# Assessing Spatial-Temporal Reasoning Capacity in Large Language Models: A Path-Finding Task in Three-Dimensional Space

**Will Calandra, Sichen Lu, Garvit Luhadia, Qing Wen**
{wcc9105, sl10911, gl2758, qw919}@nyu.edu

## Abstract

Understanding spatial correlations in language is an essential component of human-like intelligence in artificial intelligence systems. This research presents a novel approach for studying the ability of large language models (LLMs) to encode and decode spatial-temporal information purely from textual descriptions. While prior works have studied LLM reasoning through 2D spaces, we seek to fill the gap in 3D spaces. We present a new task for testing the spatial-temporal reasoning abilities of LLMs through path finding in simulated 3D environments. We feed LLMs a textual description of 3D grid dimensions, a starting point, a target point, and obstacle coordinates to test the models' abilities to plan and output a correct path sequence to the target. As natural language descriptions increase in complexity in higher dimensions, we believe LLMs would have difficulty tracking and understanding the spatial-temporal orientation of our 3D environment relative to standard 2D tasks. Corroborated by worse path finding performance in 3D compared to 2D, we show that LLMs struggle to accurately reason about 3D spaces, which is exacerbated as each dimension increases in size.

## 1 Introduction

Spatial reasoning refers to the ability to plan a path, locate objects, and visualize objects from descriptions of the environment (Byrne and Johnson-Laird, 1989). Given that LLMs are text-based, we define spatial reasoning ability in LLMs as the capacity to understand, interpret, and generate information related to spatial relationships and configurations.

LLMs' abilities to process natural language have improved significantly in recent years (Naveed et al., 2024). However, handling spatial information, including encoding and decoding spatial relationships from text, remains a struggle (Yamada et al., 2024). This performance gap has inspired our study to develop a new task that applies stricter spatial reasoning criteria. By comparing performance across different setups in our task, we aim to uncover potential thresholds and capabilities of LLMs in handling complex spatial reasoning scenarios.

We hypothesize that LLMs will perform worse on the path-finding task in 3D space for several reasons. First, the increased context size of a 3D grid space makes it more difficult for an LLM to understand. Second, describing the depth dimension of a 3D space requires more ambiguous vocabulary. Third, the greater number of divergent paths from the starting position to the target in a 3D space reduces the probability of success by random chance, posing additional challenges for an LLM to find the correct path.

Our task evaluates LLMs' pathfinding capabilities in 3D spaces by measuring the percentage of times the LLM reaches the target. Metrics include the identification of paths that lead to the final goal without leaving the 3D grid space or hitting any obstacles (i.e. valid paths) and comparison of the LLM's path to the shortest path generated by $A^*$ algorithm (i.e. optimal paths).

## 2 Related Work

While there exists a variety of tasks that require spatial reasoning abilities, path-finding or navigating in an environment appears to be one of the most straightforward ways to evaluate spatial reasoning capability of LLMs according to our definition.

In their paper "Path Planning from Natural Language (PPNL)," Aghzal et al. (2024) designed a spatial reasoning task for LLMs, where the model needs to navigate through a simulated 2D environment to reach a goal based on text descriptions. Their results demonstrated that naive LLM achieved an average of $50\%$ accuracy and the accuracy could bhigher than $95\%$ with fine-tuned models.

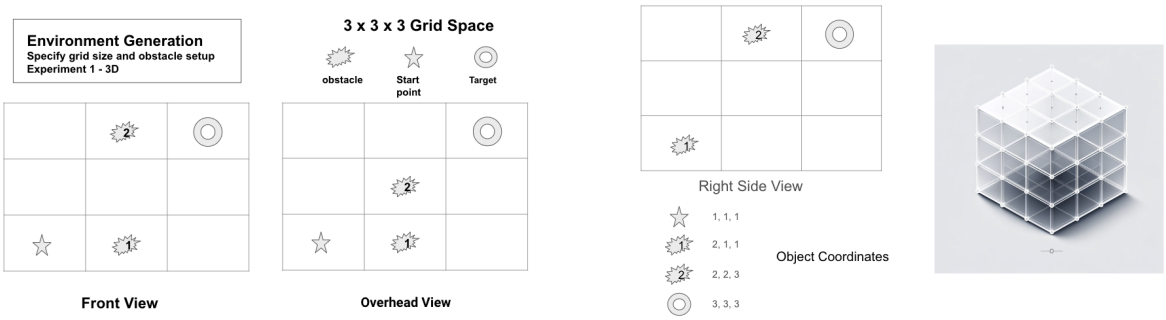


Figure 1: Example of environment setup with front view, overhead view, and right side view of the 3D grid space. Object coordinates are provided as example

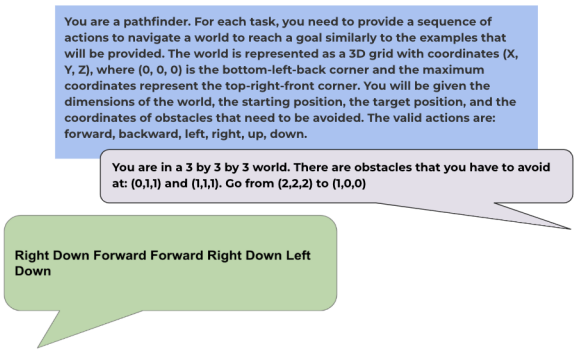


Figure 2: Example input data and expected output

3D spaces are inherently more complex than 2D scenarios as they offer more degrees of freedom. Conner et al. (1992) reveals challenges they have encountered when designing 3D user interfaces from 2D counterparts. In the context of path finding, an additional dimension results in a richer set of spatial relationships, navigational paths, and dimensional considerations. Accordingly, calculations and path-finding in 3D spaces are more computationally intensive. Liang et al. (2018) shows that Genetic Algorithms (GA) require significantly more iterations and computational resources in 3D environments than in 2D one. Consequently, any tasks that can be easily performed in 2D settings may become obscure and challenging in 3D contexts.

Regarding the scaling up of path-finding algorithms, research by Chen et al. (2024) shed light on the difficulty of the multi-agent path finding (MAPF) problem when using language models as agents. The task aims for agents to minimize the time (measured as distance) to reach their goal. The authors conclude that LLMs perform poorly compared to reinforcement learning counterparts, as LLM agents fail to generate multi-step plans, do not minimize travel distance, and collide with other agents during path planning. This suggests that LLMs have poor spatial reasoning performance in multi-stage, multi-agent environments. Therefore, we believe LLMs will struggle with navigating a 3D grid without prompting feedback.

## 3 Path Finding in 3D Space

### 3.1 Task Description

In our task, LLMs are required to find a viable path from a designated starting point to a target location within a 3D grid space. This space is explicitly defined with various obstacles that the model must avoid on its path to the target. As shown in Figure 2, the task is presented to the LLMs through textual descriptions, which detail the size of the grid, the locations of obstacles, the starting point, and the target. We prompt the LLMs using a system prompt that provides the task and coordinate system, in addition to three few-shot prompting examples that supply the ground truth description for our task. This description describes the path in terms of relative positions (up, down, left, right, forward, backward) to simulate how a human might verbally describe directions in a 3D environment.

By feeding the model textual data including coordinates of the objects and a description of the 3D grid, we expect the model to output a sequential relational path such as "right down forward forward right down left down."

### 3.2 Environment Setup

As shown in Figure:1, we define a $3 \times 3 \times 3$ grid with coordinates from (1, 1, 1) to (3, 3, 3). We place a starting point at (1, 1, 1), a target at (3, 3, 3), and obstacles at (2, 1, 1) and (2, 2, 3). We ask the model to find the path through the space.

| Setting | Template |
|---|---|
| World Description | You are in a {N} by {N} world. There are obstacles that you have to avoid at {obstacles}. |
| Enumerating Goals | $p\{i\}$ is located at $\{(x_i, y_i)\}$ |
| Single goal | Go from $\{(x_{init}, y_{init})\}$ to $\{(x_{goal}, y_{goal})\}$. |
| Initial Location | You are at $\{(x_{init}, y_{init})\}$ |

### 3.3 Data Generation

We generate our datasets containing 3D grid world environments with obstacles using heuristic methods, wherein we create a base 3D grid environment of a specified size with randomly placed obstacles. We then generate variations by randomly placing a starting position (agent) and one or more target positions (goals) within the grid, avoiding obstacle locations. Finally, we generate a corresponding natural language description. The description includes the grid size, obstacle locations, starting position, and target locations. Additionally, we compute the optimal path between the start and each target using the A* algorithm, providing the solution as coordinates, and ground truth point-to-point directions.

### 3.4 Evaluation Metrics

The evaluation of LLM performance on this task involves a multi-step process:

**Textual Interpretation**: Initially, the LLM must correctly interpret the spatial layout as described in the text, understanding the relative positions and dimensions of obstacles, start points, and end points. We carefully tailor prompting templates to best communicate our task to each LLM tested.

**Path Planning**: Subsequently, the model must generate a sequence of moves that leads from the start to the target without intersecting any obstacles. This sequence should ideally represent the shortest possible path, reflecting efficient spatial reasoning.

**Obstacle Avoidance**: The model must demonstrate an understanding of the three-dimensional nature of obstacles, planning paths that consider vertical as well as horizontal dimensions.

We use a number of metrics, defined as follows: *success rate*, *optimal rate*, and *reached rate*. The success rate is analogous to the task completion accuracy, and the optimal rate is the fraction in which the LLM plans the shortest path to the target. Further, the reached rate is the fraction of times in which the LLM reaches the target, even if it does not end at the target. We included this metric to relax the requirements for success as LLMs occasionally bypassed the target in its path. One can think of these three metrics as encompassing the accuracy, precision, and recall of LLMs in the 3D path-finding task.

## 4 Experiments and Analysis

In order to define a task that tests models on increasing levels of spatial complexity, we take a three-pronged approach where we tune the dimensionality of the data, the size of the search space (the grid), and the number of obstacles placed to obfuscate the model. We primarily test decoder-only architectures like the Llama model as these offer unparalleled efficiency in inference and in real-time applications Since we are resource constrained and could only test models with a smaller number of parameter counts, this was an essential consideration. We run permutations of the three spatial challenges to test the models' path finding ability on varying levels of difficulty. The results of our experiments 3 4 show that the models behave in consistent harmony when the search space is increased.

We also utilize advancements in autoregressive models like Command-r-plus from CohereForAI, which touts itself in grounded generation capabilities and is optimized for a variety of use cases including reasoning and question answering. In our analysis, we run inference on these models on the HPC cluster. To achieve efficiency, we employ methods like data parallelism, model quantization, and model parallelism to be able to run exhaustive experiments on our task to test its robustness in evaluating a LLM's output when prompted on our task. In order to attempt to decode our results, we run further tests where we try to understand the sudden dropoff in performance on spatial reasoning ability by all the models we ran our analysis on.

## 5 Results

We observe in 3 that the vanilla LLaMA-3 implementation outperforms all other models in both 2D and 3D spaces, but in general, all models perform poorly across our success, optimal, and reached rate metrics. We notice that the models perform worse in 3D compared to 2D spaces, which matches our hypothesis that LLMs are poor reasoners in 3D. In order to provide further evidence that LLMs struggle with spatial complexity, we see that within each dimension, an increase in the grid size

leads to decreased performance across all language models.

While our success rates and optimal rates have different definitions, the results are generally the same across all language models. We observe that when LLMs are able to plan an accurate path given few-shot examples, they align directly with the optimal path. However, the language models also traverse the goal and continue in the grid space, as evidenced the increase in our reached rates compared to our success rates.

As our task definition requires precise path planning abilities, we explored the relative spatial reasoning of LLMs through a second experiment. We parsed the language output from the LLM and measured the proportion of words that corresponded to the correct direction along a path. This experiment was informed by our observations that LLMs often had the right start in their path sequence, but lost it along the way, which is expected given the literature in poor state tracking. We find in 3 that LLMs often have the right set of directions in 2D space, but are unable to compose them together to chain a correct path. In contrast, in 3D, LLMs have lower directional success rates, which supports our hypothesis that LLMs exhibit poor spatial reasoning skills in complex spatial environments.

| Model Name | Dim | Success | Optimal | Reached |
|---|---|---|---|---|
| Cmd-R | 3x3 | 1.64 | 1.64 | 16.39 |
| Cmd-R | 5x5 | 0.75 | 0.00 | 2.99 |
| Cmd-R | 3x3x3 | 0.00 | 0.00 | 2.03 |
| Cmd-R | 5x5x5 | 0.00 | 0.00 | 0.68 |
| LLaMA2 | 3x3 | 2.00 | 2.00 | 14.00 |
| LLaMA2 | 5x5 | 0.00 | 0.00 | 6.71 |
| LLaMA2 | 3x3x3 | 0.81 | 0.81 | 7.32 |
| LLaMA2 | 5x5x5 | 0.00 | 0.00 | 0.68 |
| LLaMA3 | 3x3 | 12.67 | 12.67 | 14.67 |
| LLaMA3 | 5x5 | 2.67 | 2.67 | 9.33 |
| LLaMA3 | 3x3x3 | 2.00 | 2.00 | 8.00 |
| LLaMA3 | 5x5x5 | 0.00 | 0.00 | 1.33 |
| Nvidia | 3x3 | 4.67 | 4.67 | 12.67 |
| Nvidia | 5x5 | 0.00 | 0.00 | 5.15 |
| Nvidia | 3x3x3 | 3.26 | 3.26 | 7.61 |
| Nvidia | 5x5x5 | 0.00 | 0.00 | 0.00 |

Table 1: Experiment Compilation: Aggregated Results

## 6 Conclusion

In this initial rendition of our results, we find that the models perform significantly worse on both accuracy and optimality than what they did for 2 dimensions. Unsurprisingly, we find that Meta AI's

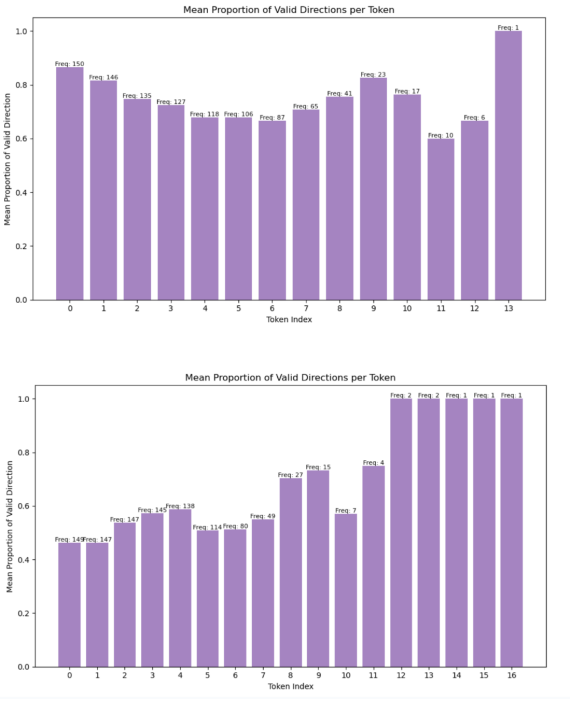


Figure 3: Experiment 2

newer Llama3 outperforms Llama2. Interestingly, our results show that the much larger 35B parameter by CohereAI performs worse than Llama2-7B-chat (et al., 2023). In our experimentation, wherever possible, we have tried to use instruction-tuned or chat variants of models to give them the best chance at our defined task.

Our finding of poor LLM performance in 3D path finding tasks align with the limitations and challenges identified in the cited literature. In our experiments, we find that the tested models, regardless of their intended use case, exhibit behavior that suggests that language models find it difficult to navigate the three-dimensional world, which points to their poor spatio-temporal reasoning abilities. The prompting styles required for the Mixture of Expert model and the use of instruction-tuned or chat variants highlight the importance of prompting and model architecture in improving spatial-temporal reasoning abilities.

## 7 Limitations and Future Work

In the future, we plan to integrate newer SOTA techniques like using state-space models derived from Mamba, like the Jamba-v0.1 (Lieber et al., 2024) from AI-21 labs.

We will also integrate methodologies such as (Jiang et al., 2023) LLM blending introduced by the Allen Institute for AI in ACL2023, where we

aim to create an ensembling framework that leverages the complementary capabilities of different LLMs to generate consistently superior results on our instruction following task.

## 8 Contribution Statement

**Will Calandra:** Documentation/extension of code for the 3D task, wrote and ran experiments, compiled results.

**Sichen Lu:** Formulated project ideas, hypothesis, and experimental setup; Developed codes for data generation and LLM output evaluation; Drafted Introduction section for proposal; Created tables for the paper.

**Garvit Luhadia:** Helped implement and come up with the logic; Developed the codebase into a coherent data distributed and parallelized script for inference on the HPC cluster; Conducted the experiments; Drafted the Methodology and Discussion sections of the paper.

**Qing Wen:** Discovered the codebase that is used for our extension for the 3D task; Constructed the paper framework; Drafted Abstract, Introduction, figures, task description, and references; Developed illustration to demonstrate our task and environment set up; Contributed to discussion of project ideas.

## References


Mohamed Aghzal, Erion Plaku, and Ziyu Yao. 2024. Can large language models be good path planners? a benchmark and investigation on spatial-temporal reasoning. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Ruth M.J Byrne and P.N Johnson-Laird. 1989. Spatial reasoning. *Journal of Memory and Language*, 28(5):564–575.

Weizhe Chen, Sven Koenig, and Bistra Dilkina. 2024. Why solving multi-agent path finding with large language model has not succeeded yet. *Preprint*, arXiv:2401.03630.

Brookshire D Conner, Scott S Snibbe, Kenneth P Herndon, Daniel C Robbins, Robert C Zeleznik, and Andries Van Dam. 1992. Three-dimensional widgets. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 183–188.

Marc-Alexandre Côté, Akos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. 2019. Textworld: A learning environment for text-based games. In *Computer Games: 7th Workshop, CGW 2018, Held in Conjunction with the 27th International Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, July 13, 2018, Revised Selected Papers 7*, pages 41–75. Springer.

Hugo Touvron et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *Preprint*, arXiv:2306.02561.

Xiao Liang, Guanglei Meng, Yimin Xu, and Haitao Luo. 2018. A geometrical path planning method for unmanned aerial vehicle in 2d/3d complex environment. *Intelligent Service Robotics*, 11:301–312.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. 2024. Jamba: A hybrid transformer-mamba language model. *Preprint*, arXiv:2403.19887.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2024. A comprehensive overview of large language models. *Preprint*, arXiv:2307.06435.

Roma Patel and Ellie Pavlick. 2021. Mapping language models to grounded conceptual spaces. In *International conference on learning representations*.

Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M Lake. 2020. A benchmark for systematic generalization in grounded language understanding. *Advances in neural information processing systems*, 33:19861–19872.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020a. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020b. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2022. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). *arXiv preprint arXiv:2206.10498*.

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

# A APPENDIX

| Benchmark | Task Environment |
|---|---|
| (Patel and Pavlick, 2021) | 2D Grid |
| (Ruis et al., 2020) | 2D Grid |
| (Shridhar et al., 2020a) | Embodied Environment |
| (Côté et al., 2019) | Embodied Environment |
| (Shridhar et al., 2020b) | Human-annotated Environment |
| (Valmeekam et al., 2022) | Blocksworld |
| (Aghzal et al., 2024) | 2D Grid |

Table 2: Benchmark Comparison: Existing Spatial Reasoning Task Environments

| Model Name | Dim | Success | Optimal | Reached |
|---|---|---|---|---|
| Cmd-R | 3x3 | 1.64 | 1.64 | 16.39 |
| LLaMA2 | 3x3 | 2.00 | 2.00 | 14.00 |
| LLaMA3 | 3x3 | 12.67 | 12.67 | 14.67 |
| Nvidia | 3x3 | 4.67 | 4.67 | 12.67 |
| Cmd-R | 5x5 | 0.75 | 0.00 | 2.99 |
| LLaMA2 | 5x5 | 0.00 | 0.00 | 6.71 |
| LLaMA3 | 5x5 | 2.67 | 2.67 | 9.33 |
| Nvidia | 5x5 | 0.00 | 0.00 | 5.15 |
| Cmd-R | 7x7 | 0.67 | 0.67 | 2.01 |
| LLaMA2 | 7x7 | 0.00 | 0.00 | 6.12 |
| LLaMA3 | 7x7 | 2.67 | 2.67 | 4.00 |
| Nvidia | 7x7 | 0.00 | 0.00 | 0.00 |

Table 3: Experimental Analysis: Results on 2D gridspace

| Model Name | Dim | Success | Optimal | Reached |
|---|---|---|---|---|
| Cmd-R | 3x3x3 | 0.00 | 0.00 | 2.03 |
| LLaMA2 | 3x3x3 | 0.81 | 0.81 | 7.32 |
| LLaMA3 | 3x3x3 | 2.00 | 2.00 | 8.00 |
| Nvidia | 3x3x3 | 3.26 | 3.26 | 7.61 |
| Cmd-R | 5x5x5 | 0.00 | 0.00 | 0.68 |
| LLaMA2 | 5x5x5 | 0.00 | 0.00 | 0.68 |
| LLaMA3 | 5x5x5 | 0.00 | 0.00 | 1.33 |
| Nvidia | 5x5x5 | 0.00 | 0.00 | 0.00 |
| Cmd-R | 7x7x7 | 0.00 | 0.00 | 0.00 |
| LLaMA2 | 7x7x7 | 0.67 | 0.67 | 3.33 |
| LLaMA3 | 7x7x7 | 0.00 | 0.00 | 1.34 |
| Nvidia | 7x7x7 | 0.00 | 0.00 | 3.13 |

Table 4: Experimental Analysis: Results on 3D gridspace

| Dim | Obstacles | Success | Optimal | Reached |
|---|---|---|---|---|
| 3x3 | Low | 21.05 | 21.05 | 21.05 |
| 3x3 | High | 11.45 | 11.45 | 13.74 |
| 5x5 | Low | 3.33 | 3.33 | 8.33 |
| 5x5 | High | 2.22 | 2.22 | 10.00 |
| 7x7 | Low | 5.63 | 5.63 | 5.63 |
| 7x7 | High | 0.00 | 0.00 | 2.53 |
| 3x3x3 | Low | 5.26 | 5.26 | 5.26 |
| 3x3x3 | High | 0.89 | 0.89 | 8.93 |
| 5x5x5 | Low | 0.00 | 0.00 | 2.74 |
| 5x5x5 | High | 0.00 | 0.00 | 0.00 |
| 7x7x7 | Low | 0.00 | 0.00 | 1.22 |
| 7x7x7 | High | 0.00 | 0.00 | 1.49 |

Table 5: Evaluation on LLaMA 3