One Cognitive Loop Is Enough: SODA unlocks Pure-Text Spatial Reasoning in Large Language Models

Shunwen Bai[1*], **Jiahuan Zhang**[2*], **Haoran Huang**[1], **Yurun Wang**[1], **Jiale Liu**[1], **Yanxi Wu**[1], **Ningzhe Yu**[1], **Yudong Gao**[3†], **Mingjun Cheng**[1†]
[1]Zhejiang University
[2]Tianjin University
[3]Hong Kong University of Science and Technology
shunwenbai@zju.edu.cn, ygaodj@connect.ust.hk, mkellerc@outlook.com

Abstract

Currently, large language models (LLMs) have significant limitations in spatial reasoning, particularly in the absence of visual input. To address this issue, we introduce **SODA** (Spatial OODA), which draws inspiration from the OODA cognitive loop (Observe, Orient, Decide, Act), originally designed to enhance human decision-making in dynamic environments. Specifically, we embed the OODA loop into multiple control tasks, generating the *SPOD-143k* dataset, and successfully integrate it into LLMs through a two-phase and spatia-aware training strategy (SFT and GRPO). Furthermore, to fill the gap in evaluating spatial reasoning in purely text-based LLMs, we introduce the *SPOD-Bench* benchmark, including multiple tasks divided into three levels of difficulty. Experimental results show that **SODA** significantly enhances the spatial reasoning capabilities of LLMs across testing scenarios including *SPOD-Bench*, SPACE and applications, providing a replicable and effective paradigm for improving the spatial cognition of LLMs.

1 Introduction

Despite their strong general language capabilities (Brown et al., 2020; Achiam et al., 2023), Large Language Models (LLMs) are hindered by hallucinations and insensitivity to logical symbols (Ramakrishnan et al., 2025; Huang et al., 2025; Bai et al., 2025), leading to errors in basic numerical comparisons like distinguishing 9.11 from 9.9 (Xie, 2025; Liu and Fang, 2025). LLMs also show significant limitations in spatial cognitive tasks requiring spatial relations, directionality, and geometric reasoning (Bubeck et al., 2023; Li et al., 2024; Hendrycks et al., 2021), such as natural language navigation (Tikhonov, 2024). Their performance varies across spatial structures, and they struggle with complex spatial reasoning (Wu et al., 2024). Current research often relies on costly vision-language or robotic sensory data, which leads to high development expenses, poor transferability, and a disregard for logical consistency in reasoning (Yamada et al., 2024; Chen et al., 2024; Stogiannidis et al., 2025; Yin et al., 2025; Brohan et al., 2023; Li et al., 2022; Shu et al., 2025; Khemlani et al., 2025). Consequently, existing LLMs face significant challenges in performing text-driven spatial reasoning without external sensory input (Driess et al., 2023; Ahn et al., 2022; Surís et al., 2023).

To address this, we draw inspiration from the OODA (Observe, Orient, Decide, Act) loop, a cognitive model that provides a structured framework for simulating biological spatial cognitive processes (Enck, 2012). Spatial cognition is fundamentally a Perception-Action Cycle (Fuster, 2004; Egelhaaf and Lindemann, 2025; Takahashi et al., 2024; Ji et al., 2025; Bollig et al., 2024; Dong and Fiete, 2024; Collett et al., 2025). By integrating the OODA loop, LLMs can simulate this process using only language: "Observe" to parse the environment, "Orient" to localize, "Decide" to plan paths, and "Act" to simulate task completion. Building on this, we introduce SODA (Spatial OODA) to systematically enhance LLM spatial reasoning. First, we created *SPOD-143k*, a large-scale pure-text spatial dataset of 143,000 question-answer pairs across 17 categories, integrating the OODA framework into the intermediate reasoning process of the data. We then embed this framework into LLMs via Supervised Fine-Tuning (SFT) and Group Relative Policy Optimization (GRPO(Shao et al., 2024)) on *SPOD-143k*, enabling structured spatial reasoning purely through language.

Furthermore, to address the current lack of dedicated benchmarks for spatial reasoning without visual input, we introduce *SPOD-Bench*, a comprehensive text-only benchmark covering 13 task

[*]Co-first authors.
[†]Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 29974–29996
July 2-7, 2026

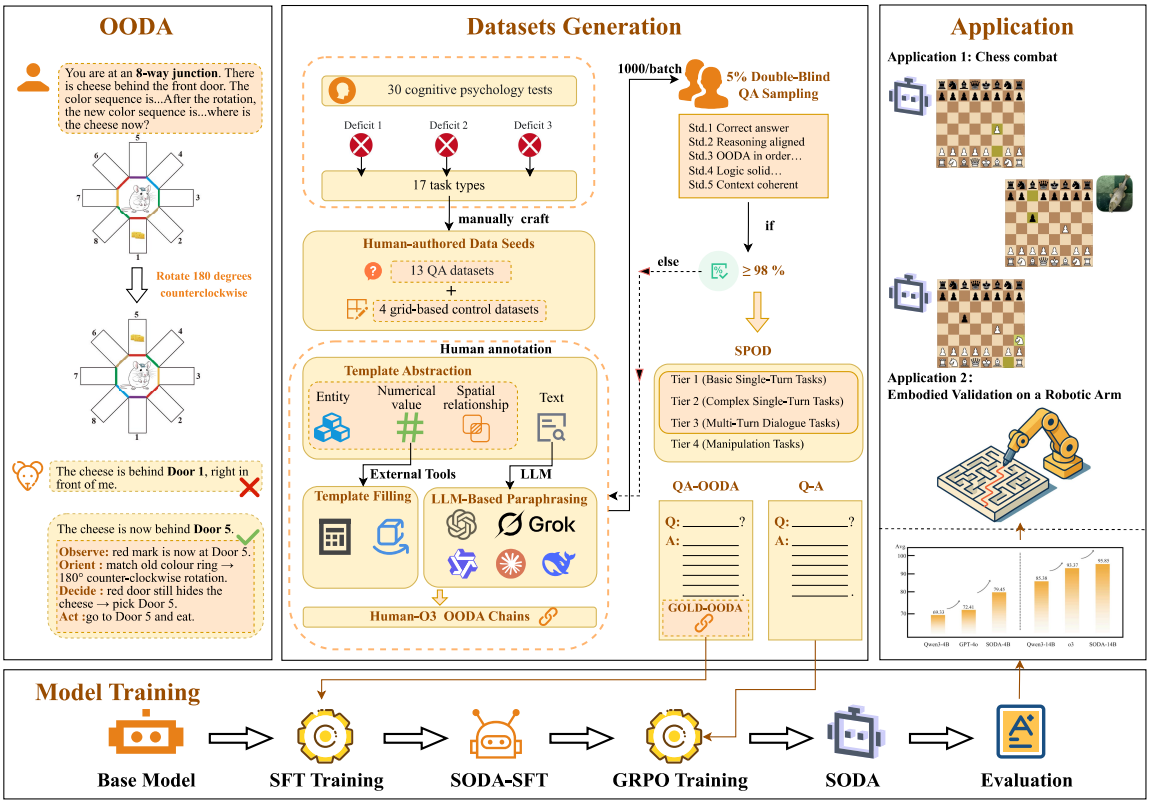

Figure 1: SODA workflow: OODA reasoning, SPOD dataset generation, training and evaluation pipeline, and applications. Left: An example OODA loop (Observe → Orient → Decide → Act) that lets the model correct a spatial mistake after a 180-degree rotation. Middle: How the SPOD dataset is built—from small human-written seeds, through template expansion and LLM paraphrasing, with 5 % double-blind quality checks—covering 17 spatial task types. Right: Two applications that use the trained model: playing chess against Stockfish and guiding a real robotic arm through a maze. Bottom: Training pipeline—Base model → SFT → SODA-SFT → GRPO fine-tuning → final SODA model → evaluation.

categories across 3 difficulty levels. We applied SODA to Qwen3-4B and Qwen3-14B and conduct extensive comparative experiments on *SPOD-Bench*, the SPACE dataset (Ramakrishnan et al., 2025), and a Chess Combat test. As illustrated in Figure 1, SODA-enhanced models perform on par with or even surpass existing LLMs. These results highlight an effective, practical, and reproducible paradigm for improving the spatial cognition capabilities of smaller LLMs. In view of the challenges discussed above, the main contributions of this work are summarized as follows: (i) We introduce the OODA cognitive loop, offering a new paradigm for pure-text spatial reasoning in LLMs. (ii) We constructed a large-scale, pure-text dataset, *SPOD-143k*, and deeply integrated the OODA framework into model reasoning via a two-stage training pipeline. (iii) We developed the first diverse and hierarchical pure-text spatial reasoning benchmark, *SPOD-Bench*. (iv) Extensive experiments on *SPOD-Bench*, public datasets, and real-world applications demonstrate the performance of the proposed method.

2 Related Work

2.1 Spatial Cognition and Reasoning in LLMs

Spatial cognition is a vital component of human intelligence, encompassing spatial perception, reasoning, control, and language understanding. In the context of LLMs, spatial perception involves sensing the spatial position of objects (Xuan et al., 2024; Xu et al., 2024), spatial reasoning pertains to the logical deduction of spatial relationships (Rodionov et al., 2025; Wang et al., 2024), spatial control is manifested in object path planning and navigation (Aghzal et al., 2025; Koh et al., 2021), and spatial language understanding is the core ability for LLMs to comprehend spatial concepts (Du et al., 2024). Human children gradually develop spatial cognitive abilities through activities such as playing with building blocks and cultivating a sense of direction (Lee et al., 2012; Yang and Pan, 2021). These abilities help humans perceive and understand the world. Similarly, for large language models, they are also crucial for understanding spatial environments. To this end, we have constructed a

dataset covering spatial perception, reasoning, path planning and navigation, and spatial language understanding to train the spatial cognitive abilities of LLMs.

Spatial reasoning refers to the construction and manipulation of spatial relationships between objects based on linguistic expressions or symbolic inputs to complete inferential and decision-making tasks such as path planning, scene comprehension, and layout judgments. While some studies indicate that current cutting-edge models face challenges in large-scale spatial cognition, with performance below human levels (Ramakrishnan et al., 2025), LLMs are capable of building limited spatial cognition from unstructured language to support reasoning. Thus, we introduce the OODA loop to enable models to excel in spatial perception, reasoning, path planning, and spatial language understanding.

2.2 OODA

OODA was originally proposed to enhance human decision-making capabilities in adversarial and dynamic environments (Boyd, 1996; Brehmer, 2005) . It consists of four parts: Observe, Orient, Decide, and Act. Observe and Orient correspond to the Perception stage in the Perception-Action Cycle, while Decide and Act correspond to the Action stage (Endsley, 2017). The Observe phase collects internal and external environmental data to build a knowledge graph of the current situation (Bala et al., 2024). The Orient phase analyzes, interprets, and localizes the observed information, with its core lying in the cognitive understanding and interpretation of the task (Abdollahian and Jeffries, 2024). The Decide phase involves choosing among different courses of action based on the understanding of the environment and context, determining the next plan of action. Finally, the Act phase implements the decision and feeds its results back into the next cycle (Bhatt and Ganatra, 2022). In spatial cognition and movement processes, OODA can rapidly iterate information (Bala et al., 2024), greatly improving action accuracy, especially when used in agent collaboration to achieve intelligent spatial decision-making without any extra external prompts (Nguyen et al., 2024; Chang et al., 2024).

3 Approach

We propose **SODA (Spatial OODA)**, which enhances the spatial cognition and reasoning abilities of LLMs by constructing a large-scale text-based spatial dataset, *SPOD-143k*, and embedding the OODA framework. To address the lack of spatial reasoning benchmarks for models without visual input, we designed *SPOD-Bench*. Next, we will first introduce the premise of **SODA**, how OODA-guided prompt chains are used for reasoning, and then describe the composition and construction process of the two datasets.

3.1 Preliminary (OODA-guided Prompt Chain)

We propose a reasoning framework based on the OODA loop that does not involve training or changes to the model's internal structure. A question extractor generates an initial blueprint for the four stages (Observe, Orient, Decide, Act), and a self-instructing prompt generator creates customized prompts for each stage. During the reasoning process, large language models execute the four stages sequentially, enhancing the accuracy and consistency of spatial text tasks without the need for manual prompt tuning, while ensuring traceability.

As shown in Table 1, the experimental results indicate that, without additional training or reinforcement learning, this framework significantly improves the accuracy of GPT-5 and Qwen3-4B across five test sets, with first-answer accuracy rising from 0.252 to 0.813, validating the effectiveness of OODA. However, due to the inference latency caused by multiple agent calls, response efficiency still needs improvement. Therefore, we propose SODA, embedding OODA internally into the model. Below, we first introduce how to construct appropriate training and testing datasets using OODA, and then describe the two-stage training pipeline.

3.2 *SPOD*

To assess the spatial reasoning capabilities of LLMs without visual input, we designed the *SPOD-Bench* based on comparative and developmental psychology. *SPOD-Bench* explores the core representations of language models in a text-based mode, ensuring traceability of the reasoning process. At the same time, to deeply integrate OODA abilities into the LLM, we propose *SPOD-143k*, which contains 64k more data entries than SPOD-Bench and integrates datasets for four additional control tasks (which we also constructed ourselves, with the process outlined later). Each data entry of *SPOD-143k* is designed to include a gold-standard OODA chain,

Model	DR	MPR	DPR	MRM	CMM
GPT-5					
Without OC	77.5	64.5	65.2	30.3	49.8
With OC	94.7	80.6	83.3	50.9	87.0
Gain	**+17.2**	**+16.1**	**+18.1**	**+20.6**	**+37.2**
Qwen3-4B					
Without OC	42.0	38.3	70.6	34.3	34.5
With OC	78.1	56.9	88.2	52.8	51.5
Gain	**+36.1**	**+18.6**	**+17.6**	**+18.5**	**+17.0**
o3					
Without OC	52.0	95.0	92.0	99.9	92.6

Table 1: Performance gains with trainfree O-O-D-A 4-steps Reasoning Chain (OC) enhancement across parts of spatial reasoning tasks with o3 model performance provided for reference. Tasks abbreviations refer to the Dataset Overview below.

thereby providing complete cognitive tracking and guiding the model's internal OODA reasoning capabilities.

3.2.1 Dataset Overview

The *SPOD-Bench* includes 13 task types, covering basic spatial perception to complex multi-step control tasks. Specifically, based on the three common defects identified in LLM spatial reasoning (detailed in the appendix), the dataset is divided into three layers according to the difficulty of the defects:

Tier 1 (Basic Single-Turn Tasks). Inspired by geometric foundational defects, these tasks involve the model misjudging basic concepts, such as Euclidean distances and left-right relationships. To rebuild this foundational knowledge, we provide four single-step datasets: *Memory Path* (**MP**), *Object Location Distance* (**OLD**), *Relative Direction* (**RD**), *Shortest Path* (**SP**).

Tier 2 (Complex Single-Turn Tasks). Inspired by unstable spatial transformation defects, such as rotation, reflection, and coordinate translation, which often disrupt the model's internal reference framework, we provide five datasets that require explicit mental transformations: *Packing Shapes* (**PS**), *Mental Rotation* (**MR**), *Double-Point Relation* (**DPR**), *Multi-Point Relation* (**MPR**), *Door Rotation* (**DR**).

Tier 3 (Multi-Turn Dialogue Tasks). Inspired by long-term planning defects, these tasks reflect the premature interruption of multi-step decision chains. Four datasets require continuous state tracking and planning depth: *Door Rotation Multi-turn* (**DRM**), *Spatial Relation Multi-turn* (**SRM**), *Coordinates Movement Multi-turn* (**CMM**), *Mental Rotation Multi-turn* (**MRM**).

Additional control tasks of SPOD-143k. Based on the *SPOD-Bench*, *SPOD-143k* includes four interactive control **grid world maze tasks**. They are parameterized by grid sizes (3, 5, 7, 9) and obstacle density, covering four different setups: OP, FOP, MCF, and MCO. Examples of tasks for Tier 1-3, and grid-world maze are shown in Figure 2, along with the data size for each *SPOD-143k*.

3.2.2 *SPOD-Bench* Construction

We use external tools and algorithmic programs to generate values, entities, and spatial relationships in QA tasks. A lightweight pipeline in Python handles this: Shapely (Gillies et al., 2024) creates and transforms geometric objects for ground-truth relations, GeoPandasc̃itegeopandas0.15.0 manages CRS conversion and nearest-neighbour queries, and NetworkX (Hagberg et al., 2025) provides shortest paths on graphs. Scalar results are checked with NumPy and manually verified before writing each QA pair and its gold OODA chain to JSON for training and evaluation.

3.2.3 *SPOD-143k* Construction

To ensure scalability and quality, we constructed a high-quality dataset containing OODA reasoning through three steps: data seed annotation, extended generation of instances, and quality control. This process resulted in approximately 143k instances.

- **Step 1: Data Seed Annotation.** For the 13 basic tasks, we generated 100 entries per task as golden data seeds, similar to the test set construction. For control tasks, we use the A* algorithm to generate four interactive grid-world maze datasets (details in the appendix).

- **Step 2: Human-Machine Data Expansion.** We expanded the dataset using high-quality seed data through a human-machine pipeline. Each seed instance was abstracted into a general template, separating logical structure from lexical form. These templates were scaled using scripts and external tools, with LLMs rewriting the instances for lexical diversity. Finally, the golden standard OODA chains were generated by providing questions and real answers with powerful LLMs.

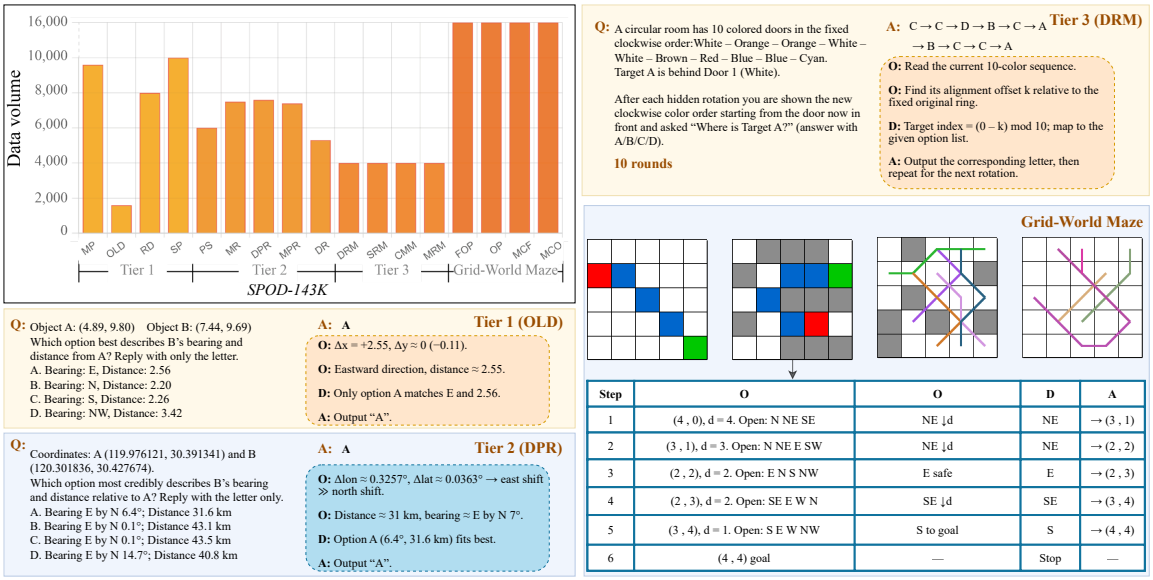

Figure 2: Data Volume and Examples of *SPOD-143k*.

- **Step 3: Iterative Quality Control.** We generated 1,000 entries at a time and randomly selected 5% (50 items) for double-blind manual inspection. A batch was retained only if at least 98% of the inspected items met the following five standards: (i) correct answer; (ii) reasoning matches the answer; (iii) correct OODA stage order; (iv) logically rigorous reasoning chain; (v) consistent spatial context tracking in multi-turn tasks.

3.3 *SODA*

3.3.1 Supervised Fine-Tuning (SFT)

We begin with a pre-trained language model and fine-tune it on *SPOD-143k*. This step provides the model with core spatial concepts (such as orientation, rotation, pathfinding, and stacking) and integrates the OODA process into the model's reasoning while retaining the strengths of the base model.

3.3.2 Group Relative Policy Optimization (GRPO)

Based on the SFT checkpoint, we applied GRPO for reinforcement learning, also on *SPOD-143k*. For each question, the model proposes multiple candidate answers, evaluates their correctness and OODA chain quality, ranks them, and updates its parameters based on this ranking. The group-ranking signal provides a lightweight, effective optimization without needing a critic. The reward function uses a dual-reward system, focusing on both "Conclusion Accuracy" and "Format Accuracy."

$$r_i = r_{\text{ans}}(o_i) + r_{\text{ooda}}(o_i).$$

- The Conclusion Accuracy Reward r_{ans}. We use a regularization matching method, assigning a reward of $+1.0$ for completely correct answers, -1.0 for incorrect answers, and -1.4 for answers with no reasoning.

- The Format Accuracy Reward r_{ooda}. We check whether the generated text sequentially includes the four reasoning stages: Observe, Orient, Decide, and Act. Each correctly included stage is rewarded with $+0.1$, while missing stages, reversed order, or logical jumps are penalized with -0.1, so $r_{\text{ooda}} \in [-0.4, 0.4]$.

Thus, the original reward range for a single candidate is $[-1.8, 1.4]$. This design aims to suppress the model's tendency to miss the OODA thinking chain, extend the model's thinking length within the OODA process, and further improve accuracy on top of the original training foundation.

4 Experiments

This section assesses the performance of **SODA** on *SPOD-Bench* and SPACE, highlighting its advantages. It further demonstrate the generalizability by evaluating its performance in practical applications, including Chess Combat and robotic arm tasks. Additionally, the effectiveness of SODA is validated through sensitivity and ablation analysis.

4.1 Experiment Setup

We trained the Qwen3-4B and Qwen3-14B models during the SFT process, which ran for 3 epochs using the AdamW optimizer. In the GRPO process, 2 epochs were set with a learning rate of 2e-5,

Model	Basic Single-Turn				Complex Single-Turn					Multi-Turn			
	OLD	SP	RD	MP	PS	MR	MPR	DPR	DR	DRM	SRM	MRM	CMM
Non-Reasoning Models													
GPT-5 nano	97.6	93.5	92.8	96.2	65.1	56.2	46.5	58.9	68.1	29.1	47.0	25.0	34.1
DeepSeek-v3	97.5	93.9	92.9	98.6	66.7	61.8	46.9	57.9	70.3	27.1	48.3	26.3	36.9
Qwen2.5-14B-Instr.	98.4	83.7	91.2	93.6	98.1	74.9	60.4	65.8	45.3	40.2	49.4	28.6	42.8
GPT-5	98.9	95.2	96.5	97.9	81.5	88.6	63.0	64.1	76.2	47.1	62.3	28.3	47.5
Claude-3-7-sonnet	98.4	**98.1**	84.9	95.8	**100.0**	92.9	86.7	75.3	77.3	76.2	87.1	27.8	48.2
Reasoning Models													
o4-mini	**100.0**	97.3	97.1	95.8	**100.0**	94.7	97.4	88.4	26.9	96.9	93.2	99.7	94.1
Grok 4.1	99.3	96.8	98.1	98.7	95.0	95.1	70.0	80.2	83.1	55.0	61.2	33.1	44.8
Claude Opus 4.5	99.7	97.5	98.2	98.5	98.8	95.5	96.0	93.0	94.9	98.0	95.5	97.5	94.5
o3	**100.0**	97.8	98.4	93.9	**100.0**	94.1	95.3	92.5	52.9	99.6	96.8	**99.9**	92.6
DeepSeek-R1	**100.0**	90.7	99.3	94.2	99.6	92.8	94.8	85.2	33.4	83.1	92.7	98.9	**95.9**
Qwen3-4B	93.8	66.3	95.7	89.8	98.4	91.9	73.6	87.9	42.1	38.3	54.7	34.3	34.5
Qwen3-14B	**100.0**	96.6	**100.0**	94.2	**100.0**	95.7	91.6	93.3	63.9	80.1	79.7	42.8	72.1
Our Models based on Qwen3													
SODA-4B	99.5	82.1	97.3	88.9	**100.0**	93.6	98.3	98.6	46.7	60.3	59.4	31.4	76.8
SODA-14B	**100.0**	97.6	99.8	**99.3**	**100.0**	**98.2**	**100.0**	**100.0**	**95.1**	**99.8**	**97.3**	63.9	95.1

Table 2: Overall accuracy (%) of baseline LLMs, reasoning-enhanced LLMs, and our SODA models on the 13-task SPOD-Bench. Tasks are grouped by tier: Basic Single-Turn (**OLD**, **SP**, **RD**, **MP**), Complex Single-Turn (**PS**, **MR**, **MPR**, **DPR**, **DR**), and Multi-Turn (**DRM**, **SRM**, **MRM**, **CMM**). The **best** score in each column is shown in **bold**; the best score in each sub-column is underlined. higher values indicate better spatial-cognition performance.

0.03 warm-up steps, and cosine learning rate decay. The training data for both stages are derived from *SPOD-143k*. The training data for control abilities included tasks involving the movement of multiple objects.

4.2 Evaluating in *SPOD-Bench*

To establish a performance baseline, we evaluated several mainstream large-scale language models, including mainstream Reasoning and non-Resoning in OpenAI (OpenAI, 2023, 2025), Claude (Anthropic, 2025,?), DeepSeek (Guo et al., 2025; DeepSeek-AI et al., 2025), Qwen (Yang et al., 2025) and Grok (xAI, 2025). The results are shown in Table 2.

Overall Performance Overview Both the 4B and 14B models demonstrate a comprehensive performance improvement with **SODA**, particularly in movement and path planning tasks. The SODA-4B model achieved an accuracy of 76.8% in CMM, significantly surpassing the baseline model's accuracy of 34.5%. The SODA-14B model scored 95.1%, outperforming most commercial models, proving that specialized spatial cognition training enhances object control and path planning abilities.

Impact of Model Size Both the SODA-4B and SODA-14B models show advantages after training, especially in complex multi-turn tasks. SODA-14B excels with an accuracy of 99.8%, 97.3%, and 95.1% on DRM, SRM, and CMM tasks, respectively. The SODA-4B model still performs strongly in complex single-turn tasks (such as MPR and DPR), with accuracies of 98.3% and 98.6%, surpassing many larger models. These results show that **SODA** enhances spatial reasoning in both large and small models.

Task-Level Performance Analysis Model performance varies across task levels. In Tier 1 (basic single-turn tasks), models showed high performance, with most exceeding 90%. As tasks became more complex in the Tier 2 (complex single-turn) and Tier 3 (multi-turn), model differences emerged. Non-reasoning models dropped significantly in performance with increased complexity. While most reasoning models performed better on harder tasks, some, like the grok 4.1, still declined in performance, highlighting the challenge of sustained reasoning over extended contexts.

4.3 Evaluating in SPACE

To demonstrate the generalizability of the **SODA**, we selected SPACE (Ramakrishnan et al., 2025)

Model	PTT_text	SAtt_text	CBTT_text	SAdd_text	MRT_text	MPFB_text	JLO_text
GPT-5	58.2	99.0	83.0	94.1	43.5	51.0	67.0
GPT-5 nano	51.5	94.8	58.0	61.2	31.0	35.5	30.5
o3	98.0	**100.0**	**100.0**	**100.0**	**97.5**	**97.0**	**76.0**
Qwen2.5-14B-Inst.	39.5	91.0	60.0	81.0	32.5	34.5	33.0
Qwen3-4B	97.0	**100.0**	76.0	96.5	58.8	31.5	59.5
Qwen3-14B	98.0	**100.0**	93.0	97.5	64.4	37.0	61.5
SODA-4B	98.5	**100.0**	84.0	99.0	70.0	37.5	63.5
SODA-14B	**98.8**	**100.0**	96.0	**100.0**	76.3	39.5	68.5

Table 3: Model performance (%) on the SPACE benchmark. Task abbreviations: DE = DirectionEstimation_BEVtext, DisE = DistanceEstimation_BEVtext, PTT = PerspectiveTaking_text, SA = SelectiveAttention_text, Corsi = CBTT_text, SAdd = SpatialAddition_text, MRT = MentalRotation_text, MS = MapSketching_BEVtext, MPFB = MPFB_text, JLO = JLO_text. "–" indicates the model was not evaluated on the corresponding task. The **best** and the second-best score in each column is shown in **bold** and underlined.

as an additional validation. SPACE encompasses multiple tasks, such as perspective taking and selective attention. The results in Table 3 show that, whether for SODA-4B or SODA-14B, both models performed excellently in most tasks, surpassing their base models, Qwen3-4B and Qwen3-14B, while also achieving comparable or superior performance to closed-source models, highlighting the advantage of SODA in spatial reasoning tasks.

4.4 Evaluation onto Reallife Application

4.4.1 Chess Combat

Model	Max Legal Moves	Min Legal Moves	Avg Legal Moves	Completion Rate (%)
Baseline Models				
GPT-5	22	2	8.50	5
Claude-3.7	6	0	2.46	0
Claude Opus 4.5	24	3	13.80	10
Deepseek-R1	21	0	6.60	0
Qwen3-4B	20	0	10.50	4
Qwen3-14B	26	1	13.50	12
o3	**56**	0	**24.10**	27
Our Models				
SODA-4B	28	3	14.87	86
SODA-14B	35	**4**	16.01	**94**

Table 4: Performance comparison on the Chess Combat task. Legal moves indicate the number of valid chess moves made during gameplay. Higher completion rates demonstrate better strategic understanding.The **best** and the second-best score in each column is shown in **bold** and underlined.

To evaluate the model's generalization ability in complex spatial control tasks, we introduced chess combat, shown in Figure 4, as a test platform. This platform is based on chess rules and emphasizes coordination and confrontation among multiple pieces, requiring the model to integrate spatial perception, reasoning, and control. We used the Stockfish engine as the opponent and conducted 100 matches to assess the model's average number of legal moves, completion rate, and maximum/minimum legal moves. As shown in Table 4, the results indicate that general-purpose large language models perform poorly in chess combat, whereas the SODA models perform better. SODA-4B achieved 14.87 legal moves with an 86% completion rate, and SODA-14B further improved to 16.01 legal moves with a 94% completion rate, surpassing several closed-source systems. Without any specialized training, the SODA models successfully transferred to the chess task through spatial cognition training via OODA.

Error analysis revealed that LLMs struggle with board state awareness and adherence to chess rules, such as losing track of piece positions or violating fundamental rules. These errors collectively underscore the persistent gaps in long-range spatial tracking and rule-constrained planning.

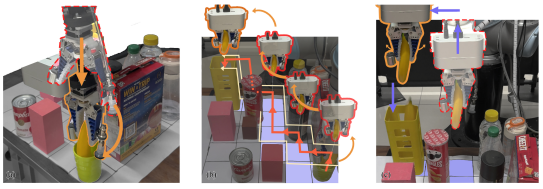

Figure 3: Robotic-arm Applicaiton: the model plans and executes obstacle-avoiding pick-and-place motions.

4.4.2 Embodied Validation on a Robotic Arm

To further assess the model's capabilities and bridge the gap between abstract textual reasoning and embodied intelligence, we conducted experiments on a real robotic-arm platform, shown in

Model	Basic Single-Turn				Complex Single-Turn					Multi-Turn			
	OLD	SP	RD	MP	PS	MR	MPR	DPR	DR	DRM	SRM	MRM	CMM
SODA-4B(SFT only)	86.7	94.8	94.4	92.6	67.9	73.6	70.6	62.5	66.7	99.2	82.5	31.4	34.5
SODA-4B	99.5	82.1	97.3	88.9	**100.0**	93.6	98.3	98.6	46.7	60.3	59.4	31.4	76.8
SODA-14B(SFT only)	**100.0**	**99.7**	99.2	**99.4**	99.8	97.5	98.1	96.6	**98.3**	99.6	96.8	50.7	**95.7**
SODA-14B	**100.0**	97.6	**99.8**	99.3	**100.0**	**98.2**	**100.0**	**100.0**	95.1	**99.8**	**97.3**	**63.9**	95.1

Table 5: Abalation Results: SPOD-Bench performance (%) on SODA model(4B and 14B) after SFT and SFT-GRPO.

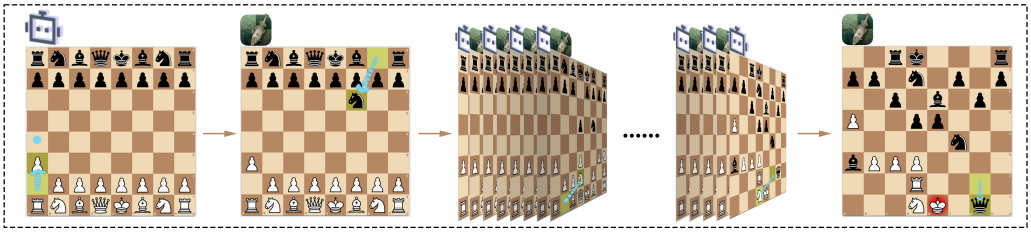

Figure 4: Chess Combat Evaluation & Application: an 8 × 8 grid-world chess setting that tests models' integrated spatial perception, reasoning, and control against the Stockfish engine.

Figure 3. The results demonstrate that the model can precisely control the arm to avoid obstacles and move target objects to their designated positions. More detailed content can be found in the appendix.

4.5 Sensitivity Analysis

Figure 5 shows the training dynamics of the **SODA** model during the GRPO stage. Whether for the 4B or 14B model, answer accuracy steadily improved after about 2,000 steps, with the 14B model converging faster and achieving higher final accuracy. This indicates that the GRPO strategy ensures stable optimization across different model sizes. Figure 5(b) shows that the average reward increased rapidly and synchronized with accuracy, staying consistently between -1.0 and 1.5, confirming that the dual-channel reward (answer correctness + OODA integrity) aligns with the task objectives and validating the stability of the reward strategy and the controllability of the training process. Figure 5(c) and (d) show that the token count for Observe and Orient rapidly increased during the first 1,000 steps and then saturated, while Decide and Act remained shorter. This suggests that the GRPO strategy helps the model prioritize allocating reasoning steps to perception and analysis, reducing redundant thinking and improving accuracy.

4.6 Ablation Experiment

The comparison between models using only SFT and the complete **SODA** framework highlights the effectiveness of the approach. As shown in the

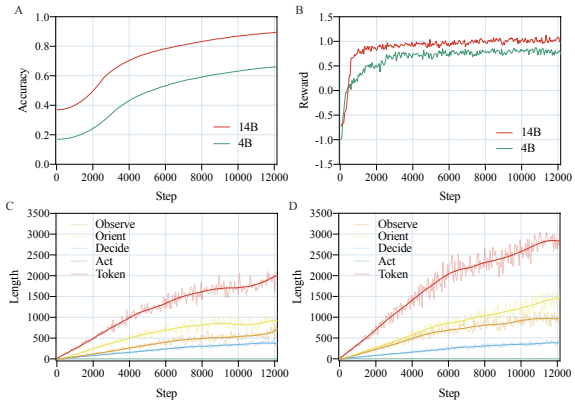

Figure 5: GRPO training curves

Table 5, the SODA-4B and SODA-14B models, combining SFT and GRPO, significantly outperform models using only SFT, especially in complex single-turn tasks and multi-turn tasks, with accuracy substantially improved. SODA-14B demonstrates high accuracy in basic tasks, complex tasks, and multi-turn tasks, indicating that the full integration of the **SODA** training framework (SFT and GRPO) greatly enhances the model's reasoning and spatial cognition capabilities.

5 Conclusion

This study presents **SODA**, a spatial-aware training framework that embeds the OODA reasoning mechanism into LLMs via two-stage training, using our conducted OODA-based dataset *SPOD-143k*. Additionally, we introduce *SPOD-Bench*, the first diverse and hierarchical pure-text spatial reasoning benchmark, covering tasks from perception to complex control and long-term planning. Results on both SPOD-bench and public datasets confirm

that **SODA** not only outperforms its backbone models but also surpasses leading commercial models, also demonstrating its generalizations from strategic planning in chess to demonstrating its spatial control abilities through a real-world robot, offering new insights in a non-multimodal perspective.

6 Limitations

While this work offers a novel, non-multimodal perspective on enhancing the spatial capabilities of large language models (LLMs), our approach has several limitations. Our unimodal framework primarily provides insights into text-based spatial reasoning. However, the field is increasingly moving toward integrated systems that incorporate diverse sensory inputs to understand and interact with the world. To expand the model's perceptual dimensions, future research should explore spatial understanding in other modalities, such as auditory perception from audio and visual perception from 2D, 3D, and 4D data. Building on the unimodal spatial-aware framework proposed in this paper, there is a clear need for more comprehensive and spatial-aware unified multimodal training strategies, which are crucial for developing the next generation of LLMs with physically-grounded understanding (Liu et al., 2025).

Additionally, while our model generalizes to existing benchmarks, its robustness in complex or Out-of-Distribution (OOD) scenarios requires further investigation. Current evaluation methods are limited in scope and lack comprehensive quantitative metrics. This necessitates next-generation benchmarks focused on quantitative reasoning (e.g., distance, size, angles) and dynamic transformations like motion. Dynamic reasoning, in particular, remains a significant challenge.

Finally, our exploration of real-world applications was limited. Future work can address application-level engineering, such as lightweight deployment and enhanced human-computer interaction. However, achieving true spatial intelligence will likely require fundamental shifts in training and architecture. Promising directions include joint multimodal pre-training and integrating components like persistent memory to build explicit, updatable world models.

Acknowledgments

This work is partially supported by the Key R&D Program of Zhejiang (No. 2024C03247).

References

Mark Abdollahian and Chasen Jeffries. 2024. Simulating boyd's ooda loop: Towards an abm of human agency and sensemaking in dynamic, competitive environments. In *Proceedings of the 17th International Conference on Advances in Computer-Human Interactions (ACHI)*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Mohamed Aghzal, Erion Plaku, and Ziyu Yao. 2025. Can large language models be good path planners? a benchmark and investigation on spatial-temporal reasoning. *Preprint*, arXiv:2310.03249.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, and 1 others. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Anthropic. 2025. Claude 3: Next-generation capabilities for helpful and harmless language models. *arXiv preprint arXiv:2503.00001*, 1(1):1–45.

Anthropic. 2025. Introducing claude opus 4.5. https://www.anthropic.com/news/claude-opus-4-5.

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2025. Hallucination of multimodal large language models: A survey. *Preprint*, arXiv:2404.18930.

Mihir Bala, Aditya Chanana, Xiangliang Chen, Qifei Dong, Thomas Eiszler, Jingao Xu, Padmanabhan Pillai, and Mahadev Satyanarayanan. 2024. The ooda loop of cloudlet-based autonomous drones. In *2024 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 178–190. IEEE.

Anant Bhatt and Amit Ganatra. 2022. Explosive weapons and arms detection with singular classification (wardic) on novel weapon dataset using deep learning: enhanced ooda loop. *Engineered Science*, 20(3):252–266.

Antonio Bollig, Marília Freire, Konrad Bücking, Jana Kühnapfel, and Markus Knaden. 2024. Aggressive conflict impacts path integration in homing desert ants. *bioRxiv*, pages 2024–11.

John R. Boyd. 1996. The essence of winning and losing. Unclassified briefing. Last version 28 June 1996.

Berndt Brehmer. 2005. The dynamic ooda loop: Amalgamating boyd's ooda loop and the cybernetic approach to command and control. In *10th International Command and Control Research and Technology Symposium*, McLean, VA.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, and 35 others. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *Preprint*, arXiv:2307.15818.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, and 1 others. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Cheng Chang, Siqi Wang, Jiawei Zhang, Jingwei Ge, and Li Li. 2024. Llmscenario: Large language model driven scenario generation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 54(11):6581–6594.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. *Preprint*, arXiv:2401.12168.

Thomas Collett, Paul Graham, and Stanley Heinze. 2025. The neuroethology of ant navigation. *Current Biology*, 35(3):R110–R124.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Ling L Dong and Ila R Fiete. 2024. Grid cells in cognition: mechanisms and function. *Annual Review of Neuroscience*, 47.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, and 1 others. 2023. Palm-e: An embodied multimodal language model.

Mengfei Du, Binhao Wu, Zejun Li, Xuanjing Huang, and Zhongyu Wei. 2024. Embspatial-bench: Benchmarking spatial understanding for embodied tasks with large vision-language models. *Preprint*, arXiv:2406.05756.

Martin Egelhaaf and Jens P Lindemann. 2025. Path integration and optic flow in flying insects: a review of current evidence. *Journal of Comparative Physiology A*, pages 1–27.

Robert E. Enck. 2012. The ooda loop. *Home Health Care Management & Practice*, 24(3):123–124.

Mica R Endsley. 2017. Toward a theory of situation awareness in dynamic systems. In *Situational awareness*, pages 9–42. Routledge.

Joaquın M Fuster. 2004. Upper processing stages of the perception–action cycle. *Trends in cognitive sciences*, 8(4):143–145.

Sean Gillies, Allan Doyle, Cal Smith Barber, and 1 others. 2024. Shapely: Manipulation and analysis of geometric objects. Version 2.0.5.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Aric A. Hagberg, Daniel A. Schult, Pieter J. Swart, and the NetworkX Developers. 2025. NetworkX: Network analysis in python. `https://github.com/networkx/networkx`. Version 3.5.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Zilong Ji, Eleonora Lomi, Kate Jeffery, Anna S Mitchell, and Neil Burgess. 2025. Phase precession relative to turning angle in theta-modulated head direction cells. *Hippocampus*, 35(2):e70008.

Sangeet Khemlani, Tyler Tran, Nathaniel Gyory, Anthony M. Harrison, Wallace E. Lawson, Ravenna Thielstrom, Hunter Thompson, Taaren Singh, and J. Gregory Trafton. 2025. Vision language models are unreliable at trivial spatial cognition. *Preprint*, arXiv:2504.16061.

Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. 2021. Pathdreamer: A world model for indoor navigation. *Preprint*, arXiv:2105.08756.

Sang Ah Lee, Valeria A Sovrano, and Elizabeth S Spelke. 2012. Navigation as a source of geometric knowledge: Young children's use of length, angle, distance, and direction in a reorientation task. *Cognition*, 123(1):144–161.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2024. Seed-bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13299–13308.

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. 2022. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *Preprint*, arXiv:2203.17270.

MingShan Liu and Jialing Fang. 2025. Enhancing mathematical reasoning in large language models with self-consistency-based hallucination detection. *Preprint*, arXiv:2504.09440.

Weichen Liu, Qiyao Xue, Haoming Wang, Xiangyu Yin, Boyuan Yang, and Wei Gao. 2025. Spatial reasoning in multimodal large language models: A survey of tasks, benchmarks and methods. *Preprint*, arXiv:2511.15722.

Zooey Nguyen, Anthony Annunziata, Vinh Luong, Sang Dinh, Quynh Le, Anh Hai Ha, Chanh Le, Hong An Phan, Shruti Raghavan, and Christopher Nguyen. 2024. Enhancing qa with domain-specific fine-tuning and iterative reasoning: A comparative study. *Preprint*, arXiv:2404.11792.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 1(1):1–60.

OpenAI. 2025. Introducing gpt-5. `https://openai.com/index/introducing-gpt-5/`.

Santhosh Kumar Ramakrishnan, Erik Wijmans, Philipp Kraehenbuehl, and Vladlen Koltun. 2025. Does spatial cognition emerge in frontier models? *Preprint*, arXiv:2410.06468.

Fedor Rodionov, Abdelrahman Eldesokey, Michael Birsak, John Femiani, Bernard Ghanem, and Peter Wonka. 2025. Planqa: A benchmark for spatial reasoning in llms using structured representations. *Preprint*, arXiv:2507.07644.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Dong Shu, Haiyan Zhao, Jingyu Hu, Weiru Liu, Ali Payani, Lu Cheng, and Mengnan Du. 2025. Large vision-language model alignment and misalignment: A survey through the lens of explainability. *Preprint*, arXiv:2501.01346.

Ilias Stogiannidis, Steven McDonagh, and Sotirios A. Tsaftaris. 2025. Mind the gap: Benchmarking spatial reasoning in vision-language models. *Preprint*, arXiv:2503.19707.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11888–11898.

Susumu Takahashi, Fumiya Sawatani, Kaoru Ide, Takaaki K Abe, Takashi Kitagawa, and Yuya Makiguchi. 2024. Mapping spatial memory in teleosts: a new frontier in neural logging techniques. *Frontiers in Physiology*, 15:1499058.

Alexey Tikhonov. 2024. Plugh: A benchmark for spatial understanding and reasoning in large language models. *Preprint*, arXiv:2408.04648.

Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Yixuan Li, and Neel Joshi. 2024. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. *Preprint*, arXiv:2406.14852.

Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. 2024. Mind's eye of llms: visualization-of-thought elicits spatial reasoning in large language models. *Advances in Neural Information Processing Systems*, 37:90277–90317.

xAI. 2025. Grok 4.1. `https://x.ai/news/grok-4-1`.

Zikai Xie. 2025. Order matters in hallucination: Reasoning order as benchmark and reflexive prompting for large-language-models. *Preprint*, arXiv:2408.05093.

Liuchang Xu, Shuo Zhao, Qingming Lin, Luyao Chen, Qianqian Luo, Sensen Wu, Xinyue Ye, Hailin Feng, and Zhenhong Du. 2024. Evaluating large language models on spatial tasks: A multi-task benchmarking study. *arXiv preprint arXiv:2408.14438*.

Li Xuan, Zhang Haoxiang, Jiang Baozheng, Li Yanxia, and Li You. 2024. A benchmark dataset for evaluating spatial perception in multimodal large models. In *Proceedings of the First International Workshop on IoT Datasets for Multi-modal Large Model*, pages 37–43.

Yutaro Yamada, Yihan Bao, Andrew K. Lampinen, Jungo Kasai, and Ilker Yildirim. 2024. Evaluating spatial understanding of large language models. *Preprint*, arXiv:2310.14540.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Xiaoli Yang and Yuejuan Pan. 2021. Spatial language of young children during block play in kindergartens in urban china. *Frontiers in Psychology*, 12:568638.

Zhiyu Yin, Kehai Chen, Xuefeng Bai, Ruili Jiang, Juntao Li, Hongdong Li, Jin Liu, Yang Xiang, Jun Yu, and Min Zhang. 2025. Asurvey: Spatiotemporal consistency in video generation. *Preprint*, arXiv:2502.17863.

A Appendix

B Datasets and Conversation Examples

Tier 1 (**Basic Single-Turn Tasks**):

memory_path(MP):

A similar paradigm was introduced by Mittelstaedt et al. while investigating the path-integration mechanism that supports homing behavior in rodents. Gerbils had to correctly account for spatial changes caused by translations or rotations of the arena. In recent years, several reading-comprehension and large-language-model benchmarks have required models to handle analogous position-update mechanisms (Weston et al., 2016).

In our implementation, we randomly sample a starting point (x_0, y_0) and then generate an action sequence of 5–15 moves drawn from (N, S, E, W), with each move shifting the position by ± 1 unit. The sequence is constructed so that it uniquely determines an end point. The model must compute $\Sigma(\Delta x, \Delta y)$ and return the final coordinates.

Conversation Examples:

memory_path(MP):

Q:Starting coordinates (1, 1), follow the action sequence W E N S W E N W S N to walk; which option below is the final
coordinate? Please select and answer only the letter: A:(1, 3); B:(2, 0); C:(0, 2); D:(8, 8)
A:C

object_location_distance (OLD):

A distance-estimation task. This type of exercise dates back to the foundations of analytic geometry, where points are expressed in coordinates and the Pythagorean theorem is used to compute the distance between two points, establishing the methodological basis for later "coordinate-distance" drills. In contemporary psychology and neuroscience, the task is regarded as a classic paradigm for assessing metric spatial representations, used to test the accuracy with which cognitive maps encode distances (Shepard, 1987).

Our implementation randomly generates the coordinates of two points. Given these coordinates, the model must calculate the Euclidean distance between the two points.

Conversation Examples:

object_location_distance:

Q:The position of object A is (3.44, 8.17), and the position of object B is (6.72, 7.32). Which option below is closest to the Euclidean distance between them? Please select and answer only the letter: A:3.48; B:3.38; C:2.19; D:4.73
A:B

relative_direction (RD):

This task falls under "relative direction–distance estimation." It simultaneously tests the ability to represent discrete directions and continuous distances, and is commonly used to assess two-dimensional geometric reasoning and numerical-approximation skills in humans or large language models. Relative-direction training is also employed in the computational cognitive model of direction judgments by Phillip M. Newman et al.

Our implementation randomly provides reasonable coordinates for two points A and B; the model must judge which of the eight compass sectors (N/NE/E/SE/S/SW/W/NW) point B occupies relative to point A, then compute the Euclidean distance, and finally select, from several options, the combination closest to the correct answer.

Conversation Examples:

relative_direction:

Q:The coordinates of object A are (8.50, 2.60), and the coordinates of object B are (0.74, 5.94).Which option most accurately describes the direction and distance of B relative to A? Please select and answer only the letter: A:Direction: NE; Distance: 5.14 units; B:Direction: S; Distance: 5.69 units; C:Direction: W; Distance: 10.15 units; D:Direction: NW; Distance: 8.45 units
A:D

shortest_path (SP):

This task resembles the traveling salesman problem (TSP) task, discussed in detail by MacGregor and Ormerod (1996), and is used in laboratory settings to test graph-based spatial problem solving. Participants see on paper a lattice layout of 10 or 20 points and must freely choose a starting point, draw the shortest closed path that visits all points and returns to the start, and clearly mark both the starting point and the travel direction; the solution is presented as a hand-drawn path. In recent years, this task has been used to evaluate

the graph-navigation performance of large neural networks capable of learning structured representations (Kool et al., 2018). Our implementation randomly generates an undirected graph with 5–15 nodes and an unequal number of edges, randomly selects distinct start and end points, and requires computation of the unique shortest path.

Conversation Examples:
shortest_path:
Q:Given an undirected graph, the node set is ['A', 'B', 'C', 'D', 'E', 'F', 'G'], and the edge set is {A-D, A-F, B-C, B-D, B-E, B-G}. Write the shortest path from node E to node A. Which option below is the shortest path? Please select and answer only the letter: A:E-D-A-B; B:E-B-D-A; C:B-D-E-A; D:E-A-B-D
A:B

B.0.1 Tier 2(Complex Single-Turn Tasks):

packing_shapes (PS):
Okagaki and Frensch proposed a similar problem; it is a computerized, two-dimensional bin-packing task used to study the acquisition of visuo-spatial skills and strategic planning. In this task, participants train mental rotation and visualization skills with Tetris-like stimuli: the screen displays a fixed-size rectangular "container" together with several geometric shapes, and participants must translate and rotate all shapes to fit them into the container while minimizing unused blank area. This paradigm has recently served as a spatial-reasoning benchmark for deep-reinforcement-learning agents learning packing strategies (Wang et al., 2022). Our implementation randomly generates a rectangular plane of reasonable, random size and simultaneously generates five rectangles and circles that can fit inside the plane, then instructs the model to compute the remaining blank area.

Conversation Examples:
packing_shapes:
Q:Given a 9.43 × 9.62 rectangular packing area, and the following objects: Object 1: Rectangle, width 2.72, height 2.88; Object 2: Rectangle, width 1.19, height 1.74; Object 3: Rectangle, width 2.33, height 1.64; Object 4: Circle, radius 1.08; Object 5: Circle, radius 1.78. Which option below is closest to the correct remaining space? Please select and answer only the letter: A: Remaining space: 64.30 square units; B: Remaining space: 72.95 square units; C: Remaining space: 63.32 square units; D: Remaining space: 54.15 square units.
A:C

mental_rotation (MR):
This paradigm originates from Shepard & Metzler's (1971) pioneering study on matching rotated 3-D objects and was quantified by Cooper & Shepard (1973) as the classic effect wherein reaction time increases linearly with rotation angle θ. In our task, the model must apply a specified axis rotation to an object's orientation vector held in working memory, then judge which option among several best matches the resulting vector. The task assesses abilities in 3-D coordinate transformation, spatial orientation updating, and mental-rotation speed.
Conversation Examples:
mental_rotation:
Q:The forward vector of a cube is (1, 0, 0). It is rotated counterclockwise by 274° around the axis vector (-0.599, +0.289, +0.746). Which of the following is the forward vector after rotation? Please select and answer only the letter: A:(-0.641, +0.244, -0.728); B:(+0.041, +0.054, +0.998); C:(+0.404, -0.906, -0.128); D:(+0.113, +0.296, -0.948).
A:C

Doublepoint-relation Test (DPR) & Multipoint-relation Test (MPR):
This paradigm is adapted from the original "multi-segment direction-distance reasoning" task used to study human way-finding and cartographic cognition (Kozhevnikov et al., 2001). In the test, participants are shown a 2-D array of objects, imagine adopting a perspective within the array, and indicate the direction of a target object from that viewpoint. In recent years it has also been adapted to evaluate large-language-model (LLM) spatial-reasoning ability (Danish et al., 2024). Unlike the classical test, the MPR is entirely text-based and leverages the model's computational capacity, making it easy to administer to language models.

Our implementation is as follows:
•**Doublepoint-relation Test:** randomly generate two geographic coordinates within a fixed range; the model must compute the geographic distance between the two points and infer their precise spatial relation.

• **Multipoint-relation Test:** take an arbitrary seed coordinate as the center, randomly sample four real POIs within a given radius, and provide the spatial and distance relations $POI_1 \rightarrow POI_2$, $POI_2 \rightarrow POI_3$, $POI_3 \rightarrow POI_4$; the model must infer the composite relation $POI_1 \rightarrow POI_4$.
Conversation Examples:
doublepoint_relation:
Q:Coordinates A(120.030685,30.246885), and B(120.317380,30.404990).Which option most reliably describes the direction and distance of B relative to A? Please select and answer only the letter: A:Direction: N86.2°E; Distance: 28.4 km; B:Direction: N61.1°E; Distance: 32.7 km; C:Direction: N63.7°E; Distance: 24.0 km; D:Direction: N31.9°E; Distance: 27.5 km.
A:B

multipoint_relation:
Q:Zhejiang Deren Financial Co., Ltd. is located 1.5 km south-southeast (19.2°) of Xinghuo Apartment Building 2. Xinghuo Apartment Building 2 is 27.1 km south-southeast (83.0°) of Hangda Equipment. Hangda Equipment is 19.4 km south-southwest (48.3°) of Chengdong Primary School (Bus Stop). What is the direction and distance from A to D? Please select and answer only the letter: A:Direction: S22.3°E; Distance: 27.9 km; B:Direction: S15.3°E; Distance: 30.4 km; C:Direction: S36.1°E; Distance: 21.9 km; D:Direction: S41.8°E; Distance: 24.8 km.
A:C

door_rotation(DR):
Circular–orientation updating task (COU):
This paradigm stems from research on human self-motion heading maintenance. Wang & Spelke (2000) showed that, in enclosed spaces, individuals can rely on a "post-rotation updating" mechanism to maintain an instantaneous representation of a target's direction. The task is a circular viewpoint-updating paradigm inspired by studies on self-motion and heading maintenance, especially the spatial-updating model for self-rotation in closed environments proposed by Wang et al. (2000), which has informed cognitive psychology. Our implementation presents, in verbal form, an experimental scene with a circular array of randomly segmented, sequentially colored door panels. The initial target is hidden behind the first door directly in front of the model;then the sytem rotates the model environment. the model must rely on memory to track its own orientation changes and continuously infer the current index of the target door.

Conversation Examples:
door_rotation:
Q:You are currently at a 9-door hub with doors surrounding you. Target A is behind the door directly in front of you. The colors of the 9 doors (starting directly in front of you and going clockwise) are: Cyan, Red, Orange, Red, Blue, Blue, Cyan, Orange, Green. Then, you are rotated **counterclockwise** by 3 doors (shifted 3 positions left). The new colors of the doors starting directly in front of you and going clockwise are: Red, Blue, Blue, Cyan, Orange, Green, Cyan, Red, Orange. Which door is Target A now behind? Please select one option and answer only the letter: A: Door 5; B: Door 2; C: Door 4; D: Door 6.
A:C

B.0.2 Tier 3

door_rotation_multiturn(DRM):
Based on the previous door_rotation(DR) method, the difference is that in the door_rotation_multiturn task the environment is rotated multiple times, and the model must rely on memory to track its own orientation changes and infer the target door's number in real time.

Conversation Examples:
door_rotation_multiturn:
Q:In a circular space with 13 doors, each door has a unique color. Target A is behind the 1st door directly in front of you. The colors of the doors clockwise from your position are: Orange, Black, Blue, Red, Orange, Purple, Blue, Brown, Black, Green, White, Cyan, Yellow. After being rotated, the new colors clockwise from your position are: White, Cyan, Yellow, Orange, Black, Blue, Red, Orange, Purple, Blue, Brown, Black, Green. Where is Target A now? A. Door 7 B. Door 4 C. Door 5 D. Door 10.
A:B
Q:You were rotated again. After the rotation, the new colors clockwise from your position are: Black, Green, White, Cyan, Yellow, Orange, Black, Blue, Red, Orange, Purple, Blue, Brown. Where is Target A now? A. Door 2 B. Door 6 C. Door 13 D. Door 5
A:B

Q:You were rotated again. After the rotation, the new colors clockwise from your position are: Yellow, Orange, Black, Blue, Red, Orange, Purple, Blue, Brown, Black, Green, White, Cyan. Where is Target A now? A. Door 6 B. Door 2 C. Door 11 D. Door 5.
A:B
Q:You were rotated again. After the rotation, the new colors clockwise from your position are: Blue, Brown, Black, Green, White, Cyan, Yellow, Orange, Black, Blue, Red, Orange, Purple. Where is Target A now? A. Door 4 B. Door 3 C. Door 8 D. Door 10
A:C
Q:You were rotated again. After the rotation, the new colors clockwise from your position are: Orange, Black, Blue, Red, Orange, Purple, Blue, Brown, Black, Green, White, Cyan, Yellow. Where is Target A now? A. Door 1 B. Door 7 C. Door 6 D. Door 9
A:A
Q:You were rotated again. After the rotation, the new colors clockwise from your position are: Orange, Black, Blue, Red, Orange, Purple, Blue, Brown, Black, Green, White, Cyan, Yellow. Where is Target A now? A. Door 4 B. Door 1 C. Door 10 D. Door 5
A:B
Q:You were rotated again. After the rotation, the new colors clockwise from your position are: Black, Blue, Red, Orange, Purple, Blue, Brown, Black, Green, White, Cyan, Yellow, Orange. Where is Target A now? A. Door 8 B. Door 13 C. Door 1 D. Door 11
A:B
Q:You were rotated again. After the rotation, the new colors clockwise from your position are: Black, Green, White, Cyan, Yellow, Orange, Black, Blue, Red, Orange, Purple, Blue, Brown. Where is Target A now? A. Door 13 B. Door 11 C. Door 6 D. Door 2
A:C
Q:You were rotated again. After the rotation, the new colors clockwise from your position are: Black, Blue, Red, Orange, Purple, Blue, Brown, Black, Green, White, Cyan, Yellow, Orange. Where is Target A now? A. Door 11 B. Door 13 C. Door 7 D. Door 3
A:B
Q:You were rotated again. After the rotation, the new colors clockwise from your position are: Cyan, Yellow, Orange, Black, Blue, Red, Orange, Purple, Blue, Brown, Black, Green, White. Where is Target A now? A. Door 3 B. Door 12 C. Door 1 D. Door 13
A:A

spatial_relation_multiturn(SRM):
Based on the aforementioned relative-direction approach, we introduce multi-turn dialogue operations to increase training complexity and enhance the model's ability to comprehensively retain key contextual information.

Conversation Examples:
spatial_relation_multiturn: Q:The coordinates of object A are (40.77, 31.69). Object B is located SW of object A at a distance of 9.61 units. What is the direction and distance of object B relative to object A? A: Direction: W; Distance: 13.27 units B: Direction: NE; Distance: 10.50 units C: Direction: SW; Distance: 9.61 units D: Direction: E; Distance: 9.61 units A:C Q:The coordinates of object A are (40.77, 31.69). Object C is located NE of object B at a distance of 9.72 units. What is the direction and distance of object C relative to object A? A: Direction: NE; Distance: 0.10 units B: Direction: SE; Distance: 0.12 units C: Direction: E; Distance: 0.10 units D: Direction: W; Distance: 0.08 units A:A Q:The coordinates of object A are (40.77, 31.69). Object D is located SW of object C at a distance of 3.10 units. What is the direction and distance of object D relative to object A? A: Direction: SW; Distance: 3.00 units B: Direction: E; Distance: 1.82 units C: Direction: N; Distance: 3.76 units D: Direction: SE; Distance: 2.70 units A:A Q:The coordinates of object A are (40.77, 31.69). Object E is located S of object D at a distance of 6.03 units. What is the direction and distance of object E relative to object A? A: Direction: E; Distance: 11.31 units B: Direction: SW; Distance: 6.61 units C: Direction: NW; Distance: 6.53 units D: Direction: S; Distance: 8.42 units A:D Q:The coordinates of object A are (40.77, 31.69). Object F is located NW of object E at a distance of 6.77 units. What is the direction and distance of object F relative to object A? A: Direction: N; Distance: 9.47 units B: Direction: NE; Distance: 8.71 units C: Direction: NW; Distance: 4.89 units D: Direction: SW; Distance: 7.68 units A:D Q:The coordinates of object A are (40.77, 31.69). Object G is located SW of object F at a distance of 8.10 units. What is the direction and distance of object G relative

to object A? A: Direction: W; Distance: 11.24 units B: Direction: SW; Distance: 15.57 units C: Direction: NE; Distance: 12.62 units D: Direction: SE; Distance: 15.37 units A:B Q:The coordinates of object A are (40.77, 31.69). Object H is located NW of object G at a distance of 4.96 units. What is the direction and distance of object H relative to object A? A: Direction: N; Distance: 11.81 units B: Direction: E; Distance: 17.73 units C: Direction: W; Distance: 17.08 units D: Direction: NE; Distance: 20.00 units A:C Q:The coordinates of object A are (40.77, 31.69). Object I is located NE of object H at a distance of 2.43 units. What is the direction and distance of object I relative to object A? A: Direction: S; Distance: 17.98 units B: Direction: W; Distance: 14.94 units C: Direction: E; Distance: 16.63 units D: Direction: NW; Distance: 14.81 units A:B Q:The coordinates of object A are (40.77, 31.69). Object J is located S of object I at a distance of 7.81 units. What is the direction and distance of object J relative to object A? A: Direction: SW; Distance: 18.56 units B: Direction: S; Distance: 14.85 units C: Direction: N; Distance: 20.61 units D: Direction: W; Distance: 21.46 units A:A Q:The coordinates of object A are (40.77, 31.69). Object K is located SW of object J at a distance of 2.59 units. What is the direction and distance of object K relative to object A? A: Direction: W; Distance: 23.38 units B: Direction: S; Distance: 15.30 units C: Direction: E; Distance: 15.77 units D: Direction: SW; Distance: 21.14 units A:D

coordinates_movement_multiturn(CMM):

This task is a two-dimensional "path-integration" coordinate-updating problem, reminiscent of the classic account of animal path integration by Mittelstaedt et al. (1980).

Within a 9×9 integer grid, the system successively issues movement commands expressed in clockwise bearings with magnitudes (e.g., 73°, 146°). After each command, the participant must instantly convert the polar displacement into Cartesian offsets, add them to the current floating-point position, and report the object's new coordinates. The system verifies the answer before providing the next instruction, and accuracy is assessed only after the final move.

The task primarily tests the subject's real-time vector decomposition and planar-geometry computation skills, as well as the ability to keep cumulative error to a minimum over multiple dialogue turns.

Conversation Examples:

coordinates_movement_multiturn:

Q:In a 9x9 grid (coordinates ranging from 0 to 8), an object starts at (1,5). Angles are measured clockwise from 0°, where 0° indicates right, 90° down, 180° left, and 270° up. The object moves 0.9 units in the direction of 73°. What is the object's new position? A. (1,7) B. (0,7) C. (1,6) D. (1,8)

A:C

Q:After moving 0.9 units in the direction of 73°, the object continues to move 3.7 units in the direction of 146°. What is the object's new position? A. (1,7) B. (0,7) C. (1,8) D. (0,8)

A:D

Q:After moving 3.7 units in the direction of 146°, the object continues to move 2.3 units in the direction of 343°. What is the object's new position? A. (4,7) B. (3,6) C. (2,6) D. (2,7)

A:D

Q:After moving 2.3 units in the direction of 343°, the object continues to move 3.9 units in the direction of 217°. What is the object's new position? A. (0,4) B. (1,6) C. (0,3) D. (0,5)

A:D

Q:After moving 3.9 units in the direction of 217°, the object continues to move 1.3 units in the direction of 328°. What is the object's new position? A. (2,5) B. (0,3) C. (1,5) D. (1,4)

A:D

Q:After moving 1.3 units in the direction of 328°, the object continues to move 1.0 unit in the direction of 341°. What is the object's new position? A. (2,6) B. (2,3) C. (2,4) D. (1,3) A:C Q:After moving 1.0 unit in the direction of 341°, the object continues to move 0.8 units in the direction of 1°. What is the object's new position? A. (4,3) B. (3,3) C. (3,4) D. (4,4)

A:C

Q:After moving 0.8 units in the direction of 1°, the object continues to move 4.6 units in the direction of 285°. What is the object's new position? A. (4,2) B. (2,0) C. (6,0) D. (4,0)

A:D

Q:After moving 4.6 units in the direction of 285°, the object continues to move 2.0 units in the direction of 248°. What is the object's new position? A. (3,1) B. (5,0) C. (1,0) D. (3,0)

A:D

Q:After moving 2.0 units in the direction of 248°, the object continues to move 4.1 units in

the direction of 351°. What is the object's new position? A. (7,0) B. (5,0) C. (7,1) D. (6,0)
A:A

mental_rotation_multiturn (MRM):
This task is a continuous, interactive three-dimensional mental-rotation assessment:The system sequentially issues counter-clockwise rotation angles about arbitrary unit vectors.After each command, the participant must compute the new, compounded forward vector based on the previous orientation and select a single answer from options A, B, C, or D.The system confirms the response, then delivers the next rotation. After the final step, overall accuracy is evaluated.
The test traces its theoretical roots to the mental-rotation experiment introduced by Shepard et al. in 1971. Modern computational implementations generally draw on Zacks' (2008) review of the cognitive mechanisms underlying rotation, ensuring that the task's algorithmic and cognitive demands remain aligned.

Conversation Examples:
mental_rotation_multiturn:
Q:A cube's initial forward vector is (1.000, 0.000, 0.000). It is rotated counterclockwise 205° around the axis (-0.943, -0.307, 0.128). Which of the following is the forward vector after rotation? Please select and respond with the letter only: A: (0.790, 0.498, -0.359); B: (0.553, 0.650, 0.520); C: (0.103, 0.010, 0.995); D: (-0.119, 0.352, 0.928).
A:A
Q:After being rotated counterclockwise 205° around the axis (-0.943, -0.307, 0.128), the cube continues to rotate counterclockwise 286° around the axis (0.105, -0.718, -0.688). Which of the following is the forward vector after rotation? Please select and respond with the letter only: A: (-0.362, 0.638, -0.680); B: (0.896, 0.331, 0.297); C: (0.501, -0.828, -0.251); D: (-0.849, -0.517, 0.109).
A:A
Q:After being rotated counterclockwise 286° around the axis (0.105, -0.718, -0.688), the cube continues to rotate counterclockwise 240° around the axis (0.733, -0.433, -0.525). Which of the following is the forward vector after rotation? Please select and respond with the letter only: A: (-0.316, 0.448, 0.836); B: (-0.667, 0.278, 0.691); C: (-0.603, -0.283, 0.746); D: (-0.567, -0.795, 0.217).
A:D
Q:After being rotated counterclockwise 240° around the axis (0.733, -0.433, -0.525), the cube continues to rotate counterclockwise 265° around the axis (-0.070, -0.834, -0.548). Which of the following is the forward vector after rotation? Please select and respond with the letter only: A: (-0.307, -0.594, 0.744); B: (0.919, 0.332, 0.212); C: (0.844, -0.155, 0.513); D: (0.620, -0.784, 0.049).
A:D
Q:After being rotated counterclockwise 265° around the axis (-0.070, -0.834, -0.548), the cube continues to rotate counterclockwise 52° around the axis (0.632, -0.688, 0.355). Which of the following is the forward vector after rotation? Please select and respond with the letter only: A: (0.823, -0.364, 0.437); B: (0.148, -0.295, 0.944); C: (0.805, -0.585, 0.105); D: (0.179, 0.324, 0.929).
A:C
Q:After being rotated counterclockwise 52° around the axis (0.632, -0.688, 0.355), the cube continues to rotate counterclockwise 114° around the axis (0.658, 0.633, 0.408). Which of the following is the forward vector after rotation? Please select and respond with the letter only: A: (0.139, 0.655, -0.744); B: (-0.754, 0.432, 0.494); C: (-0.836, -0.303, 0.457); D: (0.028, -0.370, 0.929).
A:A
Q:After being rotated counterclockwise 114° around the axis (0.658, 0.633, 0.408), the cube continues to rotate counterclockwise 201° around the axis (-0.474, 0.202, -0.857). Which of the following is the forward vector after rotation? Please select and respond with the letter only: A: (-0.125, -0.195, 0.973); B: (-0.922, -0.168, -0.351); C: (0.878, -0.465, -0.110); D: (-0.844, -0.524, 0.117).
A:B
Q:After being rotated counterclockwise 201° around the axis (-0.474, 0.202, -0.857), the cube continues to rotate counterclockwise 125° around the axis (-0.117, -0.936, 0.331). Which of the following is the forward vector after rotation? Please select and respond with the letter only: A: (-0.734, 0.085, -0.674); B: (0.816, -0.407, -0.412); C: (0.182, -0.875, -0.449); D: (-0.897, 0.294, -0.330).
A:B
Q:After being rotated counterclockwise 125° around the axis (-0.117, -0.936, 0.331), the cube continues to rotate counterclockwise 233° around the axis (0.254, 0.947, -0.195). Which of the following is the forward vector after rotation? Please select and respond with the letter only: A:

(0.941, -0.224, -0.254); B: (0.140, 0.718, 0.682); C: (-0.365, -0.756, -0.544); D: (-0.156, 0.140, 0.978).
A:D
Q:After being rotated counterclockwise 233° around the axis (0.254, 0.947, -0.195), the cube continues to rotate counterclockwise 274° around the axis (-0.951, 0.174, 0.256). Which of the following is the forward vector after rotation? Please select and respond with the letter only: A: (-0.890, -0.361, -0.279); B: (0.855, -0.344, 0.388); C: (0.995, 0.059, -0.084); D: (-0.519, -0.810, 0.274).
A:D

B.0.3 Tier 4(Manipulation Tasks):

single_object_free_obstacle_path_planning (FOP):
This task is a simplified single-agent grid-navigation benchmark. On obstacle-free discrete grids of four sizes—3 × 3, 5 × 5, 7 × 7, and 9 × 9—the participant must, through multi-turn interaction, sequentially output one of the eight compass directions, guiding the lone object from its given start cell to the target cell. Evaluation focuses on (1) whether the produced path coincides with the shortest path generated by algorithms such as A* or Dijkstra and (2) the model's decision efficiency and coherence during the interaction. The task's theoretical foundation traces back to the A* algorithm proposed by Hart, Nilsson, and Raphael in 1968, which first systematized the use of heuristic search for shortest-path planning on regular grids and established the methodological framework for subsequent grid-navigation tasks.

Conversation Examples:
free_obstacle_path_planning_3_multiturn:
Q:Task: 3×3 grid, move from (1, 2) to (0, 0). Output step by step.
A:NW
Q:Step 1 executed. Continue.
.....
Q:Step n executed. Continue.
A:NONE.Action sequence: NW, W.

free_obstacle_path_planning_5_multiturn:
Q:Task: 5×5 grid, move from (4, 2) to (3, 0). Output step by step.
A:NW
Q:Step 1 executed. Continue.
.....
Q:Step n executed. Continue.
A:NONE.Action sequence: NW, W.

free_obstacle_path_planning_7_multiturn:
Q:Task: 7×7 grid, move from (5, 5) to (2, 3). Output step by step.
A:NW
Q:Step 1 executed. Continue.
.....
Q:Step n executed. Continue.
A:NONE.Action sequence: NW, NW, N

single_free_obstacle_path_planning_9_multiturn:
Q:Task: 9×9 grid, move from (7, 6) to (2, 1). Output step by step.
A:NW
Q:Step 1 executed. Continue.
.....
Q:Step n executed. Continue.
A:NONE.Action sequence: NW, NW, NW, NW, NW

obstacle_path_planning_multiturn:
This task evaluates a model's ability to perform static-obstacle avoidance and shortest-path reasoning in a regular grid environment. It is one of the most classic spatial-reasoning benchmarks and appears widely in robot navigation, reinforcement learning, and, more recently, in planning tests for large language models (Huang et al., 2022). In our setting, the participant must strictly follow the sequence observe → orient → decide → act on maps of four sizes—3 × 3, 5 × 5, 7 × 7, and 9 × 9—step-by-step navigating the object to its destination and, finally, outputting the entire path sequence.

Conversation Examples:
single_obstacle_path_planning_3_multiturn:
Q:Task: 3×3 grid, move from (1, 1) to (2, 0). Obstacles: [(0, 1), (0, 2)]. Output step by step.
A:SW
Q:Step 1 executed. Continue.
A:NONE.Action sequence: SW

obstacle_path_planning_5_multiturn:
Q:Task: 5×5 grid, move from (0, 2) to (0, 3). Obstacles: [(1, 0), (1, 3), (2, 0), (2, 2), (2, 3), (3, 0), (3, 1), (3, 3), (4, 3), (4, 4)]. Output step by step.
A:E.
Q:Step 1 executed. Continue.
A:NONE.Action sequence: E

obstacle_path_planning_7_multiturn:
Q:Task: 7×7 grid, move from (3, 6) to (4, 5). Obstacles: [(0, 1), (0, 2), (0, 5), (0, 6), (1, 1), (1, 2), (1, 6), (2, 0), (2, 1), (2, 2), (2, 4), (2, 5), (3, 3), (4, 0), (4, 3), (4, 4), (5, 0), (5, 3), (5, 4), (5, 5), (5, 6), (6, 3)]. Output step by step.
A:SW.
Q:Step 1 executed. Continue.
A:NONE.Action sequence: SW

obstacle_path_planning_9_multiturn:
Q:Task: 9×9 grid, move from (1, 0) to (1, 1). Obstacles: [(0, 3), (0, 4), (0, 5), (0, 7), (1, 2), (1, 4), (1, 5), (1, 8), (2, 2), (2, 4), (2, 5), (2, 6), (2, 7), (3, 1), (3, 2), (3, 8), (4, 1), (4, 2), (4, 4), (4, 5), (4, 6), (4, 7), (5, 0), (5, 1), (5, 2), (5, 3), (5, 6), (5, 7), (5, 8), (6, 0), (6, 1), (6, 4), (6, 8), (7, 0), (7, 3), (7, 4), (7, 5), (8, 1), (8, 3), (8, 5), (8, 6)]. Output step-by-step.
A:E.
Q:Step 1 executed. Continue.
A:NONE.Action sequence: E

Multi-object_control_path_planning_multiturn (obstacle-free version):
This task is a "synchronous multi-goal path-planning task." On four obstacle-free grid maps—3 × 3, 5 × 5, 7 × 7, and 9 × 9—nine objects (Agents 1–9) must move in lock-step discrete time from their individual start cells to their designated goal cells while continuously avoiding inter-agent collisions. By eliminating static obstacles, the task isolates the challenges of multi-agent coordination and conflict resolution, serving as a direct counterpart to the obstacle-inclusive version. It therefore enables a systematic assessment of large language models' generalization and reasoning abilities under different spatial-constraint conditions.

Conversation Examples:
Multi-object_control_path_planning_3_multiturn:
Q:Task: Synchronously control 3 objects.\nStart/End points:\n1 (0, 2)→(0, 1)\n2 (2, 1)→(2, 0)\n3 (1, 2)→(1, 1)\nObstacles: []. Please proceed step-by-step.
A:1:W 2:W 3:W
Q:Step 1 executed. Continue.
.....
Q:Step n executed. Continue.
A:NONE.Total steps: n

Multi-object_control_path_planning_5_multiturn:
Q:Task: Synchronously control 5 objects.\nStart/End points:\n1 (2, 1)→(4, 2)\n2 (2, 0)→(0, 1)\n3 (1, 4)→(2, 2)\n4 (2, 3)→(4, 4)\n5 (3, 2)→(1, 3)\nObstacles: []. Please proceed step-by-step.
A:1:SE 2:N 3:SW 4:SE 5:N.
Q:Step 1 executed. Continue.
.....
Q:Step n executed. Continue.
A:NONE.Total steps: n

Multi-object_control_path_planning_7_multiturn:
Q:Task: Synchronously control 7 objects.\nStart and end points:\n1 (6,4)→(3,3)\n2 (5,5)→(4,3)\n3 (4,5)→(1,2)\n4 (5,0)→(3,0)\n5 (1,3)→(0,2)\n6 (3,6)→(0,1)\n7 (5,1)→(4,6)\nObstacles: []. Please proceed step by step.
A:1:N 2:W 3:NW 4:N 5:NW 6:SW 7:NE.
Q:Step 1 executed. Continue.
.....
Q:Step n executed. Continue.
A:NONE.Total steps: n

Multi-object_control_path_planning_9_multiturn:
Q:Task: Synchronously control 9 objects.\nStart and end points:\n1 (5, 2)→(6, 5)\n2 (1, 1)→(4, 3)\n3 (6, 1)→(4, 5)\n4 (1, 8)→(6, 3)\n5 (3, 0)→(2, 0)\n6 (2, 2)→(5, 4)\n7 (5, 3)→(3, 3)\n8 (3, 5)→(1, 7)\n9 (1, 5)→(4, 7)\nObstacles: []. Proceed step by step.
A:1:NE 2:SE 3:NE 4:SW 5:N 6:SE 7:N 8:NE 9:SE.
Q:Step 1 executed. Continue.
.....
Q:Step n executed. Continue.
A:NONE.Total steps: n

Multi-object_control_obstacle_path_planning multiturn:
This task requires the participant (or model) to control multiple objects simultaneously in a discrete grid world of five sizes—3 × 3, 5 × 5, 7 × 7, 8 × 8, and 9 × 9. All objects must start synchronously from their assigned origins, detour around static obstacles, and eventually reach their respective goals while avoiding inter-object collisions throughout the entire process. The task design follows the well-studied Multi-Agent Path Finding (MAPF) problem in robotics (Standley, 2010; Stern et al., 2019). By means of this benchmark, we can systematically evaluate

large language models' abilities in multi-agent coordination, concurrent action generation, and planning reasoning based on structured spatial representations within a controlled spatial setting.

Conversation Examples:
Multi-object_control_obstacle_path_planning_3 multiturn:
Q:Task: Synchronously control 3 objects.\nStart and end points:\n1 (1, 0)→(2, 0)\n2 (0, 2)→(1, 1)\n3 (0, 1)→(0, 0)\nObstacles: [(2, 2)]. Please proceed step by step.
A:1:S 2:SW 3:W
Q:Step 1 executed. Continue.
.....
Q:Step n executed. Continue.
A:NONE.Total steps: n

Multi-object_control_obstacle_path_planning_5 multiturn:
Q:Task: Synchronously control 5 objects.\nStart and end points:\n1 (3, 2)→(0, 0)\n2 (4, 1)→(4, 0)\n3 (0, 1)→(3, 0)\n4 (1, 1)→(2, 1)\n5 (0, 4)→(2, 3)\nObstacles: [(1, 0), (1, 2), (1, 4), (2, 2), (3, 1), (4, 2), (4, 4)]. Proceed step by step.
A:1:NW | 2:W | 3:S | 4:S | 5:SW \n 1:NW 2:W 3:S 4:S 5:SW
Q:Step 1 executed. Continue.
.....
Q:Step n executed. Continue.
A:NONE.Total steps: n

Multi-object_control_obstacle_path_planning_7 multiturn: Q:Task: Synchronously control 7 objects.\nStart and end points:\n1 (4, 2)→(0, 0)\n2 (1, 0)→(6, 3)\n3 (0, 5)→(3, 5)\n4 (1, 6)→(2, 4)\n5 (1, 5)→(5, 2)\n6 (6, 4)→(0, 3)\n7 (2, 1)→(3, 4)\nObstacles: [(0, 2), (0, 4), (1, 1), (1, 4), (2, 3), (3, 3), (4, 0), (4, 3), (4, 4), (4, 5), (5, 0), (5, 5), (5, 6), (6, 1), (6, 2), (6, 5), (6, 6)]. Please proceed step by step. A:1:N | 2:SE | 3:SE | 4:SW | 5:SW | 6:NW | 7:NE \n 1:N 2:SE 3:SE 4:SW 5:SW 6:NW 7:NE
Q:Step 1 executed. Continue.
.....
Q:Step n executed. Continue.
A:NONE.Total steps: n

Multi-object_control_obstacle_path_planning_8 multiturn: Q:Task: Synchronously control 16 objects.\nStart and end points:\n1 (3, 2)→(1, 6)\n2 (0, 1)→(6, 2)\n3 (1, 3)→(3, 3)\n4 (1, 5)→(0, 0)\n5 (5, 5)→(0, 6)\n6 (5, 3)→(2, 4)\n7 (6, 5)→(5, 7)\n8 (1, 0)→(1, 1)\n9 (5, 6)→(4, 5)\n10 (4, 4)→(7, 6)\n11 (3, 1)→(5, 2)\n12 (6, 3)→(5, 4)\n13 (4, 1)→(7, 1)\n14 (0, 2)→(6, 6)\n15 (5, 0)→(5, 1)\n16 (0, 7)→(0, 4)\nObstacles: [(0, 5), (1, 2), (2, 3), (2, 7), (3, 5), (3, 6), (4, 6), (6, 1), (6, 4), (7, 3), (7, 5), (7, 7)]. Proceed step by step.
A:1:E 2:S 3:SE 4:SW 5:N 6:N 7:NE 8:E 9:NW 10:SE 11:SE 12:NE 13:SE 14:SE 15:E 16:SW
Q:Step 1 executed. Continue.
.....
Q:Step n executed. Continue.
A:NONE.Total steps: n

Multi-object_control_obstacle_path_planning_9 multiturn: Q:Task: Synchronously control 9 objects.\nStart and end points:\n1 (5, 7)→(7, 6)\n2 (2, 6)→(2, 7)\n3 (0, 4)→(0, 8)\n4 (0, 0)→(3, 5)\n5 (4, 0)→(8, 6)\n6 (2, 2)→(6, 3)\n7 (6, 4)→(6, 0)\n8 (3, 2)→(6, 5)\n9 (3, 0)→(7, 7)\nObstacles: [(0, 1), (0, 2), (0, 3), (0, 6), (0, 7), (1, 1), (1, 2), (1, 4), (1, 6), (1, 7), (2, 5), (2, 8), (3, 1), (3, 3), (3, 4), (3, 6), (3, 7), (4, 3), (4, 5), (4, 6), (4, 7), (5, 0), (5, 3), (5, 4), (5, 8), (6, 1), (6, 2), (6, 6), (6, 7), (6, 8), (7, 0), (7, 1), (7, 2), (7, 3), (7, 5), (8, 0), (8, 1), (8, 3), (8, 4), (8, 5), (8, 7)]. Proceed step by step.
A:1:W 2:E 3:SE 4:S 5:E 6:S 7:W 8:S 9:SE
Q:Step 1 executed. Continue.
.....
Q:Step n executed. Continue.
A:NONE.Total steps: n

B.1 Model version

Table 6: Latest public release (before 2025-06) of each model used in our experiments.

Model	Version / Tag
GPT-5	2025-08-07
GPT-5 nano	2025-08-07
o3	2025-04-16
o4-mini	2025-04-16
DeepSeek-v3	2025-03-24
DeepSeek-R1	2025-05-28
Claude Opus 4.5	2025-11-24
Claude-3-7-thinking	2025-02-19
Grok 4.1	2025-11-17
Qwen2.5-14B-Instruct	2024-09-18
Qwen3-4B	2025-04-28
Qwen3-14B	2025-04-28

C Detail of Chess Combat

C.0.1 Error of Chess Combat

Our Chess Combat test, which pits the model against the powerful Stockfish engine, reveals key weaknesses in how standard LLMs handle complex spatial control. The models' mistakes generally fall into two clear categories: a poor awareness of the board state and a failure to correctly apply game rules. First, the models struggled to maintain an accurate mental map of where the pieces were on the board. This led to "hallucinated" moves based on an incorrect board state. For example, a model might correctly move its knight from g1 to f3 on its first turn. However, several turns later, it might try to move the same knight from g1 again, as if it had completely forgotten the piece had moved. Another common perception error was attempting to move a piece to a square already occupied by one of its own pieces, resulting in an invalid action. Second, the models frequently failed to apply the basic movement rules of chess. They would often generate plans that were physically impossible for a given piece. For instance, a model might try to move a bishop horizontally across the board (e.g., from f1 to c1), a move that is only legal for a rook. They also struggled with more nuanced rules. A frequent error involved pawn movement, where a model would attempt to capture an opponent's piece by moving one square directly forward, instead of capturing diagonally.

C.0.2 Task Setup

To evaluate the model's generalization ability in a complex, multi-step spatial control task, we designed a chess combat test. In this setup, the model plays against Stockfish, a powerful, open-source chess engine widely regarded as a benchmark for chess-playing strength. The experiment is designed not to test grandmaster-level strategy, but rather the model's ability to maintain an accurate internal representation of the board state and consistently adhere to the game's complex rules over a multi-turn interaction.

The game proceeds in turns. For each of its turns, the LLM receives the current board state and is prompted to generate a move. The model is instructed to output its move into the Universal Chess Interface (UCI) protocol, which Stockfish uses for communication. The chess notation used by the Stockfish engine is a coordinate notation based on the Universal Chess Interface (UCI) protocol, known as Long Algebraic Notation (LAN). A game is considered successful if it proceeds until a standard conclusion (e.g., checkmate or stalemate) without the model committing any rule violations. A game is terminated and recorded as a failure if the model generates an illegal move, such as attempting to move a piece from an incorrect square or violating a piece's movement rules, as detailed in Section C.0.2. Here is one example of Chess_Combat_with_Stockfish:

position startpos moves e2e4 e7e5 g1f3 b8c6 f1b5 a7a6 b5a4 g8f6 e1g1 f8e7 f1e1 a6b5 a4b3 d7d6 c2c3 e8g8 h2h3 c6a5 b3c2 c7c5 d2d4 d8c7 b1d2 c5d4 c3d4 c8d7 a2a4 b5a4 a4a5 d7d7 b2b4 a5c6 e5e9

In this practical example, the UCI engine will reject or ignore an illegal move like e5e9(llm), so the program needs to detect it and declare the game ended due to an LLM violation. It does not count as the LLM having completed the game.

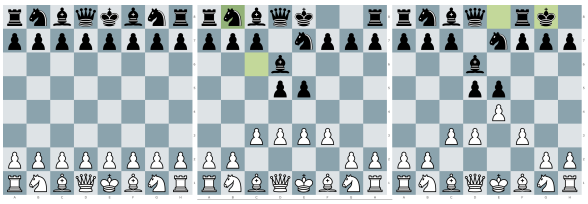

Figure 6: The Chess Combat evaluation setup visualization. The LLM (white) processes the board state to generate a move in standard notation, which is then played against the Stockfish engine (black).

D Detail of Robot Control

D.0.1 Task Setup

The experimental environment consisted of a 6-DoF robotic arm operating over grid-based mazes of varying sizes (3x3, 5x5, 7x7, and 9x9) and a local server equipped with two NVIDIA RTX 4090 GPUs . The maze contained a single target object, one or more static obstacles, and a designated target location. The task required the robotic arm to pick up a target object and transport it to a designated location, navigating around static obstacles while adhering to a strict set of movement rules.

D.0.2 Control Flow

We designed a text-only control system that operates as an iterative, closed-loop process. The workflow for each decision cycle is as follows:

1. **State and Rule Encoding:** At the beginning of each decision cycle, the system receives

Trial ID	Obstacles	Model	Result	Path Length	Avg. Step Latency (s)
1	2	Qwen-4B	Failure (Collision)	-	7.1
2	2	SODA-4B	Success	6	8.8
3	3	Qwen-4B	Success	8	7.5
4	3	SODA-4B	Success	7	9.0
5	4	Qwen-4B	Failure (Collision)	-	7.4
6	4	SODA-4B	Success	7	9.3
7	5	Qwen-4B	Failure (Invalid Move)	-	7.6
8	5	SODA-4B	Success	8	9.5

Table 7: Parts of Detailed Trial Log for 5x5 Robotic Arm Task

Model	Success Rate (%)	Avg. Path Length (steps)	Avg. Step Latency (s)
Qwen-4B	35	6.4	7.5
SODA-4B	90	7.6	9.1

Table 8: Aggregated Performance on the 5x5 Robotic Arm Task

the current world state including object, target, and obstacles location. This world state information is then encoded into a structured textual format. This state description is combined with a set of predefined rules and constraints to form a comprehensive prompt for the LLM. For example:

State: Arm at (pointA). Object at (pointB). Obstacle at (pointC). Target at (pointD). Task: Move object to target, avoid obstacle. Rule: There are three operations to choose: Move, Pickup and drop. Move only one grid horizontally or vertically each step.

2. **LLM-based Planning:** The LLM processes this prompt to determine and generate the single most appropriate action for the current step. The model's output is a discrete, low-level command consisting of an action type and its corresponding coordinates. For example:

Plan: 1. move(point1) 2. pick(point2) 3. move(point3) 4. drop(point4)

rated command is parsed and translated into motor commands for the robotic arm. After the action is physically executed, the entire cycle repeats from the first step: the system gets the new world state, generates a new prompt, and queries the LLM for the next action. This iterative process continues until the model issues a final command that completes the task.

D.0.3 Error of Robot Control

The llm-based robotic arm task revealed a significant gap between the baseline model's abstract reasoning and the requirements of physical embodiment. The failures of the untrained model were not random but systematic, highlighting fundamental deficits in its ability to ground symbolic plans in a constrained physical environment. The errors observed fall into three categories: firstly, The model frequently generated plans that resulted in direct collisions. For instance, given an obstacle at [3,3], the model would still output move [3,3] as part of a path. This indicates a critical breakdown in the Orient phase, where the model failed to incorporate the explicit obstacle locations from the prompt into its internal world model, treating the task as an unconstrained pathfinding problem. Secondly, The model struggled to adhere to the kinematic rules defined in the prompt. Despite being explicitly forbidden, it often generated kinematically invalid actions, such as diagonal moves (e.g., move [1,1] from [0,0]) or multi-cell jumps (e.g., move [0,2] from [0,0]). This suggests its internal action generation mechanism defaults to generalized notions of movement rather than the specific, constrained action space of the task. Thirdly, The model exhibited a poor ability to track its own state within the task sequence. This often led to a collapse of the planning process, where the model would get "stuck" in a repetitive loop (e.g., repeatedly outputting move [1,2] followed by move [1,3]). This behavior demonstrates a failure to form a coherent, long-term plan and an inability to recognize that its current strategy was making no progress toward

the final goal.

D.0.4 Experimental Results

To quantify the advantages of the SODA framework in a physical embodiment task, we conducted a series of simulated experiments on the 6-DoF robotic arm. The experimental environment consisted of a UR5e robotic arm operating over grid-based mazes of varying sizes (5x5) and a local server equipped with an NVIDIA RTX 5090 GPU (32GB VRAM). The maze contained a single target object, one or more static obstacles, and a designated target location. The task required the robotic arm to pick up a target object and transport it to a designated location, navigating around static obstacles while adhering to a strict set of movement rules.

We compared our SODA-4B model against the baseline Qwen-4B model across 50 trials of varying complexity. Detailed records for a subset of these trials are shown in Table 7, and the aggregated results are presented in Table 8.

The results demonstrate a substantial performance improvement in the SODA-4B model. With a 90% success rate, SODA-4B significantly outperformed the baseline Qwen-4B (35%). The baseline model's failures were consistent with the error types previously described: collisions with obstacles, generation of kinematically invalid moves, and getting trapped in repetitive action loops. The high success rate of SODA-4B indicates that its enhanced spatial reasoning, derived from the OODA framework, effectively mitigates these issues in a constrained physical environment.

Furthermore, SODA-4B demonstrated greater planning efficiency. In successful trials, it completed the task with an average path length of 7.8 steps, compared to the baseline's 6.4 steps. This suggests that SODA prefer to succeed since 8 steps means final goal-arriving in grid(5x5).

Interestingly, the enhanced reliability and efficiency of SODA-4B came with a slight increase in decision latency, averaging 9.1 seconds per step versus Qwen-4B's 7.5 seconds. This modest trade-off is attributable to the model's more deliberate OODA-based reasoning process, which involves deeper analysis during the Observe and Orient phases before acting. The significant increase in task success and planning efficiency justifies this marginal increase in processing time.

In a nutshell, these results validate that the SODA framework successfully bridges the gap between abstract reasoning and embodied intelligence, enabling the model to generate robust, efficient, and physically-grounded plans for complex robotic control tasks.