Exploring and Improving the Spatial Reasoning Abilities of Large Language Models

Manasi Sharma
Stanford University
manasis@cs.stanford.edu

Abstract

Large Language Models (LLMs) represent formidable tools for sequence modeling, boasting an innate capacity for general pattern recognition. Nevertheless, their broader spatial reasoning capabilities, especially applied to numerical trajectory data, remain insufficiently explored. In this paper, we investigate the out-of-the-box performance of ChatGPT-3.5, ChatGPT-4 and Llama 2 7B models when confronted with 3D robotic trajectory data from the CALVIN baseline and associated tasks, including 2D directional and shape labeling. Additionally, we introduce a novel prefix-based prompting mechanism, which yields a 33% improvement on the 3D trajectory data and an increase of up to 10% on SpartQA tasks over zero-shot prompting (with gains for other prompting types as well). The experimentation with 3D trajectory data offers an intriguing glimpse into the manner in which LLMs engage with numerical and spatial information, thus laying a solid foundation for the identification of target areas for future enhancements.

1 Introduction

Large Language Models (LLMs), e.g. GPT-4 [17] & PaLM [4], are massive models trained on diverse corpora with billions of tokens of text. Recent works have established the competence of LLMs in extrapolating more abstract, non-linguistic patterns, thus allowing them to serve as "general pattern machines [15]. As such, in addition to the text-based tasks for which they were trained, LLMs successfully demonstrate cross-disciplinary capabilities, such as high-level planning for robotic policies [9, 23, 1], reward function design [10, 8], and math & logic puzzles.

Labeling of various kinds of data [6] falls under the paradigm of general pattern matching, and is of utmost practical use in the prospective organization of unlabeled raw datasets. One such application is in the space of language & robotics; there are a limited number of datasets that supply language annotations for each demonstration (e.g. the trajectory is described by the instruction "pick up the blue cup"), as human labeling is costly. Intrigued by the potential for LLMs in annotating large-scale robotic datasets, we investigate LLM labeling as applied to 2D and 3D robotic trajectory data, i.e. the ability to describe a sequence of n-dimensional points with the type of motion it embodies, such as "lifting". While it is possible to explicitly train models for these capabilities, this work instead focuses on the inherent abilities of LLMs out-of-the-box, which may have downstream implications for broader numerical and spatial queries, e.g. trend analysis or time-series data.

Additionally, considering that LLMs appear to struggle with spatial reasoning abilities [2, 23, 5] (as defined by an understanding of shapes and relationships between different objects and spaces), we also gauge whether there are unique factors about this kind of numerical & spatial data that impact the improvements furnished by prompting techniques like In-context Learning (ICL) and

NeurIPS 2023 I Can't Believe It's Not Better: Failure Modes in the Age of Foundation Models Workshop.

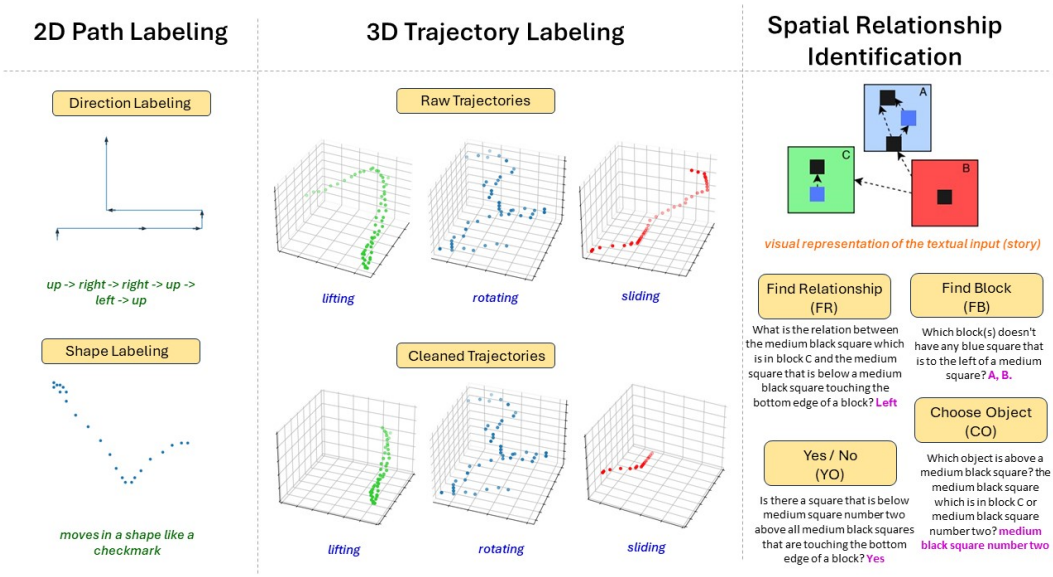

Figure 1: Illustrations of the Various Spatial Tasks: 2D Path Labeling of both directions (e.g. "up", "left", etc.) and shapes, 3D Trajectory Labeling of "lifting", "rotating" and "sliding" motions (as well as the cleaned versions), and relationship identification between blocks in an imagined setup.

Chain-of-Thought (CoT) prompting. Subsequently, we propose a strategy to pre-fix a prompt with a more general example to achieve greater performance gains for this type of application.

To summarize, we are overall interested in empirically answering the following research questions:
RQ1 Can LLMs be used to identify simple spatial patterns (e.g. circles or straightforward directions)?
RQ2 Can LLMs be used to identify and label more complex spatial patterns (such as more irregular 3D trajectories)? As such, does the irregularity of the pattern make it difficult to CoT-type reasoning-based prompting?
RQ3 Is there any knowledge transfer that can happen from simpler spatial tasks to more complex ones that can impact overall performance?

2 Related works

LLMs and Spatial Pattern Matching. Previous work has shown that LLMs can improve and complete low-level robotic action sequences or repetitive progressions like a sinusoid [15]. However, labeling of a sequence in its entirety has not been addressed, which requires long-form context retention, semantic comprehension to link a sequence to its textual annotation, and generalization to non-repetitive, complex patterns. Some works show that LLMs acceptably label one-dimensional time-series data [25, 11], but assess much shorter sequences, exclude higher dimensional analysis, or use additional token embedding models. Furthermore, on the whole, prevailing examinations of LLMs' spatial reasoning abilities [2, 23, 5] have revealed a considerably poor performance. We extend these works by tackling the inherent performance of LLMs on the underexplored subproblem of higher-dimensional trajectory identification.

LLMs & 3D Robotics. LLMs have been applied across a number of areas in robotics, most recently in originating high level step-by-step plans from task descriptions [1, 9] and robot policy code publication [10, 11]. However, our work falls into the bucket of whether LLMs can directly understand control (e.g., at the level of trajectories) in a zero-shot manner, which remains an open problem. There are also works in the embodied robotics domain where LLMs are used to reason about a 3D point-cloud scene [7, 19, 24], but these papers either use a vision model (or joint vision-language model) to integrate visual embeddings or append a finite number of 3D object positions acquired

using a detection model to the prompt. Our approach is distinguished by focusing on understanding continuous sequences of 3D points on the prompt-side.

LLMs & Prompting Several prompting approaches have been shown to improve results, such as In-context Learning [3], which supplies few-shot examples that guide the model, and Chain-of-Thought [21], which harnesses the ability of LLMs to adhere to a guided thought process for problem solving. The presence of symbols, patterns and texts are crucial to the effectiveness of CoT and ICL [12], but whether spatial trends follow such an archetype has yet to be explicitly examined.

3 Language Models as Trajectory Labelers

3.1 2D Path Labeling

Direction Labeling As discussed in [15] the ability of an LLM to pattern match is driven by in-context learning on the provided numerical tokens, which can be formulated as the problem of using the context $s_{1:k} = (s_1, ..., s_k)$, where each s_i is a symbol and using it to autoregressively predict s_{k+1} by using the factorized conditional probability $p(s_{1:k}) = \Pi_{i=1}^{n} p(s_i|s_1, ..., s_{i-1})$. Usual in-context learning examples segment the prompt into continuations of multiple examples, each a variable length sequence: $x_{1:k} = (x_1, ..., x_k)$ where each $x_i = (s_1^i, s_2^i, ..., s_{m^i}^i)$.

We adapt this paradigm for directional labeling by having x_1 state the model's expertise in spatial analysis and prompt it to generate direction labels for a newly provided sequence given the examples. Then each x_i from $i = 2$ onwards is an example, an input-output pair $D_i; L_i$ where D_i is the sequence of symbol aggregates $(d_1^i, d_2^i, ...d_j^i)$ of length j, with each d_k^i further being separated out into the symbols $(d_{k_x}^i, d_{k_y}^i)$, representing a coordinate in 2D Cartesian space. L_i is similarly a sequence of words $(l_1^i, l_2^i, ...l_{j-1}^i)$ of length $j-1$, where each l_k^i is a word representing the direction that describes the movement from d_k^i to d_{k+1}^i and is one of `[left, right, up, down]`. There are thus j points and $j-1$ segment labels (see Fig. 1).

Shape Identification Using a similar prompt framework to the one described above, x_1 states the model's expertise in spatial analysis and prompts it to identify the overall shape of the movement represented by the list of 2D coordinates; for example, "moves along a path that mirrors the pattern of a checkmark" or "moves in a circular path". Since the dataset size is fairly limited, no in-context learning is applied and the model is evaluated zero-shot by specifying x_2 as the input sequence of (x, y) coordinates $(d_1^i, d_2^i, ...d_j^i)$ (see Fig. 1).

3.2 3D Trajectory Labeling

Zero-shot Prompting To provide a baseline for the various prompting mechanisms probed, we initially implement zero-shot prompting, in which no exemplars are imparted to the model. Therefore, analogous to 2D experiments, x_1 states the model's expertise in spatial analysis and prompts it to classify the type of motion exemplified by the sequence into one of N categories, but there are no further sequences (see Fig. 2). For our set of experiments, we designate three classes that correspond to meaningfully and spatially disparate motions - lifting, rotating and sliding.

In-context Learning In-context Learning (ICL) [1] supplies a few samples that the model can generalize from. x_1 declares that the model that it is an expert in 3D trajectory labeling and prompts the model to classify the type of motion exemplified by the sequence into one of N categories. Then each x_i from $i = 2$ onwards is an input-output pair $D_i; l_i$ where D_i is the sequence of symbol aggregates $(d_1^i, d_2^i, ...d_j^i)$, with each one representing a coordinate in 3D Cartesian space $(d_{k_x}^i, d_{k_y}^i), d_{k_z}^i)$. l_i is a single word from one of the N classes describing the motion (see Fig. 2).

Chain-of-Thought Prompting Chain-of-Thought Prompting (CoT) [21], in which the few-shot examples are augmented by a step-by-step reasoning, has superseded In-context Learning on an array of textual and mathematical reasoning tasks. We extend it to our task as follows. The starting guideline for x_1 is the same as in-context learning, with the declaration that the model is an expert in

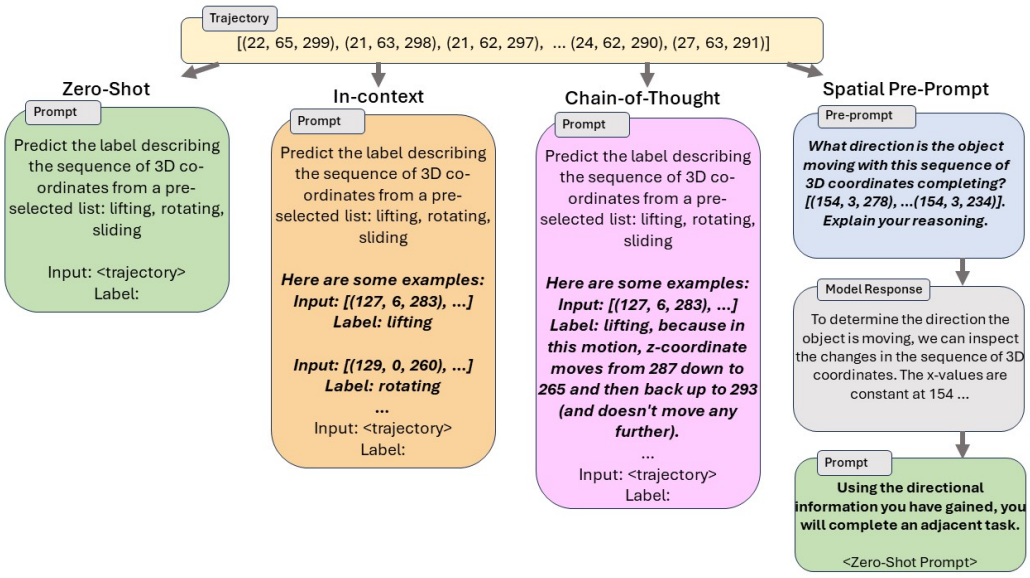

Figure 2: Different Types of Prompting Mechanisms - Zero-shot, In-context Learning, Chain-of-Thought and Spatial Prefix-Prompting. In Spatial Pre-Prompt, a tangential question is first asked, to which the model provides a response, following which the primary query is inquired.

3D trajectory labeling and a prompt for the model to classify the type of motion exemplified by the sequence into one of N categories. However in CoT, successive pairs $x_2, x_3, ..., x_{2t}, x_{2t+1}$ for some t, represent the few-shot examples. The even x_{2t} is the representative sequence of 3D coordinates and the odd x_{2t+1} is its associated answer, which is the motion-type label and accompanying reasoning steps (such as the fact that back-and-forth changes in the x and y coordinates can hint at a rotating motion, see Fig. 2).

Spatial Prefix-Prompting Anticipating the challenge with generalizing from irregular examples and bolstered by the model's performance on simpler 2D data, we also propose a new method of prompting called "Spatial Prefix-Prompting" (SPP). The method draws from a prior selection of fixed questions that instigate the model to first ponder a tangentially related spatial problem (e.g. identifying the single direction an object is moving in or checking whether a point is in the center of a circle), and then use the "knowledge gained" to answer an adjacent question, such as labeling a new, more complex 3D trajectory (see Fig. 2). This technique does not necessitate the more intensive CoT-style curation of step-by-step examples, and we hypothesize that it may build upon the more fundamental spatial concepts a model is trained on to generalize better than few-shot learning (wherein the selected examples may not be representative of all the trajectories).

4 Experiments

4.1 Implementation Details

4.1.1 2D Labeling and Description

There aren't many datasets for elementary 2D shapes and directions, and given the ease of generating such data, we decided to autogenerate the datasets. For the direction labeling task, a dataset of size 30 is generated with 10 short-horizon sequences of 2D coordinates (of length 6-8 segments), 10 long-horizon long sequences (length 35 - 40), and 10 short floating-point sequences using Python's NumPy package, with each segment's size and the directions randomly chosen based on a fixed seed. We experiment with both 1) scaling the values to integers between [0, 100] (to use fewer tokens to represent a single number) 2) scaling the values to double-digit fractions between [0, 100] (to test

Table 1: 2D Direction and Shape Labeling performance on short, long and floating-point sequences

	Direction Labeling						Shape Labeling	
	Integer (short)		Float (short)		Integer (long)		Integer	Float
LLM	Acc. (↑)	Err. # (↓)	Acc. (↑)	Err. # (↓)	Acc. (↑)	Err. # (↓)	Acc. (↑)	Acc. (↑)
ChatGPT-3.5	0.50	0.15	0.50	0.25	0.00	0.71	0.31	0.23
ChatGPT-4	**1.00**	**0.00**	**1.00**	**0.00**	**0.60**	**0.13**	**0.46**	**0.46**

whether the higher token representation translates to higher precision). Only Zero-shot and In-context Learning are applied for Shape and Direction Labeling respectively. For the shape identification, we use some previously collected hand-gesture data in which human demonstrators sign a variety of shapes, including circles and check marks, and 2D positions of the finger are recorded; the dataset tests the model on the inherent noise from human demonstrations. A subset of the dataset is "cleaned" for use by having an expert to remove extraneous points that don't belong to the shape, and it is normalized to the range [0, 100] (both integer and floating point). The size of the dataset is 13.

4.1.2 3D Trajectory Labeling

We use the CALVIN benchmark [13], a dataset for learning long-horizon language-conditioned tasks for robotics. It includes 3D end-effector positions (can be extracted from the low-dimensional state vector) and associated language descriptions of the action the robot is attempting to complete. Due to time and resource constraints from the human evaluations, we select only a small subset of the CALVIN dataset (30 samples) and three disparate subtasks ("rotate", "lift", "slide"). The trajectories are often complex and may not intuitively always resemble the action being completed, with many extraneous movements (see Fig. 1). Therefore, we also create a version of this dataset called "CALVIN-Cleaned" in which a human annotator extracts the parts of the trajectory that match with the specific action (e.g. only the lifting portion, see Fig. 1). Note, the CALVIN-Cleaned dataset retains the original "rotate" trajectories, as the task description is linked to the back-and-forth changes in the entirety of the motion, not any particular subsection. Finally, the dataset is normalized to the range $[0, 300] \in \mathbb{Z}$ to increase the granularity of the trajectory but optimize for token conservation due to the long-range of many CALVIN trajectories (upwards of 50 - 100 points).

4.2 Metrics and Models

As all of the tasks are classification / labeling tasks, we opt for traditional classification metrics, i.e. accuracy and F1-score, to reflect the balanced metrics of precision and recall. For the direction labeling, we also analyze the average number of direction misclassifications / errors per sequence, normalized by sequence length, calling this metric Err # - this is effectively how many of the directions in a single sequence are erroneously predicted. Human evaluators evaluate the ChatGPT responses for correctness while heuristics (when the answer appears in the first or last line) are used for SpartQA. We use three models for our experiments: the first two are ChatGPT 3.5 and 4 [17] and the third is Llama-7B [20] for SpartQA, chosen due to the quicker evaluation time for the size of the test dataset. We selected these models for their common use by the general public and quicker outputs.

4.3 Results

Our main results across the three tasks are given in Table 1, Table 2 and Table 3, and the main findings are as follows.

LLMs perform acceptable few-shot identification of directions As seen in Table 1, ChatGPT-3.5 and 4 succeed in achieving at least 50% classification rates, with better performance (Err. # <25) on shorter trajectories (len>5) and a neutral effect from the integer vs. floating point. We also see that parameter size and extensive training data likely play a huge role, with ChatGPT-4 hitting perfect classification for short trajectories and impressive performance (60%) for long trajectories (len>35).

Table 2: 3D Trajectory Labeling performance for ChatGPT-3.5 & 4 on CALVIN & CALVIN-Cleaned

		ChatGPT-3.5		ChatGPT-4	
Dataset	Method	F1 (↑)	Acc. (↑)	F1 (↑)	Acc. (↑)
CALVIN	Zero-shot	0.19	0.26	0.19	0.27
	In-context	0.17	0.33	**0.63**	**0.63**
	CoT	0.28	0.36	0.33	0.37
	SPP	**0.32**	**0.43**	0.55	0.60
CALVIN-Cleaned	Zero-shot	0.24	0.30	0.42	0.47
	In-context	0.16	0.34	0.62	0.67
	CoT	0.14	0.26	0.73	0.73
	SPP	**0.38**	**0.43**	**0.80**	**0.80**

The models struggle with Shape Labeling however, unsurprisingly from the lack of data in zero-shot evaluation.

LLMs demonstrate poorer capabilities on more complex 3D trajectories We find that, as seen in Table 2, LLMs achieve subpar performance when compared with the 2D scenario, especially on the raw data from the CALVIN benchmark, with the highest F1-score capped at 63%. In their current form, it is improbable that such LLMs can be used for robotic trajectory classification. A possible cause for the disparity between the CALVIN and CALVIN-Cleaned datasets could be the higher degree of irregularity in the CALVIN dataset, since as demonstrated in [15], such models excel in mimicking more repetitive patterns (e.g. sinusoidal graphs). Another factor could be that LLMs connect spatial patterns to pre-trained semantic concepts in the CALVIN-Cleaned dataset - for example, the cleaned "lifting" trajectory illustrates only a downward and upward motion in the z-dimension, matching a fundamental understanding perhaps baked in from pretraining, whereas the extraneous datapoints might muddle such comprehension.

CoT reveals a reduction in spatial reasoning performance Table 2 conveys the volatility in the performance of CoT, revealing either losses or marginal performance gains (mostly bounded at 11%) compared to In-context Learning (ICL), and even the gains are much lower than other tasks [21]. A potential reason for the diminishing returns in this application is that the CoT reasoning steps are fairly dependent on the examples chosen. We have qualitatively seen examples in which the model understands a single example in the context of its action (e.g. since an object is performing the action "lift", its z-coordinate decreases), but then witnesses a slight decrease in the z-coordinate of a "slide" trajectory due to noisiness and concludes that it belongs to the "lift" category.

5 Conclusion

We examined the performance of LLMs including ChatGPT 3.5, 4 and Llama 2 7B on a variety of spatial tasks, namely 2D direction and path labeling, 3D trajectory labeling and abstract relationship identification. We show that the selected models exhibit acceptable performance on 2D direction labeling but flounder to a greater deal on 3D trajectory labeling. We speculate on possible causes, settling on the likelihood that the irregularity of the trajectories makes classification more onerous. We also hypothesize that the brittleness of Chain-of-Thought prompting's reliance on specific examples influences its diminished yield in noisy scenarios. Finally, we propose a technique called Spatial Prefix-Prompting that first inquires a simple, related question in order to better answer more complex spatial queries. Our work could have implications in a multitude of other domains than just higher-dimensional numerical data, such as multi-variable financial trend forecasting or aggregate health data analysis. Future work includes evaluation on a larger robotic dataset, extension to other spatial tasks (e.g. segmenting trajectories), and assessment of other LLMs like PaLM 4. Overall, we establish that the domain of spatial reasoning, especially with regards to numerical data, is an underexplored realm ripe for more research.

References

M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022.

Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, Q. V. Do, Y. Xu, and P. Fung. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity, 2023.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020.

A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel. Palm: Scaling language modeling with pathways, 2022.

A. G. Cohn and J. Hernandez-Orallo. Dialectical language model evaluation: An initial appraisal of the commonsense spatial reasoning abilities of llms, 2023.

X. He, Z. Lin, Y. Gong, A.-L. Jin, H. Zhang, C. Lin, J. Jiao, S. M. Yiu, N. Duan, and W. Chen. Annollm: Making large language models to be better crowdsourced annotators, 2023.

Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan. 3d-llm: Injecting the 3d world into large language models, 2023.

H. Hu and D. Sadigh. Language instructed reinforcement learning for human-ai coordination, 2023.

W. Huang, P. Abbeel, D. Pathak, and I. Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents, 2022.

M. Kwon, S. M. Xie, K. Bullard, and D. Sadigh. Reward design with language models, 2023.

X. Liu, D. McDuff, G. Kovacs, I. Galatzer-Levy, J. Sunshine, J. Zhan, M.-Z. Poh, S. Liao, P. D. Achille, and S. Patel. Large language models are few-shot health learners, 2023.

A. Madaan and A. Yazdanbakhsh. Text and patterns: For effective chain of thought, it takes two to tango, 2022.

O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks, 2022.

S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work?, 2022.

S. Mirchandani, F. Xia, P. Florence, B. Ichter, D. Driess, M. G. Arenas, K. Rao, D. Sadigh, and A. Zeng. Large language models as general pattern machines, 2023.

R. Mirzaee, H. Rajaby Faghihi, Q. Ning, and P. Kordjamshidi. SPARTQA: A textual question answering benchmark for spatial reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4582–4598, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.364. URL https://aclanthology.org/2021.naacl-main.364.

OpenAI. Gpt-4 technical report, 2023.

Y. Razeghi, R. L. Logan IV, M. Gardner, and S. Singh. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 840–854, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.59. URL https://aclanthology.org/2022.findings-emnlp.59.

A. Takmaz, E. Fedele, R. W. Sumner, M. Pollefeys, F. Tombari, and F. Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation, 2023.

H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

S. M. Xie, A. Raghunathan, P. Liang, and T. Ma. An explanation of in-context learning as implicit bayesian inference, 2022.

Y. Xie, C. Yu, T. Zhu, J. Bai, Z. Gong, and H. Soh. Translating natural language to planning goals with large-language models, 2023.

R. Xu, X. Wang, T. Wang, Y. Chen, J. Pang, and D. Lin1. Pointllm: Empowering large language models to understand point clouds, 2023.

H. Xue and F. D. Salim. Promptcast: A new prompt-based learning paradigm for time series forecasting, 2023.

6 Appendix

6.0.1 Spatial Relationship Identification

In order to demonstrate the efficacy of the Spatial Prefix-Prompting mechanism beyond just 3D trajectory classification, we also run experiments with Llama-2-7B on the entire test set of the SpartQA dataset [16] (of size 510 instances), a textual QA benchmark for deeper spatial reasoning questions of four types: find relation (FR), find blocks (FB), choose object (CO) (see Fig. 1).

LLMs seem to enable knowledge transfer from simple to more complex tasks From Tables 2 and 3, we observe that Spatial Prefix-Prompting (SPP) often surpasses CoT and ICL, particularly on the CALVIN-Cleaned dataset and the "Find Relationship" (FR) and "Choose Object" (CO) questions in the SpartQA dataset [16]. This outcome hints that SPP might perhaps be better suited to scenarios in which the labels themselves hold morphological meaning, permitting the model to expand upon its pretrained knowledgebase (e.g. in the FR and CO questions, the labels refer to directional

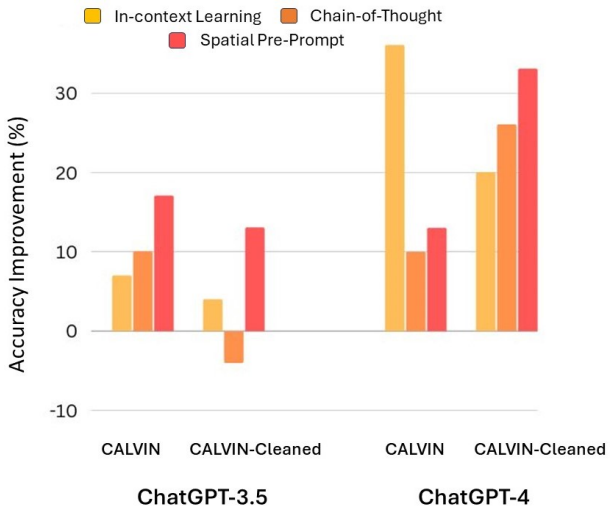

Figure 3: Accuracy Improvements (%) for In-context Learning, Chain-of-Thought and Spatial Prefix-Prompting on both the CALVIN and CALVIN-Cleaned datasets, for ChatGPT-3.5 and ChatGPT-4. As we can see, overall the accuracy gains for ChatGPT-4 are higher.

Table 3: Llama 2 7B performance on the SpartQA Test Dataset, split by subtypes FR, FB, CO, YN

		FR	FB	CO	YN	Overall
LLM	Method	Acc. (↑)	Acc. (↑)	Acc. (↑)	Acc. (↑)	Acc. (↑)
Llama 2 7B	Zero-shot	0.14	0.40	0.24	0.39	0.32
	CoT	0.21	**0.47**	0.16	**0.48**	0.36
	SPP	**0.42**	0.44	**0.42**	0.40	**0.41**

relationships like "above" or qualitative adjectives "medium black square"). Furthermore, it has previously been corroborated that, taking a Bayesian lens, ICL operates by helping the model to locate latent concepts that it learned during pretraining [22, 14, 18], i.e. if terms in a particular instance are exposed many times in the pretraining data, the model is likely to know better about the distribution of the inputs. It can be that SPP operates similarly, with a simple spatial question (e.g. direction identification) prodding the model to draw upon a more fundamental mechanism that it has been trained on (e.g. calculating numerical differences between coordinates to designate directions), in order to solve more complex questions that may use an analogous thought-process.