The Power of Bullet Lists: A Simple Yet Effective Prompting Approach to Enhancing Spatial Reasoning in Large Language Models

Ikhyun Cho[1] and **Changyeon Park**[2] and **Julia Hockenmaier**[1]
[1]University of Illinois at Urbana-Champaign [2]Seoul National University
ihcho2@illinois.edu blackco@snu.ac.kr juliahmr@illinois.edu

Abstract

While large language models (LLMs) are dominating the field of natural language processing, it remains an open question how well these models can perform spatial reasoning. Contrary to recent studies suggesting that LLMs struggle with spatial reasoning tasks, we demonstrate in this paper that a novel prompting technique, termed Patient Visualization of Thought (PATIENT-VOT), can boost LLMs' spatial reasoning abilities. The core idea behind PATIENT-VOT is to explicitly integrate *bullet lists, coordinates, and visualizations* into the reasoning process. By applying PATIENT-VOT, we achieve a significant boost in spatial reasoning performance compared to prior prompting techniques. We also show that integrating bullet lists into reasoning is effective in planning tasks, highlighting its general effectiveness across different applications.

1 Introduction

Large language models (LLMs) are massive neural networks trained on a vast and diverse range of corpora, that are currently leading the field of natural language processing (NLP) (Brown, 2020). Beyond their remarkable achievements in NLP, researchers are gradually focusing on broader goals, such as artificial general intelligence, where they envision the development of versatile, if not universal, AI assistants (Zheng et al., 2024a). In this context, LLMs play a pivotal role due to their strong reasoning capabilities, their characteristics as general pattern machines (Mirchandani et al., 2023), and their capacity to produce human-friendly explanations. However, spatial reasoning ability, one of the key requirements for these assistants, is known to be lacking in LLMs (Bang et al., 2023; Sharma, 2023). Multiple recent studies point out that even the top-performing LLMs, such as GPT4, struggle significantly with spatial reasoning tasks, including relatively simple grid-based tasks (Li et al., 2024; Yamada et al., 2023). Achieving satisfactory performance in these grid-based tasks is a necessary first step toward tackling more advanced spatial reasoning challenges.

Among various efforts to enhance LLMs' spatial reasoning abilities, a notable approach is *prompt engineering* (Bommasani et al., 2021), which aims to trigger and maximize the model's spatial reasoning capabilities by designing effective prompts. One major advantage of prompt engineering is that it does not require additional training or external resources, making it a cost-effective and generally applicable approach. While some recent studies have emerged in this field (Wu et al., 2024; Li et al., 2024; Yasunaga et al., 2023), we believe this area remains under-explored.

Our Objective and Approach In this paper, we aim to tackle the following research question from a prompt engineering perspective: *How can we effectively trigger and improve the spatial reasoning abilities of LLMs via prompting?*

To this end, we introduce Patient Visualization-of-Thought (PATIENT-VOT), a simple yet effective prompting technique designed to enhance the spatial reasoning abilities of LLMs. PATIENT-VOT builds on the Visualization-of-Thought approach (Wu et al., 2024) with two novel practical findings:

1. Patient Spatial Understanding (PSU) Summarizing information into a bullet list (done by LLMs) boosts their ability to understand information, leading to a substantial reduction in errors during subsequent reasoning steps.

2. Patient Spatial Reasoning (PSR) Combining visual-based and coordinate-based reasoning creates a synergistic effect, leading to a more effective spatial reasoning in LLMs.

In a nutshell, we find that the bullet list format is generally more *LLM-friendly* than the text-

Findings of the Association for Computational Linguistics:
NAACL 2025, pages 3047–3057

only format or other common structured formats (e.g., tables, JSON, and HTML) (see Section 3.1 and Appendix B). We then leverage this finding to address a common limitation in recent prompting techniques, such as Chain-of-Thought (Wei et al., 2022) and Visualization-of-Thought (Wu et al., 2024): their difficulty in handling information presented in a text-only format, as highlighted in recent studies (Golovneva et al., 2024). We empirically demonstrate that instructing the LLM to summarize information into a bullet list (i.e., PSU) significantly enhances its ability to understand information—not only in spatial reasoning tasks but also in planning tasks (see Section 5.1 and Appendix B). Furthermore, PSR activates an additional modality, *coordinate-based reasoning*, that enhances LLMs' spatial reasoning abilities when combined with visualization (see Table 4). PATIENT-VOT consistently boosts the performance of various models (GPT-4o, GPT-4o-mini, GPT-4-turbo, Claude-3.5-Sonnet, and Claude-3-Haiku) on a variety of challenging spatial reasoning and planning tasks.

2 Related Work

Spatial Reasoning in LLMs Several recent studies have examined the spatial reasoning capabilities of LLMs, consistently finding that LLMs continue to struggle with spatial reasoning tasks (Li et al., 2024; Bang et al., 2023). Existing research on spatial reasoning in LLMs can be broadly categorized into three approaches: (1) Analyzing LLM behavior to gain insights into their underlying mechanisms (Xie et al., 2023; Cohn and Hernandez-Orallo, 2023), (2) Augmenting spatial reasoning abilities by conducting additional training on curated datasets (Hong et al., 2023; Cheng et al., 2024), and (3) Improving spatial reasoning performance using effective prompting methods instead of further training (Wu et al., 2024; Sharma, 2023). We focus on the prompting approach, due to its cost-effectiveness, as explained in the Introduction.

Prompt Engineering Approaches to LLM Spatial Reasoning Recently, various prompting techniques have been introduced, such as chain-of-thought (Wei et al., 2022), self-consistency (Wang et al., 2022), tree-of-thought (Yao et al., 2024), and prompt template engineering (Cho et al., 2024; Shivagunde et al., 2024; Cho et al., 2023). However, these methods are primarily designed for general reasoning tasks. Given the unique challenges of spatial reasoning, some prompting techniques have been specifically tailored for this purpose (Wu et al., 2024; Sharma, 2023). Among those, visualization-of-thought (VoT) (Wu et al., 2024) has demonstrated promising results with a unified prompt. Our work aims to address a common weakness of these prompting techniques—their inability to effectively process large volumes of information when presented solely in text—by incorporating bullet lists.

3 PATIENT-VOT

3.1 Motivation

The goal of this paper is to discover a ***universal prompt*** that can effectively trigger and enhance spatial reasoning across various LLMs. Identifying such a prompt holds substantial practical value, as it allows LLM practitioners to improve task performance with a simple tweak to the input prompt.

With this motivation in mind, we present PATIENT-VOT, designed to unlock LLMs' latent spatial reasoning abilities through two novel ideas: (1) Patient spatial understanding, where LLMs are guided to first translate the information into a *bullet list* to enhance information understanding (also applicable to planning tasks); (2) Patient spatial reasoning, which activates two modalities, *visual and coordinate*, in LLMs to improve spatial reasoning performance. Detailed insights behind the use of bullet lists and coordinates are provided in Section 5.2 and Appendices B and C.

3.2 Patient Spatial Understanding (PSU)

Recent studies (Golovneva et al., 2024), as well as our preliminary experiments detailed in Appendix B, have shown that LLMs struggle with seemingly simple tasks, such as converting a natural language description of a grid into a visual representation (see Figure 1). To address this limitation, we propose a simple yet effective approach: First translating the provided information into a bullet list (done by the LLM) before converting it into a visualization. This method significantly reduces the error rate from 52% to 8% when visualizing the initial grid in the GPT-4o model. Additionally, we show that PSU is applicable to other reasoning tasks, such as planning, verifying its general efficacy (See Section 5.1). Specifically, we append the following sentence to the input: *"Before starting, convert the initial information into a detailed bullet list to effectively grasp the map's information."*

3.3 Patient Spatial Reasoning (PSR)

Visualizing the state has been shown to be effective for spatial reasoning in LLMs (Wu et al., 2024). We propose activating an additional modality: *Coordinate-based reasoning*. While LLMs may naturally engage in this type of reasoning, our observations indicate that explicitly prompting it is highly effective. Additionally, combining coordinate-based reasoning with visual-based reasoning results in a synergistic effect, leading to an additional increase in performance. Consequently, we add the following sentence into our final prompt: *"Solve the problem twice with the following approach: 'Visualize the state after each reasoning step'. In the first attempt, use coordinates instead of visualization. In the second attempt, use direct visualization and fix any errors in the first attempt."*

4 Experiments

4.1 Experimental Settings

Datasets We selected three spatial reasoning tasks from Wu et al. (2024) and three planning tasks from Zheng et al. (2024b). Specifically, the spatial reasoning tasks include Natural Language Navigation (NLN), Route Planning (RP), and Visual Tiling (VT), while the planning tasks consist of Trip Planning (TP), Calendar Scheduling (CS), and Meeting Planning (MP). Details about the datasets can be found in Appendix A.

Models and Settings We employ the GPT-4 model family, including GPT-4o, GPT-4o-mini, and GPT-4-turbo (Achiam et al., 2023) and the Claude family, including Claude-3.5-Sonnet and Claude-3-Haiku (Anthropic, 2024). For baseline prompts, we follow the approach from Wu et al. (2024), using "Let's think step by step." for the CoT baseline (Kojima et al., 2022) and "Visualize the state after each reasoning step." for the VoT baseline. Experiments are conducted using a basic greedy decoding scheme (i.e., temperature set to 0), with three different random seeds.

4.2 Results

Table 1 presents the performance of PATIENT-VOT and the baseline methods on the three spatial reasoning tasks. We observe that PATIENT-VOT significantly and consistently improves performance across all models and datasets, outperforming related prompting techniques by a substantial margin. Table 4 presents the results of several ablation studies. The top section highlights the impact of each component in PATIENT-VOT, where both PSU and PSR independently demonstrate consistent improvements, and their combination results in even greater performance gains. Notably, PSU, which involves incorporating bullet lists, proves generally effective for planning tasks as well, as described in Section 5.1 and Table 2. An additional error analysis is provided in Appendix D, offering insights into how PSU influences model outputs.

Model	Spatial Reasoning Tasks NLN	RP	VT
GPT-4o			
CoT	$8.50_{1.32}$	$7.27_{3.75}$	$28.67_{2.02}$
VoT	$26.17_{1.26}$	$5.15_{0.49}$	$29.00_{1.50}$
Ours: PATIENT-VOT	$\mathbf{83.83_{1.44}}$	$\mathbf{30.23_{0.37}}$	$\mathbf{36.33_{1.61}}$
GPT-4o-mini			
CoT	$2.67_{0.29}$	$5.15_{0.25}$	$17.33_{5.03}$
VoT	$22.17_{1.04}$	$5.80_{0.14}$	$17.67_{3.06}$
Ours: PATIENT-VOT	$\mathbf{61.00_{1.50}}$	$\mathbf{41.58_{1.48}}$	$\mathbf{24.00_{3.00}}$
GPT-4-turbo			
CoT	$21.50_{2.18}$	$5.56_{0.14}$	$21.00_{2.65}$
VoT	$25.67_{1.04}$	$3.43_{0.49}$	$19.00_{1.73}$
Ours: PATIENT-VOT	$\mathbf{51.67_{1.44}}$	$\mathbf{7.52_{0.93}}$	$\mathbf{24.33_{1.15}}$
Claude-3.5-Sonnet			
CoT	$37.67_{0.58}$	$9.90_{0.22}$	$15.17_{0.76}$
VoT	$50.67_{1.15}$	$10.00_{0.84}$	$22.33_{1.04}$
Ours: PATIENT-VOT	$\mathbf{85.67_{2.52}}$	$\mathbf{21.89_{1.74}}$	$\mathbf{24.83_{0.58}}$
Claude-3-Haiku *			
CoT	$0.00_{0.00}$	$0.70_{0.00}$	$16.00_{0.00}$
VoT	$0.00_{0.00}$	$0.49_{0.00}$	$15.00_{0.00}$
Ours: PATIENT-VOT	$\mathbf{5.00_{0.00}}$	$\mathbf{1.72_{0.00}}$	$\mathbf{25.00_{0.00}}$

Table 1: Effectiveness of PATIENT-VOT. Reported numbers are average and standard errors of three runs. We can observe that PATIENT-VOT consistently outperforms existing prompting baselines with a huge margin.

5 Main Findings

5.1 The Power of Bullet Lists

In spatial reasoning tasks As mentioned in Section 3.2, even the most advanced GPT-4 models make significant errors in translating descriptions into accurate grids (See Figure 1). This aligns with recent research showing that LLMs often struggle with simple tasks involving counting or retracing steps (Golovneva et al., 2024). We believe that converting the description into a structured format, such as a bullet list with clear delimiters, and then

*Note that Claude-3-Haiku does not have a seed parameter, resulting in same outputs across runs.

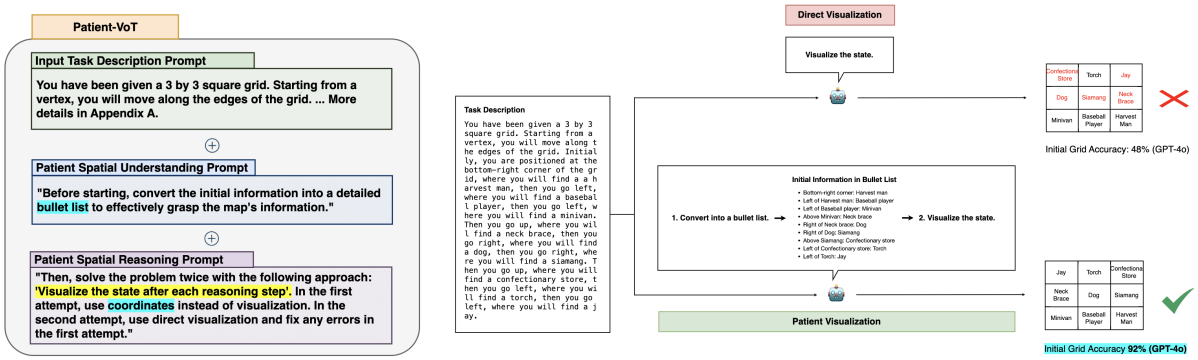

Figure 1: (Left) The overall template of PATIENT-VOT. Key trigger words, "bullet list" and "coordinates", are marked in blue, while the VoT prompt element is highlighted in yellow. (Right) The intuition behind patient spacial understanding. The structured bullet list significantly reduces mistakes when creating the initial visualization.

using this structured format for visualization, helps minimize mistakes. Quantitatively, this approach reduces the error rate of the initial grid in the natural language navigation task from 52% to 8% for GPT-4o, and from 65% to 13% for GPT-4o-mini.

In planning tasks Incorporating bullet lists into the reasoning process is not limited to spatial reasoning tasks. Therefore, we extend our experiments to another type of task, selecting *planning* tasks (Zheng et al., 2024b) due to their broad applicability.

For planning tasks, we compare (PSU+CoT) with the CoT baseline in Table 2, as incorporating coordinates (i.e., VoT and PSR) is unsuitable for these tasks and showed no performance improvement. Due to the strict rate limits of Claude-3.5-Sonnet, we excluded it from the table. Given the large volume of the datasets, we conducted the experiments using a single fixed seed of 42. The results in Table 2 demonstrate that PSU is also highly effective for planning tasks, indicating its general efficacy across different task types.

Model	Plannng Tasks CS	TP	MP
GPT-4o			
CoT	48.40	4.00	46.50
Ours: PSU+CoT	**60.40**	**5.19**	**49.40**
GPT-4o-mini			
CoT	29.40	5.56	19.60
Ours: PSU+CoT	**34.70**	**8.37**	**26.30**
GPT-4-Turbo			
CoT	45.10	29.69	36.50
Ours: PSU+CoT	**48.90**	**32.50**	**38.30**
Claude-3-Haiku			
CoT	26.50	20.94	19.50
Ours: PSU+CoT	**30.00**	**22.75**	**23.30**

Table 2: Effectiveness of bullet lists in planning tasks. We can see that integrating bullet lists is consistently effective in planning tasks as well.

5.2 Bullet list is significantly better than other structures

A fundamental question regarding PATIENT-VOT is: "Do we need to use bullet lists, or can similar performance gains be achieved with other common structures such as tables, JSON, or HTML?" To explore this, we conducted experiments to determine whether the performance improvements of PATIENT-VOT are specifically attributable to bullet lists or if they can also be obtained using alternative formats. We evaluated three formats—table, HTML, and JSON—by testing three prompts for each structure on two models (GPT-4o and GPT-4o-mini) using the natural language navigation task (detailed prompts are provided in Appendix F). The results, summarized in Table 3, indicate that although these alternative structures do offer some improvement over the baseline (i.e., the original CoT/VoT), their performance gains are significantly smaller than those achieved with bullet lists—particularly for the smaller GPT-4o-mini model. This underscores the superior effectiveness of bullet lists compared to other structures. We speculate that their advantage lies in (1) their prevalence in pretraining corpora and (2) their relative simplicity, which makes them especially LLM-friendly, particularly for smaller models.

Task: NLN Model	Prompt Variants	PATIENT-VOT Acc(%)	VoT Acc(%)	CoT Acc(%)
GPT-4o	Baseline (No structures)	31.00	26.00	8.50
	Bullet lists	**83.00**	**53.00**	**34.50**
	1. Tables (Prompt 1)	33.00	20.00	18.00
	1. Tables (Prompt 2)	33.00	22.50	17.00
	1. Tables (Prompt 3)	40.50	33.00	29.50
	2. HTML (Prompt 1)	30.50	28.50	27.00
	2. HTML (Prompt 2)	57.50	36.50	23.50
	2. HTML (Prompt 3)	60.00	42.50	13.50
	3. JSON (Prompt 1)	29.50	29.00	23.00
	3. JSON (Prompt 2)	62.50	49.50	33.50
	3. JSON (Prompt 3)	25.00	23.50	24.50
GPT-4o-mini	Baseline (No structures)	25.50	22.00	3.00
	Bullet lists	**61.00**	**37.50**	**44.50**
	1. Tables (Prompt 1)	23.00	18.50	13.00
	1. Tables (Prompt 2)	21.00	22.50	15.50
	1. Tables (Prompt 3)	4.00	5.50	5.00
	2. HTML (Prompt 1)	4.00	5.00	2.00
	2. HTML (Prompt 2)	15.00	8.50	2.50
	2. HTML (Prompt 3)	15.50	8.00	2.50
	3. JSON (Prompt 1)	10.00	3.50	2.00
	3. JSON (Prompt 2)	19.00	11.50	12.00
	3. JSON (Prompt 3)	6.50	4.00	4.50

Table 3: Bullet lists are key to PATIENT-VOT. Other widely used structures, such as tables, HTML, and JSON, perform significantly worse compared to bullet lists.

5.3 Coordinate-based reasoning and visual-based reasoning create synergy

Intuitively, LLMs can inherently use coordinates when dealing with spatial reasoning tasks. However, our findings show that explicitly prompting the LLM to employ coordinates is far from redundant. In fact, it proves effective on its own and also creates a synergistic effect when combined with visual-based reasoning. The empirical evidence supporting this claim is summarized in the bottom section of Table 4.

Specifically, we compared the following three variants: (1) PATIENT-VOT: which incorporates both coordinates and visualizations, (2) PSR (Visualization-Only): which uses only visualizations, and (3) PSR (Coordinate-Only): which relies solely on coordinates. The specific prompts for each variant are detailed in Appendix E. The results in Table 4 indicate that explicitly instructing GPT to perform coordinate-based reasoning is generally more effective than relying solely on visualizations. Most importantly, combining coordinate-based and visual-based reasoning yields even better performance than using either method alone.

6 Conclusion

This paper introduces a novel prompting technique, PATIENT-VOT, designed to enhance the spatial reasoning capabilities of LLMs. PATIENT-VOT incorporates two straightforward yet powerful concepts: patient spatial understanding and patient spatial reasoning. It demonstrates effectiveness across multiple GPT-4 and Claude-3 models on three spatial tasks and three planning tasks, achieving up to a 35% absolute improvement.

Ablation #1. Effectiveness of PSU and PSR.

Baseline: GPT-4o	NLN	RP	VT
• VoT	$26.17_{1.26}$	$5.15_{0.49}$	$29.00_{1.50}$
• VoT + PSU	$48.83_{2.57}$	$21.73_{0.57}$	$34.88_{2.21}$
• VoT + PSR	$31.33_{0.76}$	$12.17_{0.75}$	$34.33_{1.26}$
• VoT + PSU + PSR (=PATIENT-VOT)	$83.83_{1.44}$	$30.23_{0.37}$	$36.33_{1.61}$

Ablation #2. The synergy between coordinate-based and visual-based reasonings.

Baseline: GPT-4o	NLN	RP	VT
• PSR (Coordinate-Only)	$80.50_{2.65}$	$26.06_{0.51}$	$35.33_{0.76}$
• PSR (Visualization-Only)	$48.83_{2.57}$	$21.73_{0.57}$	$34.88_{2.21}$
• PSR (Both) (=PATIENT-VOT)	$83.83_{1.44}$	$30.23_{0.37}$	$36.33_{1.61}$

Table 4: A summary of two ablation study results. The upper half illustrates the individual effectiveness of each component and their combined impact, while the bottom half highlights the synergy between coordinates and visualizations.

7 Limitations

Our work has a few limitations. Firstly, our study lies in the area of "prompt engineering" which may lack strong theoretical justification for why our approach is effective. Additionally, we concentrated on greedy decoding for computational efficiency. Nevertheless, exploring the integration of PATIENT-VOT with sampling-based prompting techniques remains a promising area for future research.

References

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

Anthropic. 2024. Claude. *Anthropic Artificial Intelligence*. Large language model. Available: https://www.anthropic.com.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint ArXiv:2005.14165*.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. 2024. Spatialrgpt: Grounded spatial reasoning in vision language model. *arXiv preprint arXiv:2406.01584*.

Ikhyun Cho, Yoonhwa Jung, and Julia Hockenmaier. 2023. Sir-absc: Incorporating syntax into roberta-based sentiment analysis models with a special aggregator token. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Ikhyun Cho, Gaeul Kwon, and Julia Hockenmaier. 2024. Tutor-icl: Guiding large language models for improved in-context learning performance. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9496–9506.

Anthony G Cohn and Jose Hernandez-Orallo. 2023. Dialectical language model evaluation: An initial appraisal of the commonsense spatial reasoning abilities of llms. *arXiv preprint arXiv:2304.11164*.

Olga Golovneva, Tianlu Wang, Jason Weston, and Sainbayar Sukhbaatar. 2024. Contextual position encoding: Learning to count what's important. *arXiv preprint arXiv:2405.18719*.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 2023. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Fangjun Li, David C Hogg, and Anthony G Cohn. 2024. Advancing spatial reasoning in large language models: An in-depth evaluation and enhancement using the stepgame benchmark. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18500–18507.

Suvir Mirchandani, Fei Xia, Pete Florence, Brian Ichter, Danny Driess, Montserrat Gonzalez Arenas, Kanishka Rao, Dorsa Sadigh, and Andy Zeng. 2023. Large language models as general pattern machines. *arXiv preprint arXiv:2307.04721*.

Manasi Sharma. 2023. Exploring and improving the spatial reasoning abilities of large language models. In *I Can't Believe It's Not Better Workshop: Failure Modes in the Age of Foundation Models*.

Namrata Shivagunde, Vladislav Lialin, Sherin Muckatira, and Anna Rumshisky. 2024. Deconstructing in-context learning: Understanding prompts via corruption. *arXiv preprint arXiv:2404.02054*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. 2024. Visualization-of-thought elicits spatial reasoning in large language models. *arXiv preprint arXiv:2404.03622*.

Yaqi Xie, Chen Yu, Tongyao Zhu, Jinbin Bai, Ze Gong, and Harold Soh. 2023. Translating natural language to planning goals with large-language models. *arXiv preprint arXiv:2302.05128*.

Yutaro Yamada, Yihan Bao, Andrew K Lampinen, Jungo Kasai, and Ilker Yildirim. 2023. Evaluating spatial understanding of large language models. *arXiv preprint arXiv:2310.14540*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H Chi, and Denny Zhou. 2023. Large language models as analogical reasoners. *arXiv preprint arXiv:2310.01714*.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024a. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*.

Huaixiu Steven Zheng, Swaroop Mishra, Hugh Zhang, Xinyun Chen, Minmin Chen, Azade Nova, Le Hou, Heng-Tze Cheng, Quoc V Le, Ed H Chi, et al. 2024b. Natural plan: Benchmarking llms on natural language planning. *arXiv preprint arXiv:2406.04520*.

A Tasks and Datasets

We use three spatial reasoning tasks from Wu et al. (2024) and three planning tasks from (Zheng et al., 2024b). Specifically, spatial reasoning tasks include (1) *Natural Language Navigation (200)*, which involves visualizing a grid and tracking sequential movements within it; (2) *Route Planning (408)*, where the model must generate multi-hop navigation instructions on a 2D grid; (3) *Visual Tiling (200)*, which requires fitting appropriate tetrominoes into a square grid, similar to the game Tetris. The planning tasks include (1) *Calendar Scheduling (1000)*, which requires the model to identify possible meeting times based

on the constraints of each participant; (2) *Trip planning (1600)*, which involves generating an itinerary for visiting European cities, considering flight availability and constraints; (3) *Meeting planning (1000)*, where the model must create a plan that satisfies various constraints, including person-specific and location-specific requirements. The numbers in parantheses are the total number of instances in each dataset.

Since the original paper (Wu et al., 2024) does not provide the original datasets, we re-implemented them by following the dataset generation algorithms outlined in the paper. We made a slight modification to the Natural Language Navigation task to make the evaluation more accurate: instead of evaluating correctness based only on the final item, as done in the original paper, we evaluate based on the accuracy of all items encountered during the moves. That is, we consider the model's output correct only if all eight items during the moves are accurately identified, making this a much stricter and more precise evaluation criterion. For the planning tasks, we use the official code and metrics provided by the authors (Zheng et al., 2024b).

Examples of each task are provided below and we recommend reading the original papers (Wu et al., 2024; Zheng et al., 2024b) for full details.

Natural Language Navigation Example "You have been given a 3 by 3 square grid. Starting from a vertex, you will move along the edges of the grid. Initially, you are positioned at the bottom-left corner of the grid, where you will find a wool, then you go right, where you will find a football player, then you go right, where you will find a black-and-white colobus. Then you go up, where you will find a pot pie, then you go left, where you will find a torch, then you go left, where you will find a minivan. Then you go up, where you will find a conch, then you go right, where you will find an american dipper, then you go right, where you will find a jay.

Now you have all the information on the map. The given map is a 3 by 3 map. You start at the position where the wool is located, then you go right by one step, then you go right by one step, then you go left by one step, then you go up by one step, then you go left by one step, then you go up by one step, and then you go right by one step. For your final answer, list all eight items encountered during the moves (including the starting item and any duplicates) under the title 'Final List of Items Encountered' as a bullet list."

Route Planning Example Provided in Figure 2 below.

Visual Tiling Example Provided in Figure 3 below.

Planning Tasks We recommend referring to the original paper (Zheng et al., 2024b) for detailed information, as we have directly used their published datasets without modification.

B Motivating Experiments and Core Insights

To provide clear insights into why the idea of *integrating bullet list into reasoning* works, we summarize the core insights and present two additional experiments as motivating evidence:

Core Insights and Corresponding Evidence

Insight #1. LLMs Struggle with Text-Only Inputs: Despite employing CoT or VoT, LLMs still struggle to process information provided in text-only format, as evidenced by the poor performances when using CoT or VoT, shown in Table 1 of the paper.

Insight #2. Bullet Lists Enhance Information Understanding in LLMs: We claim that the bullet list format is a more *LLM-friendly* representation than text-only, enabling LLMs to process information more accurately. We verify this claim by conducting two experiments provided below.

Insight #3. Strong Baselines (CoT/VoT) Benefit from Bullet Lists: As a result, we believe that explicitly integrating bullet lists can complement CoT or VoT, leading to improved performance. This is supported by the performance boost observed when comparing "Baseline (CoT/VoT)" with "Bullet List + Baseline (CoT/VoT)" (as presented in Table 1 and Table 2).

Evidence for Insight #2 (Bullet Lists Enhance Information Understanding in LLMs): In addition to the downstream performance improvements, and to ensure a more comprehensive and fundamental evaluation, we conducted two experiments in simpler, distinct contexts: (1) Accurate Information Retrieval and (2) Accurate Counting.

Experiment 1 (Accurate Information Retrieval): We hypothesize that the bullet list format allows

LLMs to more easily locate and retrieve information with greater accuracy. To test this, we conducted a simple experiment where the task is to accurately retrieve specific information. We present LLMs with a sequence of N fruits and ask them to identify the M-th fruit. Despite the simplicity of this task, it is known that such tasks often pose challenges for even the most recent and advanced LLMs (Golovneva et al., 2024).

N := Number of total fruits. We used 5, 10, 15, 20.
M := Specific location of the fruit. We used 1,2,...,N. 30 random examples for each value of M.
Example prompt (text-only form): "You have the following fruits in a sequence: pineapple, mango, lychee, kiwi, persimmon, fig, peach, mulberry, strawberry, raspberry, apricot, passion fruit, orange, grapes, plum, watermelon, dragon fruit, papaya, blueberry, and apple. What is the eleventh fruit?"
Example prompt (bullet-list form): "'You have the following fruits in a sequence:\n- pineapple\n- mango\n- lychee\n- kiwi\n- persimmon\n- fig\n- peach\n- mulberry\n- strawberry\n- raspberry\n- apricot\n- passion fruit\n- orange\n- grapes\n- plum\n- watermelon\n- dragon fruit\n- papaya\n- blueberry\n- apple\n- What is the eleventh fruit?"

The experiment results are summarized in Table 5. We can observe that providing information in a bullet list format allows the LLMs to more accurately locate and retrieve relevant information.

	Accurate Information Retrieval			
Model: GPT-4o	$N = 5$	$N = 10$	$N = 15$	$N = 20$
Text-Only	100.00	99.67	92.44	85.00
Bullet-List	**100.00**	**100.00**	**96.22**	**91.67**
Model: GPT-4o-mini	$N = 5$	$N = 10$	$N = 15$	$N = 20$
Text-Only	100.00	81.00	69.78	66.83
Bullet-List	**100.00**	**85.00**	**79.78**	**72.50**

Table 5: Accurate information retrieval results on GPT-4o and GPT-4o-mini. It is evident that bullet list format is more LLM-friendly than text-only format.

Experiment 2 (Accurate Counting): We conducted an additional experiment, which is a slightly more challenging variant of the previous one. Specifically, we asked the LLM to count the number of a specific fruit, given a sequence of fruits. This allows us to further assess whether the bullet list format facilitates better manipulation of the provided information.

N := number of total fruits. We used N = 20.
M := number of counts of a specific fruit. We used 1,2,...,20, 30 random examples for each value of M.
Example Prompt (text-only form): "You have the following fruits in a sequence: raspberry, peach, blueberry, plum, cherry, apricot, pineapple, fig, watermelon, banana, strawberry, apple, papaya, plum, kiwi, passion fruit, mango, plum, guava, and orange. How many plums are there?"
Example Prompt (bullet-list form): "You have the following fruits in a sequence:\n- raspberry\n- peach\n- blueberry\n- plum\n- cherry\n- apricot\n- pineapple\n- fig\n- watermelon\n- banana\n- strawberry\n- apple\n- papaya\n- plum\n- kiwi\n- passion fruit\n- mango\n- plum\n- guava\n- orange\n- How many plums are there?"

	Accurate Counting	
Model	Text-Only	Bullet-List
GPT-4o	72	**80.33**
GPT-4o-mini	46.8	**50.17**

Table 6: Accurate counting results on GPT-4o and GPT-4o-mini. It is evident that bullet list format is more LLM-friendly than text-only format.

The experiment results are summarized in Table 6. We can observe that providing information in a bullet list format allows the LLMs to more accurately process the information.

Conclusion: We believe these two experiments support our claim that bullet lists could be more LLM-friendly than text-only formats.

C Why bullet lists and coordinates?

Given the performance improvements from incorporating bullet lists, a natural question arises: "What about other structures?" To explore this, we experimented with various common structures, including tables, JSON, and HTML, and evaluated their performance on the natural language navigation task using GPT-4o. We present the results in Table 7.

We observe that bullet lists are significantly more effective than other structures; only the bullet list format improved performance, while the other structures resulted in a significant decline.

We speculate that this effectiveness is due to their widespread inclusion in training data, as well as their simplicity, which makes them easier for models to learn and process.

Model: GPT-4o	NLN
Ablation #1. Effectiveness of Bullet Lists	
VoT	21.00
VoT + PSU (using bullet lists)	**46.00**
VoT + PSR (using tables)	7.00
VoT + PSR (using JSON)	4.50
VoT + PSR (using HTML)	0.50
Ablation #2. Effectiveness of Coordinates	
PATIENT-VOT	**61.00**
PATIENT-VOT with cardinal directions	39.00
PATIENT-VOT with no specific instructions	36.00

Table 7: Results of replacing bullet lists with other structures (i.e., tables, JSON format, and HTML) and of substituting coordinates with alternative candidates.

Next, to justify the use of *coordinates*, we conducted additional ablation studies in which we replaced "coordinates" with other alternatives in our final Patient-VoT prompt. We tested the use of "cardinal directions" instead of "coordinates", as well as a case in which no specific instruction was provided, to assess the effectiveness of coordinates under the fair compute-matched settings. The results are summarized in Table 7. We observe that coordinates are much more effective than other alternatives.

D Error Analysis

We have conducted an error analysis, to offer insights into how integrating bullet list affects the model's outcome. We used GPT-4o-mini and the Natural Language Navigation task as representatives. We compared PSU with VoT using 100 random samples.

Finding #1. Bullet lists significantly improve information understanding (i.e., getting correct initial grid representation): As briefly mentioned in Section 5.1 of our paper, the initial grid accuracy increases dramatically when GPT-4o converts text-only information into a bullet list as an intermediate step towards visualization. Likewise, the initial grid accuracy for GPT-4o-mini improves from 35% to 87% as shown below. Furthermore, a detailed analysis reveals that VoT produced 10 instances where question marks ("?") were placed in the grid, meaning the LLM does not know what to place, while Patient-VoT had no such cases. These numbers (87% vs. 35%) and the qualitative differences in output demonstrate the effectiveness of bullet lists in enhancing information understanding.

Finding #2. Coordinate-based reasoning enhances the reasoning process: Additionally, among cases with correct initial grid representations, the ratio that resulted in correct final answers was 60% (21/35) for VoT and 70.1% (61/87) for Patient-VoT, indicating that incorporating coordinate-based reasoning aids the reasoning process. We provide further details on different types of errors the model made across 100 samples in Table 8 and Table 9 below:

Model: GPT-4o-mini + **VoT**	Counts
Wrong initial grid representations • 10 out of 65 initial grid representations contain one or more "?" marks, whereas PATIENT-VOT has none	65
Correct initial grid representations, but wrong afterwards:	
• Omitting one move (out of 7 move directives)	3
• Wrong item recall during moves	2
• Moving in wrong directions	5
• Wrong coordinate calculation during moves (Note that coordinates are often naturally evoked when using VoT)	4

Table 8: Error analysis results of VoT.

Model: GPT-4o-mini + **PSU**	Counts
Wrong initial grid representations	13
Correct initial grid representations, but wrong afterwards:	
• Omitting one move	5
• Moving one additional move than directed	1
• Wrong item recall during moves	3
• Using different (x,y)-coordinate axes for understanding and reasoning, resulting in wrong final answer:	11
• Moving in wrong directions	2
• Wrong coordinate calculation during moves	4

Table 9: Error analysis results of PATIENT-VOT.

E Prompt Templates Used in Ablation Study #2

Variant 1: (=PATIENT-VOT): "Before starting, convert the initial information into a detailed bullet list to effectively grasp the map's information. Then, solve the problem twice with the following approach: 'Visualize the state after each reasoning step'. In the first attempt, use coordinates instead of visualization. In the second attempt, use direct visualization and fix any errors in the first attempt."
Variant 2: PSR (Visualization-Only): "Before starting, convert the initial information into a

detailed bullet list to effectively grasp the map's information. Then, solve the problem with the following approach: 'Visualize the state after each reasoning step'."
Variant 3: PSR (Coordinate-Only): "Before starting, convert the initial information into a detailed bullet list to effectively grasp the map's information. Then, solve the problem with the following approach: 'Visualize the state after each reasoning step'. Use coordinates instead of visualization."

F Prompt Templates Used in Section 5.2

Table prompts

Prompt 1: "Before starting, convert the initial information into a 3 by 3 table to effectively grasp the map's information."
Prompt 2: "Before starting, convert the initial information into a table where each column and row corresponds to a key aspect, to effectively grasp the map's information."
Prompt 3: "Before starting, convert the initial information into a table with appropriate column names to effectively grasp the map's information."

HTML prompts

Prompt 1: "Before starting, convert the initial information into an HTML format using appropriate semantic tags to effectively grasp the map's information."
Prompt 2: "Before starting, convert the initial information into an HTML format using the <ul> or <ol> tags to effectively grasp the map's information."
Prompt 3: "Before starting, convert the initial information into an HTML format using the list tag to effectively grasp the map's information."

JSON prompts

Prompt 1: "Before starting, convert the initial information into a JSON format, with keys representing the main aspects and values containing the detailed information, to effectively grasp the map's information."
Prompt 2: "Before starting, convert the initial information into a JSON format with key as the summary and value as the detailed information to effectively grasp the map's information."
Prompt 3: "Before starting, convert the initial information into a JSON format using descriptive keys for key elements and associating each with detailed information as values to effectively grasp the map's information."

Navigation Task: for a provided map, 🏠 is the home as starting point, 🏢 is the office as the destination. ⬜ means the road, 🚧 means the obstacle. There exists one and only one viable route for each map. Each step you choose a direction and move to the end of the continuous road or the destination.

map:

Starting from 🏠, provide the steps to navigate to 🏢.

Figure 2: Route planning example.

Task: given a set of polyominoes and corresponding variations of each polyomino, fit them into the empty squares (⬜) in the target rectangle without overlapping any existing polyominoes or going outside the rectangle. The variations allow only translation, not rotation or reflection.

Target rectangle with 8 empty squares:

Provided polyominoes:

1. Tetromino L (🟪)
2. Tetromino I (🟨)

Variations for Tetromino L:

Variation 1 fitting into its bounding box:

Variation 2 fitting into its bounding box:

Variants for Tetromino I:

Variation 3 fitting into its bounding box:

Variation 4 fitting into its bounding box:

To fit all the provided polyominoes into the empty squares (⬜), what's the correct variations of Tetromino L and I, respectively?

A. 1 and 3
B. 1 and 4
C. 2 and 3
D. 2 and 4
E. neither

Figure 3: Visual tiling example.