Thinking with Imagination: Agentic Visual Spatial Reasoning with World Simulators

Chenming Zhu[1,2*], **Jingli Lin**[2,3*], **Yilin Long**[2,4], **Peizhou Cao**[2,5],
Tai Wang[2], **Jiangmiao Pang**[2], **Xihui Liu**[1†]
[1]The University of Hong Kong, [2]Shanghai AI Laboratory
[3]Shanghai Jiao Tong University, [4]Fudan University, [5]Beihang University
chaimzhu@connect.hku.hk

Abstract

While Vision-Language Models (VLMs) have shown strong visual reasoning capabilities, their spatial reasoning abilities remain largely constrained to the observed images and text-oriented chain-of-thought. They often struggle to infer unobserved layouts, maintain cross-view consistency, and reason from alternative viewpoints when only limited egocentric observations are available. In this work, we study this problem as *thinking with imagination*, where a VLM actively acquires imagined visual evidence by interacting with a world simulator during reasoning. We propose **Astra**, an agentic spatial reasoning framework that empowers VLMs with action-conditioned visual imagination. Specifically, Astra couples Astra-VL, an RL-trained VLM policy, with Astra-WM, a Bagel-based world simulator that generates novel-view observations from context images and natural-language camera motions. To provide reliable imagined evidence, Astra-WM is trained with view consistency tuning to improve pose and content consistency across views. In the RL stage, we propose a world-simulator-in-the-loop two-phase RL curriculum to stabilize tool-use exploration and advance the model's ability to invoke the simulator only when imagined observations improve over direct answering. Experiments show complementary benefits from simulator tuning and policy training: Astra-WM improves simulator-augmented Gemini-3-Flash [15] on MMSI-Bench from 45.1 to 49.5, while Astra-VL improves the Qwen3-VL backbone from 29.8 to 38.8 on MMSI-Bench and from 36.8 to 42.7 on MindCube-tiny. These results show that imagined observations can provide useful spatial evidence, with gains depending on simulator consistency, action selection, and evidence integration.

1 Introduction

Spatial reasoning from multi-view images requires more than recognizing visible objects [41, 48, 24, 23, 21, 27, 45]. When only a few egocentric observations are available, an agent must infer unobserved layout, maintain cross-view consistency, and reason from alternative perspectives. A spatial relation may be ambiguous from the current views but become evident after a small viewpoint change, such as moving forward to inspect a camera-object relation, rotating to check viewpoint alignment, or looking beyond the current field of view to infer scene layout. Humans naturally handle such uncertainty by maintaining spatial mental models that can be updated and mentally manipulated beyond the observed pixels. In contrast, current vision-language models (VLMs) are largely tied to

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

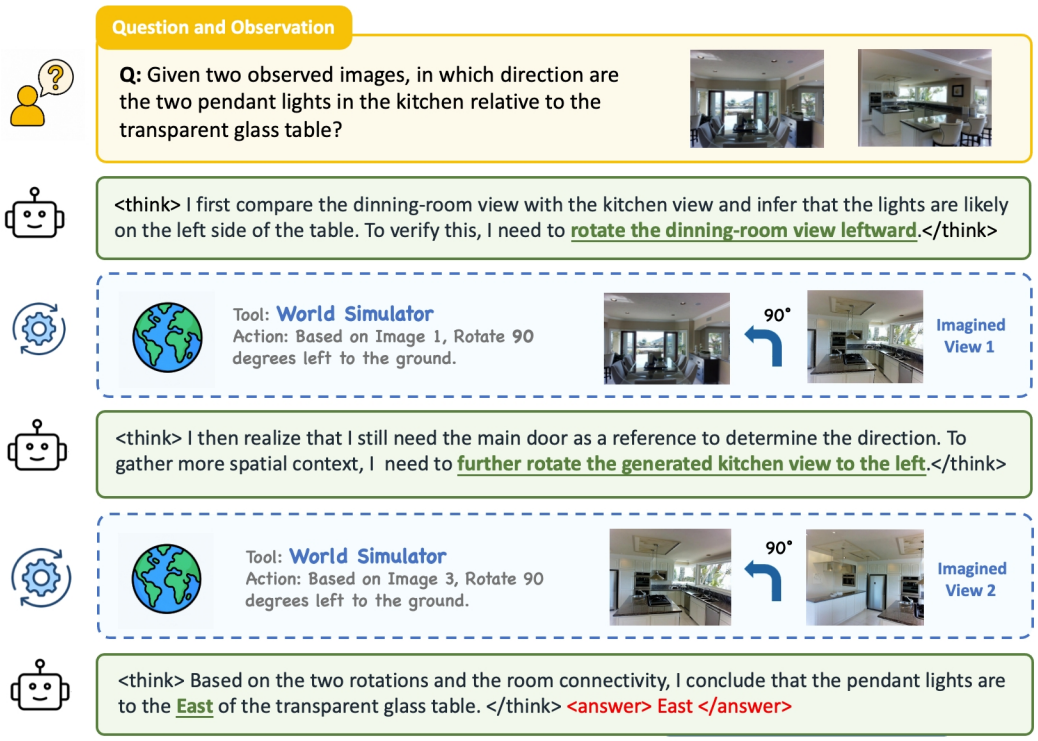


Figure 1: Reasoning trajectory of **Astra**. The agent queries a world simulator for action-conditioned observations and reasons over the original and imagined views.

the images they are given, and often struggle to reason about spatial evidence that is missing from the input observations.

Recent studies have explored spatial reasoning scaffolds such as reasoning chains, intermediate views, and cognitive maps.[41, 46] These approaches suggest that VLMs may benefit from explicit spatial reasoning processes, but most of them still reason over a fixed visual context or rely on pre-defined intermediate representations. Recent world-model approaches also explore active viewpoint acquisition [49, 6, 51]; the remaining challenge is to make this interaction spatially reliable and useful for the answering policy. This motivates an interactive formulation of spatial reasoning, where the model can seek additional evidence instead of only interpreting the observations it already has. In this paper, we study this problem as *thinking with imagination*: a VLM issues camera-motion queries to a world simulator, obtains imagined novel-view observations, and continues reasoning over both observed and imagined evidence. This transforms spatial mental modeling from a purely internal reconstruction problem into an interactive evidence-acquisition problem.

However, simply connecting an existing VLM to a generation model does not guarantee useful spatial evidence. First, the world simulator must provide reliable imagined evidence: its outputs should be not only visually plausible, but also spatially consistent with the requested camera motion and the original scene. Otherwise, the generated view may mislead the reasoning policy. Second, the VLM policy must govern simulator use, including when additional evidence is needed, what viewpoint to request, and how to ground the returned observation in the original context. Our preliminary analysis confirms both bottlenecks. Off-the-shelf generation models can produce visually plausible images, but often lack spatial consistency to serve as reliable world simulators. Meanwhile, naively connecting Qwen3-VL to a world simulator can degrade performance, motivating separate analysis of invocation, action selection, and use of the returned observation.

To address these challenges, we propose **Astra**, an agentic spatial reasoning framework that integrates an agentic VLM policy with an action-conditioned world simulator. The framework consists of two components: Astra-VL, a Qwen3-VL-based agentic reasoning model, and Astra-WM, a Bagel-

based world simulator for generating action-conditioned novel views. During reasoning, Astra-VL adaptively decides when additional visual evidence is needed, plans a camera-motion query, invokes Astra-WM, and integrates the returned imagined observation into subsequent reasoning. This enables Astra to dynamically acquire alternative-view evidence through iterative reasoning and planning, rather than relying only on the initially observed images. We train both components to support reliable agentic imagination. For Astra-WM, we construct quality-verified SFT data for world simulation and fine-tune Bagel with View Consistency Tuning, enabling it to generate spatially consistent novel views conditioned on natural-language camera-motion instructions. To verify the simulator, we design pose-consistency and content-consistency evaluations, which measure whether generated views follow the requested motion and preserve scene content and spatial layout. For Astra-VL, we construct the Spatial QA data corpus and train the policy with a world-simulator-in-the-loop two-phase RL curriculum. The first phase teaches valid simulator invocation and keeps tool-use trajectories in the on-policy distribution, while the second phase encourages selective imagination by comparing tool-augmented reasoning with direct no-tool answering.

Our experiments show that world-simulator-augmented spatial reasoning requires both reliable imagined evidence and a policy that can govern its use. First, simulator-quality analysis shows that directly using off-the-shelf Bagel brings limited benefit, whereas Astra-WM achieves stronger pose and content consistency and improves simulator-augmented Gemini-3-Flash on MMSI-Bench from 45.1 to 49.5. A matched simulator swap with the trained policy likewise favors view consistency tuning over the original Bagel simulator. Second, simulator access alone does not reliably improve the evaluated open-source backbones. This is not a universal requirement for RL: Gemini benefits from the simulator without additional policy training. In contrast, Astra-VL learns world-simulator interaction through RL, and the full Astra framework improves the Qwen3-VL backbone from 29.8 to 38.8 on MMSI-Bench and from 36.8 to 42.7 on MindCube-tiny. Together, these results show that useful imagination depends on both the quality of the world simulator and the agentic policy that decides when, where, and how to use it.

2 Related Work

VLMs for Spatial Intelligence. Spatial reasoning, which requires understanding and manipulating 3D relationships from visual observations, remains a fundamental challenge for vision-language models (VLMs). Existing efforts mainly improve spatial capability through large-scale training on specialized spatial datasets or by injecting geometric priors into model architectures, such as explicit 3D representations, depth cues, or structure-aware visual features [9, 47, 55, 40, 13, 19]. While effective, these approaches often rely on static visual inputs and supervised spatial annotations, making it difficult for models to actively resolve ambiguity through interaction. Another line of work equips VLMs with external tools to perform deterministic geometric computation or visual analysis [8, 16, 53]. For instance, tool-augmented agents can decompose a spatial query into a sequence of module calls and use the returned results for answer prediction. However, these methods typically treat tools as passive solvers for predefined geometric operations, rather than as interactive environments that can provide new observations conditioned on the model's own actions. In contrast, our work studies agentic spatial reasoning with the simulator, where the VLM learns to decide when and how to act, query a world simulator for imagined observations, and integrate these observations into multi-turn reasoning to reduce spatial uncertainty.

Thinking with Images. The thinking with images paradigm [31, 17, 37, 52, 36] has recently advanced agentic visual reasoning by enabling VLMs to interleave textual reasoning with iterative visual operations. Representative systems such as DeepEyes [54] show that reinforcement learning can induce such image-text reasoning behaviors and improve fine-grained visual understanding. Pixel Reasoner [39] identifies a learning-trap problem, where models avoid using immature visual tools, and therefore adopts a two-stage strategy to bootstrap tool use before encouraging exploratory reasoning. Similarly, Mini-o3 [20] finds that hard visual search requires deep trajectories that are difficult to discover through pure RL, motivating staged training for multi-turn capability activation. SenseNova-MARS [9] further extend the tools from perceptual image operations to open-web access and external knowledge fetching. While these methods make important progress, their tools mainly operate in image space, such as inspecting or manipulating existing visual inputs. In contrast, our work studies agentic spatial reasoning with the world simulator, where the model actively plans actions and uses action-conditioned observations to resolve visual spatial uncertainty.

World Models for Visual Imagination. MindJourney [49] searches imagined camera trajectories with a controllable world model without policy fine-tuning. SpatialDreamer [6] learns active imagery through tree-structured sampling and geometric step-level rewards. AVIC [51] assesses evidence sufficiency and adaptively plans world-model exploration; its AVIC-R variant learns both gating and planning with GRPO, combining QA correctness with imagination-cost penalties while keeping the world model and QA model frozen. In navigation, Perincherry et al. [34] generate instruction-conditioned landmark images as navigational cues, rather than camera-action-conditioned views for spatial QA. Our work shares the goal of acquiring useful imagined evidence and focuses on view consistency tuning of a Bagel-based simulator coupled with a two-phase curriculum that first maintains tool exploration, then rewards gains relative to direct answering. Matched controls further separate simulator quality, action proposal, and answer integration, without establishing superiority over methods evaluated with different components and protocols.

3 Method

In this section, we describe how **Astra** couples a spatially consistent world simulator with an agentic VLM policy for interactive spatial reasoning. We first formulate visual spatial reasoning as an interactive decision process in Sec. 3.1. We then introduce Astra-WM, a Bagel-based world simulator trained with view consistency tuning to produce spatially consistent imagined observations, in Sec. 3.2. Finally, we present the world-simulator-in-the-loop two-phase RL curriculum for Astra-VL in Sec. 3.3, followed by the data construction pipeline in Sec. 3.4.

3.1 Problem Formulation

Task and Objective. Given a spatial question q and an initial set of context images $O_0 = \{I_1, \ldots, I_n\}$, the agent must produce the correct answer by reasoning over the available visual evidence. At each turn, it can either answer directly or query a world simulator for an imagined observation from an alternative viewpoint.

Observation Space. The interaction trajectory $\mathcal{T}_t$ records all information available to the agent up to turn t, including the question, original context images, previous reasoning steps, simulator actions, and simulator outputs. We therefore represent the agent state directly as the trajectory:

$$s_t = \mathcal{T}_t.$$

After t simulator calls, the trajectory contains t imagined observations $\{\hat{I}_1, \ldots, \hat{I}_t\}$. If the agent invokes the simulator at state s_t, the simulator returns a new tool observation

$$o_{t+1} = (m_{t+1}, \hat{I}_{t+1}),$$

where $\hat{I}_{t+1}$ is the generated novel view and m_{t+1} describes the reference image and executed camera motion. The trajectory is then updated by appending the new observation:

$$\mathcal{T}_{t+1} = \mathcal{T}_t \circ o_{t+1},$$

where $\circ$ denotes ordered concatenation. This motion provenance is important because the policy must distinguish original context images from imagined views and reason about which reference viewpoint and camera motion each generated observation corresponds to.

Action Space. At each turn, the agent first produces its reasoning process and then selects one of two high-level actions: INVOKE, which queries the world simulator for an imagined observation, or ANSWER, which outputs the final answer and terminates the trajectory. Thus, the action space is

$$a_t \in \{\textsc{Invoke}(\rho_t), \textsc{Answer}(y_t)\},$$

where ρ_t denotes the parameters of a simulator query and y_t denotes the final answer.

An INVOKE action is parameterized by a reference image, a motion type, and a motion magnitude. The motion type is drawn from a compact camera-control vocabulary, including lateral movement, forward/backward movement, yaw rotation, vertical movement, and pitch rotation. These parameters are converted into a natural-language camera-motion instruction and sent to the world simulator.

An ANSWER action produces the final response and ends the trajectory. Trajectories with malformed simulator queries or missing reasoning steps in a turn are treated as invalid.

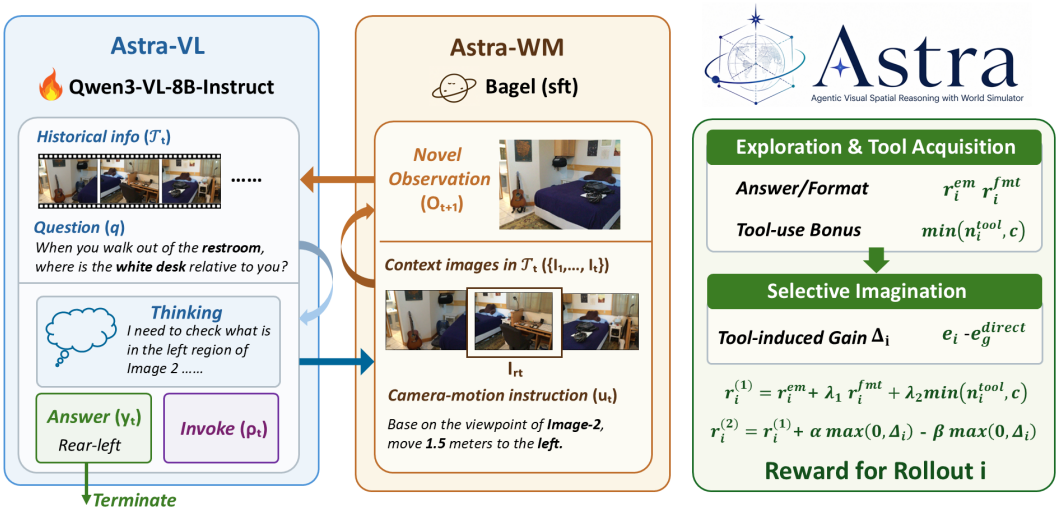


Figure 2: **Astra** consists of two components: the Astra Vision-Language Model (VL) and the Astra World Model (WM). The overview illustrates the input-output details of both models during training and inference, as well as the two-phase reinforcement learning training pipeline.

3.2 View Consistency Tuning

Astra-WM is the world-simulation component of **Astra**, serving as the external visual imagination tool for Astra-VL. Given the context images, a reference image, and a camera-motion query from the agent, Astra-WM generates a novel-view observation that approximates what the scene would look like after executing the requested motion. Formally, when Astra-VL issues an INVOKE action with query ρ_t, the query specifies a reference image index r_t, a motion type, and a motion magnitude. We convert ρ_t into a natural-language camera-motion instruction u_t and generate the imagined view as

$$\hat{I}_{t+1} = \mathcal{W}(\mathcal{I}_t, r_t, u_t),$$

where $\mathcal{I}_t$ denotes the context images available at this turn, r_t indexes the reference image selected by the agent, and $\mathcal{W}$ denotes the Bagel-based Astra-WM.

A useful world simulator must provide spatially consistent evidence rather than merely visually plausible images: the generated view should preserve scene identity, follow the requested camera motion, and maintain the relative layout of visible objects across viewpoints. However, off-the-shelf Bagel is not explicitly optimized for this role, and our consistency evaluation shows that its generated views often fail to follow the requested motion or preserve scene content reliably, which can mislead Astra-VL when simulator outputs are treated as visual evidence.

To address this limitation, we introduce *view consistency tuning*, where Bagel is fine-tuned on quality-verified world-simulator SFT data to generate spatially consistent novel views conditioned on context images and natural-language camera-motion instructions. This converts Bagel from a generic multimodal generation model into Astra-WM, an action-conditioned world simulator that better aligns imagined observations with the requested viewpoint change and provides more reliable evidence for downstream spatial reasoning.

3.3 Two-Phase Reinforcement Learning

We train Astra-VL with a world-simulator-in-the-loop two-phase RL pipeline: the first phase keeps simulator-invoking trajectories in the on-policy distribution and teaches valid tool interaction, while the second phase encourages the policy to use the simulator only when the imagined observation improves over direct answering. For rollout i, let r_i^{em} denote the final-answer exact-match reward, r_i^{fmt} denote the format-validity reward, and n_i^{tool} denote the number of simulator calls in the trajectory. Let λ_{fmt} and λ_{use} be the weights for the format reward and simulator-use reward, respectively, and let c denote the maximum number of tool calls eligible for the usage bonus.

Phase 1: Exploration and Tool Acquisition. The first phase trains Astra-VL to acquire the mechanics of valid interaction with Astra-WM. At this stage, the goal is not yet perfectly selective imagination, but to prevent the policy from collapsing to short direct answers before it has learned how to invoke the simulator. We therefore combine the answer reward and format reward with a small capped simulator-use bonus:

$$r_i^{(1)} = r_i^{\text{em}} + \lambda_{\text{fmt}} r_i^{\text{fmt}} + \lambda_{\text{use}} \min(n_i^{\text{tool}}, c).$$

The cap prevents the model from being rewarded for repeatedly calling the simulator, while still encouraging valid simulator interaction.

Phase 2: Selective Imagination. The second phase shifts the objective from invoking the simulator to using it only when it improves spatial reasoning. For each rollout group g, we additionally evaluate a direct no-tool baseline using a prompt that forbids simulator calls. This baseline is used only to estimate the benefit of simulator use and is not included in the policy update. For a normal rollout i in group g, let e_i be its exact-match score after possible simulator use, and let e_g^{direct} be the exact-match score of the corresponding no-tool baseline. We define the tool-induced gain as

$$\Delta_i = e_i - e_g^{\text{direct}}.$$

The phase-two reward combines answer correctness, format validity, optional simulator-use shaping, and the relative gain over direct answering:

$$r_i^{(2)} = r_i^{\text{em}} + \lambda_{\text{fmt}} r_i^{\text{fmt}} + \lambda_{\text{use}} \min(n_i^{\text{tool}}, c) + \alpha \max(0, \Delta_i) - \beta \max(0, -\Delta_i).$$

Here, α rewards trajectories whose simulator interaction improves over direct answering, while β penalizes trajectories whose simulator use hurts the answer.

3.4 Data Construction

World Simulator SFT Data. Training Astra-WM requires action-conditioned novel-view supervision in the form of context images, a reference image, a camera-motion instruction, and the corresponding target observation. We construct 544k quality-verified SFT samples from posed multi-view indoor scenes, including IsaacSim [29], ScanNet++ [50], ScanNet [10], Matterport3D [7], DL3DV [25] and ARKitScenes [4]. These samples train Astra-WM to synthesize target views conditioned on scene context and explicit camera-motion queries, matching the interface used by Astra-VL during reasoning. More details are provided in our Appendix.

Agentic RL Data. To enhance Qwen3-VL's ability to invoke tools for improved spatial reasoning, we construct an agentic reinforcement learning dataset from multiple spatial reasoning sources. Specifically, we downsample raw samples from different spatial reasoning categories in SenseNova-800K [9], VST-500K [47], and the training splits of our self-constructed Hard-UMMQA dataset. For each sample, we query Qwen3-VL-8B-Instruct, the model we aim to improve, five times under a high sampling temperature of 1.5. We then retain only the samples for which Qwen3-VL-8B-Instruct produces correct answers in no more than one out of the five trials. These retained examples are regarded as challenging samples and are used for RL training, resulting in 6k final training samples.

4 Experiments

4.1 Implementation Details

Model and Training Setup. Our Astra-VL is initialized from a Qwen3-VL-8B checkpoint and trained only with the RL stage. We implement RL training with veRL and optimize the policy using GRPO. During training, we deploy the Astra-WM through vLLM-Omni [33, 35] as the online world simulator. For RL training, we use a prompt batch size of 128, a rollout group size of 4, a PPO mini-batch size of 128, and a learning rate of 1×10^{-5} for 5 epochs. The vision tower is frozen, while the policy is trained with bfloat16 FSDP and gradient checkpointing. We adopt the Clip-Higher strategy with $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$, and disable the explicit KL penalty. Each trajectory supports up to 3 tool rounds and 10 assistant turns. We set $\lambda_{\text{fmt}} = 0.5$, $\alpha = 0.1$, $\beta = 0.03$, $c = 1$, and $\lambda_{\text{use}} = 0.02$. Empirically, this exposes a key tradeoff in agentic spatial RL: sparse gain rewards can cause tool use to collapse, whereas dense usage rewards preserve exploration but may encourage over-imagination.

Table 1: **Experimental Results on Spatial Reasoning Benchmarks.** We compare *Direct Answer*, *Forced Tool-Use*, and *Agentic Tool-Use* settings. Values in parentheses denote absolute changes over the corresponding *Direct Answer* result of the same model. **All** denotes the average accuracy for each benchmark. More details on the benchmark subcategories, such as **PR.**, are provided in the Appendix.

Type	Model	MMSI-Bench					MindCube-tiny			
		PR.	Attr.	Mot.	MSR	All	Rot.	Ard.	Amg.	All
Direct Answer										
Open-source	Qwen3-VL-8B-Instruct [3]	30.8	30.1	27.7	28.1	29.8	53.6	38.0	31.1	36.8
	Qwen3-VL-30B-Instruct [3]	31.2	35.8	25.9	29.1	30.6	39.9	47.5	38.5	41.8
	Bagel-7B-MoT [12]	33.5	27.7	25.3	30.8	31.0	34.5	31.4	42.8	34.7
	SpatialLLM [28]	24.5	23.1	22.7	30.8	25.3	34.0	26.8	33.0	31.1
	Spatial-MLLM [42]	28.5	25.4	18.0	26.3	26.1	33.8	34.5	28.3	32.1
	SpatialLadder [22]	30.3	23.3	16.0	21.2	25.4	30.5	39.8	47.8	42.3
	SpaceR [32]	29.1	29.4	21.9	22.5	26.9	29.8	30.0	26.8	28.3
	Video-R1 [14]	30.5	25.4	22.0	26.8	27.8	30.0	30.5	41.3	35.8
	RoboBrain-2.0 [38]	28.9	28.8	22.5	28.0	28.9	29.7	35.8	45.2	39.6
	VILASR [43]	35.9	26.0	21.0	23.2	29.8	34.4	25.7	29.4	29.1
	VLaser [44]	29.8	26.9	26.0	18.9	27.3	31.5	24.8	38.2	32.6
Proprietary	GLM-4.5V [18]	35.6	36.9	29.3	30.3	33.8	60.0	25.5	42.2	39.6
	GPT-4o [30]	28.0	32.3	36.0	30.8	30.3	33.5	35.0	37.2	35.8
	Gemini-2.5-Pro [11]	39.0	36.2	33.3	34.3	36.9	89.5	54.5	48.8	57.5
	Gemini-3-Flash [15]	45.6	45.4	44.0	46.0	45.4	93.0	72.0	61.7	70.5
Forced Tool-Use (zero-shot)										
Open-source	Qwen3-VL-8B-Instruct [3]	30.4 (-0.4)	29.5 (-0.6)	19.6 (-8.1)	30.8 (+2.7)	28.6 (-1.2)	31.1 (-22.5)	23.7 (-14.3)	26.8 (-4.3)	27.6 (-9.2)
	Qwen3-VL-30B-Instruct [3]	31.5 (+0.3)	28.7 (-7.1)	21.6 (-4.3)	28.1 (-1.0)	28.9 (-1.7)	34.7 (-5.2)	32.7 (-14.8)	38.1 (-0.4)	35.7 (-6.1)
	Bagel-7B-MoT [12]	31.3 (-2.2)	25.6 (-2.1)	24.7 (-0.6)	28.7 (-2.1)	29.7 (-1.3)	33.9 (-0.6)	26.8 (-4.6)	31.8 (-11.0)	29.2 (-5.5)
Proprietary	Gemini-3-Flash [15]	50.4 (+4.8)	51.5 (+6.1)	43.4 (-0.6)	50.3 (+4.3)	49.5 (+4.1)	93.0 (+0.0)	70.3 (-1.7)	65.0 (+3.3)	72.7 (+2.2)
Agentic Tool-Use										
Open-source	Astra $_{\text{Qwen3-VL-8B-Instruct}}$	**42.3** (+11.5)	**41.0** (+10.9)	**32.1** (+4.4)	**33.6** (+5.5)	**38.8** (+9.0)	**60.1** (+6.5)	**43.5** (+5.5)	**36.8** (+5.7)	**42.7** (+5.9)

Evaluation setup. We evaluate spatial reasoning on two held-out benchmarks: MMSI-Bench [48] and the MindCube-tiny split [41]. MMSI-Bench contains 1,000 multi-view spatial reasoning examples across diverse spatial-relation categories, while MindCube-tiny evaluates multi-view spatial reasoning in structured 3D environments. We report exact-match answer accuracy (EM) as the main task metric. For agent behavior, we report the tool-call rate and the average number of tool calls. Neither evaluation benchmark is used for RL training or hard-example selection. For world-simulator quality, we report *pose consistency*, which measures whether generated views follow the requested camera motion, and *content consistency*, which measures object-level precision, recall, and topology consistency to assess scene-content and layout preservation. More details about benchmark subcategories and simulator-quality metrics are provided in the Appendix.

Workflow Modes. We evaluate our model against several strong baselines. These include proprietary models, such as GLM-4.5V, Gemini-3-Flash, as well as open-source general VLMs and spatial VLMs. We test the models under the following three workflow settings, which control how a model is allowed to access the world simulator:

- *Direct Answer*: The model answers directly from the original context images without invoking the world simulator.
- *Forced Tool-Use*: The model is required by an explicit system prompt to invoke the world simulator for a predefined number of maximum interaction steps.
- *Agentic Tool-Use*: The model autonomously decides when and how to interact with the world simulator in the rollout reasoning process.

4.2 Main Results

Effectiveness of Astra-WM. We first evaluate whether generated observations from the world simulator provide useful evidence for spatial reasoning under the *Forced Tool-Use* setting. In this setting, the reasoning model is required to invoke the world simulator for a predefined number of steps before producing the final answer. We evaluate Gemini-3-Flash with different world simulators under the forced two-step tool-use setting. As shown in Tab. 2, off-the-shelf Bagel provides only limited gains, whereas our trained Astra-WM substantially improves simulator-augmented Gemini-3-Flash on MMSI-Bench, increasing the overall accuracy from 0.451 to 0.495 and the spatial-relation

Table 2: **Simulator Quality and Zero-Shot Simulator-Augmented Reasoning on MMSI-Bench.**

Model	Pose Cons.	Content Cons.	All	PR.	Cam.–Obj.	Cam.–Reg.	Cam.–Cam.	Obj.–Obj.	Obj.–Reg.	Reg.–Reg.
Gemini-3-Flash	–	–	0.451	0.458	0.442	**0.602**	0.430	0.436	0.447	0.395
+ Bagel	9.0/3.0	0.356/0.396/0.102	0.458	0.469	**0.523**	0.434	0.443	0.483	0.459	0.468
+ Astra-WM$_{30k}$	**72.5**/70.5	0.532/0.560/0.230	0.463	0.471	0.510	0.470	0.411	0.511	0.458	0.470
+ Astra-WM$_{60k}$	69.0/**75.0**	**0.534/0.561/0.234**	**0.495**	**0.504**	0.500	0.554	**0.554**	**0.522**	**0.470**	**0.494**

Table 3: **Effectiveness of the Two-Phase RL Curriculum on MMSI-Bench.**

Type	Method	Tool Behavior		MMSI-Bench				
		Tool rate	Calls/row	PR.	Attr.	Mot.	MSR	All
Single-stage	Tool-gain only ($\lambda_{\text{use}} = 0$)	4.9	0.049	36.5	36.8	28.4	31.2	34.3
	Usage bonus only ($\lambda_{\text{use}} = 0.02$)	98.1	1.400	39.4	38.7	30.2	32.5	36.1
Two-phase	Phase 1 only	98.0	1.120	40.1	39.2	30.8	32.9	36.8
	Phase 1 → Phase 2 (full)	**61.5**	**0.780**	**42.3**	**41.0**	**32.1**	**33.6**	**38.8**

accuracy from 0.458 to 0.504. This shows that imagined observations can provide useful evidence for downstream spatial reasoning, but their benefit depends on the spatial consistency of the simulator. However, a reliable simulator alone is not sufficient. Prompted forced tool-use requires the model to interact with the simulator mechanically, but does not teach it whether additional observations are needed, which viewpoint would be most informative, or how the returned observation should be grounded in the original context. As shown in Tab. 1, open-source VLMs such as Qwen3-VL and Bagel can underperform their direct-answer counterparts under forced tool-use, indicating that they often fail to generate useful simulator actions or integrate imagined observations effectively.

Effectiveness of Astra-VL. We next study whether an open-source VLM can be trained to use the world simulator in an agentic manner. Unlike forced tool-use, which prescribes a fixed number of simulator interactions, **Astra** makes tool-use decisions conditioned on the current reasoning state. This requires the policy to solve three coupled problems: deciding whether additional evidence is needed, selecting an informative camera-motion query, and grounding the returned imagined observation in the original context before answering. As shown in Tab. 1, simply prompting Qwen3-VL to use the simulator does not reliably improve performance and can even underperform direct answering, suggesting that simulator access alone is insufficient for open-source VLMs. In contrast, after world-simulator-in-the-loop RL training, Astra-VL learns to interact with the simulator effectively, and the full **Astra** framework improves the Qwen3-VL-8B backbone from 29.8 to 38.8 on MMSI-Bench and from 36.8 to 42.7 on MindCube. These gains indicate that the improvement comes not from tool access alone, but from learning an agentic policy that can decide when to imagine, where to query, and how to use imagined evidence.

4.3 Ablation Study

We ablate three factors that determine the effectiveness of **Astra**: the spatial consistency of the world simulator, the reward design for maintaining useful tool exploration, and the inference-time control over simulator interaction. Together, these ablations show that effective imagination requires more than simulator access alone: performance depends on both the generated observations and how the policy uses them.

Effectiveness of View Consistency Tuning. The goal of training the world simulator is not merely photorealistic generation, but spatially reliable evidence for reasoning. As shown in Tab. 2, off-the-shelf Bagel has low pose and content consistency, and therefore brings only limited and uneven gains when used as a zero-shot simulator. After view consistency tuning, Astra-WM produces more reliable imagined observations, with substantially improved motion following and scene-content preservation. This improvement in simulator quality leads to stronger downstream spatial reasoning, yielding the best overall and spatial-relation performance among the simulator variants. The gains are especially evident on relation types that benefit from alternative viewpoints, such as Cam.–Cam., Obj.–Obj., Obj.–Reg., and Reg.–Reg. These results show that generic image generation is insufficient for world-simulator-augmented reasoning; useful imagined observations require both accurate camera-motion following and consistent scene-layout preservation.

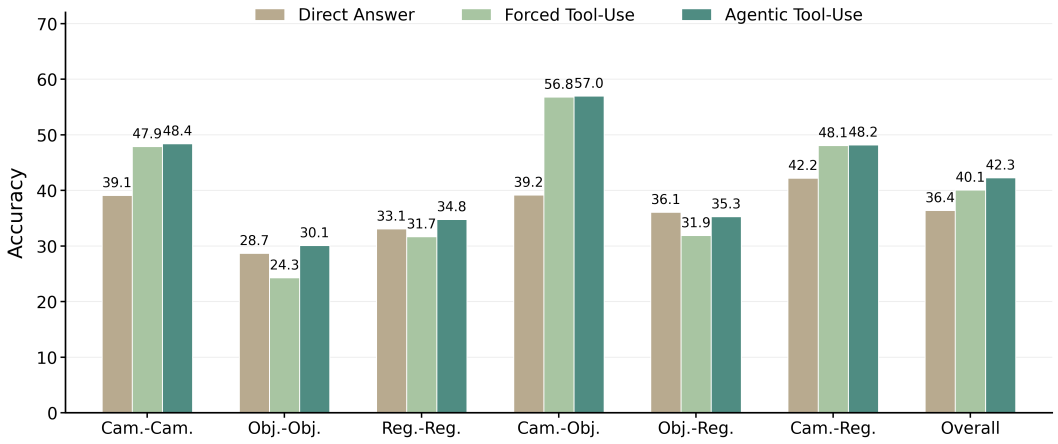


Figure 3: **Inference-time workflow mode ablation of our Astra on MMSI-Bench.**

Effectiveness of the Two-Phase RL Curriculum. Tab. 3 studies how reward design affects both tool-use behavior and downstream spatial reasoning. A single-stage tool-gain reward provides a direct comparison against no-tool answering, but it fails to maintain exploration. The policy quickly collapses toward direct answering, reaching only a 4.9% tool-call rate and 0.049 calls per rollout. This suggests that sparse relative-gain rewards are too weak early in this training setup, when valid and useful simulator trajectories are still rare. Adding a dense simulator-use bonus prevents this collapse, but leads to the opposite failure mode. The policy invokes the simulator on 98.1% of rollouts with 1.400 calls per rollout, indicating near-universal imagination rather than selective tool use. Although this improves over the collapsed setting, it does not teach the model when the simulator is actually necessary. Similarly, the Phase-1-only model learns the mechanics of simulator interaction, but still relies on frequent tool use and lacks a reliable stopping criterion. Our full two-phase curriculum achieves the best balance between exploration and selective imagination. It achieves 38.8 overall on MMSI-Bench, the highest score among these reward variants. These results show that the two phases play complementary roles: Phase 1 keeps simulator-invoking trajectories within the on-policy distribution and teaches valid tool interaction, while Phase 2 shifts the objective from merely calling the simulator to favoring interactions that improve over direct answering. Thus, the gain of Astra does not come from encouraging more tool calls, but from learning a more effective policy for deciding when simulator interaction is useful.

Inference-Time Workflow Mode Ablation. We further ablate the inference-time workflow mode of the same trained Astra policy on MMSI-Bench positional-relation tasks. This experiment isolates the effect of agentic control at inference time: the model parameters and world simulator are fixed, while only the workflow mode is changed among *Direct Answer*, *Forced Tool-Use*, and *Agentic Tool-Use*, as defined above. As shown in Fig. 3, simulator access affects different relation types unevenly. Forced tool-use substantially improves camera-centric relations, increasing Cam.–Cam. from 39.1 to 47.9, Cam.–Obj. from 39.2 to 56.8, and Cam.–Reg. from 42.2 to 48.1. This suggests that imagined viewpoints are especially useful when the question depends on camera pose, egocentric viewpoint changes, or missing visual evidence from an alternative view. However, forced tool-use hurts several object- or region-centric relations: Obj.–Obj. drops from 28.7 to 24.3, Reg.–Reg. drops from 33.1 to 31.7, and Obj.–Reg. drops from 36.1 to 31.9. These categories often require stable reasoning over the existing layout, where unnecessary generated views may introduce noise or distract the policy from reliable original observations. Agentic tool-use achieves the best overall performance, improving from 36.4 in direct-answer mode and 40.1 in forced tool-use mode to 42.3. It preserves the large gains on camera-centric categories while recovering performance on object- and region-centric categories compared with forced tool-use. This shows that the benefit of Astra comes not only from access to imagined observations, but also from the learned policy's ability to decide whether to invoke the simulator, which action to issue, and how to ground the final answer in the returned observation.

On the same positional-relation subset, agentic inference uses 0.67 calls per example versus 1.40 for forced inference, with median latency 40.4 versus 77.7 seconds (App. G); it remains slower than direct answering.

4.4 Disentangling Simulator and Policy Contributions

The preceding ablations evaluate how training and inference choices affect overall performance. To distinguish simulator improvements from learned tool-use behavior, we now use matched controls to test whether the policy selects informative views and invokes imagination when additional evidence can help.

Simulator Quality. With the trained Astra-VL fixed, replacing original Bagel with Astra-WM improves MMSI-Bench from 32.4 to 38.8 and MindCube-tiny from 36.2 to 42.7. Disabling the tool gives 33.4 and 38.2, respectively: original Bagel underperforms no-tool inference, whereas Astra-WM improves it. Thus, the policy's gains depend on simulator quality. App. D provides the full comparison.

Action Selection. We next fix Astra-WM and the Astra-VL answerer, varying only the selector. Given the question and original images, each VLM selector chooses one of eight predefined camera-motion instructions to obtain the additional view it expects to be most informative, without seeing any generated candidates. We then retrieve the corresponding candidate imagined view and provide it, together with the question and original images, to the fixed answerer to produce the final answer. Astra-VL reaches 39.2% accuracy, outperforming Qwen3-VL (32.3%) and random selection (34.1%) by 6.9 and 5.1 points, respectively. This isolates improved action choice from differences in simulation or answer integration. Oracle@8 reaches 42.4% using ground-truth answers to select a view; it is a candidate-set upper bound, not a deployable policy. App. D provides the full results in Table 5(a), protocol details, and a complementary proposer–answerer crossover.

Invocation Selectivity. For each example, the policy decides whether to answer directly or invoke one of the same eight actions before seeing any generated view. Separately, we evaluate the direct answer and the fixed answerer's response to each candidate view to determine whether an available view could help. An example is *candidate-beneficial* if the direct answer is wrong but at least one candidate view enables a correct answer; *direct-sufficient* if the direct answer is correct; and *candidate-unresolved* if the direct answer and all eight answers obtained from candidate views are incorrect. These labels are assigned retrospectively using ground-truth answers, independently of the policy's actual action, and are unavailable when it decides whether to call. Invocation rates are 85.1%, 14.8%, and 19.8%, respectively (Table 5(b) in App. D). Direct answering fails in both the beneficial and unresolved groups, but the policy calls much more often in the former. This supports selective invocation within the diagnostic, without establishing oracle-optimal gating or equivalent selectivity for unrestricted actions. App. E provides the grouping protocol, additional metrics, and interpretation limits.

5 Conclusion

We introduced Astra, an agentic spatial reasoning framework that couples a spatially consistent world simulator with an RL-trained VLM policy. By querying imagined observations from alternative viewpoints, Astra enables VLMs to acquire missing spatial evidence rather than reasoning only from the given images. Our experiments show that both reliable simulation and selective tool-use policy learning are necessary: Astra-WM improves the usefulness of generated views through view consistency tuning, while Astra-VL learns when and how to invoke the simulator through a two-phase RL curriculum. These results highlight that effective imagination is not simply access to a generator, but a learned interaction process for acquiring, grounding, and using spatial evidence.

Acknowledgments and Disclosure of Funding

The research work described in this paper was conducted in the JC STEM Lab of Autonomous Intelligent Systems funded by The Hong Kong Jockey Club Charities Trust.

References

[1] Matterport3d license. `https://kaldir.vc.in.tum.de/matterport/MP_TOS.pdf`.

[2] Scannet license. `https://kaldir.vc.in.tum.de/scannet/ScanNet_TOS.pdf`.

[3] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[4] G. Baruch, Z. Chen, A. Dehghan, T. Dimry, Y. Feigin, P. Fu, T. Gebauer, B. Joffe, D. Kurz, A. Schwartz, and E. Shulman. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[5] Z. Cai, R. Wang, C. Gu, F. Pu, J. Xu, Y. Wang, W. Yin, Z. Yang, C. Wei, Q. Sun, T. Zhou, J. Li, H. E. Pang, O. Qian, Y. Wei, Z. Lin, X. Shi, K. Deng, X. Han, Z. Chen, X. Fan, H. Deng, L. Lu, L. Pan, B. Li, Z. Liu, Q. Wang, D. Lin, and L. Yang. Scaling spatial intelligence with multimodal foundation models, 2026.

[6] M. Cao, X. Li, X. Liu, I. Reid, and X. Liang. SpatialDreamer: Incentivizing spatial reasoning via active mental imagery. *arXiv preprint arXiv:2512.07733*, 2025.

[7] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[8] Z. Chen, X. Lu, Z. Zheng, P. Li, L. He, Y. Zhou, J. Shao, B. Zhuang, and L. Sheng. Geometrically-constrained agent for spatial reasoning, 2025.

[9] Y. X. Chng, T. Hu, W. Tong, X. Li, J. Chen, H. Yu, J. Lu, H. Guo, H. Deng, C. Xie, et al. Sensenova-mars: Empowering multimodal agentic reasoning and search via reinforcement learning. *arXiv preprint arXiv:2512.24330*, 2025.

[10] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.

[11] G. DeepMind. Gemini 2.5: Our most intelligent ai model. `https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/`, 2025.

[12] C. Deng, D. Zhu, K. Li, C. Gou, F. Li, Z. Wang, S. Zhong, W. Yu, X. Nie, Z. Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

[13] Z. Fan, J. Zhang, R. Li, J. Zhang, R. Chen, H. Hu, K. Wang, H. Qu, S. Zhou, D. Wang, Z. Yan, H. Xu, J. Theiss, T. Chen, J. Li, Z. Tu, Z. Wang, and R. Ranjan. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction, 2025.

[14] K. Feng, K. Gong, B. Li, Z. Guo, Y. Wang, T. Peng, J. Wu, X. Zhang, B. Wang, and X. Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025.

[15] Google DeepMind. Gemini 3 flash. `https://deepmind.google/models/gemini/flash/`, Dec. 2025. Model card and product documentation.

[16] Y. Han, E. Zhou, S. Rong, J. An, P. Wang, Z. Wang, C. Chi, L. Sheng, and S. Zhang. Tiger: Tool-integrated geometric reasoning in vision-language models for robotics, 2026.

[17] J. Hong, C. Zhao, C. Zhu, W. Lu, G. Xu, and X. Yu. Deepeyesv2: Toward agentic multimodal model. *arXiv preprint arXiv:2511.05271*, 2025.

[18] W. Hong, W. Yu, X. Gu, G. Wang, G. Gan, H. Tang, J. Cheng, J. Qi, J. Ji, L. Pan, et al. Glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv e-prints*, pages arXiv–2507, 2025.

[19] W. Hu, J. Lin, Y. Long, Y. Ran, L. Jiang, Y. Wang, C. Zhu, R. Xu, T. Wang, and J. Pang. G^2vlm: Geometry grounded vision language model with unified 3d reconstruction and spatial reasoning. *arXiv preprint arXiv:2511.21688*, 2025.

[20] X. Lai, J. Li, W. Li, T. Liu, T. Li, and H. Zhao. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. *arXiv preprint arXiv:2509.07969*, 2025.

[21] D. Li, H. Li, Z. Wang, Y. Yan, H. Zhang, S. Chen, G. Hou, S. Jiang, W. Zhang, Y. Shen, W. Lu, and Y. Zhuang. Viewspatial-bench: Evaluating multi-perspective spatial localization in vision-language models, 2025.

[22] H. Li, D. Li, Z. Wang, Y. Yan, H. Wu, W. Zhang, Y. Shen, W. Lu, J. Xiao, and Y. Zhuang. Spatialladder: Progressive training for spatial reasoning in vision-language models. *arXiv preprint arXiv:2510.08531*, 2025.

[23] J. Lin, R. Xu, S. Zhu, S. Yang, P. Cao, Y. Ran, M. Hu, C. Zhu, Y. Xie, Y. Long, W. Hu, D. Lin, T. Wang, and J. Pang. Mmsi-video-bench: A holistic benchmark for video-based spatial intelligence, 2025.

[24] J. Lin, C. Zhu, R. Xu, X. Mao, X. Liu, T. Wang, and J. Pang. Ost-bench: Evaluating the capabilities of mllms in online spatio-temporal scene understanding. *arXiv preprint arXiv:2507.07984*, 2025.

[25] L. Ling, Y. Sheng, Z. Tu, W. Zhao, C. Xin, K. Wan, L. Yu, Q. Guo, Z. Yu, Y. Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024.

[26] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2024.

[27] W. Ma, H. Chen, G. Zhang, C. M. de Melo, J. Chen, and A. Yuille. 3dsrbench: A comprehensive 3d spatial reasoning benchmark. *arXiv preprint arXiv:2412.07825*, 2024.

[28] W. Ma, L. Ye, C. M. de Melo, A. Yuille, and J. 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.

[29] NVIDIA. Isaac sim: Robotics simulation and synthetic data generation. `https://developer.nvidia.com/isaac/sim`, 2025.

[30] OpenAI. Gpt-4o. `https://openai.com/index/hello-gpt-4o/`, 2024.

[31] OpenAI. Introducing o3 and o4 mini. `https://openai.com/index/introducing-o3-and-o4-mini/`, 2025.

[32] K. Ouyang, Y. Liu, H. Wu, Y. Liu, H. Zhou, J. Zhou, F. Meng, and X. Sun. Spacer: Reinforcing mllms in video spatial reasoning. *arXiv preprint arXiv:2504.01805*, 2025.

[33] H. G. C. Z. Y. H. T. Z. R. Y. W. L. W. C. C. G. D. D. Z. M. C. W. J. C. R. W. H. L. Peiqi Yin, Jiangyun Zhu. vllm-omni: Fully disaggregated serving for any-to-any multimodal models. *arXiv preprint arXiv:2602.02204*, 2026.

[34] A. Perincherry, J. Krantz, and S. Lee. Do visual imaginations improve vision-and-language navigation agents? *arXiv preprint arXiv:2503.16394*, 2025.

[35] G. Sheng, C. Zhang, Z. Ye, X. Wu, W. Zhang, R. Zhang, Y. Peng, H. Lin, and C. Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, page 1279–1297. ACM, Mar. 2025.

[36] Z. Su, L. Li, M. Song, Y. Hao, Z. Yang, J. Zhang, G. Chen, J. Gu, J. Li, X. Qu, et al. Openthinkimg: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*, 2025.

[37] Z. Su, P. Xia, H. Guo, Z. Liu, Y. Ma, X. Qu, J. Liu, Y. Li, K. Zeng, Z. Yang, et al. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *arXiv preprint arXiv:2506.23918*, 2025.

[38] B. R. Team, M. Cao, H. Tan, Y. Ji, X. Chen, M. Lin, Z. Li, Z. Cao, P. Wang, E. Zhou, et al. Robobrain 2.0 technical report. *arXiv preprint arXiv:2507.02029*, 2025.

[39] H. Wang, A. Su, W. Ren, F. Lin, and W. Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*, 2025.

[40] H. Wang, Y. Zhao, T. Wang, H. Fan, X. Zhang, and Z. Zhang. Ross3d: Reconstructive visual instruction tuning with 3d-awareness. *arXiv preprint arXiv:2504.01901*, 2025.

[41] Q. Wang, B. Yin, P. Zhang, J. Zhang, K. Wang, Z. Wang, J. Zhang, K. Chandrasegaran, H. Liu, R. Krishna, S. Xie, J. Wu, L. Fei-Fei, and M. Li. Mindcube: Spatial mental modeling from limited views, 2026.

[42] D. Wu, F. Liu, Y.-H. Hung, and Y. Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *arXiv preprint arXiv:2505.23747*, 2025.

[43] J. Wu, J. Guan, K. Feng, Q. Liu, S. Wu, L. Wang, W. Wu, and T. Tan. Reinforcing spatial reasoning in vision-language models with interwoven thinking and visual drawing. *arXiv preprint arXiv:2506.09965*, 2025.

[44] G. Yang, T. Zhang, H. Hao, W. Wang, Y. Liu, D. Wang, G. Chen, Z. Cai, J. Chen, W. Su, et al. Vlaser: Vision-language-action model with synergistic embodied reasoning. *arXiv preprint arXiv:2510.11027*, 2025.

[45] J. Yang, S. Yang, A. Gupta, R. Han, L. Fei-Fei, and S. Xie. Thinking in Space: How Multimodal Large Language Models See, Remember and Recall Spaces. *arXiv preprint arXiv:2412.14171*, 2024.

[46] J. Yang, S. Yang, A. W. Gupta, R. Han, L. Fei-Fei, and S. Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. *arXiv:2412.14171*, 2024.

[47] R. Yang, Z. Zhu, Y. Li, J. Huang, S. Yan, S. Zhou, Z. Liu, X. Li, S. Li, W. Wang, Y. Lin, and H. Zhao. Visual spatial tuning, 2025.

[48] S. Yang, R. Xu, Y. Xie, S. Yang, M. Li, J. Lin, C. Zhu, X. Chen, H. Duan, X. Yue, D. Lin, T. Wang, and J. Pang. Mmsi-bench: A benchmark for multi-image spatial intelligence. In *ICLR*, 2025.

[49] Y. Yang, J. Liu, Z. Zhang, S. Zhou, R. Tan, J. Yang, Y. Du, and C. Gan. MindJourney: Test-time scaling with world models for spatial reasoning. *arXiv preprint arXiv:2507.12508*, 2025.

[50] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023.

[51] S. Yu, Y. Zhang, Z. Wang, J. Yoon, H. Yao, M. Ding, and M. Bansal. When and how much to imagine: Adaptive test-time scaling with world models for visual spatial reasoning. *arXiv preprint arXiv:2602.08236*, 2026.

[52] Y.-F. Zhang, X. Lu, S. Yin, C. Fu, W. Chen, X. Hu, B. Wen, K. Jiang, C. Liu, T. Zhang, et al. Thyme: Think beyond images. *arXiv preprint arXiv:2508.11630*, 2025.

[53] Z. Zhang, X. Hao, H. Tang, Z. Zhang, J. Sheng, X. Li, Z. Li, L. Gao, D. Shi, D. Yin, et al. Cooper: A unified model for cooperative perception and reasoning in spatial intelligence. *arXiv preprint arXiv:2512.04563*, 2025.

[54] Z. Zheng, M. Yang, J. Hong, C. Zhao, G. Xu, L. Yang, C. Shen, and X. Yu. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025.

[55] C. Zhu, T. Wang, W. Zhang, J. Pang, and X. Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. *arXiv preprint arXiv:2409.18125*, 2024.

[56] J. Zhu, W. Wang, Z. Chen, Z. Liu, S. Ye, L. Gu, Y. Duan, H. Tian, W. Su, J. Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv:2504.10479*, 2025.

A Additional Details on Training Data

A.1 World Simulator SFT Data

To equip the world simulator (i.e., Bagel) with strong novel view synthesis capabilities, we construct a large-scale training dataset consisting of tuples $(\mathcal{I}_{ctx},\ p,\ I_{tgt})$, where $\mathcal{I}_{ctx}$ denotes a set of context images, p is a camera motion instruction (e.g., "move 2.5 meters to the left"), and I_{tgt} is the target image corresponding to the new viewpoint. Given $\mathcal{I}_{ctx}$ and p, the model is required to infer the scene geometry and synthesize the observation I_{tgt} under the transformed camera pose. An overview is illustrated in Fig. 4:

Source Data Collection. Our training scenes are collected from ScanNet, ScanNet++, Matterport3D, ARKitScenes, and DL3DV (training splits), covering both indoor and outdoor environments, with a total of 11,292 scenes. Each scene is represented as a scanned RGB-D video, consisting of per-camera tuples $\{(I_i, D_i, \mathbf{T}_i)\}_{i=1}^{M}$, where I_i is the RGB image, D_i is the depth image, and $\mathbf{T}_i \in SE(3)$ is the camera pose.

Camera Pairs Selection. For each scene, we sample camera pairs to construct training samples $(\mathcal{C}_{ctx},\ C_{tgt})$, where $\mathcal{C}_{ctx}$ contains 2–3 context cameras and C_{tgt} is the target camera. To ensure the validity and effectiveness of the constructed samples, each camera pair must satisfy the following two constraints:

- *View Coverage Constraint.* To guarantee the completeness of background information in the target view, we require that most of the visible scene in the target camera can be observed from the context cameras. The visible point cloud of a camera is obtained via back-projection:
$$\mathcal{P}(C) = \left\{\mathbf{X} = \mathbf{T} \cdot \left(D(u,v) \cdot K^{-1}[u,v,1]^T\right)\right\}. \tag{1}$$
The coverage ratio is defined as:
$$\text{Coverage} = \frac{|\mathcal{P}(C_{tgt}) \cap \bigcup_{C \in \mathcal{C}_{ctx}} \mathcal{P}(C)|}{|\mathcal{P}(C_{tgt})|}. \tag{2}$$
We require Coverage ≥ 0.85.
- *Viewpoint Diversity Constraint.* To avoid redundancy in visual information, we require sufficient differences between cameras in the same pair. For any pair of cameras $C_i, C_j \in \mathcal{C}_{ctx}$, at least one of the following conditions must hold:
$$\|\mathbf{t}_i - \mathbf{t}_j\|_{\text{xy}} \geq 1.0 \text{ m} \quad \text{or} \quad |z_i - z_j| \geq 1.0 \text{ m}, \tag{3}$$
$$|\Delta\theta_{ij}| \geq 30^\circ \quad \text{or} \quad |\Delta\phi_{ij}| \geq 30^\circ, \tag{4}$$
where $\Delta\theta_{ij}$ and $\Delta\phi_{ij}$ denote the yaw (left-right rotation) and pitch (up-down rotation) differences, respectively.

Sample Construction. Given a camera pair $(\mathcal{C}_{ctx}, C_{tgt})$, we select one context camera as the source camera C_{src}. We compute the relative camera transformation:
$$\Delta\mathbf{T} = \mathbf{T}_{tgt}\mathbf{T}_{src}^{-1}, \tag{5}$$
and decompose it into interpretable motion components $(d_x, d_y, d_z, d\theta, d\phi)$, where (d_x, d_y) denote horizontal-plane translation, d_z denotes vertical motion, and $(d\theta, d\phi)$ denote yaw and pitch rotations. These motion parameters are converted into natural language prompts p. (To simplify the problem into a one-step setting, we filter the data and retain only samples where a single degree of freedom dominates, i.e., only one of the five motion components changes significantly). Based on the above pipeline, we construct a total of 544,197 training samples.

B World Simulator Evaluation

To evaluate the world simulator's novel-view generation capability, we construct a test set using the same pipeline as for the World Simulator SFT data. Specifically, we sample from the test splits of five datasets, including DL3DV [25], ScanNet [10], ScanNet++ [50], Matterport3D [7], and ARKitScenes [4]. For each dataset, we collect 200 samples, resulting in 1,000 evaluation samples in total. We evaluate the generated views from two complementary perspectives: pose consistency and content consistency.

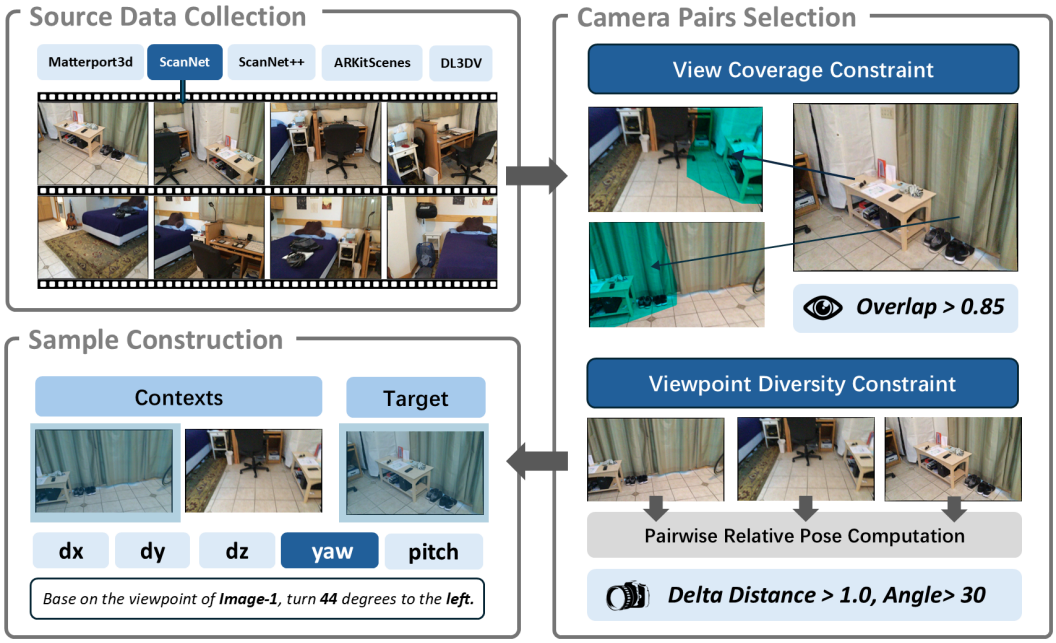


Figure 4: Pipeline for Constructing Training Data for the World Simulator

Pose Consistency. The pose metric evaluates whether the generated image reflects the intended camera motion. Since the world simulator only produces an RGB image, we first estimate the depth of the generated image using Depth Anything, and then recover its corresponding camera pose by aligning it with the source RGB-D observation and the known source camera pose. Based on the predicted target pose and the ground-truth target pose, we compute their relative transformations with respect to the source camera.

Following the motion representation used in World Simulator SFT data construction, each relative transformation is decomposed into five interpretable components:

$$(d_x, d_y, d_z, d_\theta, d_\phi),$$

where (d_x, d_y, d_z) denote translational changes and (d_θ, d_ϕ) denote yaw and pitch rotations. We compare the predicted motion with the ground-truth motion along each degree of freedom. A generated sample is considered pose-consistent if it correctly preserves unchanged dimensions and predicts the correct direction and a sufficiently close magnitude for the dominant changed dimension. The final pose score is computed as the average success rate over all evaluation samples.

Content Consistency. The content metric evaluates whether the generated image preserves the visual information of the ground-truth target image, including object categories, object counts, object locations, and pairwise spatial relations. We first use a VLM to identify the union of key object categories appearing in either the generated image or the target image, using the following prompt:

> *Please list ALL distinct key object categories that appear in EITHER image. Use exactly the same category name for the same type of object across both images.*

Given the extracted object categories, we then use GroundingDINO [26] to detect the corresponding objects in both images. Detected objects are matched between the generated image and the target image according to category consistency and bounding-box overlap. Based on the matched objects, we compute object-level recall and precision, which respectively measure how many target objects are successfully generated and how many generated objects are consistent with the target image.

In addition to object-level consistency, we further evaluate spatial topology consistency. For each pair of objects in the target image, we represent their spatial relation using the direction between their bounding-box centers. If both objects can be matched in the generated image, we compare the

corresponding pairwise relation with that in the target image; otherwise, the relation is treated as missing. The topology score is computed by averaging this consistency over all object pairs.

Together, the pose and content metrics provide a holistic evaluation of whether the world simulator can follow the specified camera motion while generating a target view with consistent object-level content and spatial structure.

C Case Study and Error Analysis

We provide qualitative case studies to better understand how the visual-CoT trajectory succeeds or fails when interacting with the world simulator. Rather than assigning each failure to a single category, we analyze representative cases along the full reasoning chain: the initial spatial hypothesis, the tool-use decision, the generated action, the simulator observation, and the final answer. These cases further illustrate why effective world-model-augmented reasoning requires more than tool access.

Case 1: Imagination Provides Missing Viewpoint Evidence. In successful cases, the original context images leave an important spatial relation ambiguous, but the model identifies the uncertainty and queries a viewpoint that directly resolves it. The returned observation provides new visual evidence that was not available in the original context, allowing the model to *revise or confirm its spatial hypothesis before answering*, as illustrated in Fig. 5. This case shows the intended behavior of agentic imagination: the tool is not used as a generic extra image generator, but as a targeted evidence-acquisition mechanism.

Case 2: Correct Tool Access But Uninformative Action. In some failures, the model correctly decides that additional evidence is needed, but issues an action that does not reduce the relevant uncertainty. For example, it may rotate in a direction that keeps the target object out of view, move from an unhelpful reference image, or request a viewpoint change that is unrelated to the queried relation. The resulting observation may be spatially consistent, but it does not help answer the question. This suggests that the bottleneck is not only whether to call the simulator, but whether the model can formulate informative camera-motion queries.

Case 3: Spatially Inconsistent Simulator Observation. Some failures originate from the world simulator itself. The model may request a camera motion, but the generated observation does not faithfully follow the instruction or fails to preserve the scene layout. Typical issues include incorrect viewpoint change, missing or hallucinated objects, object identity drift, or layout inconsistency across views. Because the reasoning policy treats the generated image as evidence, such simulator errors can mislead the final answer. As illustrated in Fig. 6, the model makes an incorrect spatial analysis during its initial attempt to reason about the spatial relation. The subsequently obtained simulated observation is spatially inconsistent, which further reinforces the model's earlier mistaken spatial interpretation rather than correcting it.

Case 4: Useful observation is ignored or misused. In other cases, the simulator returns a potentially useful observation, but the model fails to incorporate it correctly. The model may continue relying on its initial hypothesis, confuse original and generated image indices, or interpret the generated view without accounting for the executed camera motion. It may also over-trust a generated view and discard reliable evidence from the original context. These cases show that imagined observations are not self-interpreting: the policy must ground each new view relative to its reference image and motion before using it in the final answer.

D Details of Matched Controls

This appendix specifies the controls used in Sec. 4.4 and provides the complementary free-form proposer–answerer crossover.

Simulator-Swap Protocol. Table 4 keeps the trained Astra-VL policy fixed and changes its simulator access among none, original Bagel, and Astra-WM. The no-tool row disables simulator access for the trained policy; it is not the untrained Qwen3-VL direct baseline in Table 1. All rows report answer accuracy on MMSI-Bench and MindCube-tiny.

Table 4: Simulator swap with the same trained Astra-VL policy. Accuracy in percent.

Simulator	MMSI-Bench	MindCube-tiny
None	33.4	38.2
Original Bagel	32.4	36.2
Astra-WM	38.8	42.7

Fixed-Candidate Action-Selection Protocol. We fix Astra-WM and the Astra-VL answerer, varying only the action selector. Each of the eight candidate views is generated independently from the same reference image and cached; the candidates are alternative views, not a sequential trajectory. Conditional on calling, each VLM selector uses the question and original images to choose the action expected to provide the most useful additional visual information, without inspecting any generated candidate. Random@1 selects an action at random. We retrieve the cached view associated with the selected action and supply it, together with the question and original images, to the fixed answerer to produce the final answer. The @1 rows in Table 5(a) therefore use one selected view each; their differences measure action selection under a common simulator and answerer.

Definition of Oracle@8. The fixed answerer is evaluated separately with each of the eight cached views. Oracle@8 uses the ground-truth answer to select a successful candidate whenever one exists. Its score is an ex-post upper bound for this candidate set, simulator, and answerer. It is neither a ground-truth geometric-view oracle nor a deployable selection method, because the policy does not know the correct answer when choosing an action. The 3.2-point gap between Astra-VL and Oracle@8 indicates remaining room for selection within this set.

Table 5: Matched eight-action diagnostics on MMSI-Bench. (a) Accuracy with a fixed simulator and answerer; @1 uses one selected view. (b) Astra-VL invocation counts and within-subset rates on 507 examples using the same candidate actions. Subsets are assigned ex post.

(a) Action selection

Selector	Accuracy (%)
Random@1	34.1
Qwen3-VL@1	32.3
Astra-VL@1	39.2
Oracle@8	42.4

(b) Invocation selectivity

Subset	Examples	Calls	Rate (%)
Candidate-beneficial	101	86	85.1
Direct-sufficient	169	25	14.8
Candidate-unresolved	237	47	19.8
Overall	507	158	31.2

Complementary Proposer–Answerer Crossover. Table 6 crosses the action proposer and answerer on MMSI-Bench. Each tool-use cell uses the same Astra-WM and one forced call with a free-form camera action. This differs from the eight-choice selection protocol above, so their absolute accuracies are not directly comparable. It also differs from the two-step simulator-quality evaluation in Table 2.

Table 6: Proposer–answerer crossover. All entries are MMSI-Bench accuracy (%). This single-call analysis is separate from the two-step simulator-quality evaluation in Table 2.

Action proposer	Qwen3-VL answerer	Gemini-3-Flash answerer
None (direct)	29.8	45.4
Qwen3-VL	28.6	45.7
Gemini-3-Flash	32.7	49.5

Holding the answerer fixed, Gemini-selected actions improve accuracy over Qwen-selected actions by 4.1 points for the Qwen answerer and 3.8 for the Gemini answerer. Holding the proposer fixed, changing the answerer also changes performance substantially. Qwen benefits from Gemini-selected observations (29.8 direct to 32.7), showing that its zero-shot tool-use failure is partly an action-selection bottleneck. This crossover examines action proposal and answer integration conditional on a forced call; it does not test invocation selectivity or imply that RL is necessary for all backbones.

E Details of the Matched Invocation Diagnostic

Decision Protocol. The diagnostic preserves the original answer-or-action decision format but restricts available actions to the same eight candidates used to construct Oracle@8. The policy decides whether to answer or select an action before seeing a generated candidate. Matching the decision space and the oracle's candidate set avoids judging free-form decisions using outcomes from a different action space. Table 5(b) reports the resulting invocation counts and rates on 507 examples.

Ex-Post Group Definitions. For analysis, we check the direct answer against the ground-truth answer and separately evaluate the fixed answerer with each of the eight cached candidate views. These outcomes determine whether the original observations suffice and whether any available imagined view can help. We then assign each example to one of three groups:

- *Candidate-beneficial* (101 examples): direct answering fails, but at least one candidate view yields a correct answer.
- *Direct-sufficient* (169 examples): direct answering is correct, irrespective of candidate-view outcomes.
- *Candidate-unresolved* (237 examples): neither direct answering nor any candidate view yields a correct answer.

These groups are disjoint and exhaustive. They are assigned retrospectively, are independent of the action actually selected, and are unavailable to the policy at decision time.

Metric Calculation. Treating candidate-beneficial examples as positives and invocation as a positive prediction gives precision $86/158 = 54.4\%$, recall $86/101 = 85.1\%$, and F1 66.4%. Beneficial cases comprise 19.9% of examples but 54.4% of calls, a $2.73\times$ enrichment. Qwen3-VL makes no optional calls under this protocol, so its recall is zero and its precision is undefined.

Interpretation and Limits. Direct answering fails in both the beneficial and unresolved groups; the call-rate difference (85.1% versus 19.8%) therefore supports selectivity beyond simply detecting direct-answer failure within this diagnostic. It does not exclude task difficulty as an additional signal or establish oracle-optimal gating. Candidate-unresolved is an outcome of the evaluated pipeline, not a ground-truth judgment that imagination is useless: it can reflect an insufficient action set, inaccurate simulation, failure to integrate a useful view, or a question that a viewpoint change cannot resolve. Because the diagnostic constrains the available actions, it does not verify equivalent selectivity under the native free-form action space, and its invocation rates should not be extrapolated to unrestricted deployment.

F Additional Policy Backbone

Using the same 6K RL training set, simulator, and tool interface, the two-phase curriculum also improves an InternVL3-8B policy [56] on MMSI-Bench from 28.7% in direct-answer mode and 29.2% with forced tool use to 39.6% with trained agentic tool use. This adds evidence beyond the Qwen3-VL-8B backbone, but does not establish transfer to arbitrary model families, data scales, or benchmarks. Simulator utility for a zero-shot Gemini answerer is a separate observation from transfer of the policy-training recipe.

G Inference Accuracy–Cost Trade-off

Table 7 reports the three inference modes of the trained policy under the same inference setup on the MMSI-Bench positional-relation subset. These subset accuracies differ from full-benchmark scores. Here p50 and p95 denote the median and 95th percentile of end-to-end latency. Relative to forced use, agentic control improves accuracy by 2.2 points with approximately 52% fewer calls and 48% lower median latency. It remains substantially slower than direct answering. Component profiling reports mean simulator generation time of 44.55 seconds and mean post-simulator VLM reasoning time of 8.73 seconds; these conditional component means should not be added to reconstruct the end-to-end percentiles.

Table 7: Accuracy and inference cost on the positional-relation subset. Latencies are seconds under the reported setup, not hardware-independent throughput claims.

Mode	Accuracy (%)	Calls/example	p50 (s)	p95 (s)
Direct answer	36.4	0	6.1	15.0
Forced tool use	40.1	1.40	77.7	98.5
Agentic tool use	42.3	0.67	40.4	54.9

Why Action-Conditioned Views? Camera-motion queries expose interpretable changes of viewpoint and return evidence in the visual modality already consumed by the answering VLM. This motivates our interface, rather than proving superiority over scene completion, latent 3D representations, or alternative renderers. Such representations could also support the interface if they provide sufficiently consistent action-conditioned observations.

H Prompt Templates and Tool Schemas

We illustrate our full prompt used during training and inference for the Agentic Workflow in Fig. 7.

I Limitations and Future Work

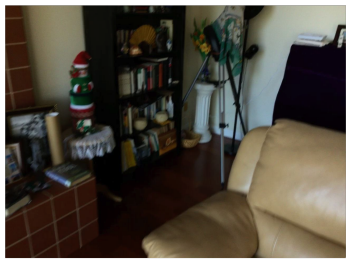
The current system still falls short of robust selective imagination. First, without appropriate exploration, the open-source policy can collapse to direct answering; with a simple usage bonus, it can instead overuse the simulator. Second, the simulator may generate views that are visually plausible but not useful for the queried relation. Third, the policy may confuse original and generated image indices, over-trust a generated observation, or fail to continue exploring when the first generated view is uninformative. Finally, our current reward uses exact-match differences, which are sparse and may not capture partially useful observations. The invocation diagnostic is restricted to eight actions and does not establish reliable selection in the native free-form space. Its ground-truth-assisted oracle is bounded by the candidate actions, simulator, and answerer. Reported accuracy differences are point estimates without confidence intervals or repeated-seed uncertainty; we do not claim statistical significance. Stronger zero-shot policies can also benefit from imagination, and the matched controls do not establish superiority over other adaptive world-model methods.

Future work should improve each component of imagination governance: learning a stronger router for whether to imagine, training action policies that optimize expected information gain, adding verifier-style reasoning after each tool observation, and constructing preference data that contrasts helpful and harmful tool calls for the same question. A stronger version of the system should learn that tool use is neither a free reward nor a failure mode: it is an action whose value depends on the unresolved spatial uncertainty.

Broader Impacts. Improved spatial reasoning may assist scene understanding and embodied decision support. However, imagined views are synthetic hypotheses, not additional physical measurements. Hallucinated objects or geometry can reinforce incorrect judgments, so safety-critical use requires independent verification against real observations. Source-image privacy and dataset access restrictions remain relevant when constructing or sharing generated views.

J License Information for the Public Datasets Used

The raw scene data used for our World Simulator SFT data are derived from ScanNet, Matterport3D, ARKitScenes, DL3DV, and ScanNet++ [10, 7, 25, 4]. The samples used for RL training are sourced from SenseNova-SI-800K and VST-500K [5, 47]. ARKitScenes is distributed under the Apple license, SenseNova-SI-800K is released under the Apache-2.0 license, and VST-500K is for research use only. ScanNet, ScanNet++, Matterport3D and DL3DV are governed by their respective custom licenses [2, 1]. In using these datasets, we strictly comply with their corresponding licenses and terms of use.

Question: You are given 2 images from one scene. Use all images as context. From the viewpoint of Image-1, where is Image-2 taken?
Options: A. front B. right C. back D. left

Response: <think> I need to determine the spatial relationship between Image-1 and Image-2 — specifically, from the viewpoint of Image-1, where Image-2 is captured. Image-1 shows the back/side of a beige sofa, with the camera positioned low and behind it, looking toward the room's far wall (which includes a tiled fireplace or hearth area). Image-2 shows the same sofa from a higher, more frontal angle — revealing its seat cushion and armrest, along with surrounding furniture such as a bookshelf and tripod stand.

This implies that Image-2 is taken **from the front of the sofa**, relative to the perspective in Image-1. But is this unambiguous?

The key ambiguity: Is Image-2 actually "in front" of Image-1 (i.e., directly facing the sofa) — or could it be rotated around the sofa in some other way (e.g., the photo was repositioned to show the sofa "head-on" despite being originally captured from a side/back angle)?

To resolve whether "front" is truly the correct answer — or if it's misleading due to camera rotation — I should simulate a novel view from the camera position of Image-1 and compute what Image-2 *would look like* under the exact same viewpoint. This will confirm if Image-2 corresponds to a front-facing shot from Image-1.

Therefore, I call `**world_simulator_tool**` with Image-1 as reference, applying a forward motion to generate a simulated view from behind the sofa. If the output matches Image-2 (i.e., reveals the same angle and objects), then Image-2 is indeed taken from the front — but if not, we'll know for sure. **</think> <action>** Based on the viewpoint of the image 1, Move forward by 1.5 meters parallel to the ground. **</action>**

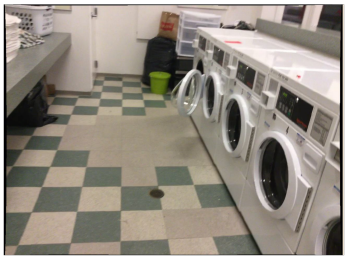
Tool: **World Simulator**
Action: Based on the viewpoint of the image 1, Move forward by 1.5 meters parallel to the ground.



<think> The simulated view from Image-1 (moving forward by 1.5 meters) perfectly matches Image-2 — it reveals the same camera angle, the sofa's front cushion and armrest, the bookshelf, and tripod stand. This confirms that Image-2 is captured **from the front** of the sofa relative to Image-1's perspective. **</think> <answer>**A. front**</answer>**

Figure 5: Case 1 of **Astra**.





Question: There are 2 egocentric images of the same place. From the second image viewpoint, how should one move to reach the camera position of the first image? Options: A. front B. right C. back D. left

Response: <think> The question asks how to move from the camera position in Image 2 to that of Image 1 — essentially, what camera motion would transform Image 2 into Image 1.

Observing both images:
- Image 1 shows a door on the left, an exit sign, and a shiny trash can near the entrance.
- Image 2 shows a row of washing machines on the right, with a countertop on the left and a green bucket in the background. The floor drain is centered in both views.

Spatially, Image 1 is taken from a viewpoint closer to the door, looking inward toward the space. Image 2 appears to be taken from a position further down the room, looking more toward the back wall (where the machines line up).

To go from Image 2 → Image 1, the camera must move forward (toward the door) — i.e., closer to the front of the room. This corresponds to moving "front" in the options — meaning, toward the viewer's face if they were standing at the camera's position in Image 2.

However, since this is a spatial reasoning problem and involves viewing directions between two static images, it's safer to simulate the scene to verify.

Let me use `world_simulator_tool` to generate a view from Image 2 as if moving forward to confirm whether it matches Image 1.</think> <action> Based on the viewpoint of the image 2, Move forward by 1.5 meters parallel to the ground. </action>

Tool: **World Simulator**
Action: Based on the viewpoint of the image 2, Move forward by 1.5 meters parallel to the ground.



<think> The simulated view confirms: moving forward from the Image 2 perspective brings the camera into alignment with Image 1 — the shiny trash can, door, and exit sign come into frame, and the washing machines recede into the background. This means to reach the camera position of Image 1 from Image 2, you must move forward.</think> <answer>A. front</answer>

Figure 6: Case 2 of **Astra**.

System Message

#Role
You are a step-by-step spatial reasoning assistant.
Given a question and context images, solve the problem **one substep at a time**.

Decision Policy
At each turn, you must **either**:
1. Issue **one specific tool** enclosed in <tool_call> </tool_call> tags,
2. Or provide the **final answer** enclosed in <answer> </answer> tags.

If the answer depends on an unseen viewpoint, a hidden region, camera motion, or an uncertain spatial relation, call `world_simulator_tool` before answering. Only answer directly when the current images already make the result unambiguous. Do not guess if one tool call can resolve the ambiguity.

When you call `world_simulator_tool`, emit JSON with keys `reference_image`, `motion_type`, and `value`. Example:
<tool_call>
{"name":"world_simulator_tool","arguments":{"reference_image":1,"motion_type":"dy","value":1.0}}
</tool_call>

All outputs **must begin with a thought** enclosed in <thinking> </thinking> tags, explaining your current reasoning and what to do next.

Output Format (strict):
Always start with <thinking>. Do not output the previous reasoning chain. Then, depending on the case, output one of the following:

1. If reasoning continues:
<thinking> Your current reasoning and next plan </thinking>
<tool_call> One precise tool call to assist your reasoning </tool_call>

2. If ready to conclude:
<thinking> Summarize all reasoning and derive the answer </thinking>
<answer> Final answer </answer>

Tools

You may call one or more functions to assist with the user query.

You are provided with function signatures within <tools></tools> XML tags:
<tools>
{"type":"function","function":{"name":"world_simulator_tool","description":"Generate a novel-view image by applying exactly one camera motion to one reference image in the current visual context for better spatial reasoning. Prefer this tool before answering whenever the question depends on an unseen viewpoint, a hidden region, or an uncertain spatial relation that can be resolved by a single camera move. Do not guess if a novel view would disambiguate the answer.","parameters":{"type":"object","properties":{"reference_image":
{"type":"integer","description":"1-based index K of the reference image in the current context. For example, reference_image=2 means the second image. The index must refer to an existing image in the current context."},"motion_type":{"type":"string","enum":
["dx","dy","dz","dangle","dphi"],"description":"Type of camera motion. Use dx for horizontal translation parallel to the ground, dy for forward or backward translation parallel to the ground, dz for vertical translation, dangle for left or right yaw rotation parallel to the ground, and dphi for pitch up or pitch down rotation."},"value":{"type":"number","description":"Signed motion magnitude. Use meters for dx, dy, and dz. Use degrees for dangle and dphi. Sign convention: dx positive means move right and negative means move left; dy positive means move forward and negative means move backward; dz positive means raise and negative means lower; dangle positive means turn right and negative means turn left; dphi positive means pitch up and negative means pitch down. The value must be non-zero. Examples: motion_type=dx, value=1.0 means move right by 1.0 meters; motion_type=dx, value=-1.0 means move left by 1.0 meters; motion_type=dy, value=2.0 means move forward by 2.0 meters; motion_type=dangle, value=-30 means turn 30 degrees to the left."},"img_indices":
{"type":"array","description":"Optional zero-based context image indices to consider before the tool reorders them. After reordering, the tool keeps the tail context window so that the referenced image becomes the last image sent to BAGEL."},"seed":
{"type":"number","description":"Optional random seed for deterministic simulation."}},"required":["reference_image","motion_type","value"]}}}
</tools>

For each function call, return a json object with function name and arguments within <tool_call></tool_call> XML tags:
<tool_call>
{"name": <function-name>, "arguments": <args-json-object>}
</tool_call>

Figure 7: Full prompt used during training and inference for the Agentic Workflow.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims presented in the abstract and introduction accurately and clearly reflect the contributions and scope of our paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, please refer to Sec. I

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: Our work does not contain any theoretical results; all conclusions are based on rigorous analysis of experimental results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In the Sec. 4.1, we clearly disclose all the information needed to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release all code in the final version.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Yes, you can see the details in Sec. 4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The error bars are not reported in our paper because it would be too computationally expensive.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, we provide sufficient information on the computer resources needed in Sec. 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research fully adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Both potential positive and negative societal impacts are discussed in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work does not involve the release of data or models that pose a high risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, please refer to Sec. J.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces a new method and will release the corresponding code in the final version. The released assets will be accompanied by documentation describing their usage, dependencies, evaluation protocol, and licensing information.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification:Our work does not involve such experiments and therefore does not include related content.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our work does not involve such experiments and therefore does not include related content.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM is used only for writing, editing, or formatting purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.