TEST-TIME REASONING WITH COMPUTATIONAL EVIDENCE FOR SPATIAL QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) have made substantial progress, while the growing interest in embodied AI underscores the need for reliable spatial reasoning. Existing approaches have attempted to improve spatial reasoning by enhancing visual reasoning capabilities or augmenting VLMs with external visual tools. However, visual reasoning can be unreliable when the VLM lacks sufficient spatial sensing capability. Although tool augmentation can potentially address this limitation, invoking a large set of tools or reconstructing an entire 3D scene is often computationally expensive, inefficient, and unnecessary. Our analysis reveals a capability imbalance: VLMs show strong object-level perception and reasoning but struggle with spatial sensing, suggesting that explicit geometric evidence can help bridge this gap. We propose *Test-Time Spatial Reasoning (TTSR)*, a training-free framework that supports spatial question answering through computational evidence. TTSR constructs a question-specific geometric plan, then grounds the entities and acquires the spatial information using the VLM and minimal auxiliary tools. It performs reference-frame transformations and compositional spatial computations, producing explicit evidence that the VLM interprets using its own reasoning capabilities to get the answer. Experiments on recently proposed challenging spatial reasoning benchmarks demonstrate that TTSR substantially outperforms existing state-of-the-art methods and generalizes consistently across diverse spatial reasoning settings.

1 INTRODUCTION

Vision-Language Models (VLMs) have achieved substantial progress, providing a strong foundation for multimodal understanding and interaction (Alayrac et al., 2022; Li et al., 2023; Liu et al., 2023; Radford et al., 2021). By jointly processing visual and textual information, VLMs enable users to query images through natural language and allow models to reason over visual observations. This capability is particularly important for agentic AI, where an agent must perceive its surroundings, understand the current context, and make decisions based on information from the real world (Driess et al., 2023; Brohan et al., 2023). These capabilities have the potential to support a wide range of real-world applications, including embodied intelligence, robotics, and autonomous systems.

With the emergence of embodied AI, spatial intelligence has become increasingly important for systems that must perceive and reason about the physical world (Chen et al., 2024). However, complex spatial reasoning remains challenging for current VLMs (Lee et al., 2026; Li et al., 2026b; Zhang et al., 2025). For example, answering a question such as "Who is the person closest to the left of the person wearing a blue T-shirt?" requires the model to first identify the reference person and then reason about the relative positions of other individuals in the scene. Reliable reasoning of this kind is important in high-stakes applications such as security, where incorrect spatial judgments may lead to erroneous decisions. It is also essential for robotic systems, which must accurately perceive their surroundings and reason about spatial relationships in order to interact effectively with real-world environments. These applications motivate the development of spatially aware VLMs with stronger capabilities for grounded perception and spatial reasoning.

Existing approaches improve spatial reasoning by enhancing VLM reasoning or introducing external geometric tools (Chen et al., 2024; Lee et al., 2025). However, our study shows that stronger reasoning alone is insufficient when the underlying geometric evidence is unreliable, as observed with

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.

Qwen3-VL (Yang et al., 2025). External tools such as Grounding DINO (Liu et al., 2024) and Depth Anything (Yang et al., 2024b) provide explicit geometric cues, but directly exposing these outputs to a VLM does not guarantee correct spatial reasoning, particularly under changing reference frames (Lee et al., 2025). These observations suggest that the key challenge is not only acquiring geometric information, but also transforming it into explicit evidence that can be reliably used for reasoning. In addition, Guo et al. (2026) relies on proprietary external models such as ChatGPT or Gemini, which may introduce additional inference cost.

Recently, multimodal spatial question answering involving multi-hop reasoning and perspective changes has emerged as an active research area. A growing number of benchmarks, including MultihopSpatial (Lee et al., 2026), ViewSpatial (Li et al., 2026b), SPHERE (Zhang et al., 2025), and so on, have been introduced to evaluate these increasingly challenging spatial reasoning capabilities. Results on these benchmarks consistently show that existing spatial reasoning methods still struggle to match the performance of leading closed-source models, even when extensively leveraging external perception and geometric tools. More importantly, existing methods are often designed to address only specific aspects of spatial reasoning, to jointly handle multi-hop reasoning, perspective changes, and visually grounded spatial inference is still a challenge.

In this paper, we investigate why existing VLMs fail on spatial reasoning tasks and how these failures can be addressed at test time. We consider three potential sources of failure: (1) insufficient 2D visual perception, which prevents models from identifying and localizing relevant objects. (2) limited reasoning capability, which prevents models from correctly composing spatial relations, and (3) insufficient 3D scene understanding, which prevents models from estimating the spatial structure. Through a series of controlled experiments, we find that current VLMs exhibit strong 2D perception and consistent logical reasoning capabilities. In contrast, their ability to estimate 3D scene structure is limited, with performance on our 3D perception diagnostics dropping significantly.

Inspired by these observations, we propose Test-Time Spatial Reasoning, a tool-augmented framework for spatial reasoning. Existing methods rely on multiple external tools such as SAM (Kirillov et al., 2023), Grounding DINO (Liu et al., 2024), and Depth Pro (Bochkovskiy et al., 2025). In contrast, our framework exploits the perception and reasoning abilities already available in the VLM. External tools are only used to provide complementary capabilities that the VLM lacks. We further find that simply appending tool outputs to the reasoning chain is not sufficient. The key is to convert the external evidence into a structured representation that is compatible with the model's reasoning process. To achieve this, we use 2.5D lifting to transform visual observations into explicit geometric evidence. We then construct executable spatial reasoning programs over this representation. These provide the VLM with grounded computational evidence for spatial inference.

Beyond the human-designed method, we also test an **AI-native research paradigm** through an Auto-Research experiment. We provide only a high-level research objective for autonomous experimentation. Across 15 trials, the Auto-Research process converges with our human design on several key ideas, including depth cues and prompt tuning. It further explores a broader set of techniques such as constraint decoding. However, these improvements are largely metric-driven and lack the deeper diagnostic analysis provided by human research. Despite this limitation, Auto-Research shows strong potential for exploring a broader design space and for more advanced ML problems.

We summarize our contributions as follows: (1) We systematically analyze the failure characteristics of a widely used open-source VLM and identify a clear capability imbalance: the model shows strong reasoning and object-level visual perception, but weaker spatial sensing, particularly in basic depth estimation. (2) Building on this analysis, we propose **Test-Time Spatial Reasoning (TTSR)**, a training-free framework that provides explicit computational evidence to the VLM while leveraging its reasoning capability. TTSR follows a planner-guided grounding and execution pipeline with minimal auxiliary tool usage. We evaluate TTSR on three recently proposed spatial question-answering benchmarks, where it substantially improves over existing methods.

2 Related Work

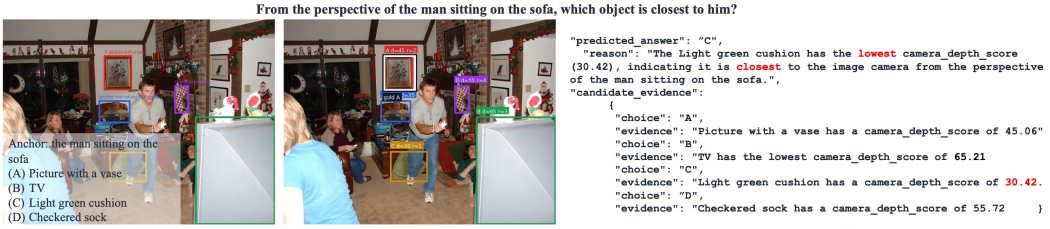
Visual and Tool Augmented Spatial Reasoning Prior work improves spatial reasoning through enhanced visual representations, grounded reasoning, or external tools. Visual CoT (Shao et al., 2024), ViGoRL (Sarch et al., 2026), LVR (Li et al., 2026a), TreeVGR (Wang et al., 2026a), and SpatialVLM (Chen et al., 2024) strengthen visual reasoning through region grounding, intermediate

representations, or spatial instruction tuning. Tool augmented methods introduce explicit geometric information: APC VLM (Lee et al., 2025) performs reference frame transformation, SpatialPIN (Ma et al., 2024) leverages 3D reconstruction, MSSR (Guo et al., 2026) iteratively curates geometric evidence, and GCA (Chen et al., 2026) executes geometry constrained computations. TTSR differs by planning the required operations first and converting only question relevant grounded observations into explicit computational evidence.

World models and scene exploration Other methods acquire additional viewpoints through generation or reconstruction. MindJourney (Yang et al., 2026) generates novel views, Think3D (Zhang et al., 2026) reconstructs 3D scenes and camera poses, and ViSA (Jha et al., 2025) combines world models with view verification. In contrast, TTSR avoids full scene exploration and computes only the spatial evidence required by the question. Due to the page limit, a detailed comparison is provided in Section A.

3 MOTIVATION: CAPABILITY BOUNDARY TESTING

In this section, we investigate why VLMs fall short on spatial question answering. To conduct a systematic analysis, we consider three possible sources of failure: **H1**: Limited object perception. VLMs may fail to reliably detect the entities required for answering the question. **H2**: Limited spatial sensing. VLMs may lack sufficient capability to perceive geometric information, such as depth. **H3**: Limited reasoning capability. VLMs may fail to correctly integrate the available perceptual and spatial evidence to derive the final answer. For each hypothesis, we present a case study based on a randomly selected example from the MultiHop Spatial benchmark. We follow conduct a systematic experiment to examine whether the observed failure pattern generalizes beyond the individual example. To ensure that our findings are broadly relevant and publicly reproducible, we use Qwen3-VL-8B (QWen3) (Bai et al., 2025) as the primary model in our analysis, as it represents a recent and capable open-source VLM whose behavior can be independently verified.



(a) H1 test (b) H2 test (c) H3 test

Figure 1: Case study for hypothesis testing. For **H1**, we prompt Qwen3-VL to predict the bounding boxes of the entities from the question and answer options, showing accurate object grounding. For **H2**, we conduct two independent runs in which Qwen3-VL estimates the depth of each candidate object, where d denotes the estimated distance from the camera and r denotes the corresponding depth rank. The results show limited depth accuracy and inconsistent estimates across the two runs. For **H3**, we provide the inferred depth information and ask Qwen3-VL to answer the question. The model associates the smallest depth values with the closest objects, showing effective reasoning over explicit spatial evidence. However, this reasoning remains in the camera coordinate frame, indicating that Qwen3-VL does not independently perform the coordinate transformation required for perspective-dependent reasoning.

Table 1: Results of QWen 3-VL-8B on the different capability boundary tests.

Experiment	Dataset	Comparison	Metric	Qwen3-VL	Comparison Performance
Object detection	COCO 2017 (Lin et al., 2014)	Faster R-CNN (Ren et al., 2016)	F1@IoU > 0.5	79.9	79.2
Reasoning	GSM8K (Cobbe et al., 2021)	DeepSeek-R1-8B (Guo et al., 2025)	Acc@Pass1	94.8	82.4
Reasoning	BBH Logical Deduction (Suzgun et al., 2023)	DeepSeek-R1-8B (Guo et al., 2025)	Acc@Pass1	94.0	89.5
Spatial	NYU Depth V2 (Silberman et al., 2012)	Depth Anything V2 Metric (Yang et al., 2024c)	Pairwise Ordering Acc.	80.8	98.0

3.1 QWEN3-VL HAS STRONG OBJECT PERCEPTION AND REASONING CAPABILITIES

We evaluate the object detection capability of Qwen3. Given a spatial question and its corresponding image, we ask the model to localize both the anchor entity and the target entities required by the question, as shown in Figure 1(a). Qualitatively, Qwen3 accurately identifies and localizes the relevant objects. We further conduct a systematic evaluation on COCO 2017 by prompting Qwen3

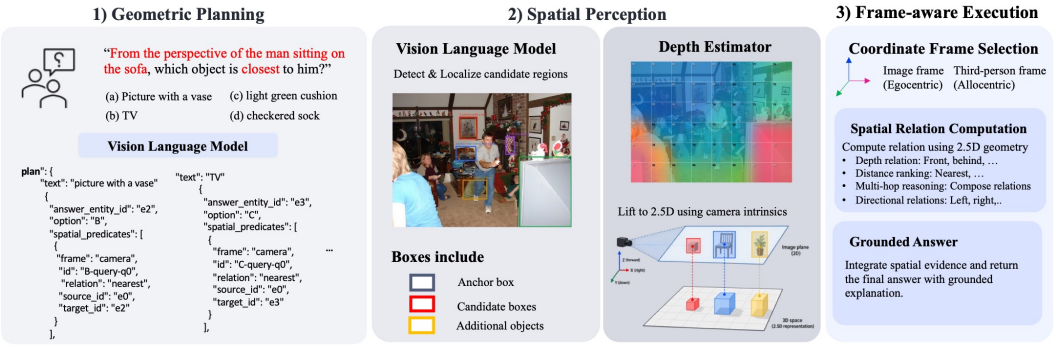


Figure 2: The framework consists of three modules. Geometric planning identifies the question specific entities and spatial evidence. Spatial perception grounds the relevant entities and extracts their 3D information. During execution, TTSR applies the required spatial transformations and computations to provide the VLM with explicit spatial evidence for the final prediction.

to predict bounding boxes for object classes. As a strong vision-based detection baseline, we evaluate Faster R-CNN. The results are reported in Table 1, showing that **H1:** Qwen3 achieves object localization performance comparable to the dedicated visual detector.

We observe a similar pattern when examining **H3**. Qwen3's reasoning procedure is largely consistent with the expected human reasoning process, as the closest is to find the smallest value in the evidence. To further quantify its reasoning capability, we evaluate Qwen3 on a logical deduction benchmark and compare it with DeepSeek-R1-8B, a model of comparable scale specifically designed for reasoning. We find that Qwen3 keeps strong reasoning capability in the textual modality.

3.2 Qwen3-VL shows Limited Spatial Sensing Capability

For **H2**, we find that Qwen3 has limited capability in recovering depth information from a single 2D image. Specifically, we randomly sample pairs of image locations from the NYU Depth dataset, and ask the model to determine their relative depth ordering. As shown in Table 1, Qwen3 shows a substantial performance gap compared with specialized depth estimation models, indicating limited capability in perceiving depth-dependent spatial relationships. We further evaluate Qwen3 under a zero-shot setting by asking it to estimate depth over entity and rank the entity based on depth, as shown in Figure 1(b). The resulting depth predictions remain inaccurate and inconsistent across two independent runs. This suggests that the model lacks reliable spatial sensing capability for supporting spatial question answering. In addition, Qwen3 has limited capability in perspective transformation, consistent with the observations reported by (Lee et al., 2025).

Takeaway. Under our diagnostic protocols, Qwen3-VL-8B performs competitively on object localization and text-only reasoning, while remaining substantially less accurate than a specialized depth estimator on relative-depth judgments.

4 Test-Time Spatial Reasoning with Computational Evidence

Overview Spatial reasoning requires both reliable visual perception and a reasoning procedure that integrates grounded visual evidence to derive the prediction. Motivated by the preliminary findings in Sections 3, we introduce Test-Time Spatial Reasoning (TTSR), a training-free framework that combines query-conditioned visual grounding with spatial computation. TTSR leverages the intrinsic object detection capabilities of the VLM, supplemented by minimal tool usage for spatial sensing, depending on computation to resolve spatial relations that require strong geometric reasoning. The overview of our method is presented in Figure 2. Given a question q, image $\mathcal{I}$, and a set of candidate answers $\mathcal{A}$, our proposed method first constructs a semantic reasoning plan, then acquires visual evidence, and evaluates supported constraints. Throughout the entire procedure, all VLM and auxiliary tool models remain frozen.

4.1 Geometry Planning

Spatial questions specify the entities of interest, the spatial constraints among them, and the comparisons required to derive an answer. To reduce the bias from the image and faithfully answer the question, TTSR determines the entities and spatial relations that are required to answer the question.

This separates question interpretation from image-dependent cues and generates a question-specific execution plan:

$$\Pi_q = \big(\mathcal{E}_q, \mathbf{O}_q, \prec_q\big), \tag{1}$$

where $\mathcal{E}_q = \{e_1, \ldots, e_M\}$ contains symbolic entity descriptions, $\mathcal{O}_q$ specifies operations, and $\prec_q$ specifies the prerequisites of a comparison. The operations $\mathcal{O}_q$ include visual filtering, spatial filtering, and comparison. The entity description is used to plan the entity to be located. $\prec_q$ is designed to efficiently perform multi-hop reasoning, where the dependency order needs to be preserved in the compositional structure of a question. For example, in *Which red object to the right of the table is closest to the person?*, the question is decomposed using color and position to establish a candidate pool, and then the distance is compared among eligible objects. $\mathbf{O}_q$ is used to perform color filtering and a position comparison.

To establish a spatial relationship among entities, we use a spatial relation representation $\rho = (e_a, r, e_i, f)$, denoting a relation r between an anchor e_a and a target e_i, interpreted in reference coordinate frame f. The anchor specifies the origin of comparison, and the frame specifies the coordinate used in the question. The decomposition preserves the reference frame and the scope of each constraint. For example, *left of the person in the image*, *on the left side of the image* induce different spatial conditions with different reference frames.

4.2 Entity grounding and reference-frame geometry

To make the response trustworthy, the second step is to ground symbolic entities in Π_q. Motivated by the findings in Section 3.1, which demonstrate the strong capability of VLMs to detect and localize entities in an image, we use the VLM to ground the entities specified by $\mathcal{E}_q$. Specifically, for each e_i, where $i \in 1, \ldots, M$, we apply the VLM to localize the entity and obtain its bounding box $b_i = (x_i^1, y_i^1, x_i^2, y_i^2)$, where (x_i^1, y_i^1) and (x_i^2, y_i^2) denote the pixel coordinates of the top-left and bottom-right corners, respectively. For efficient inference, we represent each entity by (u_i, v_i), where $u_i = \frac{x_i^1 + x_i^2}{2}$ and $v_i = \frac{y_i^1 + y_i^2}{2}$, corresponding to the median coordinates of its bounding box. To support questions involving perspective changes, we estimate the orientation of each entity by using the VLM to infer its body azimuth θ_i in a zero-shot manner.

A 2D bounding box is insufficient for answering spatial questions, reconstructing the entire scene in 3D is computationally costly, inefficient, and unnecessary. To provide the required spatial information, we lift each grounded entity into a compact 2.5D representation with minimal tool usage. Specifically, we use Depth Anything V2 (Yang et al., 2024b) to estimate a dense depth map over the image. For each grounded entity e_i, we take the median depth within the region defined by its bounding box and denote it as d_i, which represents the depth of object i. The image-plane center (u_i, v_i) does not determine the spatial position of an object, because its location also depends on depth. Two objects with the same displacement in image coordinates can correspond to different lateral distances in 3D if they lie at different depths. We combine (u_i, v_i) with the estimated depth d_i and apply pinhole-camera back-projection to recover the spatial coordinates of each entity in the camera frame (Tsai, 1987)

$$\begin{gathered} \mathbf{p}_{e_i} = d_i \mathbf{K}^{-1} [u_i, v_i, 1]^\top, \\ \mathbf{K} = \begin{bmatrix} f_k & 0 & \frac{W}{2} \\ 0 & f_k & \frac{H}{2} \\ 0 & 0 & 1 \end{bmatrix}, \qquad f_k = \frac{W}{2\tan(\phi/2)}, \end{gathered} \tag{2}$$

where W and H denote the width and height of image $\mathcal{I}$, respectively, and ϕ denotes the configured horizontal field of view. $\mathbf{P}_{e_i}$ represents the homogeneous coordinates of entity e_i. $\mathbf{K}$ is the intrinsic parameter of a camera. All entities in an image share a common camera model and depth scale.

Reference-Frame Transformation The reference frame f in the planned tuple determines how the grounded entities are compared. Inspired by rigid-body transformations in robot kinematics, we perform reference-frame transformations to provide computational evidence for perspective-dependent spatial questions. For a reference frame f associated with the question, we follow the convention that the camera coordinate system uses $+X$ to the right, $+Y$ downward, and $+Z$ into the scene. The corresponding homogeneous transformation matrix is defined as:

$$\mathbf{T}_{e_i}^{f} = \begin{bmatrix} \mathbf{R}_{e_i}^{f} & \mathbf{t}_{e_i}^{f} \\ \mathbf{0} & 1, \end{bmatrix} \tag{3}$$

where the rotation matrix $\mathbf{R}_{e_i}^{f}$ maps from coordinate f to the camera coordinates, and $\mathbf{t}_{e_i}^{f}$ is the displacement of f in camera coordinate. When the question requires reasoning from an entity's perspective (Third person perspective), the VLM estimates the orientation of that entity. For camera-relative relations, we use the camera-aligned reference frame, such that $\mathbf{R}_{e_i}^{f} = \mathbf{I}_3$. For entity-relative relations, the corresponding rotation matrix is obtained from the estimated entity orientation as:

$$\mathbf{R}_{e_i}^{f} = \begin{bmatrix} \cos\theta_i & 0 & \sin\theta_i \\ 0 & 1 & 0 \\ -\sin\theta_i & 0 & \cos\theta_i \end{bmatrix}. \tag{4}$$

Here, the entity-relative frame uses its positive x-axis for the entity's right, its positive y-axis for downward, and its positive z-axis for the entity's facing direction. The rotation matrix maps vectors from this local frame to the camera frame. Accordingly, $\theta_i = 0$ corresponds to the entity facing along the camera's positive depth axis (away from the camera), whereas $\theta_i = \pi$ corresponds to the entity facing toward the camera. For $(e_a, r, e_i, f) \in \mathbf{O}_q$, the anchor entity e_a determines the origin, the planned reference frame f determines the coordinate system. The relative displacement of entity e_i from the anchor position, expressed in reference frame f, is computed as:

$$\mathbf{\Delta}_a^{e_i} = [\mathbf{I}_3 \quad \mathbf{0}] \left(\mathbf{T}_{e_a}^{f}\right)^{-1} \begin{bmatrix} \mathbf{p}_{e_i} \\ 1 \end{bmatrix}. \tag{5}$$

This gives the displacement of entity e_i relative to the anchor e_a, expressed in the reference frame f. Each component of this displacement corresponds to the relative position of the target along one axis of f. Therefore, the planned relation (e_a, r, e_i, f) is converted into geometric evidence that can be directly evaluated. We use the signs and magnitudes of these displacement components to determine whether the spatial relation r is satisfied.

4.3 Compositional and computational spatial execution

During the execution stage, TTSR applies the operations in $\mathbf{O}_q$ to the grounded entity information from Section 4.2, and $\prec_q$ specifies the dependency order among the constraints. This design converts the grounded geometric information into explicit evidence that supports the answer candidates and can be interpreted by the VLM.

Direction templates are defined in the reference frame f specified by the planned relation. Let $\mathbf{e}_j$ denote the j-th standard basis vector in this frame. Right, left, up, and down correspond to $\mathbf{e}_1$, $-\mathbf{e}_1$, $-\mathbf{e}_2$, and $\mathbf{e}_2$, respectively. The interpretation of front depends on the reference frame: for camera-relative relations, "in front of" denotes a position nearer to the camera; for entity-relative relations, it denotes a position ahead of the reference entity along its facing direction. Accordingly,

$$\mathbf{v}_{\text{front}} = \begin{cases} -\mathbf{e}_3, & \text{camera-relative,} \\ \mathbf{e}_3, & \text{entity-relative.} \end{cases}$$

Thus, the planner selects $\mathbf{v}_r$ according to both the relation r and the reference frame f, for example, $\mathbf{v}_r = \mathbf{v}_{\text{front}}$. We compute the directional evidence as follows:

$$h_r(a, e_i, f) = \frac{\mathbf{v}_r^{\top} \mathbf{\Delta}_a^{e_i}}{d_a + \varepsilon}, \tag{6}$$

where $d_a = \text{median}\{D(u,v) \mid (u,v) \in b_a\}$ denotes the estimated depth of the anchor entity e_a. We normalize the spatial evidence by the anchor depth d_a, yielding a dimensionless measure of relative displacement. A positive $h_r(\cdot) > \tau$ indicates that the target entity e_i has a positive displacement component along the direction specified by relation r in reference frame f. ϵ is a stabilizer. For simplicity, we set $\tau = 0.08$.

For distance relations, such as *nearest* and *farthest*, we use the magnitude of the relative displacement as the computational evidence and compare it across candidate entities. For each eligible

candidate e_i, we compute

$$s_i = \frac{\|\mathbf{\Delta}_a^{e_i}\|_2}{d_a + \varepsilon}. \tag{7}$$

For each answer candidate, TTSR collects the measurements associated with the claims specified in the execution plan. Each claim is represented by its numerical evidence together with a corresponding verdict. For directional relations, such as *right*, the evidence is the signed displacement component along the queried direction. For distance-based relations, such as *nearest*, the evidence consists of the relative distance and the margin with respect to the competing candidates. These measurements summarize the quantities derived from $\mathbf{\Delta}_a^e$ in a question-specific form that can be directly interpreted by the VLM.

5 Experiments

In this section, we evaluate TTSR by addressing the following research questions: **RQ1.** How effectively does TTSR improve performance on spatial reasoning tasks? **RQ2.** How well does TTSR generalize across different VLM backbones? **RQ3.** How effectively can an autonomous research platform discover solutions to the same spatial reasoning problem? **RQ4.** Which components of TTSR are most critical for improving the spatial reasoning capability of VLMs?

Datasets. We evaluate all comparison methods on three recently proposed spatial reasoning QA benchmarks that include challenging multi-hop reasoning and perspective-change questions: the MultiHop Spatial benchmark (Lee et al., 2026), ViewSpatial (Li et al., 2026b), and SPHERE (Zhang et al., 2025). We report fine-grained accuracy results for each benchmark .

Comparison methods. We compare TTSR with methods from three categories: closed-source models, visual reasoning methods, and spatial-specific reasoning methods. For visual reasoning methods, we include Latent Visual Reasoning (LVR-16) Li et al. (2026a) and Traceable evidence enhanced visual grounded reasoning (TreeVGR) Wang et al. (2026a). For spatial-specific methods, we include Abstract Perspective Change (APC-VLM) Lee et al. (2025), Spatial VLM Chen et al. (2024), SpatialPIN Ma et al. (2024), and MSSR Guo et al. (2026). To ensure a fair comparison, we evaluate all methods using their officially released or reproduced implementations on the three benchmarks. For each method, we keep its original backbone, such as Qwen3-VL-8B, Qwen2.5-VL-7B (Yang et al., 2024a). We report the performance gain of each method relative to its corresponding backbone. For MSSR, which calls the Gemini API, we report the original paper results for the configuration using the same backbone as ours in ViewSpatial.

Auto-Research Experiments. Before presenting the main experimental results, we investigate whether a state-of-the-art generative AI coding agent can independently address the same spatial reasoning problem, and how far such an automated research process can progress relative to human-designed approaches. Motivated by this question, we use a coding agent to develop a spatial reasoning method based on Qwen3-VL-8B. Our design principle is to minimize human intervention and prevent the agent from exploiting shortcuts or accessing information outside the designated experimental environment. Due to space constraints, we defer the complete Auto-Research procedure and discussion to Section B, with implementation details provided in Section G. The method produced by the Auto-Research process is evaluated on the same benchmark dataset as our methods. In the result tables, we denote this method as GPT-5.6 AR (OpenAI, 2026).

5.1 Main Results

RQ1: Our proposed method achieves substantial performance gains over state-of-the-art methods. We report the results in Tables 2 and 3. For each benchmark, we report the overall performance and use *Bench Avg.* to summarize the average performance across the three benchmarks. Although all three datasets evaluate general spatial reasoning capability, each benchmark emphasizes different reasoning skills.

On MultiHop Spatial, Gemini 2.5 Flash (Google, 2025) performs best among closed-source models, but its accuracy drops to 38.7% on 3 hop questions, showing the challenge of compositional spatial reasoning. Existing reasoning methods such as TreeVGR and LVR 16 provide only limited gains over their backbones. In contrast, TTSR uses explicit planning and compositional spatial execution to support attribute filtering and multi hop reasoning, improving overall accuracy by 23.9% over its

Table 2: Results on the Multihop Spatial and ViewSpatial benchmarks. **Bold** and underlined values denote the best and second-best results, respectively. For spatial-reasoning methods, superscripts indicate the backbone VLM: Q3-8B, Q2.5-7B, and Q2.5-3B denote Qwen3-VL-8B, Qwen2.5-VL-7B, and Qwen2.5-VL-3B, respectively. Values in parentheses report absolute performance changes over the corresponding backbone baseline. green and red indicate improvements and degradations, respectively. Light row shading indicates methods sharing the same backbone.

Method	*Multi-Hop Spatial*				*ViewSpatial*					
	1-Hop	2-Hop	3-Hop	**Overall**	Rel. Dir. (C)	Obj. Ori. (C)	Obj. Ori. (P)	Rel. Dir. (P)	Sec. Sim.	**Overall**
Closed-Source Models										
GPT-4o	55.0	42.3	31.5	42.9	41.5	19.6	43.0	40.9	26.8	35.0
Claude Opus 4.5	52.1	45.9	37.1	45.1	–	–	–	–	–	–
Gemini 2.5 Flash	61.9	51.3	38.7	50.6	**59.0**	23.6	52.8	**44.5**	**31.4**	44.3
Backbone Baselines										
Qwen2.5-VL-3B	38.5	35.1	28.6	34.0	43.7	31.8	42.5	32.8	27.0	36.6
Qwen2.5-VL-7B	43.0	39.2	26.4	36.2	44.2	28.0	38.0	33.6	29.1	35.8
Qwen3-VL-8B	46.9	36.0	26.4	36.4	52.7	29.3	46.9	37.2	26.7	40.3
Open-Source Spatial-Reasoning Methods										
Spatial VLM$^{Q2.5\text{-}3B}$	41.3 (+2.8)	35.1 (+0.0)	28.9 (+0.3)	35.1 (+1.1)	41.1 (-2.6)	31.0 (-0.8)	55.2 (+12.7)	33.0 (+0.2)	28.6 (+1.6)	38.2 (+1.6)
SpaceOM$^{Q2.5\text{-}3B}$	38.7 (+0.2)	35.3 (+0.2)	27.7 (-0.9)	33.9 (-0.1)	37.6 (-6.1)	24.6 (-7.2)	52.9 (+10.4)	39.8 (+7.0)	27.3 (+0.3)	36.3 (-0.3)
LVR-16$^{Q2.5\text{-}7B}$	40.7 (-2.3)	41.2 (+2.0)	31.8 (+5.4)	37.9 (+1.7)	45.9 (+1.7)	24.1 (-3.9)	38.1 (+0.1)	38.4 (+4.8)	27.1 (-2.0)	36.0 (+0.2)
TreeVGR$^{Q2.5\text{-}7B}$	44.5 (+1.5)	40.4 (+1.2)	30.8 (+4.4)	38.6 (+2.4)	46.1 (+1.9)	29.0 (+1.0)	44.1 (+6.1)	34.6 (+1.0)	24.0 (-5.1)	36.8 (+1.0)
APC-VLM$^{Q3\text{-}8B}$	31.5 (-15.4)	26.7 (-9.3)	25.9 (-0.5)	28.0 (-8.4)	47.9 (-4.8)	19.7 (-9.6)	64.0 (+17.1)	37.4 (+0.2)	26.4 (-0.3)	40.1 (-0.2)
MSSR$^{Q3\text{-}8B}$	47.1 (+0.2)	33.5 (-2.5)	27.8 (+1.4)	36.1 (-0.3)	–	–	–	–	–	46.5 (+6.2)
GCA$^{Q3\text{-}8B}$	46.8 (-0.1)	34.3 (-1.7)	31.9 (+5.5)	37.7 (+1.3)	47.5 (-5.2)	15.0 (-14.3)	58.3 (+11.4)	40.7 (+3.5)	20.5 (-6.2)	37.5 (-2.8)
GPT-5.6 AR$^{Q3\text{-}8B}$	49.3 (+2.4)	36.7 (+0.7)	27.4 (+1.0)	37.8 (+1.4)	53.5 (+0.8)	27.7 (-1.6)	46.6 (-0.3)	37.7 (+0.5)	30.3 (+3.6)	41.0 (+0.7)
Ours$^{Q3\text{-}8B}$	**74.1** (+27.2)	**61.3** (+25.3)	**45.6** (+19.2)	**60.3** (+23.9)	57.0 (+4.3)	**40.6** (+11.3)	**74.7** (+27.8)	38.5 (+1.3)	28.5 (+1.8)	**48.9** (+8.6)

Table 3: Results on the SPHERE benchmark. **Bold** and underlined values denote the best and second-best results, respectively. **Bench Avg.** reports the average Overall performance across Multi-Hop Spatial, ViewSpatial, and SPHERE.

Method	*Single Skill*				*Multi-Skill*			*Reasoning*		**Overall**	**Bench Avg.**
	Pos.	Count	Dist.	Size	P+C	D+C	D+S	Occl.	Manip.		
Closed-Source Models											
GPT-4o	71.7	76.6	71.3	89.4	54.4	62.7	58.8	66.0	63.3	68.9	48.9
Gemini 2.0 Flash	68.6	82.1	75.2	86.9	50.3	52.5	49.7	48.0	49.2	63.4	–
Gemini 2.5 Flash	74.2	**85.6**	**83.2**	86.9	**68.6**	**74.1**	**68.8**	58.5	62.8	**73.8**	56.2
Backbone Baselines											
Qwen2.5-VL-3B	57.9	69.9	60.2	84.2	59.4	46.7	50.6	61.3	55.2	60.7	43.8
Qwen2.5-VL-7B	65.2	79.1	70.5	86.0	58.6	61.4	44.8	54.5	49.2	63.5	45.2
Qwen3-VL-8B	64.6	84.6	61.4	83.0	48.5	51.9	51.1	58.2	60.7	63.3	46.7
Open-Source Spatial-Reasoning Methods											
Spatial VLM$^{Q2.5\text{-}3B}$	61.0 (+3.1)	73.6 (+3.7)	60.9 (+0.7)	79.8 (-4.4)	59.2 (-0.2)	48.1 (+1.4)	51.5 (+0.9)	56.1 (-5.2)	51.7 (-3.5)	60.6 (-0.1)	44.6 (+0.8)
SpaceOM$^{Q2.5\text{-}3B}$	57.1 (-0.8)	57.8 (-12.1)	52.9 (-7.3)	53.5 (-30.7)	54.8 (-4.6)	39.2 (-7.5)	43.2 (-7.4)	60.9 (-0.4)	56.2 (+1.0)	53.5 (-7.2)	41.2 (-2.6)
LVR-16$^{Q2.5\text{-}7B}$	54.8 (-10.4)	67.4 (-11.7)	50.9 (-19.6)	63.5 (-22.5)	36.0 (-22.6)	34.2 (-27.2)	33.4 (-11.4)	55.8 (+1.3)	53.8 (+4.6)	51.0 (-12.5)	41.6 (-3.6)
TreeVGR$^{Q2.5\text{-}7B}$	67.6 (+2.4)	81.1 (+2.0)	69.1 (-1.4)	84.5 (-1.5)	56.8 (-1.8)	39.2 (-22.2)	40.4 (-4.4)	57.2 (+2.7)	54.8 (+5.6)	62.3 (-1.2)	45.9 (+0.7)
APC-VLM$^{Q3\text{-}8B}$	71.6 (+7.0)	84.3 (-0.3)	55.5 (-5.9)	51.1 (-31.9)	48.2 (-0.3)	51.8 (-0.1)	41.8 (-9.3)	49.5 (-8.7)	49.2 (-11.5)	57.4 (-5.9)	41.8 (-4.9)
MSSR$^{Q3\text{-}8B}$	50.6 (-14.0)	–	58.3 (-3.1)	47.9 (-35.1)	–	–	49.5 (-1.6)	51.2 (-7.0)	42.6 (-18.1)	–	–
GCA $^{Q3\text{-}8B}$	**75.0** (+10.4)	81.7 (-2.9)	81.7 (+20.3)	71.5 (-11.5)	41.2 (-7.3)	56.7 (+4.8)	33.3 (-17.8)	57.1 (-1.1)	65.6 (+4.9)	64.2 (+0.9)	46.5 (-0.2)
GPT-5.6 AR$^{Q3\text{-}8B}$	71.0 (+6.4)	83.6 (-1.0)	78.1 (+16.7)	**89.5** (+6.5)	63.9 (+15.4)	67.7 (+15.8)	51.2 (+0.1)	**69.3** (+11.1)	**67.2** (+6.5)	71.5 (+8.2)	50.1 (+3.4)
Ours$^{Q3\text{-}8B}$	65.8 (+1.2)	84.4 (-0.2)	73.9 (+12.5)	88.7 (+5.7)	65.0 (+16.5)	66.6 (+14.7)	55.6 (+4.5)	57.6 (-0.6)	60.6 (-0.1)	68.6 (+5.3)	**59.3** (+12.6)

backbone and achieving the highest 3 hop accuracy. On ViewSpatial, which emphasizes perspective and reference frame changes, general visual reasoning methods show limited improvement. APC explicitly transforms a coarse 3D scene into the queried reference frame, but the final relation is still inferred by the VLM. MSSR combines 3D reconstruction with an agentic pipeline and uses Gemini 2.5 Pro for orientation estimation, introducing reliance on a proprietary model. TTSR instead transforms grounded geometry into the target reference frame and directly computes relation specific spatial evidence, achieving the best performance without closed source API calls.

SPHERE evaluates diverse spatial skills, including counting, position, distance, and size. Gemini 2.5 Flash shows balanced performance, while latent visual reasoning methods may degrade accuracy on basic tasks and spatial-specific methods can hurt non-geometric skills such as counting. TTSR achieves consistent improvements across categories, with large gains in distance and size estimation. The method discovered by the GPT-5.6 Sol Auto-Research platform also performs strongly on SPHERE, despite not being explicitly instructed to optimize for this benchmark. **RQ3. Auto-Research shows strong potential for metric-driven optimization.** It autonomously identified SPHERE as a relevant target and iteratively improved performance. These results also highlight

a tradeoff: TTSR is most effective when explicit constraint composition and geometric execution are required, whereas on heterogeneous tasks in SPHERE, additional planning and grounding can introduce unnecessary failure modes.

5.2 Ablation Study

In this section, we first validate the key design choices of TTSR and then study the sensitivity of final performance to different VLM backbones and auxiliary models. All ablation experiments are conducted on the MultiHop Spatial test set Lee et al. (2026).

Design ablation. We first evaluate the contribution of the computational spatial evidence in TTSR. Specifically, we only remove the directional evidence defined in Eq. (6) and (7) and provide the VLM with the raw geometric quantities computed in Eq. (5). This ablation tests whether explicitly converting grounded geometric quantities into relation specific directional evidence, such as translating a directional relation to the sign of the corresponding scalar component, improves the final prediction. We isolate the effect of the explicit directional evidence from the underlying geometric representation itself.

Second, we remove the question-conditioned attribution filtering in $\mathcal{O}_q$. In this setting, computational evidence for all candidate entities is provided to the VLM without applying the constraints specified by the execution plan. This ablation evaluates whether the structured filtering and compositional execution are necessary compared to simply providing a complete set of geometric evidence. The results are reported in the first three rows of Table 4.

Table 4: Result of the ablation study.

Ablation	Method	1-Hop	2-Hop	3-Hop	Acc.
Design	Base	46.9	36.0	26.4	36.4
	w/ Direct evidence	51.3 (+4.4)	43.5 (+7.5)	31.3 (+4.9)	42.0 (+5.6)
	w/ Attribute Filtering	72.7 (+25.8)	50.5 (+14.5)	40.6 (+14.2)	54.6 (+18.2)
	Full	74.1 (+27.2)	61.3 (+25.3)	45.6 (+19.2)	60.3 (+23.9)
	+ Moge2ViT-s	**75.3** (+28.4)	**61.3** (+25.3)	**45.9** (+19.5)	**60.8** (+24.4)
Backbone	Qwen2.5-VL	43.0	39.2	26.4	36.2
	Ours + Qwen2.5-VL	54.8 (+11.8)	48.3 (+9.1)	35.0 (+8.6)	46.0 (+9.8)
	InternVL-3.5-4B	49.1	39.4	30.6	39.7
	Ours + InternVL	62.3 (+13.2)	51.0 (+11.6)	38.9 (+8.3)	50.7 (+11.0)

Motivated by Section 3, which identifies depth sensing as a vital limitation of Qwen3-VL, we further study the role of the auxiliary depth estimator. We replace the Depth Anything 2 model with Moge2Vit-s (Wang et al., 2026b) and measure the changes in spatial reasoning performance.

RQ4: we observe that direct computational spatial evidence plays an important role in accurate prediction. Without this step, performance drops and approaches that of the baseline model, supporting the effectiveness of our pipeline design. Removing attribute grounding and filtering has only a limited effect on 1 hop questions, but leads to larger performance degradation on 2 hop and 3 hop questions, where each additional hop introduces another grounding step. This suggests that grounding and filtering become increasingly important as the reasoning chain becomes more compositional. We further observe that using different depth estimation models has only a limited impact on the final prediction, indicating that TTSR is relatively robust to the choice of depth estimator.

Backbone ablation. We evaluate TTSR with different backbone VLMs, including Qwen2.5-VL-7B and InternVL3.5 4B, to assess whether the proposed framework generalizes across model architectures. The results are reported in Table 4. **RQ2: TTSR is applicable across different VLM backbones.** TTSR consistently improves performance with both Qwen2.5-VL-7B and InternVL3.5, yielding overall accuracy gains of 9.8% and 11.0%, respectively. However, the improvement is substantially larger with Qwen3-VL-8B, where TTSR achieves a gain of more than 23%. We hypothesize that this difference is partly attributable to the visual perception capability of the backbone model. This interpretation is consistent with the observations of (Park et al., 2026), who report a performance gap of up to approximately 10% between the Qwen3-VL and Qwen2.5-VL on object detection tasks. These results suggest that stronger visual grounding capabilities can further amplify the benefit of explicit spatial reasoning in TTSR.

6 Conclusion

In this paper, we present TTSR, a test-time spatial reasoning framework for VLMs. We first systematically characterize the capability boundaries of existing VLMs on the fundamental skills required for spatial reasoning and diagnose the key sources of their failures. Motivated by these findings, TTSR integrates a geometry-aware planner, grounded perception through targeted tool use, and explicit computational spatial evidence to support compositional reasoning at test time. Experiments demonstrate that TTSR substantially outperforms prior state-of-the-art methods across spatial reasoning benchmarks while remaining broadly applicable to different VLM backbones.

AI USE STATEMENT

In this work, we used generative AI tools for literature search, coding and testing assistance, and language polishing. We did not use generative AI tools to develop the motivation, research ideas, or proposed methods, to draft the manuscript, or to generate figures or plots, except for Figure 9. All AI-assisted code was carefully reviewed and validated by human researchers. Specifically, we performed unit tests at the module level and manually audited the code through researcher-led code review. Before the human review, we conducted two rounds of AI-agent code review. The implementation was independently tested in four rounds and passed all designed tests. For literature search, generative AI was used only to help identify relevant paper titles. Researchers independently read the original papers. For Auto-Research, the entire method and experiment are conducted by the coding agent, human verify any data leakage or unfair comparison. No scientific claims or interpretations were derived from AI-generated summaries. The authors carefully reviewed all AI-assisted outputs and take full responsibility for the accuracy, integrity, and final content of this work.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

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.

Alexey Bochkovskiy, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *International Conference on Learning Representations*, volume 2025, pp. 75602–75637, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

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.

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.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Google. Gemini 2.5 flash. `https://ai.google.dev/gemini-api/docs/models/gemini-2.5-flash`, 2025. Accessed: 2026-09-25.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Yejie Guo, Yunzhong Hou, Wufei Ma, Meng Tang, and Ming-Hsuan Yang. Pursuing minimal sufficiency in spatial reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 121192–121222, 2026.

Saurav Jha, M Jehanzeb Mirza, Wei Lin, Shiqi Yang, and Sarath Chandar. Probing the effectiveness of world models for spatial reasoning through test-time scaling. *arXiv preprint arXiv:2512.05809*, 2025.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

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

Youngwan Lee, Soojin Jang, Yoorhim Cho, Seunghwan Lee, Yong-Ju Lee, and Sung Ju Hwang. Multihopspatial: Multi-hop compositional spatial reasoning benchmark for vision-language model. *arXiv preprint arXiv:2603.18892*, 2026.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 148076–148090, 2026a.

Dingming Li, Hongxing Li, Zixuan Wang, Yuchen Yan, Hang Zhang, Siqi Chen, Guiyang Hou, Shengpei Jiang, Wenqi Zhang, Yongliang Shen, et al. Viewspatial-bench: Evaluating multi-perspective spatial localization in vision-language models. In *European Conference on Computer Vision*, pp. 95–111. Springer, 2026b.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PmLR, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024.

Chenyang Ma, Kai Lu, Ta-Ying Cheng, Niki Trigoni, and Andrew Markham. Spatialpin: Enhancing spatial reasoning capabilities of vision-language models through prompting and interacting 3d priors. *Advances in neural information processing systems*, 37:68803–68832, 2024.

OpenAI. GPT-5.6 System Card. `https://deploymentsafety.openai.com/gpt-5-6`, July 2026. Published July 9, 2026.

Chanho Park, Daehyeon Choi, Jihyun Lee, and Minhyuk Sung. Retrieval heads meet vision: Uncovering how vlms locate and extract visual information. *arXiv preprint arXiv:2608.27417*, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

Gabriel Sarch, Snigdha Saha, Naitik Khandelwal, Ayush Jain, Michael Tarr, Aviral Kumar, and Katerina Fragkiadaki. Grounded reinforcement learning for visual reasoning. *Advances in Neural Information Processing Systems*, 38:150977–151013, 2026.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pp. 746–760. Springer, 2012.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023.

Roger Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, 1987.

Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Sule Bai, Zijian Kang, Jiashi Feng, Zhuochen Wang, et al. Traceable evidence enhanced visual grounded reasoning: Evaluation and method. In *International Conference on Learning Representations*, volume 2026, pp. 148769–148794, 2026a.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *Advances in Neural Information Processing Systems*, 38:35928–35959, 2026b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

An Yang et al. Qwen2 technical report, 2024a. URL https://arxiv.org/abs/2407.10671.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024b.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in neural information processing systems*, 37:21875–21911, 2024c.

Yuncong Yang, Jiageng Liu, Zheyuan Zhang, Siyuan Zhou, Reuben Tan, Jianwei Yang, Yilun Du, and Chuang Gan. Mindjourney: Test-time scaling with world models for spatial reasoning. *Advances in Neural Information Processing Systems*, 38:109855–109885, 2026.

Wenyu Zhang, Wei En Ng, Lixin Ma, Yuwen Wang, Junqi Zhao, Allison Koenecke, Boyang Li, and Lu Wang. Sphere: Unveiling spatial blind spots in vision-language models through hierarchical evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11591–11609, 2025.

Zaibin Zhang, Yuhan Wu, Lianjie Jia, Yifan Wang, Zhongbo Zhang, Yijiang Li, Binghao Ran, Fuxi Zhang, Zhuohan Sun, Yizhuang Peng, et al. Think3d: Thinking with space for spatial reasoning. *arXiv preprint arXiv:2601.13029*, 2026.

A Complete Related work

Existing approaches to spatial question answering can be broadly organized into three directions: methods that enhance spatial reasoning within the vision-language model, and methods that augment VLMs with external tool computation, and world model and generative methods.

A.1 Visual Reasoning Methods

One line of work improves spatial reasoning by introducing intermediate representations into the reasoning process. Visual CoT (Shao et al., 2024) incorporates bounding-box supervision and multi-turn reasoning to progressively attend to image regions relevant to the question. ViGoRL (Sarch et al., 2026) uses reinforcement learning to ask the model to inspect question-related regions through visual feedback such as zooming. LVR (Li et al., 2026a) introduces latent visual states into the reasoning trajectory by reconstructing visual tokens. TreeVGR (Wang et al., 2026a) improves traceability by grounding intermediate reasoning steps in localized visual evidence. SpatialVLM (Chen et al., 2024) constructs large-scale spatial question-answering data and trains VLMs to better answer qualitative and quantitative spatial questions. These approaches improve internal visual representations and reasoning traces, but the information required for spatial reasoning is not evaluated or grounded. In contrast, our method converts grounded visual observations into explicit numerical spatial evidence.

A.2 Spatial Reasoning with tool calling

An alternative direction augments VLMs with external perception tools. APC-VLM (Lee et al., 2025) constructs a sparse 3D abstraction containing object positions and orientations, transforms the scene into the queried reference frame, and provides the transformed coordinates to the VLM for reasoning. SpatialPIN (Ma et al., 2024) uses 3D foundation-model priors through partial scene reconstruction, pose estimation, and object geometry, providing richer 3D representations to support spatial reasoning. In contrast to providing reconstructed geometry, our method directly converts question-relevant observations into computational evidence and composes such evidence according to the spatial relations required by the question. More recent methods further integrate tool use with agentic reasoning. MSSR (Guo et al., 2026) uses both of perception and reasoning agents to iteratively acquire geometric evidence, discard irrelevant evidence, and maintain a minimally sufficient evidence set for answering the question. Our approach determines the required spatial operations through a question-conditioned planner before evidence acquisition, and calls only the tools required by this plan. This design is motivated by our capability-boundary analysis and avoids iterative exploration of the evidence space. GCA (Chen et al., 2026) introduces geometry-constrained reasoning, formalizing spatial constraints and executing geometric computations through tools. GCA retains substantial VLM agency during geometric computation: its VLM operates as a tool orchestrator, resolves ambiguities, and supplies descriptions to a code-generating component. In contrast, our methods is to push more of the post-grounding reasoning into a fixed executor. Once the entities and geometric variables have been obtained, relations such as direction, distance, filtering, and eligibility are evaluated numerically according to the planned structure. Our method represents how question-level constraints should be composed and executed through an explicit dependency structure.

A.3 World model and scene exploration method

Another line of work acquires or generates additional viewpoints to support spatial reasoning. MindJourney (Yang et al., 2026) uses a VLM to plan camera movements and a video world model to generate novel views. Think3D (Zhang et al., 2026) reconstructs a 3D point cloud together with camera poses, which can be used for the model to navigate reconstructed viewpoints and perform 3D spatial reasoning. ViSA Jha et al. (2025) similarly leverages world models for spatial reasoning and introduces a verifier to assess the consistency of generated views and reasoning trajectories. While these approaches enrich the available visual evidence through view synthesis or explicit scene reconstruction, they introduce significantly additional inference overhead by constructing or exploring representations beyond those directly required by the question. In contrast, our method performs query-specific grounding and computes only the spatial evidence required by the planned reasoning operations, avoiding reconstruction of the complete scene.

B AUTONOMOUS RESEARCH INITIATIVE

Recent advances in generative AI coding agents have achieved a huge success. Beyond presenting our designed method, we ask a complementary question: given the same spatial reasoning problem, what solution would an autonomous research system develop independently? To study this question, we task an autonomous research platform with addressing the same research objective considered in this work: improving the spatial reasoning capability of Qwen3-VL-8B.

Experimental setup. We construct a research harness that provides the coding agent only with the high-level research objective. The agent receives no task-specific datasets, methods, implementation details, or evaluation targets, and must rely on its own search and coding capabilities to develop a solution. The harness is provided in the supplementary material. The research objective is to independently develop methods for improving the spatial reasoning capability of the VLMs, with an emphasis on three aspects: (1) multi-hop spatial reasoning, (2) reasoning under viewpoint changes, and (3) grounded spatial relations between objects.

Importantly, we do not provide the auto research system with the datasets identified and used in our human-designed experiments, nor do we specify a target optimization metric. The system is responsible for identifying appropriate evaluation resources, formulating candidate methods, implementing them, and conducting its own experiments. We deploy the research harness using ChatGPT Codex and use Qwen3-VL-8B as the target backbone. We do not impose a predefined quota on the number of research iterations.

To prevent information leakage or shortcut behaviors during autonomous research, such as inspecting the held-out test set, we explicitly specify prohibited actions in the research harness and enforce them through an independent code-review agent. The reviewer examines the code produced during autonomous research to verify (i) compliance with the predefined experimental protocol and sandbox boundaries and (ii) the absence of unauthorized access to held-out datasets, human-developed implementations, or other restricted artifacts. The reviewer does not provide methodological suggestions, performance feedback, or debugging assistance to the research agent. To further prevent information exchange, the reviewer is instantiated as an independent coding agent with no communication channel to the autonomous research agent.

After the experiment is completed, human researchers examine the implementations and analyses to understand the methods discovered by the autonomous system. In addition, the autonomous research system is required to record every experimental trial, including the proposed modification, implementation, evaluation, and resulting observations, enabling a complete post-hoc analysis of its research trajectory. Details are provided in Appendix G.

Finding analysis. The AR conducts 15 rounds of experiments, it independently identifies SPHERE Zhang et al. (2025) as a closely related benchmark and iteratively develops methods to improve performance on it. We present the complete AR experiment log in the supplementary material and summarize the progression of iterations in the Appendix, with a progressive experiment plot in Fig 9. The methods found by AR use only Qwen3-VL-8B at inference time, without calling external tools. Across these experiments, AR identifies two particularly effective strategies for spatial reasoning. **(1) Multi-pass reasoning.** AR first performs an inference pass in which the VLM generates a concise visual description, limited to 48 tokens. The generated description is then fed back to the VLM in a second pass to produce the final answer. Conceptually, this strategy introduces an explicit intermediate visual-reasoning step before answer generation, related to prior methods that improve visual reasoning through intermediate representations and multi-stage inference Li et al. (2026a). However, such methods do not pass the improvement on the metric test. **(2) Constrained decoding.** AR further discovers that restricting the first decoding step to the valid answer choices specified by the question improves results. For example, for a four-way multiple-choice question, the candidate tokens are limited to $\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}$. The improvement from this strategy suggests that substantial answer-relevant information is encoded in the first-token distribution, while unrestricted generation may introduce additional decoding errors.

C Details of capability boundaries testing

To characterize the capability boundaries of a general-purpose vision–language model, we evaluated the frozen Qwen3-VL-8B-Instruct checkpoint on three distinct capability families: object localization, text-only reasoning, and relative-depth perception. All experiments are zero-shot. Vision tasks used greedy decoding, while the reasoning comparison used the same stochastic decoding configuration for Qwen and DeepSeek. Scores are reported on a 0–100 scale, so differences between models are measured in absolute percentage or F1 points.

For object localization, we sampled 1000 images from the COCO 2017 validation set and constructed 1186 category queries on 992 images. For each query, Qwen was given the image and a category name and instructed to return a bounding box for every visible instance of that category. Its output boxes used coordinates normalized to the range 0–1000. The reference predictions were produced by Faster R-CNN with a ResNet-50 FPN V2 backbone and COCO V1 weights. Predicted and ground truth boxes are matched one-to-one at $IoU \geq 0.5$. The performance is computed by

$$F_1 = \frac{2TP}{2TP + FP + FN}, \tag{8}$$

where TP, FP, and FN denote the matched boxes, unmatched predictions, and missed ground-truth objects

The reasoning experiments examine text-only reasoning. GSM8K tests multi-step mathematical word-problem solving. The evaluation used questions from the GSM8K test set, and answers were scored by exact numerical agreement. BBH Logical Deduction tests the ability to infer an ordering from a set of textual constraints.

Qwen3-VL was compared with DeepSeek-R1. Both models were instructed to reason step by step and conclude with either a single boxed number for GSM8K or a single boxed option letter for logical deduction. Each model generated two independent responses per problem using temperature 0.6, top-$p = 0.95$, and a maximum output length of 32768 tokens. The $\mathrm{Acc@Pass1}$ represents an estimate of single-sample exact-match accuracy. Correctness was first averaged over the two samples for each problem and then averaged over problems.

The spatial experiment evaluates relative-depth perception on the official 654-frame NYU Depth V2 test split, spanning 215 indoor scenes. Two pairs of spatially separated surface points were selected in each frame. Ground-truth depth came from NYU's measured rawDepths. The two points in a pair were separated both spatially and in measured depth.

Qwen received the RGB image with the two marked as A and B and was asked which had the smaller forward camera-axis depth z. The A/B assignment was randomized to prevent positional or label bias. Depth Anything V2 Metric Indoor Base received the corresponding unmarked RGB image and produced a dense metric-depth map. Its predicted depth at each evaluation point was obtained from the median of a valid 5×5 patch. The primary depth score was pairwise ordering accuracy.

D Case study

We present qualitative case studies to examine the spatial reasoning behavior of TTSR. We randomly sample examples from the benchmark and compare TTSR with state-of-the-art spatial reasoning methods. For each example, we report the question, answer options, and reasoning traces produced by different methods. The results are shown in Figure 3 and 4.

The first example requires reasoning in an entity-centric reference frame. GPT-4o and Gemini directly predict the answer without providing a spatial rationale to save tokens. TreeVGR provides only localization information as evidence, but its reasoning trace does not explicitly translate the localized entities into spatial evidence for resolving the question. TTSR grounds the entities of each candidate option and derives computational evidence from their localized bounding boxes. This produces explicit spatial evidence for each option and allows inconsistent candidates to be filtered during reasoning. In this example, the final decision is determined by the sign of the corresponding spatial evidence, highlighted in red in Figure 3. The second example also requires a change of perspective, with directional answer choices such as *back*, *front*, and *left*. TTSR explicitly constructs the computationally transformed spatial evidence, where the rightward and forward displacement

components are the dominant non-zero elements in the highlighted coordinate system. The final prediction is obtained by comparing this evidence with the directional template coordinate of each option using cosine similarity. Thus, the prediction is supported by explicit, grounded geometric evidence rather than being inferred from the visual representation. TreeVGR provides bounding-box information, but localization alone is insufficient to resolve the required perspective-dependent relation. MSSR performs a multi-stage reasoning procedure involving planning, information curation, and decision making, and converts the visual information into numerical representations. However, its reasoning does not explicitly account for the required perspective transformation, leading to an incorrect prediction.

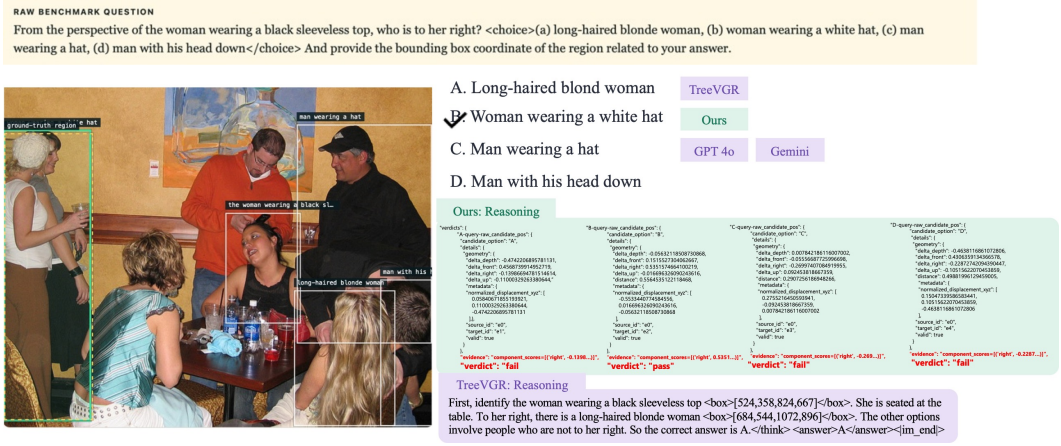


Figure 3: Qualitative comparison of perspective-dependent spatial reasoning between TTSR and baseline methods. Bounding boxes are predicted by TTSR and shown for visualization.

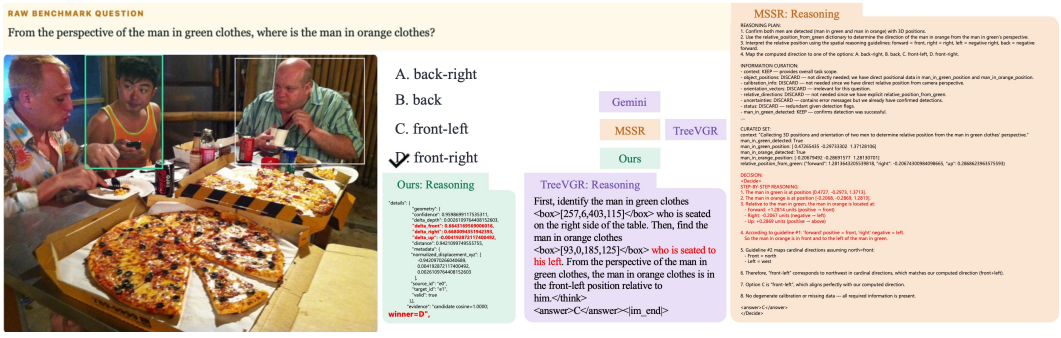


Figure 4: Case study with directional answer options. The candidate answers correspond directly to spatial directions. We compare the visual-spatial reasoning processes of TTSR with different baseline methods.

E TTSR IMPLEMENTATION

For TTSR, we implement as a plan, ground, and execute platform for spatial question answering. For the experiment section, all VLM calls use the same frozen Qwen3-VL-8B-Instruct model, with loaded in evaluation mode with bfloat16 weights, SDPA attention, and auto device placement. Input images are resized to 672×672. We use greedy decoding and set `do_sample = False`. Every structured call is instructed to return one JSON file.

The complete VLM call schedule and generation budgets are given in Table 5. Text-only calls receive no image tokens. For every visual call, images are inserted in the stated order, followed by the stage text and then the stage-specific instruction in a single prompt.

Table 5: Frozen Qwen3-VL calls used by TTSR. Token counts are maximum newly generated tokens.

Call	Visual input	Tokens	Output
Choice planner	None	1400	$\Pi_q = (\mathcal{E}_q, \mathbf{O}_q, \prec_q)$ as JSON.
Entity localization	Original image(s)	1200	One normalized box and optional orientation receipt per required entity.
Missing-entity retry	Original image(s)	700	Conditional box-only recovery for all unresolved required entities.
Recovery identity check	Unlabeled crops, scene images	512	Conditional batched match/mismatch for recovered answer entities.
Orientation estimation	Entity crop, its scene image	256	Per-entity body azimuth information.
Appearance verification	Entity crops, relevant scene images	900	One call per candidate containing an appearance claim.
Evidence-conditioned answer	Original image(s) + Grounded evidence	512	Among non-failed surviving options.

Given the input consists of one or more images I, a question q, and answer choices A. It first translates the question and choices into a semantic reasoning plan. It then leverages its perception capability and auxiliary visual tool calls to localize the entities required by the plan and estimate their geometric information. A deterministic executor converts this information into a claim level computational evidence. The final answer is produced directly when the evidence is decisive. Our method is generalizable method: dataset identity, annotation tags, and metadata are never input into the methods.

E.1 Implementation of geometry planning

The planner is a text-only call, the VLM receives q and $\mathcal{A}$. A frozen lexical parser extracts attribute (ATT), position (POS), and ranking (REL) operators and records their character span in q. The text-only planner receives the question and answer choices and returns a JSON file of Π_q. Each spatial operation in $\mathbf{O}_q$ contains the tuple $\rho = (e_a, r, e_i, f)$. For f, our TTSR is able to work under camera- or entity-relative logics.

Each entity in $\mathcal{E}_q$ will be assigned an identifier, description, and role. Each candidate in $\mathbf{O}_q$ contains spatial tuples $\rho = (e_a, r, e_i, f)$. The returned plan covers every option, references only declared entities, use a supported relation and frame, and preserves the constraints in the raw question. The implementation plan of $\prec_q$ is in a sequential order. Attribute and non-ranking spatial predicates are prerequisites. The comparative operations are executed only after these predicates create the eligible set. For example, in the example from Section 4.1, redness and table position are evaluated before the distance measurement.

Grounding We use the VLM's capability to ground the entities. We use Π_q to help identify which object in the image needs to be grounded. We use the localization prompt and feed the prompt and image to the VLM, and the VLM returns one box for each entity with the requested identifier. The model returns the bounding box for each entity in normalized coordinates $b_i = [x_i^1, y_i^1, x_i^2, y_i^2] \in [0, 1000]^4$. For each entity with the identifier, we represent the localization information (u_i, v_i) using the arithmetic midpoint of the bounding box corners. To improve the robustness of the prediction, if an entity in $\mathbf{O}_q$ returns a NaN value, we repeat the localization grounding procedure.

For each candidate with an appearance description, we use VLM to verify the claim. Concretely, the VLM receives the crop from the bounding box and determines the attribution in the range of color, identity, and material.

Depth estimation The implementation of depth estimation in Section 4.2 uses a metric method. The depth tool is `Depth-Anything-V2-Metric-Indoor-Base-hf`, whose output is positive metric depth. Each depth image is converted into RGB, processed by the depth model, and its prediction is resized to the original image resolution with bicubic interpolation and `align_corners = False` before aggregation.

Concretely, we use a pinhole back-projection, where the principal point is $c_x = \frac{W}{2}$, $c_y = \frac{H}{2}$, and $f_x = f_y = \frac{W}{2\tan(\phi/2)}$, with horizontal field of view $\phi = 60^\circ$. Given the median positive metric depth d_i, we compute $\mathbf{p}_{e_i}$ based on Equation 2.

E.2 Reference frame transformation

The implementation uses the transformation $\mathbf{T}_{e_a}^{f}$ and displacement $\Delta_a^{e_i}$ defined in Section 4.2. We define the camera coordinates are ($+X$right, $+Y$down, $+Z$Into the scene). For entity relative relations, the model estimates an azimuth ϑ_a where 0° represents facing toward the camera, 90° represents facing camera-left. Before construction $\mathbf{R}_f$, we convert this estimation to the convention used in Section 4.2 by $\theta_a = (\frac{\pi}{180}\vartheta_a + \pi) \bmod 2\pi$,, which transfers $\theta_a = 0$ point along positive camera depth, i.e. facing away from the camera. We apply equation 5 to compute the relative displacement of two entities. The right, up, and front evidence values are the projections of this vector on the corresponding axes of f. This is equivalent to evaluating $h_r(a, e_i, f)$ in Section 4.3.

Open set counting For a question with an explicit count request, we use the same planning and evidence method. For the prediction, we use the bounding box to cover all question-related entities, and deduplicate of two entities with IoU greater than 0.95. We then use the same method to ground every spatial and attribute constraint and filter out the entity fails a planned claim. We use VLM to produce the final count of the verified entities.

Computing relation evidence. For the entity relation r, the VLM projects $\bar{\mathbf{\Delta}}_a^{e_i}$ onto the direction template $\mathbf{v}_r$. We determine the value of each direction by set a threshold value of 0.08 to determine the justification. A compound direction such as front-left, needs to ground all of the directions with the computational evidence. For nearest or farthest relative to an anchor, we use $\|\mathbf{\Delta}_a^{e_i}\|_2$ as the computational evidence.

E.3 Hyperprameters and reproducibility receipts

We summarize the fixed the hyperparameters in our experiment section: depth FOV 60°, comparison margin 0.08, crop expansion 0.02, ordinal depth weight 1.0.

F Failure analysis

We systematically analyze the failure cases of TTSR by tracing each incorrect prediction through the full execution process, including planning and semantic representation, object grounding, constraint execution, and final answer selection. We audit prediction records and further examine representative case studies by comparing the question, plan, grounded entities, and final prediction. Through this analysis, we identify three recurring failure patterns, summarized in Table 6 and illustrated with representative examples in Figures 5, 6, and 8.

Planning and semantic representation. The largest source of error arises when converting the question and answer options into an executable plan. We observe two recurring semantic errors: **information loss or distortion**, where a required entity, relation, or condition is omitted or incorrectly represented, and **frame mismatch**, where the plan adopts a reference frame that is inconsistent with the question. For example, in Figure 5a, the question asks for a woman's position from a girl's perspective, but the reference frame predicate is evaluated in the camera frame. In Figure 5b, the plan omits the condition "on the desk," causing a floor scale to be selected instead of the mouse. This category accounts for 63.30%, 57.39%, and 15.41% of answer errors on SPHERE, ViewSpatial, and MultiHop Spatial, respectively.

Spatial representation and viewpoint construction. Beyond specifying the intended reference frame in the plan, perspective-dependent reasoning requires consistent orientation estimates and a well-defined coordinate system. For example, in Figure 6c, a person is described as facing toward the camera, while the estimated body azimuth is 180°, although the adopted convention assigns 0° to this direction. The VLM then uses this inconsistent numeric estimate and produces an incorrect answer. Another challenge arises from simulated viewpoints. For example, the instruction "stand

Table 6: Failure patterns identified from reasoning traces across spatial reasoning benchmarks.

Failure Pattern	Trace Evidence	SPHERE	ViewSpatial	MultiHop Spatial
Planning / semantic representation	Lost constraints, wrong frame, or empty plan	63.30%	57.39%	15.41%
Spatial representation	Orientation conflict or unavailable cross-image geometry	0.19%	16.67%	5.49%
Perception / grounding	Missing required entities or incorrect referents	10.73%	0.34%	6.72%

at the lamp facing the sofa" requires both a viewpoint origin and a facing landmark. In Figure 6d, the sofa is omitted, while the lamp and target bag are grounded in different images without a shared geometric reference. Scene simulation is also the weakest subset of ViewSpatial, with an accuracy of 28.51% (315/1105). We further report the confusion matrix for person-relative directional questions in Figure 7. A large fraction of left and right predictions are reversed, further indicating that failures in spatial representation and viewpoint construction remain a second source of error.

Perception and object grounding. Grounding errors occur when a required entity cannot be resolved or when a description is assigned to the wrong entity. For example, the question specifies the woman holding a remote control, but the observer box localizes a different woman (Figure 8e). In another example, the gold bicycle remains unresolved after repeated localization attempts, preventing complete verification of its spatial relation to the reference person (Figure 8f). The grounding proportions reported in Table 6 reflect the frequency of this primary failure category.

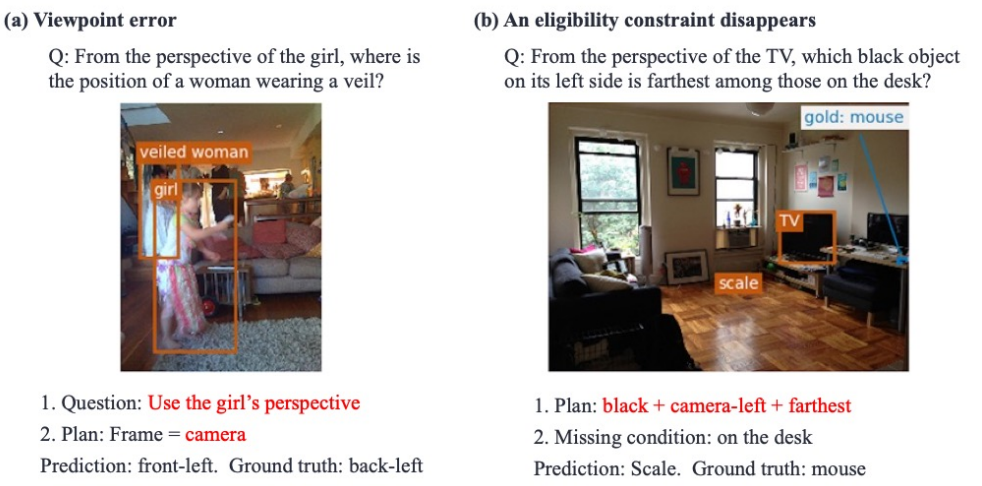


Figure 5: Representative failure traces of TTSR: (a-b) planning and semantic representation.

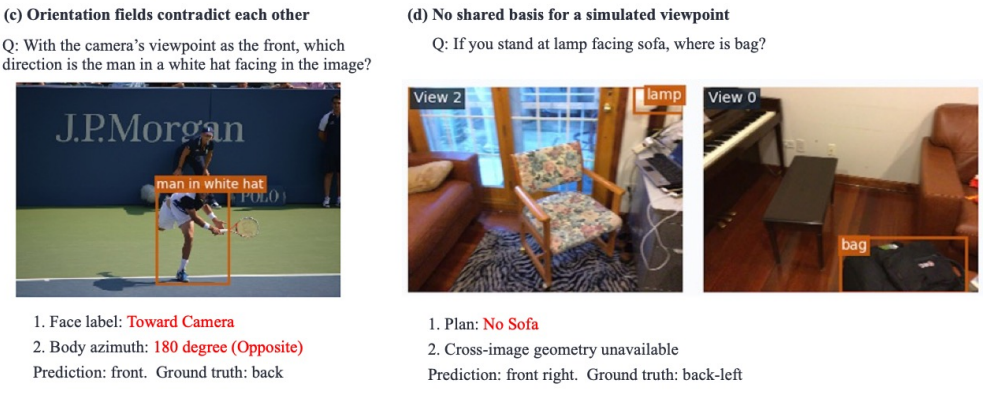


Figure 6: Representative failure traces of TTSR: (c-d) spatial representation and viewpoint construction

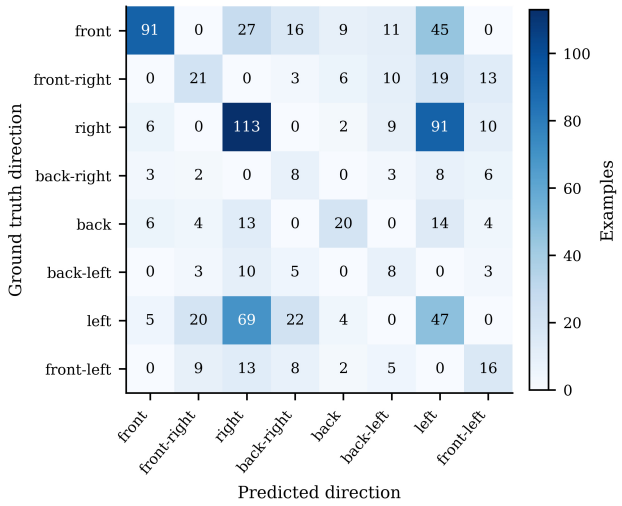


Figure 7: Directional confusion matrix for person-relative spatial reasoning.

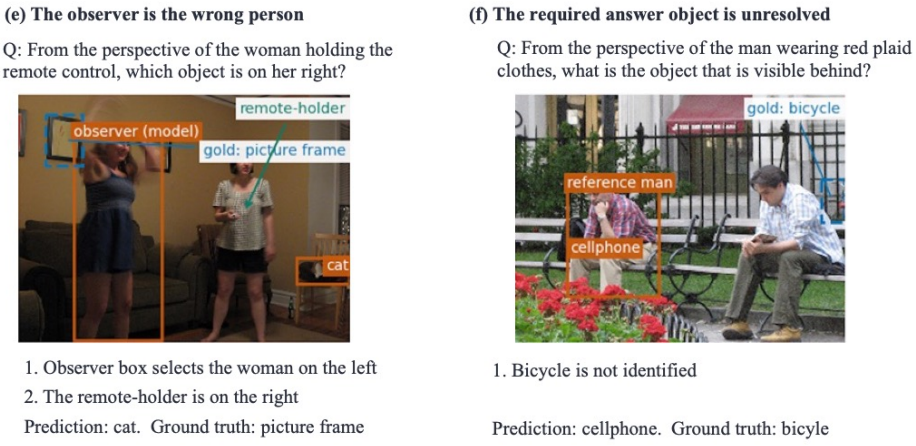


Figure 8: Representative failure traces of TTSR: (e-f) perception and object grounding.

G AUTO RESEARCH DETAILS

The auto research is conducted with OpenAI codeX platform with GPT-5.6 Sol high thinking with Goal mode. We use ChatGPT 5.6 extra thinking to draft the research plan.md with prompt `You are running an auto-research experiment. The objective is to develop a spatially aware VLM. You can use Qwen3-VL-8B on my disk. The task is to improve multi-hop spatial reasoning and perspective-change reasoning. You need to first plan the task, run the validation, and run the tests on your own. You must provide runnable code before the quota runs out. Always document what you have done, what you have observed, and all experimental results. You should do everything you need by yourself. No references will be provided.` The research .md file is provided in the Supplementary material. The entire experiment runs 3 hours and 30 minutes, with 2 failures that human researchers relaunched the experiments.

All information is interpreted from the log with authors effort, the entire experiment procedure transformed into an image is presented in Figure 9. This figure is also plotted based on the experiment log by ChatGPT 5.6 Sol. During the discovery procedure, the research agent used a sequential,

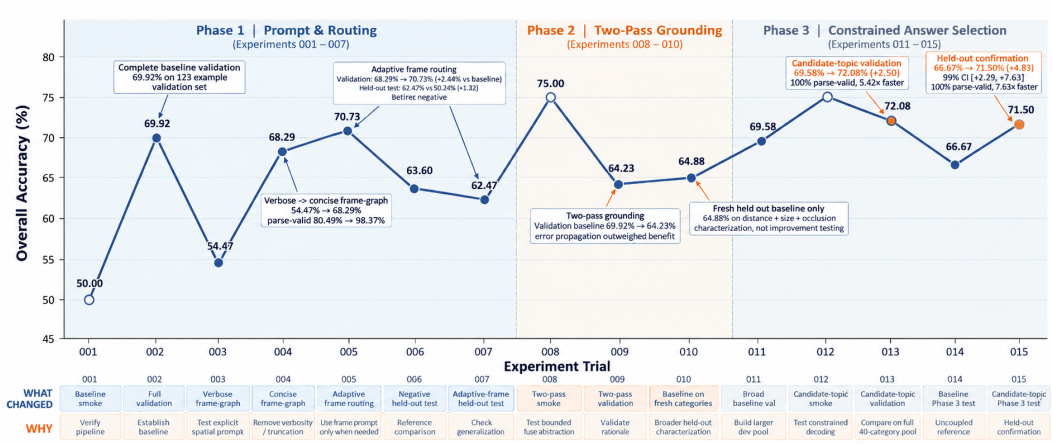


Figure 9: Auto Research progress over 15 trials measured by validation accuracy. The search exhibits three phases. In Phase 1, Auto Research explores prompt tuning as a simple initial strategy. In Phase 2, it introduces grounding evidence into the reasoning trace. In Phase 3, it further applies constrained decoding. This figure was generated by GPT 5.6 from the complete experiment log in `experiment.md`.

evidence-driven procedure. It first implemented zero-shot baselines, then evaluated the baseline on a small split to speed up the discovery. Baseline errors are grouped into viewpoint transformation, multi-hop composition, counting, and answer formatting failures. These groups are used to generate the hypothesis:

- 1. Explicit spatial reasoning. AR tested a verbose frame-graph prompt, a concise prompt, and a question-dependent router. After the 5 trials, the performance dropped from 63.6 to 62.5, therefore, the hypothesis was rejected.
- 2. Two-pass grounding. AR tested on a two-pass method that first generated one short visual fact description and then independently verified it while answering. The performance on the validation drop from 69.9 to 64.2. The intermediate fact contains perceptual errors, which cause the final prediction to be worse. This test lasted 2 rounds.
- 3. Constrained decoding. The previous trials indicated that generating additional reasoning can introduce errors. Therefore, AR removed free-form decoding and constrained the decoding only to candidate answers. This improved from 69.6 to 72.1.

Each round follows the cycle: baseline error analysis, hypothesis formulation, minimal implementation, controlled validation comparison, accept and reject decision.

What auto research final proposed Constrained decoding achieved a significant improvement on the held-out validation set. The implementation method is straightforward: For candidate set $\mathcal{C}$, prediction is given by

$$\hat{c} = \arg\max_{c \in \mathcal{C}} \log p_\theta(c|I, q). \tag{9}$$

For multiple-choice questions, candidates are the single token labels `A,B,...` For counting questions, candidates are digits `0-9`. This focus on the final prediction is a recent hop research topic.

H Computation Cost

We summarize the computational cost of the methods used in our experiments. All experiments are conducted on a mixed GPU cluster with NVIDIA A100, H100, and A6000 GPUs, and all jobs are managed using Slurm. We report the wall-clock time and GPU-hours of each method in Table 7. SpatialVLM and SpaceOM have relatively low computational costs, mainly because they use lighter backbone models. LVR is a training-based method, we directly use its released checkpoint, and

its inference cost is also relatively low. MSSR requires substantially more computation. In our experiments, we observe that MSSR repeatedly searches for minimally sufficient evidence, which involves many search and tool-calling steps. Our method requires more computation than baselines because it performs multi-step grounding and constructs explicit spatial evidence for reasoning. This additional computation leads to a substantial improvement in performance. Improving the computational efficiency of our framework remains an important direction for future work. We also report the number of API calls and API cost in US dollars for closed-source models on the MultiHop Spatial Reasoning benchmark.

Table 7: Computational cost comparison across spatial reasoning benchmarks. Wall-clock time is reported as HH:MM:SS, together with total GPU-hours.

	MultiHopSpatial		**ViewSpatial**		**SPHERE**	
Method	**Wall-clock**	**GPU-h**	**Wall-clock**	**GPU-h**	**Wall-clock**	**GPU-h**
SpatialVLM	00:14:53	0.99	00:40:57	1.33	01:10:48	1.72
SpaceOm	03:55:12	15.68	04:10:05	4.17	03:21:42	3.33
TreeVGR	06:03:56	24.26	01:33:56	6.26	03:10:04	12.67
LVR	02:22:08	9.48	03:10:54	12.73	06:28:20	25.89
APC	05:47:28	23.16	02:57:38	11.84	06:57:37	27.84
MSSR	44:09:36	223.56	145:59:05	330.61	66:42:08	382.62
Ours	23:40:17	59.10	19:44:59	39.77	19:31:38	64.71

Table 8: API usage and cost comparison.

Model	**Requests**	**Total Tokens**	**API Cost ($)**
Gemini 2.5 Flash	4500	4325107	6.97
GPT-4o	4500	2529943	6.53

I LIMITATION

Our geometry provides an approximate, question-conditioned spatial representation rather than a calibrated 3D reconstruction. Each entity is represented using the midpoint of its bounding box and the median depth within the box. These approximations may introduce errors when objects overlap, are partially occluded, or occupy irregular regions, as the estimated midpoint or depth may correspond to another object or the background. The camera intrinsics are also approximated using commonly adopted assumptions, including a fixed horizontal field of view. Consequently, the resulting directional and distance evidence can be affected by errors in object localization, depth estimation, orientation estimation, and camera parameter approximation.

Orientation estimation represents another source of error. On PASCAL3D+, we evaluate Qwen3-VL under zero-shot four-way orientation estimation and obtain an accuracy of 66.7%, indicating substantial room for improvement. Since orientation estimates directly affect perspective-dependent spatial transformations, more accurate orientation estimation could further improve the performance of our framework.

Finally, TTSR introduces additional computational overhead due to its grounded perception, auxiliary estimation, and explicit spatial computation steps. Our current implementation has not been extensively optimized for computational efficiency or high-performance GPU execution. Improving inference efficiency, for example through more efficient tool invocation and parallelized computation, is an important direction for future work.