Technical Appendices for Submission 20260: "Reinforcing Spatial Reasoning in Vision-Language Models with Interwoven Thinking and Visual Drawing"

Anonymous Author(s)
Affiliation
Address
email

A Comparison against proprietary LVLMs

In addition to the open-source LVLMs and representative multimodal reasoning methods examined in our main results, we also evaluated leading proprietary LVLMs, including GPT-4o [1] and Gemini-1.5-Flash [3][1]. Our evaluation spans two settings: direct answer generation and reasoning-based response generation (i.e., first generating a textual reasoning chain and then concluding the final answer).

The results, as showcased in Table 1, clearly underscore the competitive advantage of proprietary models, largely attributed to their large-scale architecture. Nevertheless, on these challenging benchmarks, our models not only are comparable with these industry leaders but occasionally outperform them (e.g., on MAZE, SpatialEval, VSIBench).

B Visualization results

To illustrate the effectiveness of our approach, we present two representative examples in Figure 1 and 2 from the MAZE and EmbSpatial benchmark, respectively.

In the maze navigation task (Figure 1), Qwen2.5-VL-7B provides incorrect answers directly, and GPT-4o attempts textual reasoning but fails to accurately track spatial transitions in the complex action sequence. In contrast, SPARK successfully decomposes the task into interpretable steps, using the "Path Tracer" tool to visualize and verify each movement through auxiliary lines. This visual tracking approach ensures accurate navigation through the maze, leading to the correct destination.

Similarly, for real-world spatial relationship understanding (Figure 2), SPARK demonstrates a systematic approach by first using "Object Mapper" to establish precise object locations by annotating bounding boxes, followed by "Path Tracer" to explicitly verify relative positions. This two-step visual reasoning process helps avoid ambiguity in spatial relationships, particularly in determining left-right relationships from the viewer's perspective. While GPT-4o reaches the correct conclusion through textual reasoning, our visual approach provides more explicit and verifiable reasoning steps.

These cases illustrate how drawing operations enable more reliable spatial reasoning by grounding abstract relationships in concrete visual representations.

C Complexity analysis

To thoroughly assess the computational efficiency of SPARK, we present a comprehensive analysis comparing our approach with the base model Qwen2.5-VL-7B model. This analysis encompasses

[1] Due to the high cost, we exclude OpenAI o3 [2] from large-scale evaluation ($\sim$ \$0.5 per example).

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Table 1: Performance comparison across spatial reasoning benchmarks. Gray-shaded rows represent large-sized models (>7B parameters). For non-shaded rows, **bold** and underlined numbers indicate the best and second-best results. *Italic* numbers in gray-shaded rows indicate performance below non-shaded best results. Stages 1-3 refer to cold-start training, reflective rejection sampling, and reinforcement learning. † Results from [4].

Method	Tool	Reasoning	Image			Video	Mutli-view
			MAZE	SpatialEval	EmbSpatial	VSIbench	SPAR-Bench
Proprietary LVLMs							
GPT-4o	✗	✗	*48.8*	*58.3*	*60.7*	*35.8*†	*33.6*
GPT-4o	✗	✓	*63.1*	67.8	74.2	-	38.1
Gemini-1.5-Flash	✗	✗	*32.4*	*59.2*	66.4	*42.1*†	*34.9*
Gemini-1.5-Flash	✗	✓	*58.0*	*61.5*	67.9	-	35.6
Ours							
SPARK	✓	✓	**89.0**	63.4	64.9	**42.9**	34.9
SPARK w/o Stage 3	✓	✓	85.6	63.1	64.5	38.9	34.3
SPARK w/o Stage 2	✓	✓	87.1	63.9	**65.7**	39.8	**35.1**
SPARK w/o Stage 2&3	✓	✓	85.4	**64.1**	64.3	37.1	34.6
SPARK w/o Stage 1&2&3	✓	✓	28.1	53.5	55.8	17.7	23.3

Table 2: Complexity analysis of different reasoning approaches.

Model Type	Input Length	Output Length	Time Complexity
Base Model	M	N	$\mathcal{O}(M \cdot N)$
Standard Reasoning	M	$L + N$	$\mathcal{O}(M \cdot (L + N))$
SPARK (Ours)	$M + \sum_{i=1}^{S} M_i$	$S \cdot L + N$	$\mathcal{O}\big(S^2 \cdot L \cdot (L + M) + S \cdot (L + M)(L + N)\big)$

both theoretical complexity bounds and empirical resource utilization, providing insights into the framework's scalability and computational characteristics.

C.1 Theoretical time complexity

We analyze the computational complexity across three model variants: the base model (performing direct answer generation), standard reasoning models (generating reasoning chains followed by answer derivation), and our SPARK model (employing iterative visual drawing and thinking). Let M denote the input length (including both image and text tokens), N the answer length, L the per-step reasoning path length, and S the number of reasoning steps, where $S = 1$ for both base model and standard reasoning approaches. Empirically, SPARK achieves shorter per-step reasoning paths (smaller L) compared to single-step approaches due to its decomposition of reasoning into interpretable visual operations. Table 2 presents a comparative analysis of their time complexities.

The total computational complexity is:

$$T = \sum_{k=1}^{S-1} [(M + \sum_{i=1}^{k-1} (L_i + M_i)) \cdot L_k] + (M + \sum_{i=1}^{S-1} (L_i + M_i)) \cdot (L_S + N), \tag{1}$$

where L_k is the length of the reasoning path generated at step k, and M_k is the additional multimodal context incorporated at step k.

Assuming each reasoning step has approximately equal length, i.e., $L_i \approx L$, $M_i \approx M$ for all i, we can approximate:

$$T = \mathcal{O}\left(S^2 \cdot L \cdot (L + M) + S \cdot (L + M)(L + N)\right). \tag{2}$$

C.2 Practical runtime

To provide empirical evidence for our complexity analysis, we evaluate SPARK's runtime performance across different benchmarks. Table 3 illustrates the average runtime time and the average number of reasoning steps per sample.

Table 3: Average runtime and reasoning steps of SPARK across different benchmarks.

Benchmark	Avg. Runtime (s/sample)	Avg. Reasoning Steps
Maze	4.4	10.8
EmbSpatial	1.2	3.1
VSIbench	7.8	4.5

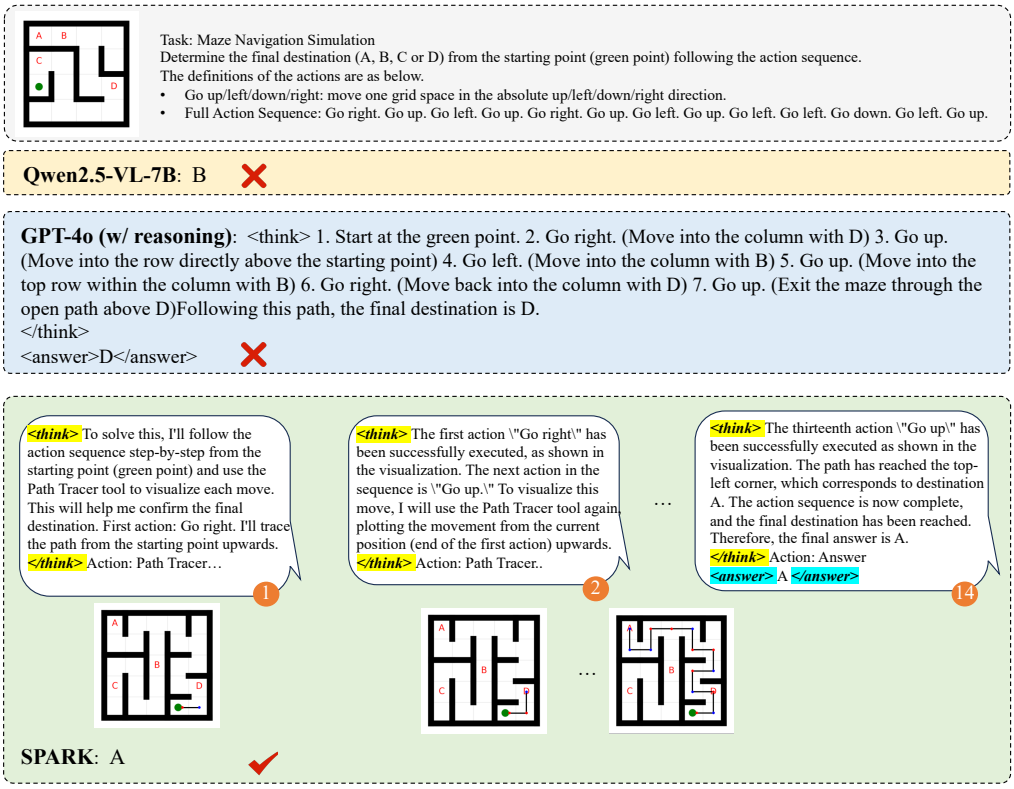

Figure 1: A visualization example of spatial reasoning approaches on the Maze benchmark, including Qwen-2.5-VL-7B, GPT-4o, and SPARK.

D Prompt templates

In this section, we present the comprehensive prompt templates utilized in SPARK, including both the system prompt and user prompt, as illustrated in Figure 3 and Figure 4, respectively. We use these prompt templates for both generating reasoning paths in cold start data and inference of SPARK.

D.1 System prompt of SPARK

The system prompt utilized in the SPARK reasoning framework is presented in Figure 3.

D.2 Query prompt template of SPARK

The query prompt utilized in the SPARK reasoning framework is presented in Figure 4. Our framework utilizes three distinct types of prompts throughout the reasoning process:

- **Initial Query Prompt** (Image/Video Query Prompt): Initiates the reasoning process by establishing the initial visual context and query.

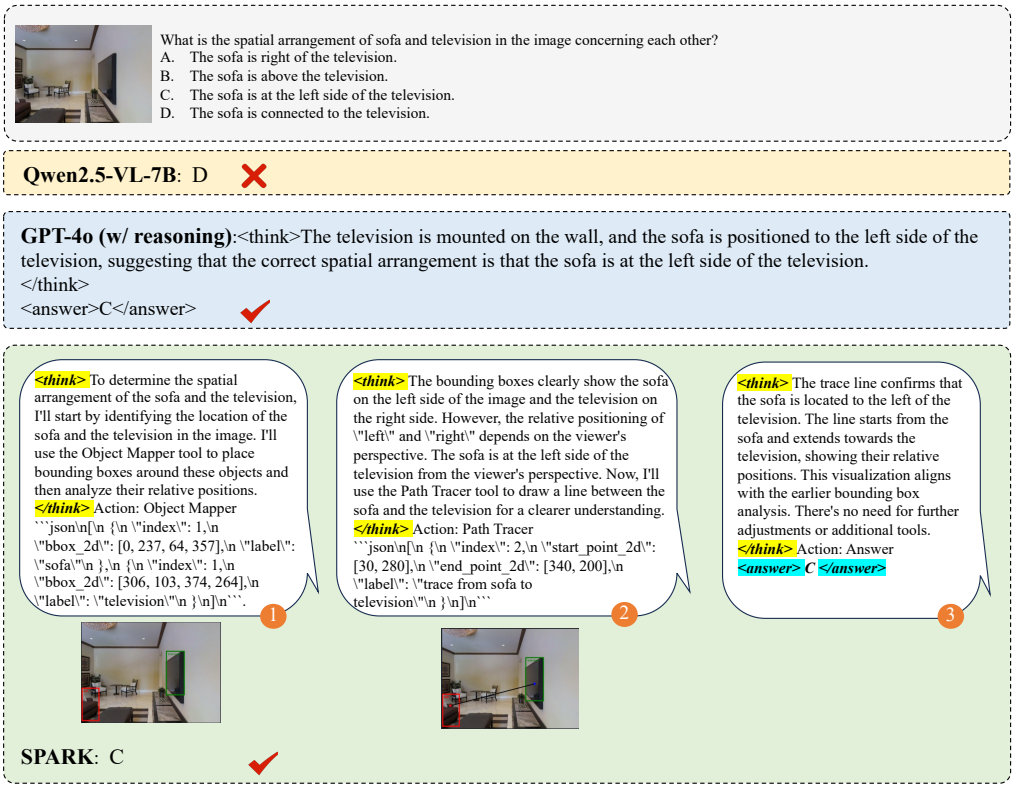

Figure 2: A visualization example of spatial reasoning approaches on the EmbSpatial benchmark, including Qwen-2.5-VL-7B, GPT-4o, and SPARK.

- **Follow-up Query Prompt**: Guides subsequent reasoning steps by incorporating intermediate results and accumulated context.
- **Final Answer Prompt**: Terminates the reasoning process and enforces answer generation when either the maximum number of processed images or reasoning steps is reached.

References

[1] OpenAI. Hello gpt-4o. In OpenAI Blog, 2024.

[2] OpenAI. Thinking with images, 2025.

[3] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530, 2024.

[4] Jihan Yang, Shusheng Yang, Anjali Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in Space: How Multimodal Large Language Models See, Remember and Recall Spaces. arXiv preprint arXiv:2412.14171, 2024.

System prompt

Guidance:
You are a spatial reasoning assistant with access to two powerful visualization tools.
Your task is to break down complex spatial problems and iteratively refine your solution through visualization feedback.

Available tools:
You can use the following two tools to visualize. After each tool usage, you must wait for and analyze the visualization feedback before proceeding.

1. **Object Mapper**
- Purpose: Identifies and maps key items in the space
- Input format: JSON
```json
[{{
"index": i, # Image index
"bbox_2d": [x1, y1, x2, y2],
"label": "object name/description"
}}]
```
- Output: Generates bounding boxes for visual inspection of the i-th image

2. **Path Tracer**
- Purpose: Plots movement or connections between points
- Input format: JSON
```json
[{{
"index": i, # Image index
"start_point_2d": [x1, y1],
"end_point_2d": [x2, y2],
"label": "trace_description"
}}]
```
- Output: Generates visual paths for verification of the i-th image

Required Output Format:
For each reasoning step, you must structure your response as follows:
<think> [Your detailed reasoning process] </think> Action: [Object Mapper/Path Tracer]
```json
[JSON format coordinates]
```

After your reasoning and iteratively refine your solution through visualization feedback, you should arrive at a final answer and structure your response as follows:
<think> [Your detailed reasoning process] </think> Action: Answer
<answer> [Your final answer] </answer>

Please NOTE the following reasoning techniques:
1. Initial Analysis
- Break down the spatial problem
- Plan your approach

2. Iterative Reasoning for Each Step
- Choose appropriate tool
- Provide absolute coordinates in JSON format (The top-left corner of the image is (0, 0) and the bottom-right corner is (**{width}**, **{height}**))
- Observe the visualization output
- Reflect on the visualization:
* Is the placement/path accurate?
* Does it align with your reasoning?
* What adjustments are needed?
- Backtrack and Adjust:
* If errors found, backtrack to previous step to modify actions or decisions as needed

Figure 3: System prompt used in SPARK.

Video Query Prompt

 The index of the given image is 1 (width: **{width}**, height: **{height}**).
…
 The index of the given image is **{number_frames}** (width: **{width}**, height: **{height}**).
These are frames from a video, numbered from 1 to **{number_frames}** in sequence. That is, the index of each image is 1, 2, 3, ..., **{number_frames}**.
Answer the question with appropriate tools:
{Question}
{Output Template}
Begin your reasoning. After each tool use, critically evaluate the visualization and adjust if needed:

Image Query Prompt

{Question}
The index of the given image is 1.
{Output Template}
Begin your reasoning. After each tool use, critically evaluate the visualization and adjust if needed:

Follow-up Query Prompt

The index of the given image is **{current_image_idx}** (width: **{width}**, height: **{height}**). Continue your reasoning. After each tool use, critically evaluate the visualization and adjust if needed:

Termination Prompt

The index of the given image is **{current_image_idx}** (width: **{width}**, height: **{height}**). Then, you can not invoke the Object Mapper or Path Tracer tool. Please answer the initial question and structure your response as required:

Output Template

"Multiple Choice": Answer with the option's letter from the given choices directly.
"Numerical": Please answer the question using a single numerical value.
"Regression": Please answer the question using a single numerical value.

Figure 4: Query prompt and output template used in SPARK.