Limits of Spatial Reasoning Using LLMs

Varis Ithivatana
Independent researcher, MSc Business Analytics Graduate, University of Exeter, UK

Brandon Bennett
School of Computing, University of Leeds, UK

Abstract

This paper investigates the limits of spatial reasoning of LLMs in a grid-based warehouse environment, with ground-truth answers computed by simulation. We test two kinds of task: executing given movement instructions, and generating shortest action sequences for planning problems. In the execution tasks, an agent must follow egocentric and allocentric actions and report its final absolute direction relative to a target object. In the planning tasks, the model must find a shortest route, sometimes with object carrying, allocentric descriptions, or ordered pallet-pushing goals. We evaluate 100 instances per experiment using **GPT-5-mini**, and compare a smaller planning set with **ChatGPT-5.2** (extended thinking), **Gemini-Pro-3.1**, and **Grok-Expert**. The results show high accuracy on short and moderately long instruction-following tasks, but reduced accuracy on long trajectories with a much sharper drop on planning tasks. Static landmark labels in the form of coloured walls improve robustness, compared with problems described in terms of cardinal directions, while planning with ordered object-pushing remains difficult even for stronger models.

2012 ACM Subject Classification Computing methodologies → Natural language processing; Computing methodologies → Spatial and physical reasoning

Keywords and phrases Spatial Reasoning, Large Language Models, Frame of Reference, Path Integration, Planning

Digital Object Identifier 10.4230/LIPIcs.COSIT.2026.17

Category Short Paper

Supplementary Material *Software (Source code and experiment results)*: `https://github.com/vi217/spatial-reasoning-llms`

Declaration on GenAI/LLM use Generative AI assistants were used during this work (2025–2026): Google **Gemini 2.5 Pro** and **Gemini 3.1 Pro** helped write a substantial part of the study's code (instance generation, the warehouse simulator and breadth-first-search ground-truth routines, and the scoring scripts), and Anthropic **Claude Opus 4.8** helped organise and refactor the code and prepare the code repository. All AI-generated code was read, executed, and tested by the authors, and the simulation and ground-truth logic was independently verified by a co-author. The authors retain full responsibility for the code, the results, and the content of this paper.

1 Introduction

Large language models (LLMs) perform well on many tasks that appear to require multi-stage inference, planning, or structured reasoning. However, it remains poorly understood what their underlying capabilities and limitations are, or how exactly such behaviour is achieved. The spatial domain provides useful test cases that involve combining pieces of information with complex interactions. Moreover, many spatial reasoning problems have clear ground-truth solutions that can be calculated by simulation or reasoning algorithms.

Our experiments use a warehouse domain with agents, pallets, racks, and forklifts occupying cells in a fixed grid. This setting is deliberately simple: it avoids visual ambiguity and allows actions to be interpreted as precise coordinate updates. At the same time, it supports several different kinds of spatial difficulty, including egocentric movement, allocentric

direction names, object movement, obstacle handling, object-relative orientation, and explicit planning. The main aim is not to introduce a new benchmark score, but to locate some points at which current LLM spatial reasoning behaviour begins to fail.

Benchmarks such as StepGame [5] and DecompSR [3] test multi-hop reasoning involving spatial relations and connect with broader work on compositional generalisation in neural models [2, 7]. In January 2024 Cohn and Blackwell [1] showed that models available at the time performed poorly on simple cardinal-direction questions. Current models are far more accurate on such short problems: in January 2026 we re-ran both of their datasets on **GPT-5-mini** and obtained 95% on the original small problem set and 88% on the larger version of the same problem set. We therefore examine longer, dynamic tasks where a model must reason about a spatial state that is updated by many actions.

2 Methodology

Our experiments concern a warehouse situation represented by a 10×10 grid, with cells indexed by x, y-coordinates. Each test specifies an initial agent position and facing, a target object, and a sequence of actions. The basic egocentric action set is $\mathbf{Ego} = \{F, B, SL, SR, TL, TR\}$, representing forward, backward, shift-left, shift-right, turn-left and turn-right. Some experiments also use diagonal moves and object-relative facing commands. For instruction-following tasks, the required output is one of $\mathbf{Card} = \{N, NE, E, SE, S, SW, W, NW\}$, expressing the direction from the agent to a named object in the final situation. Correct answers for instruction-following tasks are computed by a simulator implemented in Python. Ground-truth shortest paths for planning tasks are computed by breadth-first search. We used the same 10×10 warehouse layout for all experiments to hold the environment constant, so that performance differences reflect changes in the action rules and task requirements rather than the underlying map or grid size; varying the grid size and setting is left to future work.

2.1 Experiments

The evaluation consists of five experiments of increasing complexity. **Exp-1** is a baseline path-integration task where an agent executes a sequence of egocentric rotations and translations and reports the cardinal direction of a landmark relative to its final position. **Exp-1.1** replaces cardinal direction names by colour labels attached to the warehouse walls. A further variant, **Exp-1.2**, replaces only the cardinal direction names with coloured walls but keeps the egocentric diagonal moves; it is included for token-usage comparison at $k = 120$ only.

Exp-2 adds object movement and drop events: a robot pushes an object, a product is dropped after a specified unit push, and later agents transport it. **Exp-3** adds blocked "spill" cells and a detour rule. If a move would enter a blocked cell, it is replaced by a specified detour movement. The detour specification produces a sensible avoidance only when the worker approaches the spill from the West or East; when travelling North or South into a spill, the detour leaves the worker either one cell back from where it started or still adjacent to the obstacle. **Exp-3** therefore tests comprehension of a particular avoidance instruction rather than a general capacity to reason about avoidance and detours. **Exp-4** inherits the spill rule from **Exp-3** and also replaces some egocentric turning with object-relative orientation, so the agent must turn to face the current location of a specified object.

Whereas **Exp**s **1**–**4** mainly test state update under a supplied action sequence, **Exp**s **5**, **5.1**, **5.2** and **5.3** test the ability to construct an action sequence. This separation is useful because planning failures can otherwise be mistaken for failures of spatial interpretation. For **Exp-5** the model must output a shortest valid action sequence from a start position to a

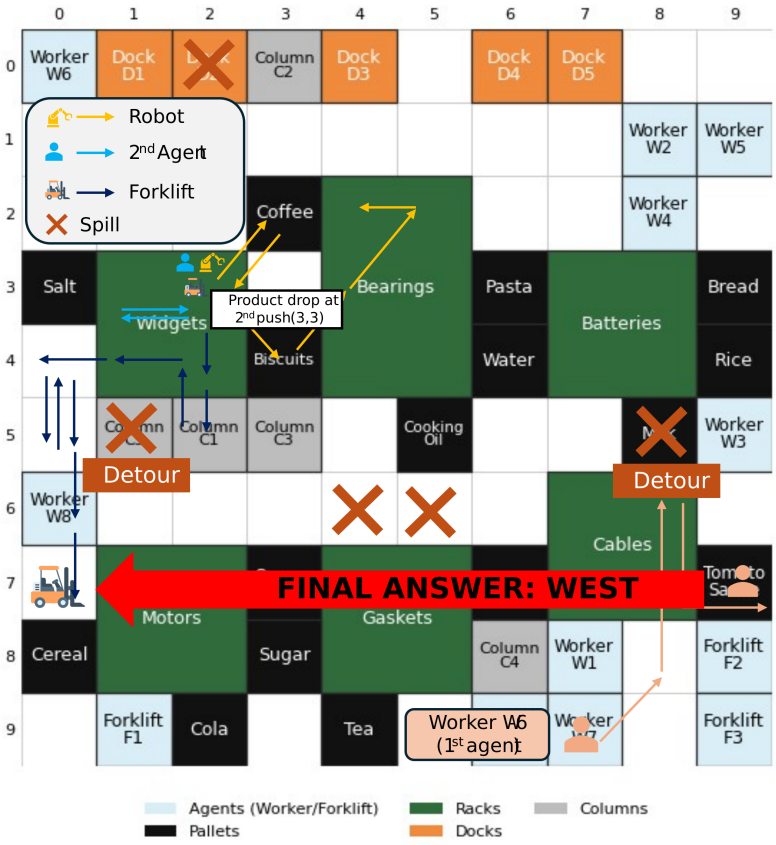

Figure 1 Warehouse grid layout, entity locations and example movement sequences used in the experiments. Crosses indicate locations of "spill" events.

target location. **Exp-5.1** adds object carrying; **Exp-5.2** adds allocentric "riddles" involving rack-facing relative to another agent; and **Exp-5.3** adds ordered pallet-pushing goals, where a push is valid only under specified hand, facing, and occupancy conditions.

2.2 Dataset, models and prompts

For each experiment, **GPT-5-mini** was evaluated on 100 instances, using two random seeds with 50 instances from each. For **Exps 1–4**, action sequences were generated from random-walk controls with a minimum length of five actions. The action distribution for the random walk was set to: turn 0.45, lateral shift 0.25, diagonal move 0.25, with the residual mass on forward/backward. These values were chosen so that all six action types appear within most instances and the agent's facing direction (the key state quantity the model must track) changes frequently. We did not perform a sensitivity analysis on these proportions and report them so that any reader can reproduce or vary them. The seeds 7 and 8 are arbitrary fixed values used for reproducibility. For the planning variants, we additionally evaluated **ChatGPT-5.2** (extended thinking), **Gemini-Pro-3.1**, and **Grok-Expert** on 30 instances per condition. These additional runs were performed through web interfaces and were therefore smaller, but they are useful for distinguishing limitations of **GPT-5-mini** from limitations that persist in stronger models.

Table 1 Summary of the experimental conditions. Unless otherwise stated, **GPT-5-mini** was evaluated on $n = 100$ instances and stronger models on $n = 30$ instances per experiment.

Experiment	Main additional requirement	Task type
1	Egocentric grid tracking, $k = 5$–300	execute
1.1	Cardinal directions replaced by coloured walls	execute
1.2	Coloured walls, keeping egocentric diagonals, $k = 120$	execute
2	Object pushed, dropped, picked up and transported	execute
3	Blocked spill cells with detour rule	execute
4	Object-relative orientation commands	execute
5	Shortest route to target	plan
5.1	Pick up, deliver, and return	plan
5.2	Allocentric rack-facing descriptions	plan
5.3	Ordered pallet-pushing goals	plan

All prompts are zero-shot. We avoided few-shot tests because in-context demonstrations risk leaking the structure of a valid trace (e.g. the JSON schema and canonical detour pattern) into the model response, converting **Exp-3** from a comprehension test into a copy-and-adapt task. We acknowledge that few-shot prompting may improve absolute accuracy and leave a controlled comparison for future work. The system prompt gives the coordinate convention, action rules, entity positions and output format. User prompts give the instance-specific start state, target, movement sequence and any additional constraints. For **Exp**s **2**–**4**, we also requested trace output to inspect error sources. Scoring, however, was based only on the final direction or plan, not the trace. The traces were purely diagnostic, allowing us to distinguish, for example, a wrong final direction caused by an early movement update from one caused by a later drop-location error.

2.3 Statistical procedure

Instances are generated independently for each experiment and condition, so all reported comparisons are between independent samples; we use two-proportion z-tests for differences and report 95% Wilson intervals per proportion. For **GPT-5-mini**, each experiment pools seeds 7 and 8 ($n = 50$ each, $n = 100$ total). Multiple comparisons among the stronger models (Exp 5–5.3) use the Holm correction.

2.4 Reproducibility considerations

Commercial LLMs may receive silent updates between versions, so our numerical results are just a snapshot of February 2026 behaviour. **GPT-5-mini** API runs (16–25 Feb. 2026) used the OpenAI `/responses` endpoint with model name `GPT-5-mini`, temperature 1.0, API `seed` 42, and top-p at the default. Every prompt and response is stored with the model name, prompt text, generation seed, token usage and elapsed time. Web-interface evaluations of **ChatGPT-5.2** (extended thinking), **Gemini-Pro-3.1** and **Grok-Expert** (20 Feb. 2026) are recorded as screenshots and transcripts; these are inherently less reproducible than API runs, so the more nuanced quantitative claims rely only on the API-based **GPT-5-mini** results. We plan to extend evaluation to open-weight models so the pipeline can be re-run independently of any commercial provider.

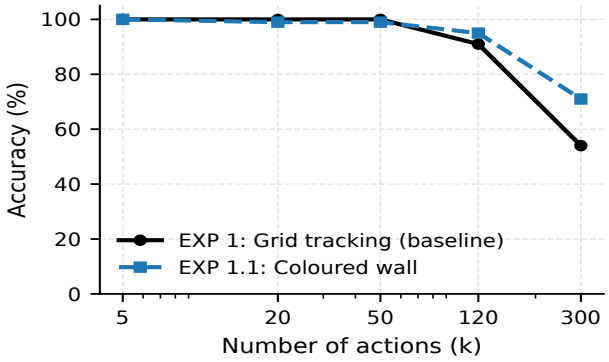

Figure 2 Accuracy with trajectory length k for **Exp-1** (baseline) and **Exp 1.1** (coloured walls).

3 Results

3.1 Instruction following

On the baseline trajectory task, **GPT-5-mini** remains highly accurate for short and medium action sequences. Accuracy is between 0.90 and 1.00 for $k = 5$–120, but falls at $k = 300$. The coloured-wall variant improves the longest-sequence result: at $k = 300$, baseline accuracy is 0.54, while coloured walls is 0.71 (Figure 2). The difference of 0.17 is statistically significant, with a 95% confidence interval of $[0.04, 0.30]$. Coloured walls also reduce median token usage relative to the baseline at $k = 120$ and above, indicating that the explicit landmark cue improves accuracy and reduces computation. **Exp-1.2**, with cardinal directions and egocentric diagonals, uses more tokens than **Exp-1.1** at $k = 120$, suggesting that replacing the diagonal moves contributes a substantial part of the token saving.

Pooled across both seeds ($n = 100$), accuracy is 0.99 in **Exp-2**, 0.88 in **Exp-3**, and 0.80 in **Exp-4**. The drop from **Exp-2** to **Exp-3** is statistically significant ($[0.04, 0.19]$). The further drop from **Exp-3** to **Exp-4** is not statistically significant at the 95% level ($[-0.02, 0.18]$, which includes zero).

Trace error analysis

For **Exp-3** and **Exp-4** we ran the trace prompt to identify where errors occur, using the seed-7 set ($n = 50$ per experiment). Errors in a small set of samples ($n = 5$ and $n = 11$ respectively) were categorised by a single annotator. Hence, we present the analysis as qualitative and illustrative rather than as a quantitative claim. To keep the categories mutually exclusive we use the following operational rules. "Incorrect movement action" applies when the trace's next coordinate disagrees with the simulator's and the step is not preceded by a spill cell; "failure to apply detour rule" applies when the trace enters a spill cell or omits the prescribed four-step detour. "Incorrect product-drop location update" applies when the recorded drop position differs from the simulator's at the specified step; "product dropped at incorrect step k" applies when the drop occurs at any step other than k. The four categories partition the failure space, so each trace step is assigned to exactly one. Table 2 shows the breakdown.

In Experiment 3, none of the five errors arose from misapplying the spill or detour rule. They came from updating the wrong agent coordinate, or tracking the product-drop position incorrectly. We interpret the Experiment 3 drop as a state-tracking failure under a longer

Table 2 Error type breakdown from full-trace analysis for Experiments 3 and 4 (seed 7, $n = 50$). Single annotator; qualitative, illustrative.

Experiment	Error type	Count	Share
Exp-3	Incorrect movement actions	2	40%
	Incorrect product-drop location update (used original forklift position)	2	40%
	Product dropped at incorrect step	1	20%
Exp-4	Incorrect movement actions (not including cases due to blockage)	7	64%
	Failure to detect blockage / apply detour rule	3	27%
	Incorrect product-drop location update	1	9%

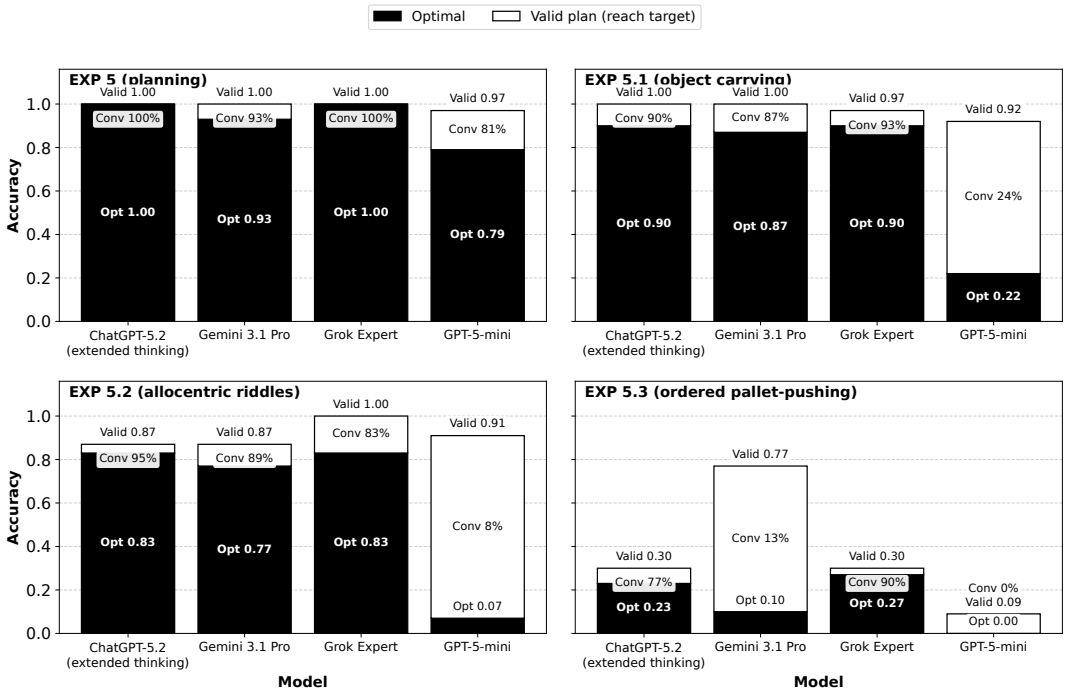

Figure 3 Planning accuracy for Experiments 5–5.3 by model. Black bars show optimal; outlined bars show valid plan minus optimal. Conversion rate = optimal / valid plan.

and more rule-laden prompt, rather than a rule-comprehension failure. In Experiment 4 the dominant pattern is incorrect movement actions (7 of 11), which suggests a separate failure mode in vector computation under variable binding.

3.2 Planning

Planning is substantially more difficult. **Exp-5** answers are scored as correct only when they reach the target by a shortest route. **GPT-5-mini** achieves 0.79 on the basic planning task, but falls to 0.22 with object carrying, drops further to 0.07 with allocentric riddles, and reaches 0.00 in the ordered pallet-pushing task (Figure 3). Its valid-plan rate is often higher than its optimal-plan rate, indicating that the model can frequently reach the target but fails to find a shortest route. The fraction of its valid plans that are also optimal, the *conversion rate* ("Conv" in Figure 3), is therefore low.

The stronger models perform much better on the first three planning variants. In **Exp-5** **ChatGPT-5.2** (extended thinking) and **Grok-Expert** both achieve optimal-route rate 1.00, and **Gemini-Pro-3.1** reaches 0.93. On **Exp-5.2**, **ChatGPT-5.2** achieves optimal-route rate 0.83,

Grok-Expert 0.83, and **Gemini-Pro-3.1** 0.77. All three nevertheless show a sharp decline on **Exp-5.3** in optimal-route rate: **ChatGPT-5.2** drops from 0.83 to 0.23 (a difference of 0.60), **Grok-Expert** from 0.83 to 0.27 (0.56), and **Gemini-Pro-3.1** from 0.77 to 0.10 (0.67), with all three differences statistically significant. A model-specific pattern is also worth noting: for **Exp**s **5.1**–**5.3**, **Gemini-Pro-3.1** consistently shows the lowest optimal-route rate among the stronger models while maintaining a high valid-plan rate, particularly on **Exp-5.3**, indicating that it reliably reaches the target without finding the shortest route. Therefore, under the specific prompting, representation, and models tested, ordered object pushing marks a limitation of current spatial planning rather than a general bound; because the collapse appears across three independent frontier models, we expect it to hold for newer ones, though confirming this is left to future work.

A qualitative pattern in the **GPT-5-mini** planning errors is avoidance of diagonal moves: the model produces a valid route composed mainly of cardinal moves, although a shorter route using diagonals exists. This implies a non-optimal route-generation strategy rather than systematic search over the action space. The optimality requirement is therefore crucial: without it, **Exp-5** would be much easier.

4 Discussion

The results suggest that generating an action sequence to achieve a goal is much more difficult for LLMs than inferring spatial relations holding after executing a specified action sequence. **GPT-5-mini** is often robust when following long sequences, even when several kinds of spatial relations and events are involved. Its limitations emerge for very long trajectories, but much more acutely for planning, where a shortest route must be found rather than merely followed. The coloured-wall result is notable: replacing cardinal names by arbitrary colour labels improves performance on very long trajectories. We had expected better performance with the more conventional way of expressing directions. Perhaps the colour coding forces the LLM into a more algorithmic mode, rather than relying on statistical correlations that may be insufficient for handling long trains of reasoning.

The planning failures are consistent with earlier observations that LLMs do not reliably implement exhaustive search [6]. A valid but non-optimal route can be generated by following a locally reasonable pattern, whereas an optimal route requires comparing alternatives. The ordered pallet-pushing task adds further constraints: the model must interleave route choice, object status, facing direction, and ordering. This combination sharply reduces performance even for stronger models. **Exp-5.3** therefore marks a limitation of current spatial planning under the specific prompting, representation, and model set tested. We note one design limitation: the later experiments add planning on top of spatial reasoning, so a failure in **Exp-5.3** is more accurately a failure of planning under spatial and object-state constraints than a pure spatial-reasoning failure. This remains relevant in practice, since many real uses of spatial understanding involve considering alternative possibilities, but the distinction matters when interpreting the results.

5 Conclusion

We evaluated LLM spatial reasoning in a grid environment with simulation-derived ground truth. **GPT-5-mini** performs well on many instruction-following tasks, but accuracy declines on very long trajectories and when dynamic object-relative orientation is added. Compared to instructions given in terms of cardinal directions, static landmarks in the form of coloured walls

improved long-sequence robustness and reduced token use at $k = 120$ and above. Planning is harder: valid routes are often produced, but shortest routes missed. Ordered pallet-pushing goals cause a sharp drop for all models tested. This is not surprising, since spatial action planning is intrinsically difficult. However, the observation that once a certain complexity threshold is reached, reasoning accuracy drops very sharply (without any indication in the actual output) could be a serious issue for any system relying on an LLM for spatial planning.

Future work should more clearly separate different aspects of spatial reasoning, as well as investigating their combinations in a systematic way; in particular, varying the grid size and setting (e.g. larger or outdoor layouts) would help disentangle sequence-length effects from grid-size effects. Going beyond simple grid-based situations to richer spatial puzzle domains [4] will further illuminate the spatial reasoning capabilities and limitations of LLMs, as will evaluating newer model snapshots (e.g. Claude Opus and post-GPT-5.4 models) to test whether the failure modes we identify persist as models improve.

References

1 Anthony G Cohn and Robert E Blackwell. Evaluating the ability of large language models to reason about cardinal directions (short paper). In *Proceedings of the 16th conference on Spatial Information Theory (COSIT)*, volume 315. Schloss Dagstuhl, LIPIcs, 2024. doi:10.4230/LIPIcs.COSIT.2024.28.

2 Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020. doi:10.1613/JAIR.1.11674.

3 Lachlan McPheat, Navdeep Kaur, Robert Blackwell, Alessandra Russo, Anthony G. Cohn, and Pranava Madhyastha. Decompsr: A dataset for decomposed analyses of compositional multihop spatial reasoning, 2025. doi:10.48550/arXiv.2511.02627.

4 Paulo E Santos, Pedro Cabalar, Zoe Falomir, and Thora Tenbrink. Representing and solving spatial problems. *Spatial Cognition & Computation*, 25(1):1–14, 2025. doi:10.1080/13875868.2024.2411052.

5 Zhengxiang Shi, Qiang Zhang, and Aldo Lipani. StepGame: A new benchmark for robust multi-hop spatial reasoning in texts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. Pre-print: arXiv:2204.08292. doi:10.48550/arXiv.2204.08292.

6 Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models: A critical investigation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

7 Zhuoyan Xu, Zhenmei Shi, and Yingyu Liang. Do large language models have compositional ability? an investigation into limitations and scalability. In *First Conference on Language Modeling*, 2024.

A Data, Code, and Further Results Availability

Our code repository including prompt sets, raw model outputs, and an extensive technical report are available from `https://github.com/vi217/spatial-reasoning-llms`.

Our simulator scripts enable the results to be reproduced as faithfully as possible while the underlying model version remains available. The same tests could also be run on future models.