SPATIALCANVAS: A VISUAL CANVAS HARNESS FOR EXPLICIT SPATIAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models can recognize local spatial relations yet still fail on compositional spatial questions that require reasoning across viewpoints or reference frames. We argue that a key failure mode is not missing visual evidence itself, but losing the association between an observation and the entity, source view, and reference frame under which it is valid. We introduce SpatialCanvas, a training-free inference framework that makes these bindings explicit for a frozen VLM. Given a spatial question, SpatialCanvas compiles the required entities, reference frames, spatial operations, and evidence dependencies into a typed program. The VLM supplies source-bound observations, while SpatialCanvas instantiates the corresponding frames, stores observations and derived relations in an image-bound state, and executes only frame-compatible spatial transformations. A structural validator checks whether the evidence required by the program is supported before producing the final answer. With frozen Qwen3-VL-4B, SpatialCanvas improves accuracy over direct inference from 28.1 to 30.5 on MMSI-Bench, from 34.9 to 44.4 on MindCube-tiny, and from 45.7 to 48.1 on OmniSpatial. These results suggest that explicitly preserving reference-frame and source-view bindings can improve compositional spatial reasoning without updating model parameters. Our code will be made publicly available soon.

1 INTRODUCTION

Recent advances in large vision-language models (VLMs) have substantially improved visual recognition and multimodal reasoning, including spatial relation understanding Bai et al. (2025); Chen et al. (2024); Cheng et al. (2024). These capabilities provide an important foundation for applications such as embodied interaction and robotics. However, a spatial question can require more than recognizing a local relation: it may require matching entities across views, interpreting a direction under a particular reference frame, composing several geometric operations, and mapping the result to an answer option. Such compositional failures remain pronounced for frozen small VLMs, where inference-time organization is more practical than updating the model parameters.

Existing efforts address different parts of this challenge. Spatial supervision, grounded region representations, depth cues, and progressive training improve the spatial capabilities available to VLMs Chen et al. (2024); Cheng et al. (2024); Li et al. (2026). Visual programming and externalized reasoning provide a complementary strategy by decomposing a request into operations whose intermediate outputs can be reused Gupta & Kembhavi (2023). Spatial benchmarks and frame-of-reference studies further show that multi-view, perspective, and ambiguous-direction questions stress different parts of this pipeline Zhang et al. (2024); Wang et al.; Jia et al. (2026). Yet stronger perception or more explicit intermediate outputs do not by themselves preserve the semantics needed for later composition. Figure 1(a) illustrates a representative direct-reasoning failure: the model identifies the two people and their image-relative positions, but answers in the image frame rather than in the woman's observer frame.

Tool-assisted reasoning exposes a related but distinct failure mode. As shown in Fig. 1(b), a pipeline may localize entities and extract orientation cues while still leaving those observations in incompatible or unspecified coordinate systems; evidence is visible, but the final relation can remain reversed. We therefore hypothesize that losing entity, source-view, and frame bindings is a recurring failure mode in compositional spatial reasoning. A model may correctly observe that an object lies on the

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

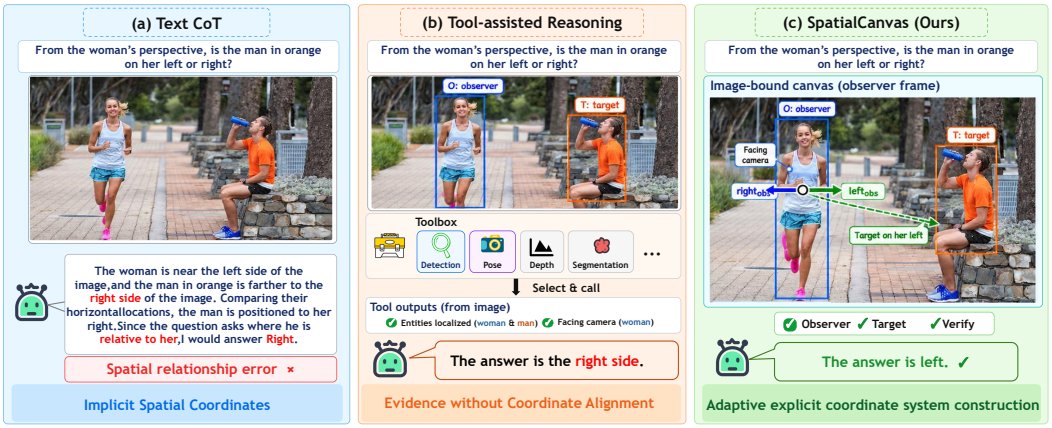


Figure 1: Motivation for SpatialCanvas. (a) Text-only reasoning compares image-relative positions without constructing the observer frame. (b) Tool-assisted reasoning exposes intermediate visual evidence but may still lack coordinate alignment. (c) SpatialCanvas constructs an image-grounded observer frame, preserves role and provenance bindings, and verifies the derived relation before answer selection.

right side of an image, but that observation does not establish its position relative to the observer or its direction in the scene. Those conclusions require an explicit alignment between image, observer, and world frames. S-Space provides related mechanistic evidence that image-relative and cardinal directions can be coupled in multimodal representations, and that explicit coordinate transformations can improve the use of internal spatial readouts MirroS Team (2026). This evidence motivates an inference-time interface that preserves observation provenance and makes frame transformations explicit; it does not imply that a structural check can certify the visual correctness of an estimated mark or orientation.

Figure 1(c) previews the design principle of **SpatialCanvas**: construct an observer frame from image-grounded orientation evidence, record the roles and coordinates of the observer and target, and verify the resulting relation before selecting an answer. To implement this principle, we propose a training-free framework that maintains an image-bound spatial state for a frozen VLM. Given a question and its images, a question-conditioned program specifies the roles to ground, the frame semantics to instantiate, the operations to perform, and the evidence obligations to satisfy. The VLM supplies short-horizon observations such as object locations, orientation cues, and cross-view correspondences. The canvas records these observations with their entity identities, source images, frame assignments, and dependencies, and renders the state back onto the source images. Before composing observations, the executor checks role, source, type, and frame compatibility and applies a transformation only when its required inputs are supported. A structural validator identifies missing or conflicting evidence; when the encoded relation uniquely matches an option, the system maps it deterministically, and otherwise the frozen VLM selects an option conditioned on the canvas and state readout.

We evaluate SpatialCanvas with frozen Qwen3-VL-4B on MMSI-Bench, MindCube-tiny, and OmniSpatial Yang et al. (2026b); Wang et al.; Jia et al. (2026). In the current preliminary results, SpatialCanvas improves each benchmark's "All" score over direct Qwen3-VL-4B inference by 2.4, 9.5, and 2.4 percentage points, respectively. We additionally use cumulative ablations and per-example diagnostics to examine proof-conditioned reliability, error correction and regression, and execution cost and program behavior. These results test whether an explicit, image-bound state can improve compositional spatial reasoning without updating the backbone.

Our contributions are summarized as follows:

- We formulate a testable account of compositional spatial errors: losing the entity, source-view, or reference-frame bindings of intermediate observations can undermine reasoning even when local observations are correct.

- We introduce SpatialCanvas, which operationalizes this account through a question-conditioned spatial program, an image-bound state, explicit frame-compatibility checks, and structural evidence validation for a frozen VLM.

- We demonstrate preliminary end-to-end improvements over direct inference on three spatial reasoning benchmarks with the same frozen backbone, together with cumulative ablations and diagnostics of proof-conditioned reliability, error correction, and execution behavior.

2 Related Work

2.1 Spatial Reasoning in Vision-Language Models

A large body of work improves spatial reasoning in VLMs through training. SpatialVLM and SpatialRGPT inject spatial supervision and region-level depth representations Chen et al. (2024); Cheng et al. (2024); SpatialLLM and Spatial-MLLM integrate 3D-informed relational data and geometric encoders Ma et al. (2025); Wu et al. (2026a); and reinforcement-learning methods such as SpaceR, SpatialLadder, and Spatial-SSRL derive spatial rewards or pretext supervision Ouyang et al. (2025); Li et al. (2026); Liu et al. (2026). These approaches improve spatial knowledge inside the model, but parameter updates do not provide an explicit inference-time record of which entities, views, and reference frames support each intermediate relation. SpatialCanvas instead operates on a frozen VLM and maintains these associations externally.

A complementary line of work studies how spatial information is interpreted inside VLMs. COMFORT shows that models are sensitive to the reference frame assumed by a question and behave inconsistently across frames Zhang et al. (2024). S-Space provides mechanistic evidence that image-relative and cardinal directions can be coupled, and that explicit coordinate transformations can improve internal spatial readouts MirroS Team (2026). These findings distinguish representing spatial information from using it consistently, and motivate an inference-time interface that preserves the scene-grounded meaning of intermediate observations without accessing hidden activations.

2.2 Visual Programming and Externalized Reasoning

Externalized reasoning delegates inference to programs, tools, or visual workspaces. PAL and Program of Thoughts use interpreters for computation Gao et al. (2023); Chen et al. (2022), and ReAct interleaves reasoning with actions Yao et al. (2022). In vision, VISPROG, ViperGPT, MM-REACT, and HuggingGPT compose perception modules or route subproblems to external tools Gupta & Kembhavi (2023); Surís et al. (2023); Yang et al. (2023); Shen et al. (2023). Visual Sketchpad, Whiteboard-of-Thought, and PyVision expose intermediate drawings or generated tools to later reasoning Hu et al. (2024); Menon et al. (2024); Zhao et al. (2025). These methods make computation explicit, but executability or visibility alone does not ensure that spatial evidence is composed under compatible interpretations: a program can execute while treating an image-relative direction as world-relative. SpatialCanvas addresses this through explicit entity and frame bindings, together with checks on evidence dependencies.

Recent spatial systems address related concerns from different angles. TIGeR learns geometric tool use through fine-tuning but does not expose an inspectable intermediate record for frame consistency Han et al. (2025). GCA fixes a task-level reference frame before execution rather than retaining per-observation frame assignments Chen et al. (2026). Allocentric Perceiver transforms reconstructed geometry into a query-conditioned frame, depending on the recovered 3D state rather than task-relevant local observations Wang et al. (2026). SpatialClaw checks frame-index alignment in a persistent workspace, which prevents source-image mismatches but does not establish compatibility between image-, observer-, and world-relative relations Cho et al. (2026). Across these systems, spatial evidence is tied to task-level constraints, a reconstructed 3D state, or index-level checks. SpatialCanvas instead keeps frame semantics in an inspectable intermediate state and checks them together with entity bindings and evidence dependencies, while keeping the backbone frozen.

3 METHOD

3.1 PROBLEM DEFINITION

A spatial relation is meaningful only with respect to a reference frame and the evidence that establishes it. Image-right, an observer's right, and world-east may denote different directions. SpatialCanvas therefore keeps the requested semantics separate from the image-grounded frame and attaches both to every derived relation.

We consider a spatial question $x = (\mathcal{I}, q, \mathcal{O})$, where $\mathcal{I} = (I_1, \ldots, I_N)$ is a set of source images, q is the question, and $\mathcal{O}$ is the set of answer options. Given a frozen VLM with parameters θ, we predict $\hat{y} \in \mathcal{O}$ through a question-conditioned program and an explicit execution state:

$$P_q = (R_q, \Gamma_q, A_q, \Omega_q), \qquad S_t = (M_t, \mathcal{F}_t, T_t, E_t). \tag{1}$$

Here R_q contains object roles, Γ_q frame requirements, A_q the ordered operations, and Ω_q the evidence obligations. The state stores grounded marks M_t, instantiated frames $\mathcal{F}_t$, derived quantities T_t, and their dependencies E_t. A requested frame is not assumed from its name: its anchor and orientation must be established from the source images.

Given x, the compiler constructs P_q and initializes the canvas before any evidence is recorded. At each subsequent step, $\mathrm{Observe}_\theta$ requests a short, source-image-bound observation, and $\mathrm{Execute}$ applies the requested deterministic operation only after checking its inputs. After the final step, SpatialCanvas renders the accumulated state, validates the evidence obligations, and selects an answer either through deterministic mapping or canvas-conditioned VLM inference. Thus visual evidence comes from the frozen VLM, while frame conversion, geometric derivation, and structural validation operate on the recorded state.

Direct answer generation entangles grounding, frame selection, geometric computation, and option mapping in one prediction. SpatialCanvas instead carries entity roles, source views, and frame assignments alongside each intermediate relation, while recording dependencies for derived quantities. This makes the inputs to deterministic execution explicit without requiring access to the VLM's hidden reasoning.

This separation determines what the canvas is meant to preserve. A textual rationale may state a relation without retaining which entity, source view, or frame supports it, while a drawing alone does not specify how an annotation should be interpreted. SpatialCanvas therefore lets the frozen VLM propose source-bound observations and lets deterministic operators perform only the supported geometric steps. The canvas does not make those observations infallible; it makes their assumptions explicit and gives the executor a well-defined point at which to reject unsupported compositions or request refinement.

3.2 PROGRAM CONSTRUCTION AND IMAGE-BOUND CANVAS

Because the required geometric operations depend on the queried entities, views, and reference frame, we first compile these requirements into a plan. The question-only router identifies the queried roles, relevant views, frame semantics, and required operations without using dataset labels or ground-truth answers. Its output is

$$\begin{aligned} \tau_q &= (R_q, \Gamma_q, V_q, A_q, \Omega_q), \quad & V_q &= \mathrm{SelectViews}_q(\mathcal{I}), \\ \gamma &= (\kappa, \rho, \nu, \eta) \in \Gamma_q, & a_t &= (\mathrm{op}_t, \xi_t, \gamma_t, \omega_t), \end{aligned} \tag{2}$$

where V_q is the active view subset, γ specifies frame kind, anchor role, source views, and axis directions, and a_t includes operation inputs and its evidence obligation. View selection is question-conditioned: single-view questions need not invoke matching, while multi-view questions activate only the views required by their program.

SpatialCanvas uses typed operators for grounding, frame construction, cross-view matching, coordinate transformation, geometric solving, and answer mapping. The implementation names these operations `LOCATE`, `GROUND_SET`, `SOURCE_FRAME`, `CAMERA_ROTATION`, `MATCH_SET`, `VIEW_TRANSFORM`, `MEASURE`, `SOLVE`, and `MAP`. Here $\mathcal{R}$ denotes the task-specific relation specification, and α denotes a heading or camera rotation supplied by the relevant observation. The VLM

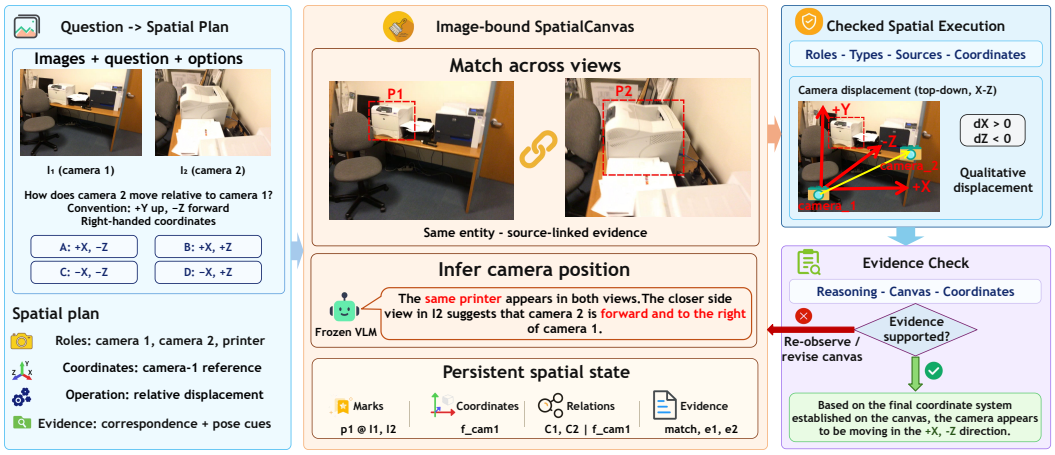


Figure 2: Overview of SpatialCanvas. A question-conditioned program specifies roles, frame requirements, operations, and evidence obligations. The canvas stores source-bound observations and renders them with frames, correspondences, and derived relations before answer selection.

proposes marks, orientation cues, and correspondence candidates; deterministic operators consume them only when the required source, unit, and frame metadata are present.

Compilation determines which operators are needed and in what dependency order. For a multi-view displacement question, a typical chain is `GROUND_SET` → `MATCH_SET` → `SOURCE_FRAME` → `VIEW_TRANSFORM` → `SOLVE` → `MAP`; a single-view relation can skip matching and transformation. Each step carries its own evidence obligation ω_t. To meet these obligations, we record each grounded observation together with its source image and entity role:

$$m_i = (\mathrm{id}_i, j_i, \rho_i, b_i, \pi_i), \qquad I_{j_i} \in \mathcal{I}. \tag{3}$$

Here b_i is a point or region in the original source image and π_i is its provenance. Cross-view correspondence is stored separately, so sharing an entity identity does not make image coordinates interchangeable. Frame grounding then binds the requested semantics to observed anchors and orientation cues:

$$\begin{aligned} f &= \mathrm{SOURCE_FRAME}(\gamma, M^{\mathrm{anchor}}, E^{\mathrm{orient}}), \\ f &= (\gamma, \mathcal{D}_f, o_f, U_f, \pi_f). \end{aligned} \tag{4}$$

The frame stores its domain, origin, directed axes, and supporting provenance. If an anchor or orientation is unavailable, the frame remains incomplete instead of defaulting to image-left or image-right.

A supported cross-frame conversion is

$$p^{(f)} = R_{f \leftarrow g}\big(p^{(g)} - o_g\big) + o_f, \qquad d_f(A, B) = U_f^{\top}\big(p_B^{(f)} - p_A^{(f)}\big), \tag{5}$$

where the transform and origins are state variables with provenance. A normalized box is sufficient for an image-relative relation, but does not by itself provide depth, camera pose, or a world coordinate. The canvas stores marks, frames, correspondences, and derived relations such as $r(A, B \mid f)$, and renders boxes, axes, vectors, and measurements on the source images.

The state consequently distinguishes observations from consequences. Marks, orientation cues, and correspondences are source-bound records supplied by the VLM; transformed points, measurements, and qualitative relations are derived records produced by typed operators; and missing inputs or conflicts are diagnostics. A rendered annotation can expose all three to the next answer step, but it cannot silently promote a derived value into a new observation.

3.3 Compatible Execution and Evidence Validation

Even with this metadata, a program could still combine records from incompatible roles, views, or frames. Before applying an operation, SpatialCanvas checks role bindings, input types and units, source-view links, and frame components:

$$K_t = \mathrm{Compat}(\bar{S}_t, a_t) = \begin{array}{l}\mathrm{Roles}(\bar{S}_t, a_t) \wedge \mathrm{Types}(\bar{S}_t, a_t) \\ \qquad \wedge \mathrm{Sources}(\bar{S}_t, a_t) \wedge \mathrm{Frames}(\bar{S}_t, a_t).\end{array} \tag{6}$$

The transition is therefore

$$S_t = \begin{cases} \mathrm{Apply}(\bar{S}_t, a_t), & K_t = 1, \\ \mathrm{Diagnose}(\bar{S}_t, a_t), & K_t = 0. \end{cases} \tag{7}$$

A failed prerequisite produces a diagnostic rather than an unsupported geometric result. When refinement is enabled, the diagnostic triggers an evidence-directed repair rather than an unconstrained second attempt: a missing role requests another grounded observation, an unsupported frame requests a new orientation cue or reconstruction, and an inconsistent correspondence requests re-matching. Only downstream records affected by an upstream replacement are invalidated and recomputed, preserving unaffected evidence. We allow at most B refinement attempts, indexed by k:

$$S_{t+1} = \mathrm{Refine}_\theta(S_t, d_t), \qquad k < B. \tag{8}$$

Operation-level checks establish that each step has compatible inputs; they do not establish that the full question has enough evidence for an answer. After execution, the validator checks whether the program obligations are structurally supported:

$$v = (b, d) = \mathcal{V}(P_q, S_L), \qquad b = \bigwedge_{\omega \in \Omega_q} \omega(S_L). \tag{9}$$

Let $\mathcal{J}_q(S_L, \mathcal{O})$ denote the options compatible with the derived relation. The final decision is

$$\hat{y} = \begin{cases} \mathrm{Map}_{\mathrm{det}}(S_L, \mathcal{O}), & b = 1 \ \wedge \ |\mathcal{J}_q(S_L, \mathcal{O})| = 1, \\ \mathrm{Select}_\theta(q, \mathcal{O}, C_L, H_L), & \text{otherwise}, \end{cases} \tag{10}$$

where $H_L = \mathrm{Readout}(P_q, S_L, v)$ contains the recorded roles, source views, frames, relations, and diagnostics. The deterministic branch is used only when the evidence chain is complete and the answer is unique; otherwise the frozen VLM selects from the rendered canvas and state readout. The validator certifies structural support and provenance, not the factual correctness of visual marks.

Design boundary and validation. The canvas is deliberately image-bound and operator-light: it preserves entity identity, source view, frame assignment, and dependencies while leaving visual recognition to the frozen VLM. This keeps textual rationale from becoming an unsupported observation and makes each derived relation inspectable. The validator is a structural gate, not a second vision model; a complete proof means that required roles, sources, frames, and operation outputs are present and compatible, not that every visual mark is factually correct. An incorrect upstream observation can therefore still produce a wrong answer, while an incomplete canvas is passed to the frozen VLM for a best-effort option choice. This distinction guides the proof-conditioned and error-transition analyses in Section 4.4.

4 Experiments

4.1 Evaluation Setup

Benchmarks. We evaluate SpatialCanvas on three spatial reasoning benchmarks with complementary task coverage. MMSI-Bench reports the PR., Attr., Mot., and MSR categories Yang et al. (2026b); MindCube-tiny reports Rot., Ard., and Amg. Wang et al.; and OmniSpatial reports Dyn. and Pers. Jia et al. (2026). We retain these benchmark-provided category labels and report the "All" score produced by each benchmark's evaluator. "All" is an official aggregate over the evaluated examples, not a simple arithmetic mean of the displayed category columns.

Models and protocol. For the controlled comparison, we keep Qwen3-VL-4B frozen Bai et al. (2025). The Direct reference receives the image and question, and is asked to reason about the task and return an answer without explicit grounding, canvas rendering, or geometric operators. SpatialCanvas uses the same backbone. All ablation variants use the same input instances, parser, per-call decoding limit, legality checks, and answer format; the Full variant may invoke additional observation rounds as part of evidence refinement.

Table 1 reports the end-to-end comparison, while Table 2 shows the cumulative configurations leading to SpatialCanvas. The Direct and Full endpoints match across the two tables.

Table 1 places open-source spatial models, including training-based methods and tool-integrated systems Wu et al. (2026a); Li et al. (2026); Ouyang et al. (2025); Feng et al. (2026); Team et al. (2025); Wu et al. (2026b); Yang et al. (2026a), alongside the same-backbone Qwen3-VL-4B reference and SpatialCanvas. The shaded lower block lists general-purpose foundation VLMs, including GPT-4o, Gemini-2.5-Pro, Qwen3-VL-Thinking, and GLM-4.5V Bai et al. (2025); Hong et al. (2025). Prior methods are shown under their reported evaluation setups as context; our controlled comparison is between direct reasoning with Qwen3-VL-4B and SpatialCanvas with the same backbone.

4.2 Main Results

Table 1 reports benchmark accuracies on MMSI-Bench, MindCube-tiny, and OmniSpatial Yang et al. (2026b); Wang et al.; Jia et al. (2026). On each benchmark's "All" score, SpatialCanvas improves over direct Qwen3-VL-4B from 28.1 to 30.5, from 34.9 to 44.4, and from 45.7 to 48.1, respectively.

The category-level results reveal a selective pattern. On MMSI-Bench, gains are largest for Mot. and MSR (+8.6 and +5.6 points), whose motion and multi-step questions require carrying observations across changes or several reasoning steps; Attr. improves by 2.3 points, while PR. decreases by 0.6. This pattern is consistent with explicit state being useful when evidence must be composed. On OmniSpatial, Dyn. improves by 8.8 points, whereas Pers. decreases by 3.2, showing that gains differ between dynamic-scene and perspective-taking tasks. MindCube-tiny shows the largest gain in its "All" score (+9.5 points), alongside a large Rot. difference (37.0 to 87.0) on rotation across views. This result highlights the value of explicit frame construction for rotation across viewpoints. Amg. improves by 2.2 points, while Ard. decreases by 5.2.

The general-purpose VLM rows provide additional context by benchmark. SpatialCanvas reaches 30.5 on MMSI-Bench, close to GPT-4o at 30.3; on MindCube-tiny, its 44.4 is above the listed GPT-4o and GLM-4.5V scores and 2.9 points below Qwen3-VL-Thinking-30B. On OmniSpatial, SpatialCanvas scores 48.1, compared with 51.0–55.8 for the listed general-purpose VLMs. These reported values contextualize the compact-backbone results; the controlled comparison remains the same-backbone Qwen3-VL-4B reference.

4.3 Cumulative Ablation Study

We evaluate the framework as a cumulative sequence. Direct gives the image and question to frozen Qwen3-VL-4B and asks it to reason about the task and answer. Each subsequent row adds one module while preserving the parser, legality checks, per-call decoding limit, answer format, and benchmark instances; Full may invoke additional observation rounds for evidence refinement.

The five configurations have the following meanings. **Direct** passes the image and question directly to the model. **+ Grounding** first records which entities occur, where they are located, which view contains each entity, and any cross-view correspondence; these observations are passed as text, without drawing boxes, constructing a reference frame, or invoking a spatial calculator. **+ Coordinate Reasoning** adds observer, camera, or world frames and performs the required projection, rotation, distance, or directional computation, with the result still passed as text and not rendered back onto the image. **+ Visual Feedback** renders the existing boxes, axes, vectors, and projections on the source images while retaining the same textual evidence rule. **+ Evidence Refinement (Full)** allows consistency checks to trigger re-observation, box correction, orientation reconstruction, supplementary observations, or re-measurement. Together, these configurations trace the progression from explicit grounding to computation, visual feedback, and evidence correction.

Method	MMSI-Bench					MindCube-tiny				OmniSpatial		
	PR.	Attr.	Mot.	MSR	All	Rot.	Ard.	Amg.	All	Dyn.	Pers.	All
Spatial-MLLM (2026a)	28.5	25.4	18.0	26.3	26.1	33.8	34.5	28.3	32.1	37.2	42.1	40.0
SpatialLadder (2026)	30.3	23.3	16.0	21.2	25.4	30.5	39.8	**47.8**	42.3	46.5	43.1	44.5
SpaceR (2025)	29.1	29.4	21.9	22.5	26.9	29.8	30.0	26.8	28.3	53.5	40.5	46.0
Video-R1 (2026)	30.5	25.4	22.0	26.8	27.8	30.0	30.5	41.3	35.8	50.0	44.2	46.7
RoboBrain2.0 (2025)	28.9	28.8	22.5	28.0	28.9	29.7	35.8	45.2	39.6	49.4	42.2	45.2
VILASR (2026b)	**35.9**	26.0	21.0	23.2	29.8	34.4	25.7	29.4	29.1	37.5	42.2	40.2
VLaser (2026a)	29.8	26.9	26.0	18.9	27.3	31.5	24.8	38.2	32.6	39.1	42.6	41.1
Qwen3-VL-4B (2025)	29.7	30.0	26.7	23.7	28.1	37.0	**46.4**	29.3	34.9	47.4	**44.4**	45.7
SpatialCanvas (ours)	29.1	**32.3**	**35.3**	**29.3**	**30.5**	**87.0**	41.2	31.5	**44.4**	**56.2**	41.2	**48.1**
General-purpose foundation VLMs												
GPT-4o (2024)	28.0	32.3	36.0	30.8	30.3	33.5	37.0	35.2	35.8	58.7	46.2	51.5
Gemini-2.5-Pro (2025)	39.0	36.2	33.3	34.3	36.9	89.5	54.5	48.8	57.5	70.7	44.6	55.8
Qwen3-VL-Thinking-30B (2025)	33.7	40.0	23.3	31.8	32.6	87.0	47.3	35.0	47.3	60.5	43.9	51.0
GLM-4.5V (2025)	35.6	36.9	29.3	30.3	33.8	60.0	25.5	42.2	39.6	58.6	47.2	52.1

Table 1: Benchmark accuracy (%) on MMSI-Bench Yang et al. (2026b), MindCube-tiny Wang et al., and OmniSpatial Jia et al. (2026). Each "All" is the benchmark evaluator's aggregate over its evaluated examples. General-purpose VLMs are included for context.

Configuration	Ground.	Coord.	Visual	Refine.	MMSI	MindCube	Omni
Direct (Qwen3-VL-4B)	–	–	–	–	28.1	34.9	45.7
+ Grounding	✓	–	–	–	27.2	30.9	43.3
+ Coordinate Reasoning	✓	✓	–	–	28.7	36.5	44.9
+ Visual Feedback	✓	✓	✓	–	29.3	37.8	46.8
+ Evidence Refinement (Full)	✓	✓	✓	✓	30.5	44.4	48.1

Table 2: Cumulative ablation of frozen Qwen3-VL-4B. Direct is prompted to reason from the image and question; Full is SpatialCanvas.

Relative to Direct, Grounding changes the three scores by -0.9, -4.0, and -2.4 points. Coordinate Reasoning recovers $+1.5$, $+5.6$, and $+1.6$ points over Grounding; Visual Feedback adds $+0.6$, $+1.3$, and $+1.9$, followed by $+1.2$, $+6.6$, and $+1.3$ from Evidence Refinement. The Full scores (30.5, 44.4, and 48.1) match SpatialCanvas in Table 1. These values summarize the cumulative progression, from an initial drop with Grounding to gains with the complete framework.

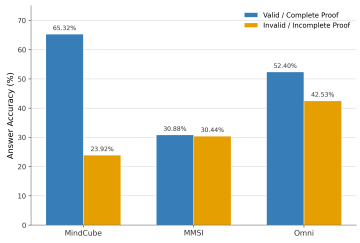
4.4 Diagnostic Analysis

The following analyses use execution traces from the current diagnostic snapshot. They address two practical questions that are not visible in an aggregate score: when the canvas state is reliable, and what computation is required to obtain it.

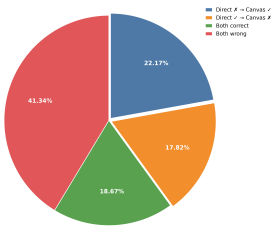
4.4.1 Reliability and Error Correction

We first examine whether a structurally complete evidence chain is associated with a more reliable answer, and whether the complete system corrects more errors than it introduces. Figure 3 reports both analyses. In panel (a), the bars show answer accuracy conditioned on the validator status of the final proof. Across the full diagnostic run, complete proofs cover 39.43% of 3,031 examples. Complete versus incomplete proof accuracy is 30.88% versus 30.44% on MMSI-Bench, 65.32% versus 23.92% on MindCube-tiny, and 52.40% versus 42.53% on OmniSpatial. The association is therefore strongest on MindCube-tiny and negligible on MMSI-Bench; proof completeness is not a uniform confidence signal. The validator checks structural support and provenance, not the factual correctness of the underlying visual observations.

Panel (b) evaluates the same 3,031 examples under Direct and the complete SpatialCanvas configuration. SpatialCanvas corrects 672 Direct errors (22.17%) and introduces 540 errors (17.82%); both systems are correct on 566 examples (18.67%) and wrong on 1,253 (41.34%). The paired comparison therefore yields a net gain of 132 examples, or 4.35 percentage points, with 1.24 corrected



(a) Proof-conditioned accuracy



(b) Paired error transitions

Figure 3: Reliability and error correction. (a) Proof-conditioned answer accuracy; complete proofs cover 39.43% of 3,031 examples. (b) Direct-to-SpatialCanvas transitions, including corrected and introduced errors.

examples for every introduced error. The large both-wrong group shows that substantial errors remain unresolved.

4.4.2 Efficiency Analysis

Table 3 focuses on generated output volume; execution traces also show how model use varies across benchmarks. SpatialCanvas averages 12.96 VLM calls per sample on MMSI-Bench, 12.86 on MindCube-tiny, and 8.62 on OmniSpatial (Appendix C). These calls include visual observations and refinement decisions, while deterministic geometric operators do not count as VLM calls.

Call count alone does not capture generation-side output volume. Table 3 shows that SpatialCanvas requests compact, structured observations (e.g., localization fields) rather than a free-form rationale, averaging 2.08 generated tokens per call and 23.91 per sample over 11.52 calls per sample. This is about 11.9× fewer generated output tokens than the one-call CoT response (283.6 tokens per sample). These figures measure generated output only: they exclude prompt and image tokens, repeated input processing across calls, and wall-clock latency, so they should not be interpreted as total inference cost.

Method	Calls/ Sample	Tokens/ Call	Tokens/ Sample
Answer-only	1	2.6	2.6
SpatialCanvas	11.52	2.08	23.91
CoT	1	283.6	283.6

Table 3: Generated output tokens under three response protocols.

Accuracy across call-count bins is non-monotonic, ranging from 26.46% to 48.13%; questions requiring more observations may also be harder. Operation use is selective: `MAP` and `GROUND_SET` are invoked in 100.0% and 86.4% of examples, while `SOURCE_FRAME`, `MATCH_SET`, and `VIEW_TRANSFORM` are invoked in 49.0%, 43.1%, and 38.8%, respectively; `MEASURE`, `SOLVE`, and `CAMERA_ROTATION` each occur in 8.9–11.2%. These trace statistics describe question-conditioned execution but do not isolate the contribution of individual operations to accuracy (Appendix C).

5 Conclusion

We presented SpatialCanvas, a training-free harness that gives a frozen VLM an explicit spatial program and an image-bound state. The canvas preserves entity, source-view, and frame bindings; deterministic operators execute only compatible transformations, and a validator checks the evidence chain. On MMSI-Bench, MindCube-tiny, and OmniSpatial, preliminary results improve over the same-backbone Qwen3-VL-4B reference, with ablations and diagnostics showing how grounding, computation, feedback, and refinement affect the pipeline. SpatialCanvas thus provides an inspectable interface for combining visual observations with geometric reasoning.

Limitations and future work. Our study focuses on static image-based spatial reasoning. Extending it to video, 3D, and embodied settings, evaluating across model families, and broadening the operator set are natural next steps, along with adaptive program compilation, operator learning, and interactive multi-turn reasoning.

AI USE STATEMENT

In this work, we used generative AI tools for language editing, organization of the manuscript, LaTeX formatting, and consistency checks. We did not use generative AI tools to generate experimental data, alter benchmark examples, run experiments, or determine the reported numerical results. All AI-assisted text and formatting were reviewed and revised by the authors, who take responsibility for the final content, claims, results, and artifacts of this work.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 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.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022.

Zeren Chen, Xiaoya Lu, Zhijie Zheng, Pengrui Li, Lehan He, Yijin Zhou, Jing Shao, Bohan Zhuang, and Lu Sheng. Geometrically-constrained agent for spatial reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38689–38699, 2026.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093, 2024.

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

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *Advances in Neural Information Processing Systems*, 38:99114–99137, 2026.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International conference on machine learning*, pp. 10764–10799. PMLR, 2023.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14953–14962. IEEE, 2023.

Yi Han, Enshen Zhou, Shanyu Rong, Jingkun An, Pengwei Wang, Zhongyuan Wang, Cheng Chi, Lu Sheng, and Shanghang Zhang. Tiger: Tool-integrated geometric reasoning in vision-language models for robotics. *arXiv preprint arXiv:2510.07181*, 2025.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *Advances in Neural Information Processing Systems*, 37:139348–139379, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

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.

Hongxing Li, Dingming Li, Zixuan Wang, Yuchen Yan, Hang Wu, Wenqi Zhang, Yongliang Shen, Weiming Lu, Jun Xiao, and Yueting Zhuang. Spatialladder: Progressive training for spatial reasoning in vision-language models. In *International Conference on Learning Representations*, volume 2026, pp. 76566–76592, 2026.

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, 2026.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17249–17260. IEEE, 2025.

Sachit Menon, Richard Zemel, and Carl Vondrick. Whiteboard-of-thought: Thinking step-by-step across modalities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20016–20031, 2024.

MirroS Team. S-space: Exploring spatial workspace in multimodal models, September 2026. URL `https://mirros.ai/blog/s-space`. Blog post.

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 preprint arXiv:2504.01805*, 2025.

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

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11854–11864. IEEE, 2023.

BAAI RoboBrain Team, Mingyu Cao, Huajie Tan, Yuheng Ji, Xiansheng Chen, Minglan Lin, Zhiyu Li, Zhou Cao, Pengwei Wang, Enshen Zhou, et al. Robobrain 2.0 technical report. *arXiv preprint arXiv:2507.02029*, 2025.

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 preprint arXiv:2602.05789*, 2026.

Qineng Wang, Baiqiao Yin, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, et al. Mindcube: Spatial mental modeling from limited views, 2026. *URL https://arxiv. org/abs/2506.21458*, 3.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *Advances in neural information processing systems*, 38: 13569–13597, 2026a.

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. *Advances in Neural Information Processing Systems*, 38:143297–143330, 2026b.

Ganlin Yang, Tianyi Zhang, Haoran Hao, Weiyun Wang, Yibin Liu, Dehui Wang, Guanzhou Chen, Zijian Cai, Junting Chen, Weijie Su, et al. Vlaser: Vision-language-action model with synergistic embodied reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 65736–65765, 2026a.

Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, et al. Mmsi-bench: A benchmark for multi-image spatial intelligence. In *International Conference on Learning Representations*, volume 2026, pp. 157051–157088, 2026b.

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

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

Zheyuan Zhang, Fengyuan Hu, Jayjun Lee, Freda Shi, Parisa Kordjamshidi, Joyce Chai, and Ziqiao Ma. Do vision-language models represent space and how? evaluating spatial frame of reference under ambiguities. *arXiv preprint arXiv:2410.17385*, 2024.

Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Ming Li, Qilong Wu, Kaipeng Zhang, and Chen Wei. Pyvision: Agentic vision with dynamic tooling. *arXiv preprint arXiv:2507.07998*, 2025.

A BENCHMARK DESCRIPTIONS

Table 4 summarizes the evaluated examples, benchmark categories, and task focus of each dataset.

Dataset	Examples	Category	Description
MMSI-Bench Yang et al. (2026b)	1,000	PR	Positional relations between objects and regions.
		Attr.	Spatial attributes such as object size and appearance.
		Mot.	Camera and object motion within a scene.
		MSR	Tasks requiring a sequence of spatial reasoning steps.
MindCube-tiny Wang et al.	1,050	Rot.	Rotation across viewpoints.
		Ard.	Reasoning about objects located around a reference.
		Amg.	Reasoning about relations among multiple objects.
OmniSpatial Jia et al. (2026)	981	Dyn.	Spatial reasoning about dynamic scenes.
		Pers.	Spatial relations viewed from different perspectives.

Table 4: Evaluation subsets, category labels, and task focus of the three benchmarks.

For OmniSpatial, the reported 981 examples are the combined evaluation subset for its Dyn. and Pers. categories.

B REPRODUCIBILITY DETAILS

The following boxes show the visual-evidence prompts. Task-specific inputs such as the question, target, marks, and role schema are inserted as needed.

2.1 Object localization. *Purpose:* locate the named observer or anchor, without answering the question.

You are the image-bound localization module of a spatial reasoning harness. Inspect the source image itself. The question names an observer or anchor. Your only job is to locate that one entity; do not find other objects, do not infer left/right, do not establish a coordinate frame, and do not answer the multiple-choice question.
Question: {question} **Target description:** {target_description}
Return exactly this schema:
`{"operation":"LOCATE","role":"observer","label":"<specific target>", "view_id":"0","bbox_2d":[x1,y1,x2,y2], "coordinate_space":"canvas_qwen_1000","confidence":0.0}`.
Use Qwen-VL `bbox_2d` coordinates in the 0..1000 coordinate system of the ENTIRE ATTACHED CANVAS. The box must tightly contain the named entity inside the source image and must not include the title, question, options, border, margins, or nearby people.

2.2 Direction and reference-frame judgment. *Purpose:* estimate the observer's physical left and forward directions from visible pose cues.

You are the observer-orientation module of a spatial reasoning harness. The attached canvas contains the original source image with exactly one highlighted observer box. Use that highlighted entity as the origin. Do not use the canvas border or generic image-left/image-right.
Estimate the observer's forward direction from visible pose cues. Distinguish the observer's physical left from the viewer's image-left: if the person faces the camera, their physical left is usually on the image-right side.
Return exactly this schema:
{"operation":"ORIENT","origin_mark":"m1","direction":"body-left",
"left_vector":[dx,dy],"forward_vector":[dx,dy],
"left_image_side":"image_left_or_image_right",
"front_image_side":"image_up_or_image_down",
"left_source":"visible_pose","source":"visible_pose","confidence":0.0}.
All vectors are nonzero directions in SOURCE-IMAGE coordinates: +x is image right and +y is image down.

2.3 Orientation-cue observation. *Purpose:* record multiple independent visible cues before frame construction.

You are the orientation-cue module. Inspect only the highlighted observer in the source image. Extract several independent visible cues before constructing any coordinate frame.
Return exactly:
{"operation":"OBSERVE_CUES","origin_mark":"m1","cues":[
{"part":"hand|shoulder|face|torso|feet|scene|affordance_heading",
"observation":"what is visible",
"image_direction":"image_left|image_right|image_up|image_down|unknown",
"facing":"facing_camera|facing_away|
facing_image_left|facing_image_right|unknown",
"supports":"physical_left|physical_front|none","confidence":0.0}].
Do not convert image_left into physical left. supports is not a confidence field: use supports=none for generic observations. Prefer 2 to 5 concrete cues and short observations.

2.4 Candidate frame construction. *Purpose:* construct at most two observer-centered candidate frames from recorded cues.

You are the frame-construction module. Construct at most two candidate coordinate frames at the highlighted observer.
Stand conceptually at the CENTER of the highlighted observer and face in the SAME physical direction as that observer. The frame is anchored to this named observer, not to the canvas, camera, target, background, or displayed image axes.
Vectors are in SOURCE-IMAGE coordinates: +x=image right, +y=image down. left_vector points to the observer's physical left, not the viewer's left. For a clear side profile, convert facing_image_left/right into both physical axes:
- facing_image_left means physical LEFT=image_down and FRONT=image_left
- facing_image_right means physical LEFT=image_up and FRONT=image_right
If facing_camera, physical LEFT is image-right; if facing_away, physical LEFT is image-left.
Return candidate identifiers (A or B), left and forward vectors, observer-facing classification, and the cue identifiers supporting each candidate.

2.5 Candidate frame consistency check. *Purpose:* check visual attachment and consistency of candidate axes.

You are the frame-consistency module. Inspect the source image with every colored candidate axis anchored at the highlighted observer. Check visual attachment and body orientation.
Reject a candidate if its vector contradicts its image-side label, its observer-facing classification disagrees with the highlighted body, its axes are not attached to the observer, or its LEFT and FRONT axes are not approximately perpendicular.
If the observer faces the camera or away from it, do not require a 2D image arrow for physical FRONT. Check that LEFT is mirrored correctly relative to the observer's body.

2.6 Candidate frame selection. *Purpose:* choose an existing candidate without introducing a new frame.

Select one existing candidate ID. Do not alter its vectors, construct a new frame, or answer the question. Prefer supported candidates with no geometric conflict. If the visual evidence is ambiguous, select the least-conflicting candidate and lower confidence.

2.7 Cross-view entity grounding. *Purpose:* localize entities across one or more requested views.

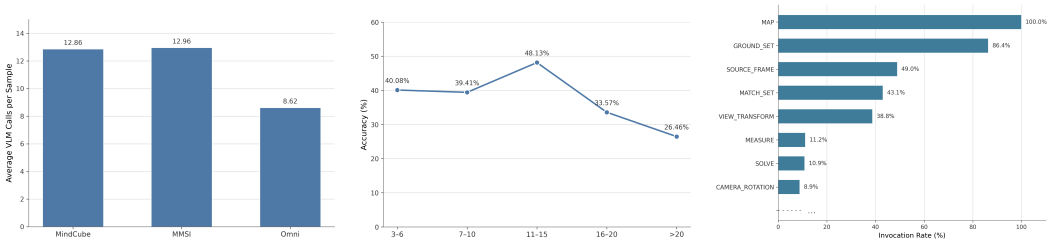
You are the grounding module of a visual proof harness. The attached canvas contains one or more source views. Your only job is semantic localization and orientation estimation. Do not answer the multiple-choice question.
Return every entity needed by this exact typed role schema:
- {role}: {description}; cardinality {min}..{max}; exactly one in every source view
Use exact role strings from the schema. Each entity format is:
{role, label, view_id, bbox:[left,top,right,bottom], confidence}.
Coordinates must use the full attached canvas Qwen 0..1000 system and set `coordinate_space:'canvas_qwen_1000'`. For a set role, return one entity per distinct visible instance; never duplicate a box. For per-view roles, use the same semantic label across views.

2.8 Bounding-box refinement. *Purpose:* tighten an existing box or re-anchor the same semantic mark to the correct nearby instance.

You are the visual box-refinement module. Inspect the SOURCE IMAGE and the one highlighted rough box. Tighten that existing box to the visible pixels of the named entity only.
If the rough box points at the wrong nearby instance, re-anchor the SAME semantic mark on the correct visible entity.
Return exactly:
`{"operation":"REFINE","mark_id":"{mark_id}","view_id":"{view_id}", "bbox_2d":[x1,y1,x2,y2], "coordinate_space":"canvas_qwen_1000","confidence":0.0}`.
The refined box should normally stay inside the rough box. If the rough box visibly cuts off part of the named entity, a modest expansion is allowed, but do not add background or a neighboring entity.

C Supplementary Execution Analysis

Figure 4 provides additional diagnostics of how SpatialCanvas uses model calls and typed operations. Panel (a) shows mean VLM calls per sample: 12.96 on MMSI-Bench, 12.86 on MindCube-tiny, and 8.62 on OmniSpatial, matching the plotted values. These calls include visual observations and refinement decisions; deterministic geometric operators do not themselves count as VLM calls.



(a) VLM calls per sample (b) Calls versus accuracy (c) Operation invocation profile

Figure 4: Supplementary execution diagnostics. Call-count bins in (b) are descriptive, not causal.

Panel (b) groups examples by their observed call count. Accuracy is 40.08% for examples requiring 3–6 calls, 39.41% for 7–10 calls, 48.13% for 11–15 calls, 33.57% for 16–20 calls, and 26.46% for more than 20 calls. Accuracy is not monotonic in call count: examples that need more observations may also be intrinsically harder. The grouping is descriptive and does not estimate the causal effect of additional calls.

Panel (c) shows that operation use varies with the question. `MAP` and `GROUND_SET` are invoked on 100.0% and 86.4% of examples, respectively; `SOURCE_FRAME`, `MATCH_SET`,

and VIEW_TRANSFORM on 49.0%, 43.1%, and 38.8%; and MEASURE, SOLVE, and CAMERA_ROTATION on 11.2%, 10.9%, and 8.9%. This profile is consistent with question-conditioned execution, with some operations appearing more often than others; invocation frequency alone does not establish an operation's contribution to accuracy.

Accuracy by number of input images. The four bins in Figure 5 partition the same 3,031-example snapshot by the number of images supplied with each question: 981 one-image, 1,084 two-image, 403 three-image, and 563 four-or-more-image examples. The bars compare Direct and Full within each bin. SpatialCanvas has similar accuracy on one- and two-image questions, a larger gain on three-image questions, and no gain on the four-or-more-image bin. This is a descriptive breakdown by input size; it does not establish that image count alone causes the differences.

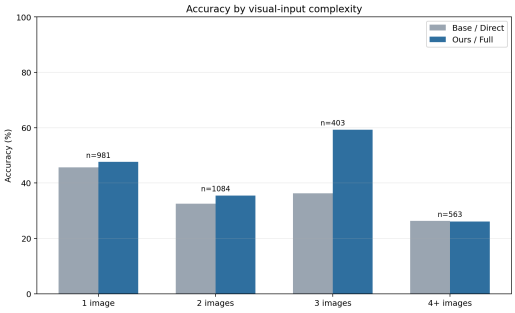


Figure 5: Direct and Full accuracy grouped by the number of input images. Labels above the groups give sample counts; all bins sum to 3,031 examples.