = -\mathbb{E}_i\left[\sum_{b:N_{i,b}>0}\frac{\lambda_b}{N_{i,b}}\sum_n m_{i,n}^b \times \min\Big(\rho_{i,n}(\theta)\widetilde{A}_i^b,\ \text{clip}(\rho_{i,n}(\theta), 1-\epsilon, 1+\epsilon)\,\widetilde{A}_i^b\Big)\right] + \beta\,\mathcal{L}_{\text{KL}}(\theta; G_x), \tag{10}$$

here, λ_b, ϵ, and β denote the branch weight, clipping parameter, and regularization strength, respectively. The shared loss combines outcome and local process feedback as a training surrogate for the trajectory-level objective in equation 2. Details of supervision construction, check matching, token masks are provided in Appendix C.

5 Experiments

Our experiments address the following six research questions:

- **RQ1 (Overall Performance):** Can UrbanSDF improve derivation faithfulness while maintaining answer accuracy on spatial reasoning tasks compared with existing baselines?
- **RQ2 (Execution Certificate Effectiveness):** Can the proposed execution certificates identify derivation errors behind correct answers across different spatial reasoning benchmarks?
- **RQ3 (Component Analysis):** How do tool-protocol alignment, targeted process supervision, and conditional dual-track optimization individually affect overall performance?
- **RQ4 (Faithfulness Optimization Analysis):** Which reasoning bottlenecks limit the sampling of faithful positive trajectories, how does targeted process supervision alleviate these limitations, and how does conditional dual-track optimization further strengthen the preference for faithfulness among answer-correct trajectories?
- **RQ5 (Cross-Backbone Applicability):** Can UrbanSDF consistently improve spatial derivation faithfulness across different vision-language backbones?
- **RQ6 (Cross-City Generalization):** Can UrbanSDF trained on seen cities generalize to unseen cities while maintaining answer correctness and derivation faithfulness?

5.1 Experimental Settings

We evaluate the faithfulness of UrbanSDF's reasoning on shared urban spatial reasoning tasks in a common GIS tool environment. Full details of dataset construction, metric definitions, tool interfaces, and runtime configurations are provided in Appendix D.

Datasets We construct approximately 6,000 task instances using OpenStreetMap across New York, Sydney, Singapore, Berlin, and Tokyo, covering four multi-hop spatial reasoning tasks: comparing total distances of multi-segment routes, assessing route constraints, analyzing road-closure impacts, and determining goal-search directions. Each instance contains an urban map image, an instruction specifying spatial constraints, and geographic information accessible through GIS tools. Tools return spatial observations rather than answers; the model must use these observations to perform derivations and produce a discrete-option answer.

Baseline Methods *Geospatial Workflow Methods.*: Spatial-Agent (Bao et al., 2026) and GeoFlow (Bhattaram et al., 2025) use spatial-concept guidance and workflow orchestration, respectively, to regulate GIS tool calls and reduce tool misuse, thereby improving execution reliability and traceability. *Evidence-Grounded Reasoning Methods.*: CodeV (Hou et al., 2026) and DeFacto (Xu et al., 2026) use tool-related rewards and counterfactual training, respectively, to guide models to answer based on actual observations and reduce unsupported inferences. *Process-Based Reinforcement Learning Methods.*: C-GRPO (Zhang et al., 2026) and Tree-GRPO (Ji et al., 2026) use citation-aware evidence-chain rewards and tree-structured relative advantages, respectively, to provide training feedback for intermediate decisions and improve the reliability of multi-step reasoning.

Evaluation Protocol We split task instances into training, development, and test sets at an approximate ratio of 8:1:1, and select training data for Tool-SFT, Process-SFT, and GRPO at an approximate ratio of 5:2:2. The development set is used for model selection and hyperparameter tuning. To control for differences in model visual capabilities, we consider two input settings: Controlled supplies correct entity bindings to focus on downstream spatial derivation, whereas Full requires autonomous visual binding and derivation to assess end-to-end performance. All methods are scored on the same test set using a unified evaluator. We report overall answer accuracy (OA), faithful accuracy (FA), wrong-answer rate (WR), the rate of invalid or unassessable outputs (IR), and the faithful (FC) and unfaithful (CBUFR) proportions among answer-correct trajectories. We also report L1–L5 failure rates within correct-but-unfaithful trajectories to localize derivation errors.

Implementation Details We conduct our main experiments with Qwen3.5-9B (Qwen) and evaluate UrbanSDF's cross-backbone applicability on InternVL3.5-8B (InternVL), Llama-3.2-11B-Vision (Llama), and LLaVA-OneVision-7B (LLaVA). All backbones are trained using LoRA on a server equipped with four RTX 4090 GPUs (24 GiB). Policy optimization samples eight trajectories per group. Main evaluations use greedy decoding and apply the verifier only for post-generation scoring.

Table 1: Overall performance and layer-wise derivation failure rates of different urban spatial reasoning methods on Qwen3.5-9B (%).

Method	Setting	OA ↑	FA ↑	FC ↑	CBUFR ↓	WR ↓	IR ↓	L1* ↓	L2↓	L3↓	L4↓	L5↓
Spatial-Agent	Controlled	57.7	4.7	8.1	91.9	40.1	2.2	–	88.0	88.8	51.6	0.0
	Full	32.4	0.0	0.0	100.0	65.9	1.7	100.0	78.7	100.0	65.5	0.0
GeoFlow	Controlled	55.4	5.9	10.7	89.3	32.1	12.4	–	92.8	90.3	72.4	0.0
	Full	29.7	0.0	0.0	100.0	67.3	3.0	100.0	99.5	100.0	75.3	0.0
CodeV	Controlled	70.5	55.4	78.7	21.3	29.1	0.4	–	0.0	1.8	98.2	0.0
	Full	46.2	13.1	28.4	71.6	53.1	0.7	61.7	0.8	15.0	64.6	0.4
DeFacto	Controlled	66.8	51.6	77.3	22.7	32.1	1.1	–	0.0	1.8	98.2	0.0
	Full	47.0	12.3	26.1	73.9	52.0	1.0	61.9	0.8	15.5	67.9	0.4
C-GRPO	Controlled	35.2	0.0	0.0	100.0	64.8	0.0	–	100.0	100.0	100.0	82.4
	Full	37.0	0.0	0.0	100.0	63.0	0.0	98.5	28.0	50.7	100.0	83.2
Tree-GRPO	Controlled	40.3	0.0	0.0	100.0	59.7	0.0	–	52.7	62.3	100.0	84.9
	Full	36.6	0.0	0.0	100.0	63.4	0.0	99.2	33.2	63.8	100.0	81.9
UrbanSDF	Controlled	**93.1**	**83.6**	**89.8**	**10.2**	**6.9**	0.0	–	1.4	5.8	94.2	0.0
	Full	**66.3**	**33.4**	**50.3**	**49.7**	**33.7**	0.0	76.6	7.1	21.3	46.4	6.7

* L1–L5 failure rates are computed only over CBUF trajectories; L1 is not evaluated in Controlled.

5.2 Overall Performance (RQ1)

Table 1 summarizes the performance of all methods on Qwen3.5-9B. UrbanSDF achieves the highest overall answer accuracy (OA) and faithful accuracy (FA) under both the Controlled and Full settings, improving FA over the strongest baseline by 28.2 and 20.3 percentage points, respectively. Meanwhile, the proportion of faithful trajectories among correct answers (FC) improves by 45.6% on average. The layer-wise results further reveal the gap between correct tool interaction and faithful spatial derivation. Under the Controlled setting, 98.2% of CodeV's correct-but-unfaithful trajectories fail the spatial derivation check, indicating that compliant tool interaction alone does not guarantee correct downstream spatial computation. In contrast, UrbanSDF uses hierarchical certificates to verify the complete execution chain from entity binding to the final answer, allowing training to identify and correct derivation errors hidden behind correct answers. These results show that UrbanSDF maintains strong task accuracy while substantially improving the consistency between model derivations and evidence obtained through actual interactions.

5.3 Execution Certificate Effectiveness (RQ2)

Figure 3 presents the evaluation results of Qwen3.5-9B across different benchmarks. Across the four benchmarks, answer accuracy exceeds faithful accuracy by an average of 21.1 percentage points, indicating that a substantial proportion of correct answers are not obtained through faithful reasoning. For example, Visual7W requires the model to autonomously call tools to retrieve and pass candidate-box and coordinate information, compute spatial distances, and map the results to answer options. Further examination of the corresponding correct-but-unfaithful trajectories shows that 100.0% contain spatial derivation errors (L4), while 22.6% fail the final-answer entailment check (L5). Similarly, MapSpatial requires the model to identify origin and target markers on a map and derive directions, angles, or distances from map coordinates. Among its correct-but-unfaithful trajectories, 33.3% fail entity-binding checks and 50.0% fail final-answer entailment checks. These results indicate that the five-layer

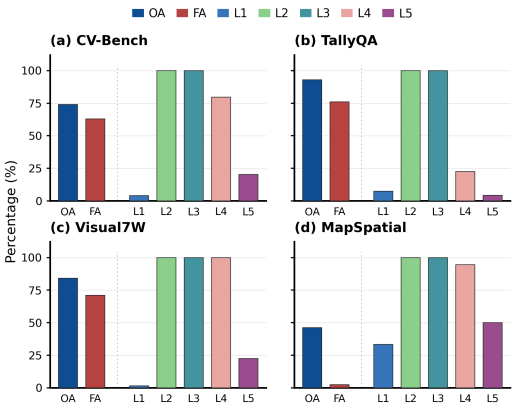


Figure 3: Execution-certificate evaluation on different spatial benchmarks (%).

Table 2: Component analysis of UrbanSDF with Qwen3.5-9B (%).

Controlled

(a) Basic training				(b) Process & online training				(c) Online optimization			
Configuration	OA ↑	FA ↑	FC ↑	Configuration	OA ↑	FA ↑	FC ↑	Configuration	OA ↑	FA ↑	FC ↑
Base	45.4	0.0	0.0	w/o Process+Outcome-GRPO	60.4	43.3	71.7	w/o Process+Dual-GRPO	79.2	65.9	83.3
w/o Process+Outcome-GRPO	60.4	43.3	71.7	w/o Outcome-GRPO	92.0	81.8	88.9	w/o Process	77.1	67.9	88.0
w/o Process+Dual-GRPO	79.2	65.9	83.3	w/o Process	77.1	67.9	88.0	w/o Dual-GRPO	89.1	79.6	89.3
UrbanSDF	**93.1**	**83.6**	**89.8**	UrbanSDF	**93.1**	**83.6**	**89.8**	UrbanSDF	**93.1**	**83.6**	**89.8**

Full

(d) Basic training				(e) Process & online training				(f) Online optimization			
Configuration	OA ↑	FA ↑	FC ↑	Configuration	OA ↑	FA ↑	FC ↑	Configuration	OA ↑	FA ↑	FC ↑
Base	35.6	0.6	1.6	w/o Process+Outcome-GRPO	51.9	11.3	21.8	w/o Process+Dual-GRPO	59.3	23.4	39.5
w/o Process+Outcome-GRPO	51.9	11.3	21.8	w/o Outcome-GRPO	62.6	28.6	45.6	w/o Process	60.7	29.4	48.4
w/o Process+Dual-GRPO	59.3	23.4	39.5	w/o Process	60.7	29.4	48.4	w/o Dual-GRPO	63.0	28.4	45.1
UrbanSDF	**66.3**	**33.4**	**50.3**	UrbanSDF	**66.3**	**33.4**	**50.3**	UrbanSDF	**66.3**	**33.4**	**50.3**

* Process: Process-SFT;
w/o Process+Outcome-GRPO: Tool-SFT;
w/o Outcome-GRPO: Tool-SFT+Process-SFT;
w/o Dual-GRPO: Tool-SFT+Process-SFT+Outcome-GRPO.
Dual-GRPO: Conditional Dual-GRPO;
w/o Process+Dual-GRPO: Tool-SFT+Outcome-GRPO;
w/o Process: Tool-SFT+Dual-GRPO;

certificate design corresponds to actual reasoning stages across different spatial tasks, providing a unified and fine-grained basis for evaluating faithfulness in urban spatial reasoning.

5.4 Component Analysis (RQ3)

Table 2 evaluates the contribution of each component across different training stages. In Table 2(a), combining Tool-SFT with Outcome-GRPO achieves an FA of 65.9%, indicating that tool supervision and outcome rewards can establish basic spatial reasoning capability; however, without explicit constraints on cross-step evidence dependencies, its FA remains 17.7 percentage points below the complete method. Table 2(b) further shows that, from the same Tool-SFT initialization, Process-SFT increases FA from 43.3% to 81.8%, whereas directly applying Dual-GRPO raises it only to 67.9%. This is because Process-SFT uses corrective demonstrations to directly learn critical spatial data dependencies and increase the probability of sampling faithful trajectories. In contrast, Dual-GRPO relies on within-group process differences from online samples; when faithful trajectories are scarce, the effective process signals available for optimization remain limited. Table 2(c) further shows that applying Outcome-GRPO after Process-SFT causes FA to decrease by 2.7% relatively, indicating that rewards based only on final answers cannot distinguish faithful derivations from CBUF trajectories among correct outputs. Conditional Dual-GRPO instead combines outcome and process advantages to further strengthen the preference for faithful derivations, enabling the complete UrbanSDF to achieve 83.6% FA. A similar trend is observed under Full: applying Outcome-GRPO directly after Tool-SFT achieves 23.4% FA, 5.2 percentage points below Process-SFT. The complete UrbanSDF further improves FA by 4.8 points while also increasing OA. These results show that Tool-SFT stabilizes tool use, Process-SFT increases faithful trajectories, and Conditional Dual-GRPO strengthens faithful preferences, jointly improving end-to-end faithful reasoning.

5.5 Faithfulness Optimization Analysis (RQ4)

To examine the model's ability to generate correct and faithful trajectories after training, we further evaluate the coverage of faithful positives. As shown in Table 3, after Tool-SFT, the L2 and L3 failure rates among correct-but-unfaithful trajectories decrease, while the spatial derivation (L4) failure rate remains high at 63.8%. After Process-SFT, FA increases to 32.3%, and Coverage@8 rises from 24.8% to 44.0%, indicating that more tasks can generate at least one correct and faithful trajectory under

Table 3: Full-setting faithfulness diagnostics with eight samples per task (%).

Stage	FA ↑	U	L1 ↓	L2 ↓	L3 ↓	L4 ↓	L5 ↓	Coverage@8 ↑
Base	2.7	2036	74.4	32.2	72.3	42.2	0.1	10.6
Tool-SFT	15.1	2030	64.5	1.5	13.6	63.8	0.0	24.8
Tool-SFT + Process-SFT	32.3	1776	81.5	5.9	21.0	44.7	0.0	44.0
UrbanSDF	**33.9**	1801	75.1	22.4	33.7	53.9	0.0	**45.1**

* Coverage@8: the percentage of tasks with at least one correct-and-faithful trajectory in eight samples; U: correct-but-unfaithful trajectory count.

the same budget of eight samples per task. With Dual-GRPO, FA is further improved by 5.0% relatively. Moreover, although the number of U trajectories remains nearly unchanged (1,776 vs. 1,801), their share among correct outputs decreases by 0.9 percentage points, further indicating that UrbanSDF improves the faithfulness of correct trajectories.

5.6 Cross-Backbone Applicability (RQ5)

Table 4 compares UrbanSDF with the outcome-only baseline. UrbanSDF consistently improves both OA and FA across all backbone models. Under the Controlled setting, OA improves by an average of 16.1 percentage points over the baseline, while FA shows a larger average gain of 31.8 percentage points, indicating that UrbanSDF not only improves answer accuracy but also more substantially enhances derivation faithfulness. Notably, under the Full setting, which requires end-to-end reasoning, the baseline OA of Llama and LLaVA is only 21.0% and 35.6%, respectively, and their correct outputs still frequently contain entity-grounding errors. Nevertheless, UrbanSDF raises their average FA from 0 to 5.1%. These results show that UrbanSDF can improve derivation faithfulness even on models with relatively weak baseline performance, further supporting its applicability across different backbones.

Table 4: Cross-backbone performance on urban spatial reasoning (%).

Backbone	Setting	Baseline		UrbanSDF	
		OA ↑	FA ↑	OA ↑	FA ↑
Qwen	Controlled	79.2	65.9	**93.1**	**83.6**
	Full	59.3	23.4	**66.3**	**33.4**
InternVL	Controlled	72.0	19.0	**87.4**	**74.9**
	Full	57.1	13.4	**65.7**	**27.0**
Llama	Controlled	64.6	28.1	**70.9**	**47.0**
	Full	21.0	0.0	**39.7**	**6.5**
LLaVA	Controlled	51.3	0.0	**80.0**	**34.5**
	Full	35.6	0.0	**46.9**	**3.6**

* Baseline: Tool-SFT+Outcome-GRPO.

5.7 Cross-City Generalization (RQ6)

Figure 4 presents the transfer performance of UrbanSDF on the held-out Sydney test set after training on data from the other cities. Under Controlled, OA decreases only slightly from 93.8% on the mixed test set to 92.3% on Sydney, while FA further increases from 79.4% to 87.8%. Under Full, the transfer gap also remains small, with OA decreasing by 2.8 percentage points and FA by only 1.2 points. Moreover, UrbanSDF substantially outperforms the outcome-only baseline on Sydney, improving FA by 16.8 points under Controlled and 17.0 points under Full. These results indicate that the learned tool dependencies and spatial derivation patterns are not tied to city-specific entities or routes, but can be transferred to an unseen city under the same tool interfaces and task specifications.

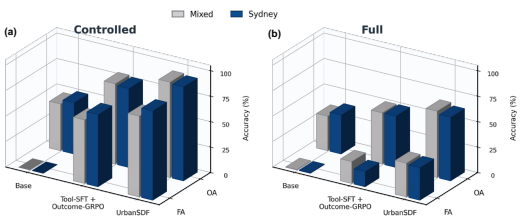


Figure 4: Cross-city generalization to Sydney under (a) Controlled and (b) Full settings.

6 Conclusion And Future Work

To address correct-but-unfaithful derivations in urban spatial reasoning, we propose UrbanSDF, which models reasoning as a complete execution chain from entity grounding to final-answer entailment and progressively applies tool supervision, certificate-guided process supervision, and Conditional-Dual GRPO to encourage spatial conclusions grounded in actual interaction evidence. In this work, we primarily use OSM-rendered maps, where entities can be explicitly annotated and execution dependencies can be reliably verified. Extending this setting to real-world urban imagery introduces additional grounding challenges due to occlusion, dense building layouts, and visually similar structures. Future work will incorporate street-view and satellite imagery to further study faithful spatial derivation under more complex visual conditions.

USE OF LARGE LANGUAGE MODELS

This study did not use generative AI tools for core research activities, including developing theoretical frameworks, mathematical derivations, hypothesis formulation, research methodology or experimental design, method implementation, data processing, or result interpretation. Generative AI was used only for language editing to improve the readability, clarity, and fluency of the manuscript, and for limited assistance with debugging experimental scripts, such as identifying potential coding issues, suggesting fixes, and improving code readability. It was not used to design or implement the proposed method.

All AI-assisted content was manually reviewed and verified by the authors. Language revisions were checked to ensure that they preserved the intended technical meaning and introduced no unsupported content, while code modifications were manually inspected, executed, and tested. The authors take full responsibility for the final text, claims, code, and other research content.

REFERENCES

Manoj Acharya, Kushal Kafle, and Christopher Kanan. TallyQA: Answering complex counting questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 2019.

Riyang Bao, Cheng Yang, Dazhou Yu, Zhexiang Tang, Gengchen Mai, and Liang Zhao. SpatialAgent: Agentic geo-spatial reasoning with scientific core concepts. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14896–14911. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.679.

Amulya Bhattaram, Justin Chung, Stanley Chung, Ranit Gupta, Janani Ramamoorthy, Kartikeya Gullapalli, Diana Marculescu, and Dimitrios Stamoulis. GeoFlow: Agentic workflow automation for geospatial tasks. In *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*, pp. 1150–1153, 2025. doi: 10.1145/3748636.3763217.

Jie Feng, Shengyuan Wang, Tianhui Liu, Yanxin Xi, and Yong Li. UrbanLLaVA: A multi-modal large language model for urban intelligence with spatial reasoning and understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Jun He, Yi Lin, Zilong Huang, Jiacong Yin, Junyan Ye, Yuchuan Zhou, Weijia Li, and Xiang Zhang. UrbanFeel: A comprehensive benchmark for temporal and perceptual understanding of city scenes through human perspective. In *The Fourteenth International Conference on Learning Representations*, 2026.

Xinhai Hou, Shaoyuan Xu, Manan Biyani, Moyan Li, Jia Liu, Todd C. Hollon, and Bryan Wang. CodeV: Code with images for faithful visual reasoning via tool-aware policy optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21500–21510, 2026.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for LLM agent reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Xiang Li, Jian Ding, and Mohamed Elhoseiny. VRSBench: A versatile vision-language benchmark dataset for remote sensing image understanding. In *Advances in Neural Information Processing Systems*, volume 37, pp. 3229–3242, 2024. doi: 10.52202/079017-0106.

MapSpatial. Map-based spatial reasoning benchmark. Hugging Face dataset, 2026. URL https://huggingface.co/datasets/mapspatial/map-spatial-benchmark.

Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Ziteng Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Xingrui Wang, Wufei Ma, Tiezheng Zhang, Celso M. de Melo, Jieneng Chen, and Alan Yuille. Spatial457: A diagnostic benchmark for 6D spatial reasoning of large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24669–24679, 2025. doi: 10.1109/CVPR52734.2025.02297.

Tianxin Wei, Ting-Wei Li, Zhining Liu, Xuying Ning, Ze Yang, Jiaru Zou, Zhichen Zeng, Ruizhong Qiu, Xiao Lin, Dongqi Fu, Zihao Li, Mengting Ai, Duo Zhou, Wenxuan Bao, Yunzhe Li, Gaotang Li, Cheng Qian, Yu Wang, Xiangru Tang, Yin Xiao, Liri Fang, Hui Liu, Xianfeng Tang, Yuji Zhang, Chi Wang, Jiaxuan You, Heng Ji, Hanghang Tong, and Jingrui He. Agentic reasoning for large language models, 2026. URL https://arxiv.org/abs/2601.12538.

Tianrun Xu, Haoda Jing, Ye Li, Yuquan Wei, Jun Feng, Guanyu Chen, Haichuan Gao, Tianren Zhang, Jing Liu, and Feng Chen. DeFacto: Counterfactual thinking with images for enforcing evidence-grounded and faithful reasoning. In *Forty-third International Conference on Machine Learning*, 2026.

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. *arXiv preprint arXiv:2505.23764*, 2025.

Jiajie Zhang, Xin Lv, Ling Feng, Lei Hou, and Juanzi Li. Chaining the evidence: Robust reinforcement learning for deep search agents with citation-aware rubric rewards. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20744–20765. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.950.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7W: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4995–5004, 2016. doi: 10.1109/CVPR.2016.540.

APPENDIX

This appendix presents supplementary experiments and provides details of the method implementation and experimental configurations. Appendix A presents the supplementary experimental settings and analyses; Appendix B gives the verification and scoring rules for the five-layer derivation certificate; Appendix C explains how certificates guide supervision-segment selection and online optimization feedback, together with the algorithm; Appendix D describes the construction of the four task types, GIS tool interfaces, evaluation metrics, and training and inference configurations.

A SUPPLEMENTARY EXPERIMENTS

Training Dynamics. Figure 5 shows how answer accuracy and faithful accuracy change across training stages. Tool-SFT obtains most of its improvements early in training, after which progress slows; introducing Process-SFT further improves FA, suggesting that targeted process supervision helps the model learn cross-step data dependencies. During subsequent policy optimization, Outcome-GRPO's OA and FA both rise initially and then gradually decline, ending below their values at the start of the stage, indicating that outcome rewards alone struggle to sustain derivation quality. In contrast, although UrbanSDF fluctuates midway through training, it subsequently recovers and maintains higher performance in the later stages. Its final OA and FA improve by 5.7% and 6.4%, respectively, over the stage initialization, highlighting the role of certificate feedback in the continued optimization of derivation faithfulness.

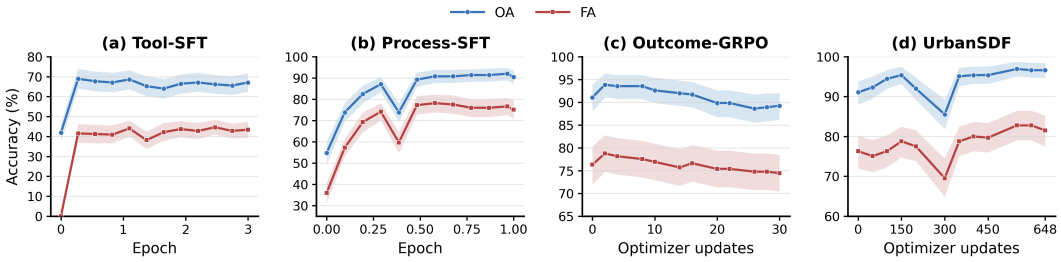


Figure 5: Training dynamics of Qwen3.5-9B under Controlled.

Online Feedback Availability Analysis. As described above, for multiple trajectories sampled from the same task, the Outcome branch compares answer correctness, while the Process branch compares local check scores for entity binding, parameter passing, and spatial derivation. To examine whether these process-level differences can provide usable feedback during online sampling, we further analyze branch activation and advantage retention in Conditional Dual-GRPO. As shown in Table 5 , the Process branch activates more frequently than the Outcome branch in both settings (Controlled: 18.5% vs. 5.6%; Full: 61.3% vs. 37.3%), indicating that local process checks more often provide comparable signals. Among process checks entering token localization, 85.6% in Controlled and 78.7% in Full retain both positive and negative advantages after filtering and can therefore contribute to local policy updates. These results show that certificates provide additional online process feedback, most of which preserves effective relative comparison signals after token localization and can participate in training.

Table 5: Credit activation and localization (%).

Measure	Controlled	Full
Groups	659	643
Outcome	37 (5.6)	240 (37.3)
Process	122 (18.5)	394 (61.3)
Mask	296 (90.8)	2064 (100.0)
Retained	279 (85.6)	1625 (78.7)

* Outcome/Process rates are computed over groups; Mask/Retained rates are computed over process checks entering localization. Retained denotes checks with both positive and negative advantages.

Case Study. Figure 6 presents the reasoning processes of different methods on the same goal-search-direction task in Sydney. The task first requires the model to identify the nearest café west of START, approximately 1–3 road junctions away, as the SOURCE. It then resolves the SOURCE on the road network, searches for hidden restaurants, computes shortest paths, and determines the initial

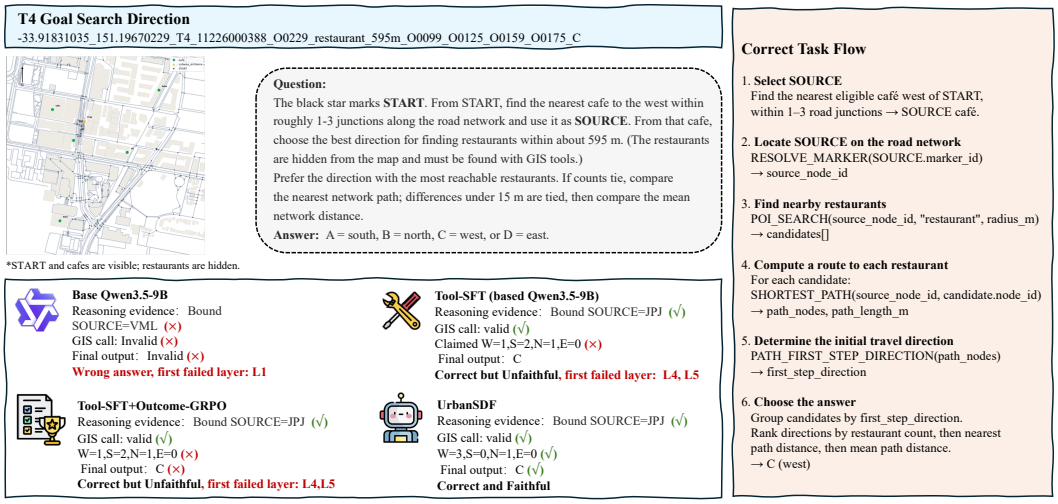


Figure 6: Case analysis of the Sydney T4 goal-search task, where the model selects a direction to search for hidden restaurants from a café west of START.

travel direction to compare the number of reachable restaurants and their path distances across different directions. The Base model incorrectly binds the SOURCE and fails to make valid tool calls, resulting in an invalid output with L1 as the first failed layer. Tool-SFT and Tool-SFT+Outcome-GRPO both bind the SOURCE correctly and complete valid tool calls, but make spatial derivation errors when aggregating the returned evidence: they claim that there is only one restaurant to the west and two to the south, whereas the tool evidence supports three to the west and none to the south. This occurs because Tool-SFT mainly supervises tool-use behavior, while Outcome-GRPO provides feedback only on the final answer; neither explicitly constrains the faithful derivation from tool observations to spatial conclusions, allowing incorrect intermediate statistics to coexist with a correct final answer. In contrast, UrbanSDF correctly uses the tool evidence to derive the directional statistics and make the final decision, producing a correct and faithful reasoning trajectory.

B Verification Rules for Five-Layer Derivation Certificates

L1–L5 all follow the task specification Ω_x. L1 checks entity bindings by canonical ID, using task-defined equivalence mappings or valid sets when needed. L2 matches calls to required operations by purpose and verifies dependencies from returned-result references; missing operations fail, while alternative orders or tools are accepted only when permitted. L3 verifies that each argument comes from an allowed instruction field, binding, or prior tool result and matches its value by type. Numerical equality alone does not establish provenance, and binding errors are not counted again as argument-passing errors.

L4 independently recomputes spatial results and dependent intermediate quantities from actual tool outputs under the task-specified scope and spatial conventions. For $\Delta_{x,k}$, numeric fields use absolute error in the specified units; identifiers and discrete decisions use exact matching, while sets and ordered sequences follow task-specified membership and order predicates. A composite result passes only if all required fields pass. L5 compares the final answer with the option entailed by the model's own conclusion rather than the reference label. For numerical checks, $\varepsilon_{x,k}(\tau)$ denotes absolute error; $\delta_{x,k}$ determines the verdict, while $\delta^{\mathrm{s}}_{x,k} > 0$ is used for graded L4 scores:

$$s_{4,k}(\tau) = \begin{cases} 1, & \varepsilon_{x,k}(\tau) \le \delta^{\mathrm{s}}_{x,k}, \\ \delta^{\mathrm{s}}_{x,k}/\varepsilon_{x,k}(\tau), & \varepsilon_{x,k}(\tau) > \delta^{\mathrm{s}}_{x,k} \end{cases}. \tag{11}$$

Missing, unparsable, or non-finite values score zero, and discrete checks are binary. Distance tolerances are 0.5 m for both verdicts and scores in T1, T2, and T4, while T3 uses 1 m for verdicts and 0.5 m for scores; scores do not override verdicts. Required omissions fail when assessable, inap-

plicable checks are excluded, and insufficient evidence leaves a check unresolved. At each layer, a confirmed failure resolves $c_\ell = 0$, all applicable checks passing resolves $c_\ell = 1$, and otherwise $c_\ell = \perp$. We compute H_x only when all five layer verdicts are resolved; if any layer remains unresolved, the trajectory is unassessable ($V_x = 0$) and counted as I.

C Certificate-Guided Training Details

Algorithm 1 presents the complete three-stage training sequence: Tool-SFT, certificate-guided Process-SFT, and online optimization. Let π_{base} be the pretrained backbone policy, $\mathcal{D}_{\text{tool}}$ the tool-call demonstrations, $\mathcal{D}_{\text{base}}$ the reference records retained for Process-SFT, $\mathcal{D}_{\text{train}}$ the task, specification, and faithful-demonstration triples $(x, \Omega_x, \tau_x^\star)$, and $\mathcal{B}_{\text{p}}$ the matched process-check branches returned by Match. Supervision masks cover selected assistant-output tokens in their original interaction contexts, excluding prompts and tool responses; each record loss is normalized by its number of supervised tokens.

Supervision-span selection. For each assessable failed check, Select maps its semantic identifier to the corresponding assistant-output span in a faithful demonstration of the same task. The union of located spans forms the token mask; prompts and tool returns remain context only, and an empty mask produces no additional local-supervision record.

During online optimization, assessable required omissions retain failure scores, while resolved checks without localized tokens still contribute to within-group advantage statistics but receive no direct gradient. Let w_{o} and w_{p} be the Outcome- and Process-track weights, and w_ℓ the weight of layer ℓ; the branch weights are $\lambda_{\text{o}} = w_{\text{o}}$ and $\lambda_{(\ell,k)} = w_{\text{p}} w_\ell$. Check losses are accumulated without normalization by the number of checks. The reference penalty applies only to generated output tokens, excluding prompts and tool responses, with each token counted once. Each eligible sampling group takes one optimizer step before policy resynchronization; because rollout behavior probabilities may differ from those recomputed during training, clipping can activate on the first update.

Cross-trajectory matching. Match aligns checks by semantic unit identifiers (layer and normalized field key), rather than call order. For each identifier observed in at least one valid trajectory, it constructs a score vector over all valid trajectories, filling an absent entry with zero. This zero is used for group comparison, not as a verifier failure verdict. A branch contributes to the token-level loss only when its score standard deviation exceeds η and its localized token mask is nonempty.

Controlled auxiliary updates. For the Qwen Controlled setting, we introduce two auxiliary updates for cases where the main Dual-GRPO signal is unavailable or insufficient. $\mathrm{NumericUpdate}$ handles isolated numerical errors: when L1–L3 and L5 pass and only one verifiable L4 numeric error remains, it supervises the corrected value with masked cross-entropy (weight 0.25), together with a correct same-field canary when available, using a separate optimizer at 0.1 times the main learning rate. $\mathrm{AnchorUpdate}$ is used when the Dual gate is inactive and all rollouts are faithful, selecting at most two rollouts to preserve correct behavior. Both auxiliary paths retain the applicable preservation loss and reference penalty, whose base coefficient is 0.05 and is multiplied by four for faithful rollouts. $\mathcal{L}_{\text{pres}}$ assigns (reference-penalty, token-NLL) weights of $(1, 0.1)$ to correct selector/lineage spans, $(0.5, 0.05)$ to correct answer spans, and $(2, 0)$ to correct continuous numeric spans, with empty masks contributing zero. The Dual update follows equation 10 with the additional $\mathcal{L}_{\text{pres}}$, whereas $\mathrm{NumericUpdate}$ and $\mathrm{AnchorUpdate}$ are applied as separate update paths; each admitted group takes one optimizer step.

D Dataset Construction and Experimental Configurations

Task Construction and GIS Tools. We construct four urban spatial reasoning task types: route-distance comparison, route-constraint assessment, road-closure impact analysis, and goal-search direction determination. Solving these tasks requires GIS interaction for road-network queries and spatial computation. Table 6 summarizes the task construction rules and answer forms, while Table 7 lists the model-callable tools and their interfaces. All tools operate on the environment-provided region and road network, with distances and angles reported in meters and degrees.

Algorithm 1 Complete UrbanSDF Training Procedure

Input: $\pi_{\text{base}}, \mathcal{D}_{\text{tool}}, \mathcal{D}_{\text{base}}, \mathcal{D}_{\text{train}}$
$\mathcal{T}, \mathcal{E}, N, \eta$, training budgets
Output: π_θ

Stage 1: Tool-SFT
Mask tool-call and argument tokens in $\mathcal{D}_{\text{tool}}$; require $\sum_n m_n > 0$ for each record
$\pi_{\text{tool}} \leftarrow \mathrm{SFT}(\pi_{\text{base}}, \mathcal{D}_{\text{tool}}; \mathcal{L}_{\text{SFT}})$ ▷ masked loss, equation 7
Stage 2: Certificate-guided Process-SFT
$\mathcal{D}_{\text{sup}} \leftarrow \mathcal{D}_{\text{base}}$
for each $(x, \Omega_x, \tau_x^\star) \in \mathcal{D}_{\text{train}}$ **do**
  $\tau \sim \mathbb{P}_{\pi_{\text{tool}}, \mathcal{E}}(\cdot \mid x, \mathcal{T})$ ▷ interaction, equation 1
  if $V_x(\tau) = 1$ **then**
    Verify L1–L5 against Ω_x
    Check L3 provenance equation 4 and L4 derivation equation 5
    Compute c_ℓ and $\Gamma_x(\tau)$ ▷ equation 3, equation 6
    Select $\mathcal{P}_x(\tau, \tau_x^\star)$ from failed checks; $m_n \leftarrow \mathbf{1}[n \in \mathcal{P}_x(\tau, \tau_x^\star)]$
    Add $(x, \tau_x^\star, m)$ to $\mathcal{D}_{\text{sup}}$ if $m \neq \varnothing$
  end if
end for
$\pi_{\text{process}} \leftarrow \mathrm{SFT}(\pi_{\text{tool}}, \mathcal{D}_{\text{sup}}; \mathcal{L}_{\text{SFT}})$ ▷ equation 7
$\theta \leftarrow \theta_{\text{process}}$; $\pi_{\text{process}} \leftarrow \mathrm{Freeze}(\pi_{\text{process}})$
Stage 3: Conditional Dual-GRPO
while online budget remains **do**
  $(x, \Omega_x, \tau_x^\star) \sim \mathcal{D}_{\text{train}}$; synchronize π_θ as π_{old}
  Sample $G_x = \{\tau_i\}_{i=1}^N$ with π_{old} ▷ equation 1
  Record behavior log-probabilities for generated tokens
  $\mathcal{V} \leftarrow \{i : V_x(\tau_i) = 1\}$
  $\Gamma_i \leftarrow \mathrm{Verify}(\tau_i; \Omega_x), \quad i \in \mathcal{V}$ ▷ equation 6
  Determine Dual and Controlled auxiliary eligibility from $\{\Gamma_i\}_{i \in \mathcal{V}}$
  if Controlled and a verified isolated numeric correction is available **then**
    $\theta \leftarrow \mathrm{NumericUpdate}(\theta, G_x)$
  else if Controlled, the Dual gate fails, and an all-faithful anchor is admitted **then**
    $\theta \leftarrow \mathrm{AnchorUpdate}(\theta, G_x, \pi_{\text{process}})$
  else if $|\mathcal{V}| \geq 2$ and the Dual group gate admits G_x **then**
    $\mathcal{B}_{\text{p}} \leftarrow \mathrm{Match}(\{\Gamma_i\}_{i \in \mathcal{V}})$
    $R_i^{\text{o}} \leftarrow C_x(\tau_i)$, $R_i^{(\ell,k)} \leftarrow s_{\ell,k}(\tau_i)$; compute A_i^b by equation 8
    Set branches with $\sigma_b \leq \eta$ to zero
    Realize branch masks and $N_{i,b} \leftarrow \sum_n m_{i,n}^b$
    Drop one-sided branches; recenter survivors as $\widetilde{A}_i^b$
    if $\exists\, i, b :\ N_{i,b} |\widetilde{A}_i^b| > 0$ **then**
      Compute $\rho_{i,n}(\theta)$ and $\mathcal{L}_{\text{KL}}$ ▷ equation 9
      $\mathcal{L} \leftarrow \mathcal{L}_{\text{Dual}}(\theta; G_x) + \mathbf{1}_{\text{Controlled}} \mathcal{L}_{\text{pres}}$
      Use equation 10 as a surrogate for equation 2
      $\theta \leftarrow \mathrm{Update}(\theta, \nabla_\theta \mathcal{L})$
    end if
  end if
end while
return π_θ

Dataset Splits. Using these task templates, we construct 6,061 distinct tasks. Ties are resolved by instruction-specified priorities; instances that remain ambiguous are removed, and reference trajectories must pass the applicable checks in Appendix B. The supervised stages use separate Tool-SFT and Process-SFT training and development records, while online optimization uses 660 tasks. Evaluation includes 325 tasks from New York, Tokyo, and Berlin and 400 Sydney tasks.

Baseline Adaptation. For fair comparison, Spatial-Agent and GeoFlow organize GIS calls from the corresponding Tool-SFT policy without additional training. CodeV and DeFacto adapt their evidence-based training to GIS from the same Tool-SFT initialization, while C-GRPO and Tree-GRPO outputs are converted to our common trajectory representation.

Evaluation Metrics. Outputs fall into four categories: correct and faithful; correct but unfaithful; valid but wrong; and invalid or unassessable. Their counts are n_F, n_U, n_W, n_I, with $N = n_F + n_U + n_W + n_I$. We compute:

$$\begin{aligned} \mathrm{OA} &= \frac{n_F + n_U}{N} \times 100\%, & \mathrm{FA} &= \frac{n_F}{N} \times 100\%, \\ \mathrm{WR} &= \frac{n_W}{N} \times 100\%, & \mathrm{IR} &= \frac{n_I}{N} \times 100\%, \\ \mathrm{FC} &= \frac{n_F}{n_F + n_U} \times 100\%, & \mathrm{CBUFR} &= \frac{n_U}{n_F + n_U} \times 100\%. \end{aligned} \tag{12}$$

Table 6: Task construction and final-answer mappings.

Task	Construction and decision rules	$R_x(\widehat{Z})$
T1: Multi-segment route-distance comparison	Specify a common source and a relay and destination for each candidate route. Compute the shortest source-to-relay and relay-to-destination walking distances and compare their sums; exact ties follow candidate order, and missing required lengths fail the check.	`route:` $\hat{l}$.
T2: Route-constraint assessment	Specify candidate destinations and route-level POI constraints. Retain routes satisfying the POI requirement, then compare major turns, path length, and junction count in that order. Length gaps within 15 m are ties; instances with no eligible route or a gap strictly between 15 and 20 m are excluded.	`target:` $\hat{l}$.
T3: Road-closure impact analysis	Specify a source, destination, and closed road segment. An unreachable normal path prevents comparison; an unreachable closed path gives C, unchanged path edges give A, and a changed path gives C when its detour reaches the instance threshold and B otherwise.	`class:` $\hat{c}$.
T4: Goal-search direction determination	Specify a target category and search range. Compare direction buckets by target count, nearest path distance, then mean distance. Empty buckets have count zero and infinite ranking distance; nearest gaps below 15 m defer to mean distance, and unresolved mean gaps below 8 m are excluded.	`option:` $m_x^{-1}(\hat{d})$.

$\hat{l}$, $\hat{c}$, and $\hat{d}$ are the model-declared label, class, and direction; m_x maps options to directions. R_x formats the declaration, while L4 checks whether it follows from tool evidence. For example, if option C denotes west and the model concludes west, then $R_x(\widehat{Z}) =$ `option: C`.

Table 7: GIS tools exposed to the model.

Tool	Inputs	Outputs
`RESOLVE_MARKER`	Marker ID.	Road node and coordinates.
`ROLE_CANDIDATE_METRICS`	Origin, candidate markers, junction mode.	Reachability, junction count, network distance.
`SHORTEST_PATH`	Source and destination nodes.	Reachability, path nodes/edges, total length.
`PATH_NAVIGATION_PROFILE`	Path nodes and markers.	Keypoints, junction count, turn angles, marker distances.
`RESOLVE_CLOSURE`	Closed-segment ID.	Closed edges and total length.
`SHORTEST_PATH_WITH_CLOSURE`	Source, destination, closed edges.	Reachability, path, and length after closure.
`POI_SEARCH`	Origin, target category, search radius.	Candidate markers, nodes, and coordinates.
`PATH_FIRST_STEP_DIRECTION`	Path-node sequence.	Initial direction and associated nodes.
`EUCLIDEAN_DISTANCE`	Two nodes.	Straight-line distance.

Let U denote the set of correct-but-unfaithful trajectories in the evaluation set, with $|U| = n_U$. Their L1–L5 failure rates are defined below; layer failures may overlap and therefore need not sum to 100%, while zero denominators and inapplicable layers are reported as N/A, including L1 under Controlled.

$$\mathrm{Fail}_{L_\ell|U} = \frac{\sum_{\tau \in U} \mathbf{1}[c_\ell(\tau) = 0]}{n_U} \times 100\%. \tag{13}$$

Training and Runtime Configurations. Table 8 reports the Qwen SFT configurations, where example counts refer to training records rather than distinct tasks. Tool-SFT selects the checkpoint

Table 8: Supervised-training configurations for Qwen.

Parameter	Tool-SFT	Process-SFT
Supervised training examples	1,819	1,036
Development examples	215	380
Learning rate	5×10^{-5}	2×10^{-5}
Optimizer	AdamW	AdamW
Microbatch size	1	1
Gradient accumulation steps	2	2
Maximum sequence length	4,096	8,192
Warmup steps	137	10
Optimizer updates	2,730	518
Training epochs	3	1

with the lowest development loss, while Process-SFT uses task-, scope-, and answer-balanced sampling with replacement for one dataset-equivalent epoch, with an effective batch size of 2. Policy optimization uses a clipping coefficient of 0.2, equal Outcome and Process weights, L1–L5 weights of $(1, 1, 1, 2, 0.5)$, and a branch-activation threshold of 0.03. During evaluation, T1–T3 include planning and derivation, while T4 additionally includes retrieval; template-level thinking is disabled, and verification is performed only after generation.