Geometrically-Constrained Agent for Spatial Reasoning

Zeren Chen[1,2*], **Xiaoya Lu**[2,3*], **Zhijie Zheng**[1,2], **Pengrui Li**[1], **Lehan He**[1,4], **Yijin Zhou**[2,3,4]
Jing Shao[2], **Bohan Zhuang**[5†], **Lu Sheng**[1†]

[1]School of Software, Beihang University [2]Shanghai AI Laboratory [3]Shanghai Jiao Tong University
[4]Shanghai Innovation Institute [5]State Key Lab of CAD&CG, Zhejiang University

{czr1604,lsheng}@buaa.edu.cn, {luxiaoya, shaojing}@pjlab.org.cn

Homepage: https://gca-spatial-reasoning.github.io

Abstract

*Vision Language Models (VLMs) exhibit a fundamental semantic-to-geometric gap in spatial reasoning: they excel at qualitative semantic inference but their reasoning operates within a lossy semantic space, misaligned with high-fidelity geometry. Current paradigms fail to bridge this gap. Training-based methods suffer from an "oracle paradox," learning flawed spatial logic from imperfect oracles. Tool-integrated methods constrain the final computation but critically leave the VLM's planning process unconstrained, resulting in geometrically flawed plans. In this work, we propose **Geometrically-Constrained Agent** (**GCA**), a training-free agentic paradigm that resolves this gap by introducing a formal task constraint. Specifically, we strategically decouples the VLM's role into two stages. First, acting as a semantic analyst, the VLM translates the user's ambiguous query into the formal, verifiable task constraint, which defines the reference frame and objective. Second, acting as a task solver, the VLM generates and executes tool calls strictly within the deterministic bounds defined by the constraint. This geometrically-constrained reasoning strategy successfully resolve the semantic-to-geometric gap, yielding a robust and verifiable reasoning pathway for spatial reasoning. Comprehensive experiments demonstrate that GCA achieves SOTA performance on multiple spatial reasoning benchmarks, surpassing existing training-based and tool-integrated methods by ∼27%.*

1. Introduction

Intelligent agents operating in real-world applications, such as robotics [37, 49, 57], AR/VR [2, 9, 27], and autonomous driving [7, 38, 46], demand a perceptual understanding of the world akin to humans. Humans intuitively comprehend their surroundings as a cohesive 3D environment, effortlessly discerning object orientations and complex spatial relationships. However, equipping Vision Language Models (VLMs) into agents with this holistic **spatial reasoning** capability remains a critical challenge [16, 48, 50, 54, 55].

As shown in Figure 1 (a), current VLMs lossily translate rich visual information into a textual semantic space, leading fine-grained geometric details to be omitted or distorted [21, 45]. This creates a fundamental semantic-to-geometric gap: *VLMs excel at probabilistic, qualitative semantic inference, but their lossy semantic space required for spatial reasoning fails to ground high-fidelity geometry.* For example, a VLM may possess the spatial commonsense (*e.g.*, intuitively knowing that "sitting on a sofa" implies a viewpoint aligned with the sofa's orientation), yet critically fail at high-precision geometric computation (*e.g.*, determining the sofa's orientation) and robust spatial imagination (*e.g.*, imagining the user's egocentric perspective). To reconcile this gap, robust **constraints** must be imposed, guiding the VLM's reasoning onto a geometrically sound and verifiable pathway.

However, effectively applying these constraints remains a formidable challenge. Recent approaches that apply implicit constraints via end-to-end training [6, 20, 26, 29, 37, 43, 45, 47] attempt to embed geometric logic by fine-tuning on massive datasets. These methods, however, face an "oracle paradox": their data generation relies on oracles like GPT-4o [15] which themselves struggle with spatial reasoning [16, 48, 50, 54, 55]. Consequently, the VLM is often trained on flawed spatial logic rather than sound geometric principles. An alternative paradigm, tool integration [10, 44, 57], attempts to bridge this gap by adopting an iterative plan-then-execute strategy, which offloads high-precision geometric computation to deterministic external tools. While this constrains the final computation process, the VLM's planning process remains unconstrained. To plan next step, the VLM must still perform spatial imagination and further decision-making within its lossy semantic space, inevitably producing geometrically flawed plans.

*Equal contribution. †Corresponding author.

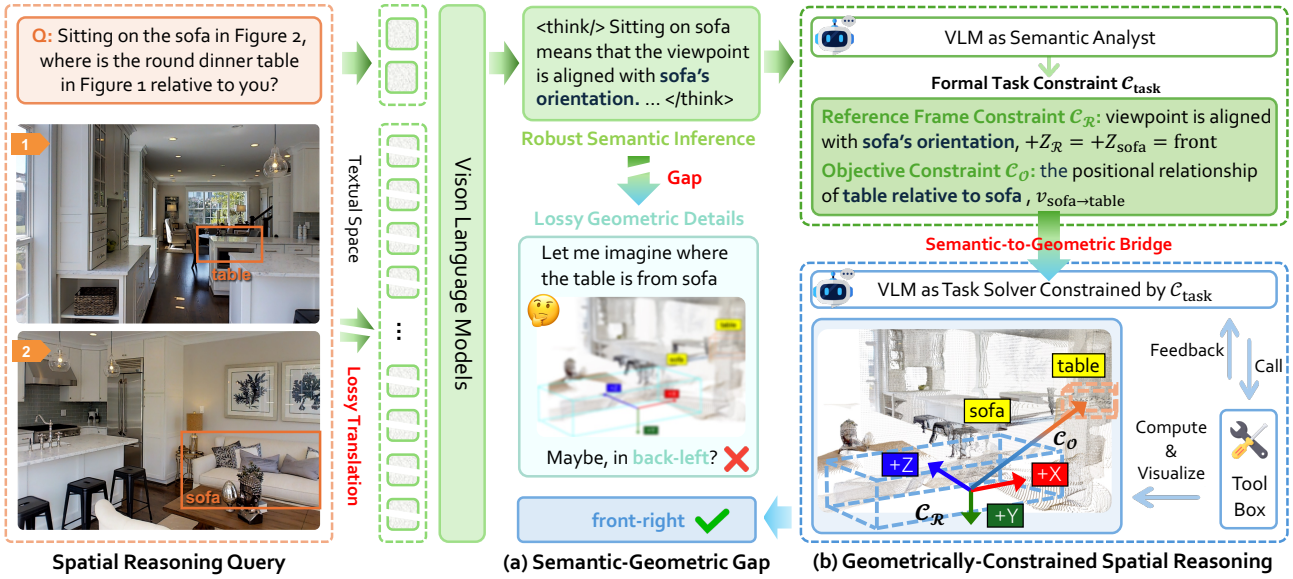

Figure 1. **Overview. (a) Semantic-Geometric Gap.** The geometric details required for spatial reasoning are lost when translating visual information into textual space, leading to VLM's flawed reasoning or unconstrained planning. **(b) Geometrically-Constrained Spatial Reasoning.** We propose a formal task constraint that serves as a deterministic bridge between semantics and geometry in spatial reasoning.

For instance, when asked to reason from the perspective of a user "sitting on the sofa" (see Figure 1), its unconstrained plan may default to the camera's viewpoint, compromising the problem definition before any tool is even called.

These challenges reveal the critical research question: *How do we bridge the VLM's semantic-to-geometric gap?* We argue the solution is not to force the VLM to reason about lossy geometric details directly, but to reframe the problem into a task that leverages its spatial commonsense to define a **formal task constraint** $\mathcal{C}_{\text{task}}$ for subsequent computation. Specifically, this $\mathcal{C}_{\text{task}}$ must be (1) grammatically rich enough to define complex spatial concepts, such as viewpoints, which elude traditional state-based formalisms, (2) semantically clear enough for a VLM to generate using its qualitative strengths, and (3) geometrically sound enough to provide a deterministic, verifiable constraint for subsequent computation.

To this end, we introduce **Geometrically-Constrained Agent** (**GCA**), a training-free agentic paradigm for geometrically-constrained spatial reasoning. As shown in Figure 1 (b), this strategy leverages a formal task constraint, $\mathcal{C}_{\text{task}}$, to decouple the reasoning process into two stages: (1) *Task Formalization.* The VLM, acting as a semantic analyst, translates the ambiguous query and visual data into the formal, verifiable task constraint $\mathcal{C}_{\text{task}}$. This stage defines what to solve, establishing immutable sub-constraints: a **reference frame constraint** and an **objective constraint**. (2) *Constrained Geometric Computation.* The VLM then, acting as a task solver, generates and executes tool calls to compute the final answer, operating strictly within the deterministic bounds defined by $\mathcal{C}_{\text{task}}$. This two-stage decoupling directly bridges the semantic-to-geometric gap. Through formulating a geometrically sound constraint, we force the VLM to solve deterministic mathematical problems, thereby avoiding the demands for directly computing or imagining about high-fidelity geometric details that are lost in its semantic space. Extensive experiments demonstrate the effectiveness and generalizability of GCA paradigm. GCA yields substantial performance gains when applied to several foundation VLMs (by an average of $\sim$37%), establishing a new state-of-the-art across a diverse suite of challenging spatial reasoning benchmarks.

2. Related Work

Spatial Reasoning with VLMs. Spatial reasoning, including comprehension and mental manipulation of 3D spatial relationships [16, 48, 50, 54, 55], remains a foundational challenge for VLMs [15, 19, 32, 39]. To address this deficit, recent research [4, 6, 20, 26, 29, 35, 43, 45, 47] focuses on large-scale, end-to-end training on specialized spatial datasets. These methods attempt to bridge the 2D-3D cognitive gap by incorporating geometric priors, such as 3D features [43], or depth maps [4], directly into the VLM's architecture, but they are often hindered by the reliance on high-quality datasets generated by flawed oracle. Another line of research introduce tool-integrated reasoning [10, 22, 25, 44, 57] to offloads deterministic geometric computation to external modules. For example, SpatialAgent [44] and TIGeR [10] focus on translating the in-

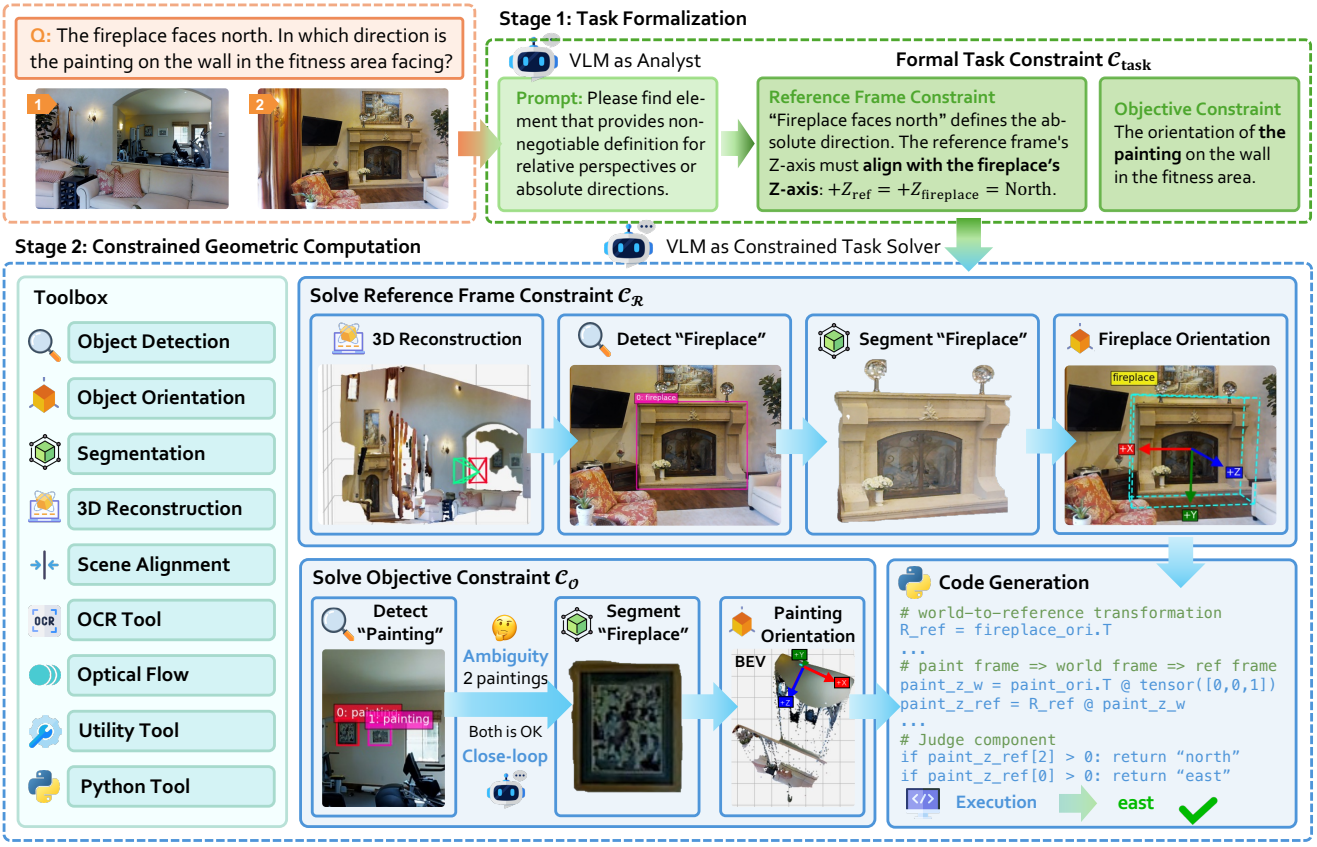

Figure 2. **Overall Paradigm of GCA.** Given a spatial reasoning query, our GCA leverages a geometrically-constrained reasoning strategy centered on the formal task constraint ($\mathcal{C}_{\text{task}}$). The VLM first translates the ambiguous query into this explicit $\mathcal{C}_{\text{task}}$, establishing a non-negotiable reference frame ($\mathcal{C}_{\mathcal{R}}$) and objective ($\mathcal{C}_{\mathcal{O}}$). Strictly constrained by $\mathcal{C}_{\text{task}}$, the VLM then orchestrates a toolbox to perform deterministic geometric computation and derive the final answer.

put query directly into an iterative sequence of tool executions. However, unconstrained planning process could lead to geometrically-flawed results, causing the agent to conflate "what to solve" with "how to solve it".

Constrained-Guided Reasoning. Constraint-guided reasoning involves restricting a search space by defining variables and the constraints governing them [34], which has been adapted to manage the probabilistic nature of LLMs and VLMs. A primary application is neuro-symbolic reasoning [1, 11, 12, 30, 36, 53], where LLM is constrained to act as a translator, converting ambiguous natural language into a formal, verifiable representation. For example, LogicLM [30] leverage LLMs to translate NL problems into task-specific formalisms. Constraint-guided reasoning can also be extended to planning [3, 14, 23, 31, 51, 56]. LLM+P [23] uses an LLM to translate an NL problem into a formal PDDL format and then applies an optimal planner to generate the plan. ReKep [14] employs a VLM to translate a free-form language into relational keypoint constraints and solves the constraints to generate final robot actions.

3. Methodology

As illustrated in Figure 2, we propose Geometrically-Constrained Agent (GCA), a training-free agentic paradigm designed for geometrically-constrained spatial reasoning. The core of GCA is the introduction of a formal task constraint $\mathcal{C}_{\text{task}}$ that serves as a deterministic bridge between semantics and geometry. Section 3.1 defines this geometrically-constrained paradigm. Section 3.2 details the formal task constraint $\mathcal{C}_{\text{task}}$ and its automated generation. Section 3.3 describes the subsequent constrained computation stage, which is strictly governed by this constraint. Finally, Section 3.4 discusses how GCA resolves the VLM's semantic-to-geometric gap in spatial reasoning.

3.1. Geometrically-Constrained Spatial Reasoning

Contemporary agentic frameworks often model reasoning as a generic, iterative policy. Those based on the ReAct framework [52], for example, can be defined by:

$$r_t = \mathcal{A}(q, \boldsymbol{v}, \mathcal{T}, r_{t-1}). \tag{1}$$

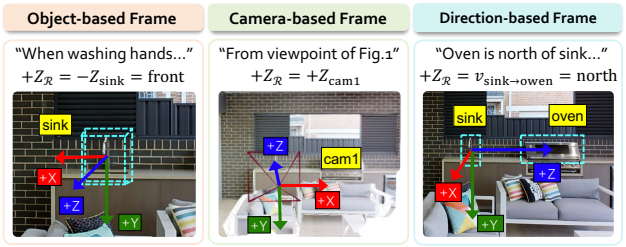

Figure 3. **Reference Frame.** Here, $v_{\text{sink}\to\text{owen}}$ denotes a vector calculated by "normalize (Centroid(owen) − Centroid(sink))".

In this framework, an agent $\mathcal{A}$ produces a response r_t based on a query q, visual information $\boldsymbol{v}$, a set of tools $\mathcal{T}$, and its past history r_{t-1}. This generic policy $\mathcal{A}$ is unconstrained, making it unreliable for high-stakes, deterministic domains like spatial reasoning. Recent work [10, 44] attempts to mitigate this by using external tools to constrain the final computation. However, they fail to constrain the VLM's planning process. The VLM may still rely on its flawed spatial imagination in lossy semantic space to formulate plan, conflating "what to solve" with "how to solve it".

We solve this by replacing the generic policy $\mathcal{A}$ with a two-stage process. This paradigm is built on the formal task constraint $\mathcal{C}_{\text{task}}$, which functions as the architectural scaffolding to align the VLM's asymmetric capabilities:

$$\begin{aligned} \mathcal{C}_{\text{task}} &\leftarrow \mathcal{F}_{\text{formalize}}(q, \boldsymbol{v}), \\ r_t &= \mathcal{F}_{\text{compute}}(\mathcal{C}_{\text{task}}, \mathcal{T}, r_{t-1}). \end{aligned} \tag{2}$$

In the $\mathcal{F}_{\text{formalize}}$ stage, the VLM acts as a semantic analyst, translating the ambiguous query q and visual information $\boldsymbol{v}$ into a formal, verifiable task constraint $\mathcal{C}_{\text{task}}$. This stage defines what to solve by establishing the necessary geometric scaffolding (*e.g.*, the reference frame and target subjects). In the $\mathcal{F}_{\text{compute}}$ stage, the VLM's role shifts to a task solver. Governed by the constraint $\mathcal{C}_{\text{task}}$ established in the $\mathcal{F}_{\text{formalize}}$ stage, the VLM iteratively executes tool calls to acquire necessary geometric data and perform final computations.

3.2. Task Constraint Formalization

3.2.1. Constraint for Spatial Reasoning

While existing constraint-guided reasoning paradigms leverage formalisms such as PDDL [23] or relational keypoint constraints [14], these constraints are insufficient for spatial reasoning. PDDL, for instance, excels at describing discrete, symbolic object states (*e.g.*, "`is_on(A, B)`") but fundamentally lacks the geometric grammar to express the continuous, relative, and perspective-dependent nature of spatial queries (*e.g.*, egocentric *vs.* allocentric viewpoints). This gap necessitates a new formalism. We thus propose a novel **formal task constraint** $\mathcal{C}_{\text{task}}$ specifically designed to capture the geometric nature of spatial reasoning. We define $\mathcal{C}_{\text{task}}$ as a tuple containing two key sub-constraints: a single, non-negotiable **Reference Frame Constraint** ($\mathcal{C}_{\mathcal{R}}$) that defines the coordinate system for answering the query, and an **Objective Constraint** ($\mathcal{C}_{\mathcal{O}}$) that specify the objective to be measured within that frame.

Reference Frame Constraint. Humans intuitively understand spatial language (*e.g.*, "north of") by grounding it within a specific coordinate system, namely a reference frame ($\mathcal{R}$). In contrast, VLM failures often stem from ambiguity in this crucial grounding step, causing them to adopt flawed geometrically flawed plans [50] (*e.g.*, defaulting to the camera's viewpoint). The $\mathcal{F}_{\text{formalize}}$ stage addresses this ambiguity by requiring the VLM to first formally anchor $\mathcal{R}$ to the scene's geometry.

We model all spatial queries as requiring a 3D cartesian coordinate system defined by an origin $O_{\mathcal{R}}$ and three orthogonal basic vectors $(\mathbf{x}_{\mathcal{R}}, \mathbf{y}_{\mathcal{R}}, \mathbf{z}_{\mathcal{R}})$. This system follows the OpenCV convention, where $+\mathbf{z}_{\mathcal{R}}$ points forward, $+\mathbf{y}_{\mathcal{R}}$ points down and $+\mathbf{x}_{\mathcal{R}}$ follows the right-hand rule. The agent's task is to anchor $\mathcal{R}$ to one of three geometric primitives (see Figure 3) derived from the visual information:

- **Object-based Frame.** $\mathcal{R}$ is defined by an object's intrinsic coordinate system. For example, the query "when the user is washing hand" implies a reference frame defined by $+\mathbf{z}_{\mathcal{R}} = -\mathbf{z}_{\text{sink}}$ (one must face a sink to wash hand).
- **Camera-based Frame.** $\mathcal{R}$ is defined by a specific camera's viewpoint. For "from the viewpoint of Figure 1", the reference frame is defined by $+\mathbf{z}_{\mathcal{R}} = +\mathbf{z}_{\text{cam1}}$.
- **Direction-based Frame.** $\mathcal{R}$ is defined by a vector connecting two locations. For "Owen is north of sink", the reference frame is defined by $+\mathbf{z}_{\mathcal{R}}$ = normalize (Centroid(owen) − Centroid(sink)) = north.

The output of this step is a human-readable and machine-parsable definition of $\mathcal{R}$, which becomes a non-negotiable constraint $\mathcal{C}_{\mathcal{R}}$ for all subsequent computation.

Objective Constraint. Concurrently, the agent identifies the objective $\mathcal{O}$ from the query. This constraint $\mathcal{C}_{\mathcal{O}}$ defines *what* must be measured relative to the established $\mathcal{R}$. For the query, "Is chair to the west of toaster?", the toaster defines $\mathcal{C}_{\mathcal{R}}$, while the positional relationship between toaster and chair is the objective constraint $\mathcal{C}_{\mathcal{O}}$.

3.2.2. Automated Formalization via VLM

We exploit the VLM's innate strength in semantic interpretation to generate $\mathcal{C}_{\text{task}}$ automatically. Acting as a semantic analyst, the VLM performs qualitative interpretation, guided by the formal definitions of $\mathcal{C}_{\text{task}}$, to generate the $\mathcal{C}_{\text{task}} = (\mathcal{C}_{\mathcal{R}}, \mathcal{C}_{\mathcal{O}})$. This formal task constraint, generated by the VLM but grounded in geometry, serves as the geometrically sound contract for the $\mathcal{F}_{\text{compute}}$ stage. In our implementation, we enforce this architectural decoupling procedurally. The VLM executes the $\mathcal{F}_{\text{formalize}}$ stage and formalizes the $\mathcal{C}_{\text{task}}$ before any computation begins.

3.3. Constrained Geometric Computation

3.3.1. Tool Integration and Code Generation

Once the formal task constraint $\mathcal{C}_{\text{task}} = (\mathcal{C}_{\mathcal{R}}, \mathcal{C}_{\mathcal{O}})$ is established, the VLM's role shifts to a constrained task solver. This $\mathcal{F}_{\text{compute}}$ stage then operates as a ReAct-style framework, consuming the $\mathcal{C}_{\text{task}}$ as an immutable constraint. This execution is not a one-shot generation but an iterative, closed-loop process involving data acquisition, ambiguity resolution, and augmented computation.

Data Acquisition. $\mathcal{C}_{\text{task}}$ dictates a set of geometric ingredients that the agent must acquire. For instance, as shown in Figure 2, to instantiate an object-based frame $\mathcal{R}$ defined by a sink, the agent must acquire the orientation of that sink. The $\mathcal{F}_{\text{compute}}$ stage begins by generating a sequence of tool calls to parameterize the geometry, and acquire all variables necessary to instantiate $\mathcal{C}_{\text{task}}$.

Tool Orchestration and Ambiguity Resolution. The VLM is responsible for managing tool feedback and resolving ambiguity, ensuring the data acquired from tools correctly binds to the symbols in $\mathcal{C}_{\text{task}}$. For example, considering $\mathcal{C}_{\mathcal{O}}$ involves an object like "leftmost chair", the perception tool returns several "chair" detections. The VLM analyzes this feedback (*e.g.*, visualizing bounding boxes) and resolves the ambiguity by determining which object index correctly corresponds to the context ("leftmost") specified. This closed-loop mechanism allows the agent to handle noisy tool outputs while ensuring the final computation remains strictly grounded in the intent of $\mathcal{C}_{\text{task}}$.

Knowledge-Augmented Code Generation. Once all variables in $\mathcal{C}_{\text{task}}$ are bound to concrete geometric data, the agent invokes a code generator for the final computation. To prevent the coder from hallucinating incorrect formulas, we leverage a knowledge-augmented strategy, which functions analogously to a static Retrieval-Augmented Generation (RAG) [8, 18] system. Specifically, when invoking the code generator, the VLM specifies a high-level requirement and the necessary bound variables (*e.g.*, object's orientation). Instead of expecting the coder to generate complex geometric formulas from memory, our framework maintains a pre-prepared, fixed library of basic, verified geometric formulas. Based on the data types of the bound variables, the system automatically retrieves the relevant, fixed set of formulas (*e.g.*, object's local-to-world transformation formula) and injects them directly into the code generator's context. This ensures the computation steps do not produce black-box guesses, but rather deterministic results, derived from a formally structured task and sound geometric principles. More details are provided in Appendix E.

3.3.2. Toolbox

We equips the agent with perceptual and computation capabilities required to execute its geometrically-constrained reasoning flow in $\mathcal{F}_{\text{compute}}$, as shown in Figure 2. Detailed APIs for all tools are provided in Appendix C.

Geometry and Perception Tools. These tools are responsible for parameterizing the visual world. "3D Reconstruction" tool leverages foundational models like VGGT [40] to build a unified, high-fidelity 3D representation of the scene. This provides the geometric context required for complex scenarios. This category also contains a suite of 2D perception tools, such as "Object Detection" for open-vocabulary object detection, "Segmentation" for instance segmentation.

Computation and Utility Tools. These tools operate on the data extracted by the perception tools and executes the final deterministic geometric computation. "Python Tool" is the core computational engine, which prompts the VLM to generate and execute Python code in a sandbox environment, using the knowledge-augmented strategy. This category also includes essential utilities ("Utility Tool"). For example, "`project_box_to_points`" bridges 2D perception to 3D computation by converting 2D bounding boxes into corresponding 3D point clouds.

3.4. Discussion

Our GCA decouples VLM's spatial reasoning through the formal constraint $\mathcal{C}_{\text{task}}$, jointly addressing two core deficiencies in spatial reasoning.

$\mathcal{F}_{\text{formalize}}$ Solves Flawed Planning and Imagination. Directly solving an ambiguous query forces the VLM to plan and perform spatial imagination within its native lossy semantic space. This is a primary failure mode, as unconstrained planning can lead to geometrically flawed assumptions before any computation even begins. Our paradigm resolves this by reframing the problem. Leveraging VLM's strength in qualitative semantic interpretation, the $\mathcal{F}_{\text{formalize}}$ stage transform the original spatial query into a deterministic mathematical problem with constraint, preventing the VLM to solve the query in its lossy semantic space directly.

$\mathcal{F}_{\text{compute}}$ Solves Flawed Execution and Computation. In this stage, the VLM acting as the task solver, orchestrating external tools to execute the plan. Crucially, its entire reasoning and execution process is bound by the formal task constraint $\mathcal{C}_{\text{task}}$ generated in $\mathcal{F}_{\text{formalize}}$. This ensures that all subsequent high-precision computations are executed strictly within the deterministic, geometrically sound constraint, effectively bridging the semantic-to-geometric gap.

4. Experiments

4.1. Experimental Setup

Implementation Details. GCA is implemented as a training-free agentic paradigm, requiring no model fine-tuning. It centers on a VLM responsible for both stages of our paradigm: acting as the semantic analyst to generate the $\mathcal{C}_{\text{task}}$ in the $\mathcal{F}_{\text{formalize}}$ stage, and as the task solver

Table 1. **Experimental Results on Several Spatial Reasoning Benchmarks.** The best and second best results are shown in **bold** and underlined, respectively. "**Avg.**" denotes the average of overall accuracy across all benchmarks. More details about these benchmarks' subcategory (*e.g.*, "PR.") are provided in Appendix.

	MMSI-Bench					MindCube-tiny				OmniSpatial			SPBench			CV-Bench			Avg.
	PR.	Attr.	Mot.	MSR	All	Rot.	Ard.	Amg.	All	Dyn.	Pers.	All	SI	MV	All	2D	3D	All	
Baseline Foundation VLMs																			
Qwen3-VL-Thinking [32]	33.7	40.0	23.3	31.8	32.6	87.0	47.3	35.0	47.3	60.5	43.9	51.0	51.9	61.2	54.1	81.9	92.6	86.8	54.4
GLM-4.5V [13]	35.6	36.9	29.3	30.3	33.8	60.0	25.5	42.2	39.6	58.6	47.2	52.1	50.0	55.1	51.3	80.7	91.6	85.6	52.5
GPT-4o [15]	28.0	32.3	36.0	30.8	30.3	33.5	35.0	37.2	35.8	58.7	46.2	51.5	42.4	48.3	43.8	69.4	84.9	76.5	47.6
Gemini-2.5-Pro [5]	39.0	36.2	33.3	34.3	36.9	**89.5**	54.5	48.8	57.5	70.7	44.6	55.8	55.6	58.3	56.3	81.2	92.5	86.3	58.5
Training-based Spatial VLMs																			
SpatialLLM [26]	24.5	23.1	22.7	30.8	25.3	34.0	26.8	33.0	31.1	59.6	42.9	49.5	32.2	26.4	30.7	51.3	78.6	64.5	40.2
Spatial-MLLM [43]	28.5	25.4	18.0	26.3	26.1	33.8	34.5	28.3	32.1	37.2	42.1	40.0	52.0	52.0	52.0	59.5	63.3	61.2	42.3
SpatialLadder [20]	30.3	23.3	16.0	21.2	25.4	30.5	39.8	47.8	42.3	46.5	43.1	44.5	**70.2**	**70.9**	**70.3**	72.4	74.9	73.7	51.2
SpaceR [29]	29.1	29.4	21.9	22.5	26.9	29.8	30.0	26.8	28.3	53.5	40.5	46.0	48.6	59.4	51.1	74.1	77.4	75.6	45.7
Video-R1 [6]	30.5	25.4	22.0	26.8	27.8	30.0	30.5	41.3	35.8	50.0	44.2	46.7	44.8	40.7	43.8	73.5	74.7	74.0	45.6
RoboBrain-2.0 [37]	28.9	28.8	22.5	28.0	28.9	29.7	35.8	45.2	39.6	49.4	42.2	45.2	49.1	46.8	48.5	77.1	90.7	83.4	49.1
VILASR [45]	35.9	26.0	21.0	23.2	29.8	34.4	25.7	29.4	29.1	37.5	42.2	40.2	50.2	57.6	51.9	75.7	77.7	76.6	45.5
VLaser [47]	29.8	26.9	26.0	18.9	27.3	31.5	24.8	38.2	32.6	39.1	42.6	41.1	53.2	69.2	56.9	79.9	87.8	83.6	48.3
Tool-Integrated Spatial Agents																			
TIGeR [10]	29.1	27.7	26.0	25.8	27.8	33.0	28.3	26.7	28.3	52.9	45.7	49.8	48.7	38.8	46.3	75.2	**95.7**	84.5	47.3
GCA (ours)	**52.8**	**45.0**	**44.7**	**38.0**	**47.6**	82.0	**61.8**	**59.8**	**64.2**	**73.6**	**58.6**	**65.1**	61.7	61.9	61.8	**83.6**	90.8	**86.9**	**65.1**

to manage a suite of off-the-shelf foundation models for perception and computation [24, 33, 40–42]. For our primary experiments, we utilize Qwen3-VL-Thinking [32] as the central VLM. To assess the paradigm's generalizability, we also evaluate other leading VLMs in our ablation studies, including GLM-4.5V [13], GPT-4o [15], and *etc*. All open-source VLMs are deployed using the vLLM inference engine [17] for efficiency. The agent's architecture is built using Ray [28] for concurrent tool execution and LangGraph for robust state management.

Evaluation Benchmarks and Counterparts. We conduct comprehensive experiments on several spatial reasoning benchmarks. As our current toolbox is primarily designed for image-based inputs, we focus on evaluations that test complex spatial logic from single and multiple images, including MMSI-Bench [50], MindCube-tiny [54], OmniSpatial (Perspective Taking + Dynamic Reasoning) [16], SPBench [20] and CV-Bench [39]. For all benchmarks, we report both overall accuracy (%) and subcategory accuracy (%). We compare our paradigm against several counterparts, including baseline foundation VLMs [5, 13, 15, 32], training-based methods [6, 20, 26, 29, 37, 43, 45, 47] and tool-integrated agents [10].

4.2. Main Results

SOTA Performance. As shown in Table 1, GCA establishes a new state-of-the-art across a wide range of spatial reasoning benchmarks, achieving an average accuracy of 64.8%. Our geometrically-constrained paradigm surpasses the strongest foundation VLM baseline (Gemini-2.5-Pro [5] by 12%) and demonstrates a massive lead over other training-based (*e.g.*, SpatialLadder [20] by 27%) or agentic approaches (*e.g.*, TIGeR [10] by 38%). These results strongly validate that our strategy, centered on the $\mathcal{C}_{\text{task}}$, successfully bridges the VLM's semantic-to-geometric gap.

Effectiveness on Challenging Benchmarks. The advantage of our constrained paradigm is most pronounced on complex, multi-step spatial reasoning benchmarks. For example, on MMSI-Bench, the performance of even SOTA foundation VLMs remain severely limited. Considering its 4-choice questions, most counterparts perform near the 25% random-guess threshold. In contrast, GCA achieves an overall accuracy of 47.6%, surpassing the strongest VLM baseline (Gemini-2.5-Pro) by a 28% relative improvement. A similar trend is evident on other challenging benchmarks like MindCube-tiny, where GCA (64.2%) also significantly outperforms the top baselines. This superior performance stems directly from our paradigm. The introduction of $\mathcal{C}_{\text{task}}$ prevents the VLM from defaulting to flawed semantic shortcuts or falling into a lossy spatial imagination.

Generalizability Across Benchmarks. Our training-free paradigm also demonstrates superior generalizability compared to training-based specialists, which often suffer from biases inherent to their training data. For example, SpatialLadder [20] is fine-tuned on data originating from the same source as the SPBench, leading to a high in-domain score of 70.3%. However, its performance on out-of-domain benchmarks is suboptimal, where GCA consistently outperforms it, often by a margin of $\sim$20 points. A similar bias affects TIGeR [10]. While its tools theoretically support multi-view processing, the model is primarily trained on single-image tasks. Consequently, it performs well on single-image benchmarks like CV-Bench but fails on multi-view benchmarks such as MMSI-Bench. GCA, in contrast, is not

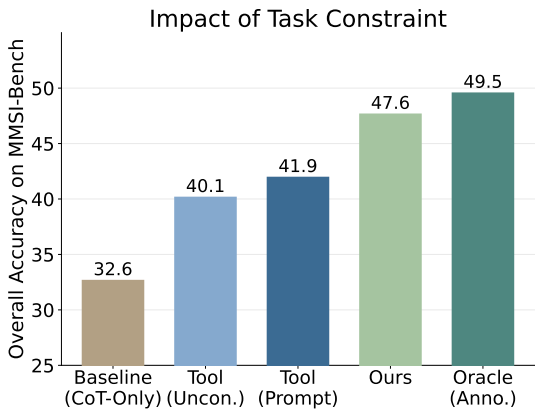

Figure 4. **Ablation Study on Formalization.** We compare our method in against several baselines: (1) no tool integration ("Baseline (CoT-Only)"), (2) unconstrained tool integration with ("Tool (Prompt)") or without ("Tool (Uncon.)") hints, (3) using a human-annotated $\mathcal{C}_{\text{task}}$ ("Oracle (Anno.)").

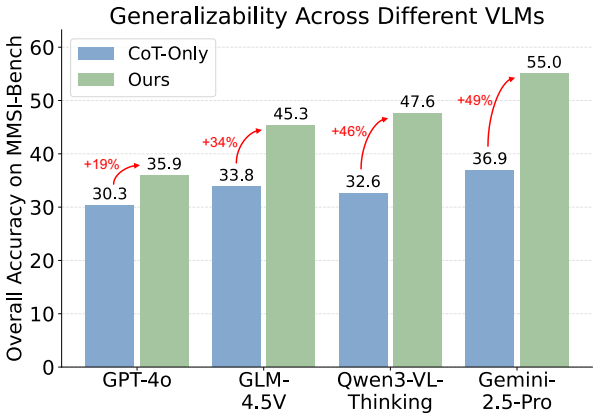

Figure 5. **Ablation Study on Generalizability across Different VLMs.** Our GCA achieves an average of 37% relative performance improvement across all tested foundation VLMs.

compromised by these training priors and leverages multi-view tools as dictated by the problem. This demonstrates that GCA, which forces the VLM to derive a geometrically sound task constraint for each new problem, provides a more generalizable pathway to spatial reasoning.

4.3. Ablation Study

In this section, we conduct extensive ablation studies to dissect the GCA paradigm and validate its core design. Our analysis aims to answer four critical questions. **(1)** How essential is the formal task constraint $\mathcal{C}_{\text{task}}$? **(2)** How generalizable is the GCA paradigm across different VLMs? **(3)** What is the contribution of each system component?

4.3.1. Formalization Analysis

We first investigate the necessity and impact of our core contribution, the $\mathcal{C}_{\text{task}}$ constraint, by comparing our method against different reasoning strategies in Figure 4. The results strongly confirm our central hypothesis. Simply prompting the VLM to "pay attention to the reference frame and objective in the query" ("Tool (Prompt)") only yields a negligible improvement on unconstrained tool integration. This empirically suggests that the VLM's unconstrained planning process remains fundamentally flawed and unreliable, even when weakly guided by hints. In comparison, the introduction of our formal $\mathcal{C}_{\text{task}}$ constraint ("Ours") delivers a substantial performance boost, far surpassing all unconstrained methods. This demonstrates that a deterministic and verifiable constraint is essential for bridging the VLM's semantic-to-geometric gap, as it forces the VLM to first establish what to solve before determining how to solve it. Furthermore, we explore the theoretical upper bound using a human-annotated oracle formalization ("Oracle (Anno.)"). The gap between our method (47.6%) and oracle (49.5%) is relative small. As revealed in Section 4.4, the $\mathcal{F}_{\text{formalize}}$ stage achieves ~70% accuracy, confirming the formalization task is well within the VLM's capabilities.

4.3.2. Generalizability Across VLMs

We assess the generalizability of our GCA paradigm by applying it to several leading foundation VLMs, including GLM-4.5V [13], GPT-4o [15], and Gemini-2.5-Pro [5]. As shown in Figure 5, GCA proves to be a highly generalizable architectural solution, substantially enhancing the spatial reasoning capabilities of every VLM tested compared to their CoT-only baselines. We observe that the magnitude of this enhancement appears to correlate strongly with the VLM's inherent agentic proficiency and their baseline spatial reasoning capability. It is most evident that Gemini-2.5-Pro, which holds the strongest CoT-only baseline on MMSI-Bench (36.9%), also achieves the most dramatic gain (+49%), rising to 55.0%. On the other hand, the improvement on GPT-4o, while significant, is more modest (+19%). We attribute it to its suboptimal agentic reasoning capability and coding skills. Through introduction of formal task constraint $\mathcal{C}_{\text{task}}$, our paradigm serves as a catalyst, successfully unlocking and guiding the VLM's powerful execution engine towards the robust spatial reasoning across a diverse set of SOTA models.

4.3.3. Component Contribution

We quantify the importance of each component in the GCA, as presented in Table 2. This analysis reveals improvements in two distinct parts. First, building a standard tool-integrated agent by adding tool integration (+4.2 points), knowledge-augmented code generation (KACG, +1.9 points), and visual feedback (+1.4 points) provides a cumulative +7.5 points gain over the CoT-only baseline. The second part, the introduction of $\mathcal{F}_{\text{formalize}}$, brings an additional massive improvement, increasing the overall accuracy by +7.5 points. This result strongly validates that con-

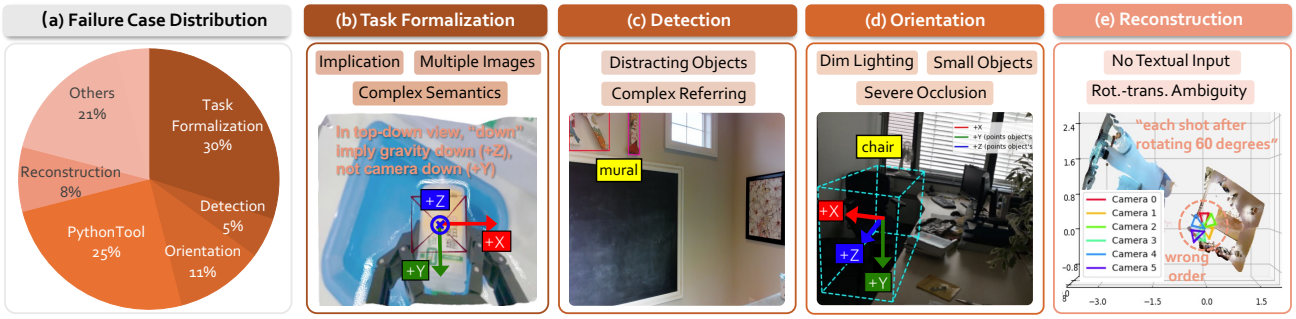

Figure 6. **Error Attribution and Failure Cases.** We provide a detailed error attribution analysis to identify the main failure modes within the VLM's reasoning trajectory.

Table 2. **Ablation Study on Each Component in GCA.** Here, "KACG" denotes applying knowledge-augmented code generation, and "Feedback" denotes applying the VLM to manage tool feedback and resolve ambiguity.

Tool Integration	KACG	Feedback	$\mathcal{C}_{\text{task}}$	MMSI-Bench
✗	✗	✗	✗	32.6
✓	✗	✗	✗	36.8
✓	✓	✗	✗	38.7
✓	✓	✓	✗	40.1
✓	✓	✓	✓	**47.6**

straining the VLM's planning via a formal $\mathcal{C}_{\text{task}}$ is essential to prevent flawed reasoning within its lossy semantic space.

4.4. Error Attribution and Failure Cases

A key advantage of GCA paradigm is its verifiable and interpretable nature, which allows us to trace the reasoning pathway and perform detailed error attribution. As shown in Figure 6 (a), this analysis pinpoints the current bottlenecks, attributing failures to either the VLM's initial formalization or the subsequent tool orchestration.

Errors in $\mathcal{F}_{\text{formalize}}$. Failures in the initial $\mathcal{F}_{\text{formalize}}$ stage account for 30% of all errors. Given this is the first step of the paradigm, it indicates the VLM achieves ∼70% accuracy in correctly formalizing the task constraint $\mathcal{C}_{\text{task}}$. A deeper analysis reveals these failures primarily lie in challenging cases involving complex semantics, ambiguity in multiple images, or ignored implications. For instance, as shown in Figure 6 (b), when asked about a top-down view, the VLM fails to grasp the query's implication that "down" referred to the direction of gravity, defaulting instead to "camera down" and establishing an incorrect reference frame.

Errors in $\mathcal{F}_{\text{compute}}$. The remaining 70% of errors occur during $\mathcal{F}_{\text{compute}}$ stage. Perception failures (∼24%) are a major bottleneck, particularly in "Reconstruction" and "Orientation". A typical reconstruction failure, shown in Figure 6 (e), is caused by the inability of the underlying VGGT [40] to accept textual input. The query's textual input, "each shot after rotating 60 degrees" provides a deterministic rotational sequence. However, the VGGT model, which cannot accept this textual input, parameterize the scene incorrectly, resulting in the "wrong order" of cameras and a flawed geometric foundation. Errors from "Python Tool" (25%) are also significant, often stemming from forgotten coordinate transformations or lacking nuanced problem-solving logic, such as identifying a principal direction. Besides, "Other" (21%) errors capture issues like incorrect parameter passing between tools, exhausting the predefined budget (*e.g.*, a maximum of 15 turns), and *etc*.

5. Conclusion

In this work, we introduce GCA, a training-free agentic paradigm designed to bridge the VLM's semantic-to-geometric gap in spatial reasoning. We address it through leveraging a formal task constraint, transforming the ambiguous spatial query into a deterministic mathematic problem with constraints, preventing the VLM reasoning about the geometric details within its lossy semantic space. As demonstrated experimentally, GCA establishes a new state-of-the-art on multiple challenging spatial reasoning benchmarks, showcasing a effective and generalizable pathway for robust spatial reasoning.

Limitations and Future Prospects. The GCA paradigm, involving iterative tool calls and VLM interactions, is computationally more costly than simple end-to-end CoT reasoning. However, this trade off yields a more robust and verifiable reasoning pathway. Furthermore, we believe the structured outputs from $\mathcal{F}_{\text{formalize}}$ and $\mathcal{F}_{\text{compute}}$ stages can serve as a valuable source of supervision for training more efficient end-to-end spatial VLMs in the future. Besides, current toolbox is primarily designed for image-based spatial reasoning. A key direction for future work is to extend this geometrically-constrained framework by incorporating tools for temporal reasoning, thereby addressing a broader range of spatial intelligence tasks.

Acknowledgement

This work was supported by National Natural Science Foundation of China (62132001), Capital's Funds for Health Improvement and Research (CFH 2024-2-40611), the Fundamental Research Funds for the Central Universities, and the Shanghai Artificial Intelligence Laboratory.

References

[1] Bikram Pratim Bhuyan, Amar Ramdane-Cherif, Ravi Tomar, and TP Singh. Neuro-symbolic artificial intelligence: a survey. *Neural Computing and Applications*, 36(21):12809–12844, 2024. 3

[2] Keshigeyan Chandrasegaran, Agrim Gupta, Lea M Hadzic, Taran Kota, Jimming He, Cristóbal Eyzaguirre, Zane Durante, Manling Li, Jiajun Wu, and Li Fei-Fei. Hourvideo: 1-hour video-language understanding. *Advances in Neural Information Processing Systems*, 37:53168–53197, 2024. 1

[3] Zeren Chen, Zhelun Shi, Xiaoya Lu, Lehan He, Sucheng Qian, Zhenfei Yin, Wanli Ouyang, Jing Shao, Yu Qiao, Cewu Lu, et al. Rh20t-p: A primitive-level robotic dataset towards composable generalization agents. *arXiv preprint arXiv:2403.19622*, 2024. 3

[4] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatial-rgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093, 2024. 2

[5] 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. 6, 7

[6] 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*, 2025. 1, 2, 6

[7] Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. Drive like a human: Rethinking autonomous driving with large language models. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pages 910–919. IEEE, 2024. 1

[8] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1), 2023. 5

[9] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18995–19012, 2022. 1

[10] Yi Han, Cheng Chi, Enshen Zhou, Shanyu Rong, Jingkun An, Pengwei Wang, Zhongyuan Wang, Lu Sheng, and Shanghang Zhang. Tiger: Tool-integrated geometric reasoning in vision-language models for robotics. *arXiv preprint arXiv:2510.07181*, 2025. 1, 2, 4, 6

[11] Yilun Hao, Yang Zhang, and Chuchu Fan. Planning anything with rigor: General-purpose zero-shot planning with llm-based formalized programming. In *The Thirteenth International Conference on Learning Representations*, 2024. 3

[12] Pascal Hitzler and Md Kamruzzaman Sarker. *Neuro-symbolic artificial intelligence: The state of the art*. IOS press, 2022. 3

[13] Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv e-prints*, pages arXiv–2507, 2025. 6, 7

[14] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. In *Conference on Robot Learning*, pages 4573–4602. PMLR, 2025. 3, 4

[15] 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. 1, 2, 6, 7

[16] 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. *arXiv preprint arXiv:2506.03135*, 2025. 1, 2, 6

[17] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626, 2023. 6

[18] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020. 5

[19] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025. 2

[20] 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. *arXiv preprint arXiv:2510.08531*, 2025. 1, 2, 6

[21] Songtao Li and Hao Tang. Multimodal alignment and fusion: A survey. *arXiv preprint arXiv:2411.17040*, 2024. 1

[22] Zefu Lin, Rongxu Cui, Chen Hanning, Xiangyu Wang, Junjia Xu, Xiaojuan Jin, Chen Wenbo, Hui Zhou, Lue Fan, Wenling Li, et al. Embodiedcoder: Parameterized embodied mobile manipulation via modern coding model. *arXiv preprint arXiv:2510.06207*, 2025. 2

[23] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023. 3, 4

[24] 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*, pages 38–55. Springer, 2024. 6

[25] 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. 2

[26] Wufei Ma, Luoxin Ye, Celso M de Melo, Alan Yuille, and Jieneng Chen. Spatialllm: A compound 3d-informed design towards spatially-intelligent large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17249–17260, 2025. 1, 2, 6

[27] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023. 1

[28] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. Ray: A distributed framework for emerging {AI} applications. In *13th USENIX symposium on operating systems design and implementation (OSDI 18)*, pages 561–577, 2018. 6

[29] 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. 1, 2, 6

[30] Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. 3

[31] Mingjie Pan, Jiyao Zhang, Tianshu Wu, Yinghao Zhao, Wenlong Gao, and Hao Dong. Omnimanip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17359–17369, 2025. 3

[32] QwenTeam. Qwen3-vl: Sharper vision, deeper thought, broader action. https://qwen.ai/blog?id=99f0335c4ad9ff6153e517418d48535ab6d8afef&from=research.latest-advancements-list, 2025. 2, 6

[33] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *The Thirteenth International Conference on Learning Representations*, 2024. 6

[34] Stuart Russell, Peter Norvig, and Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25(27):79–80, 1995. 3

[35] Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15768–15780, 2025. 2

[36] Oren Sultan, Eitan Stern, and Dafna Shahaf. Towards reliable proof generation with llms: A neuro-symbolic approach. *arXiv preprint arXiv:2505.14479*, 2025. 3

[37] 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. 1, 6

[38] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, XianPeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. In *Conference on Robot Learning*, pages 4698–4726. PMLR, 2025. 1

[39] Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024. 2, 6

[40] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 5, 6, 8

[41] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5261–5271, 2025.

[42] Zehan Wang, Ziang Zhang, Tianyu Pang, Chao Du, Hengshuang Zhao, and Zhou Zhao. Orient anything: Learning robust object orientation estimation from rendering 3d models. In *Forty-second International Conference on Machine Learning*, 2024. 6

[43] 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*, 2025. 1, 2, 6

[44] Haoning Wu, Xiao Huang, Yaohui Chen, Ya Zhang, Yanfeng Wang, and Weidi Xie. Spatialscore: Towards unified evaluation for multimodal spatial understanding. *arXiv preprint arXiv:2505.17012*, 2025. 1, 2, 4

[45] 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*, 2025. 1, 2, 6

[46] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via

large language model. *IEEE Robotics and Automation Letters*, 2024. 1

[47] 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. *arXiv preprint arXiv:2510.11027*, 2025. 1, 2, 6

[48] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643, 2025. 1, 2

[49] Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. In *Forty-second International Conference on Machine Learning*, 2025. 1

[50] 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. *arXiv preprint arXiv:2505.23764*, 2025. 1, 2, 4, 6

[51] Zhutian Yang, Caelan Garrett, Dieter Fox, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Guiding long-horizon task and motion planning with vision language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16847–16853. IEEE, 2025. 3

[52] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022. 3

[53] Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. Satlm: Satisfiability-aided language models using declarative prompting. *Advances in Neural Information Processing Systems*, 36:45548–45580, 2023. 3

[54] Baiqiao Yin, Qineng Wang, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, et al. Spatial mental modeling from limited views. In *Structural Priors for Vision Workshop at ICCV'25*, 2025. 1, 2, 6

[55] Songsong Yu, Yuxin Chen, Hao Ju, Lianjie Jia, Fuxi Zhang, Shaofei Huang, Yuhan Wu, Rundi Cui, Binghao Ran, Zaibin Zhang, et al. How far are vlms from visual spatial intelligence? a benchmark-driven perspective. *arXiv preprint arXiv:2509.18905*, 2025. 1, 2

[56] Enshen Zhou, Qi Su, Cheng Chi, Zhizheng Zhang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, and He Wang. Code-as-monitor: Constraint-aware visual programming for reactive and proactive robotic failure detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6919–6929, 2025. 3

[57] Filippo Ziliotto, Tommaso Campari, Luciano Serafini, and Lamberto Ballan. Tango: Training-free embodied ai agents for open-world tasks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24603–24613, 2025. 1, 2