

CHAIN-OF-SYMBOL PROMPTING FOR SPATIAL RELATIONSHIPS IN LARGE LANGUAGE MODELS

Anonymous authors

Paper under double-blind review

ABSTRACT

While conventional Chain-of-Thought prompting shows promising performance on various language tasks for LLMs, the spatial scenarios are nearly unexplored. In this paper, we first investigate the performance of LLMs on complex spatial understanding and planning tasks that require LLMs to understand a virtual spatial environment simulated via natural language and act or reason correspondingly in text. By evaluating on classic spatial planning scenarios, we found that current popular LLMs such as ChatGPT still lack abilities to handle spatial relationships in texts. This raises a question: Is the natural language the best way to represent complex spatial environments for LLMs, or are other alternatives such as symbolic representations more efficient and effective for LLMs? To this end, we propose a novel method called COS (Chain-of-Symbol Prompting) that represents the spatial relationships with condensed symbols during the chained intermediate thinking steps. COS is easy to use and does not need additional training on LLMs. Extensive experiments indicate that COS clearly surpasses the performance of the Chain-of-Thought (CoT) Prompting described in natural language in all three spatial planning tasks and existing spatial QA benchmark, with even fewer tokens used in the inputs compared with CoT. The performance gain is strong, by up to 60.8% accuracy (from 31.8% to 92.6%) on Brick World for ChatGPT. CoS also reduces the number of tokens in the prompt obviously, by up to 65.8% of the tokens (from 407 to 139) for the intermediate steps from demonstrations on Brick World.

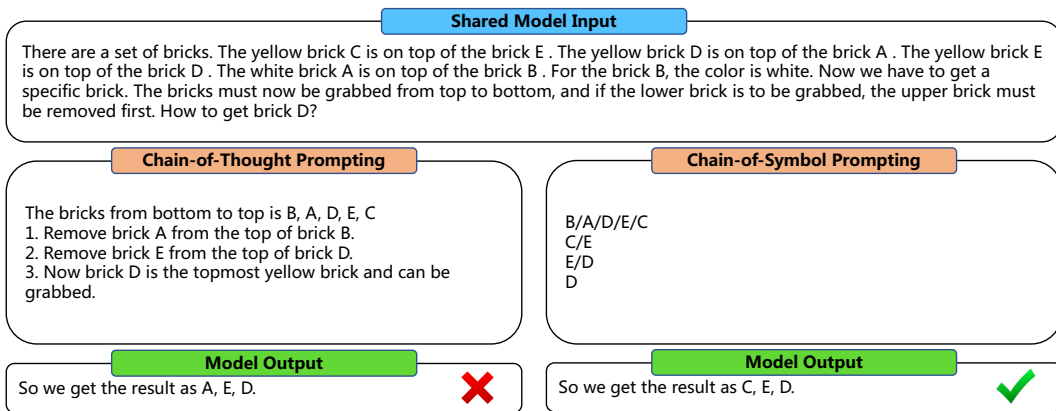


Figure 1: An example for comparison between Chain-of-Thought (CoT) and Chain-of-Symbol (CoS) that elicits large language models in tackling complex planning tasks with higher performance and fewer input tokens. We let the model generate CoT/CoS during inference in a few-shot manner. Results were taken in May 2023 with ChatGPT and can be subject to change.

1 INTRODUCTION

Given a set of target behaviour examples, large language models (LLMs) demonstrate exceptional abilities to accomplish a wide range of tasks, frequently exhibiting performance that surpasses that of

humans (Brown et al., 2020; Srivastava et al., 2022). Specifically, LLMs exhibit impressive sequential textual reasoning ability during inference, resulting in a significant boost in their performance when encountered with reasoning questions described in natural languages (Nye et al., 2021; Wei et al., 2022). This phenomenon can be clearly observed with a multi-step chain of intermediate thinking procedure, i.e., a "Chain of Thought" (CoT, Wei et al. 2022).

Conventional CoT usually leverages **natural languages** as intermediate thinking steps in prompting. Although CoT can enhance LLMs' ability in many cases, redundant natural languages and irrelevant information also can hamper the performance of LLMs (Shi et al., 2023) in some cases. For example, spatial languages and descriptions can be hard for language models to understand Mirzaee et al. 2021; Mirzaee and Kordjamshidi 2022 due to complex spatial relationships. Aligning symbols and representing spatial relationships by symbols in word sequences can be a neater representation and thus can be potentially easier to understand by LLMs. We thus explore the use of **symbols** for LLM prompting, which is still an understudied topic. This is important to study which implies understanding abilities beyond language models for language understanding per se.

To explore the role of symbolic representations in prompting, we take the complex spatial understanding and planning as the evaluation scenarios, which require LLMs to understand the virtual spatial environments described through natural language as well as planning and achieving certain goals in such environments. Inspired by existing classic planning competitions and spatial reasoning datasets, we present three domains: (i) Brick World (ii) NLVR-based Manipulation and (iii) Natural Language Navigation. Figure 1 illustrates an example for Brick World 1D, and all these three tasks are described in detail in Section 2.1. These three tasks are all described in natural language. And we also evaluate one existing spatial question answering dataset SPARTUN (Mirzaee and Kordjamshidi, 2022) which uses human-generated questions thus closer to realistic situations. For these tasks, LLMs need to understand a virtual environment in natural language, with the spatial relationship between the objects to be operated on and the restrictions on the operation, which is easy for real humans. However, we found that there are still places for improvement in the performance of LLMs on the tasks.

As a major contribution to this study, we investigate the symbolic representations for spatial relationships, and propose a novel method called **Chain-of-Symbol (CoS)** prompting to elicit spatial understanding and planning abilities on LLMs. As in Figure 1, instead of using intermediate thinking steps described in **natural language** in CoT prompts shown on the left-hand side, the CoS prompts remove the redundant text description but only using a set of **symbols** to represent spatial relationships between objects in complex environments. CoS achieves noticeable improvement in both **performance** and **efficiency** (by up to 60.8% improvements in accuracy and 65.8% for the number of input tokens). We speculate that such an improvement is benefited by the more efficient symbolic representation produced by CoS. Our main contributions are three-fold:

- We evaluate LLMs on both existing classic spatial understanding tasks and our proposed synthetic spatial planning tasks. We spot that there is still room for performance improvements on current LLMs even with CoT.
- We propose a novel method called CoS, which prompts LLMs to convert the complex environment described with natural language into condensed symbolic representations. CoS drastically improves LLMs on the spatial tasks. The accuracy gain of CoS is large, also with a good reduction in the token consumption for LLMs.
- We conduct an in-depth analysis on CoS to explore the effect of using different symbols, on different LLMs, and different languages to show the robustness of our method.

2 SPATIAL PLANNING AND UNDERSTANDING TASKS

2.1 NATURAL LANGUAGE SPATIAL PLANNING

Inspired by classic planning domains and tasks described in Liu et al. (2023) and existing spatial reasoning dataset Suhr et al. (2017), we explore the performance of LLMs in three natural language spatial planning tasks. For all three tasks, we can formulate the problem as given a virtual scenario described by natural language, and a planning question. LLMs should take both the scenario and the question as the input and output correspondingly to solve the question. Such a solution usually

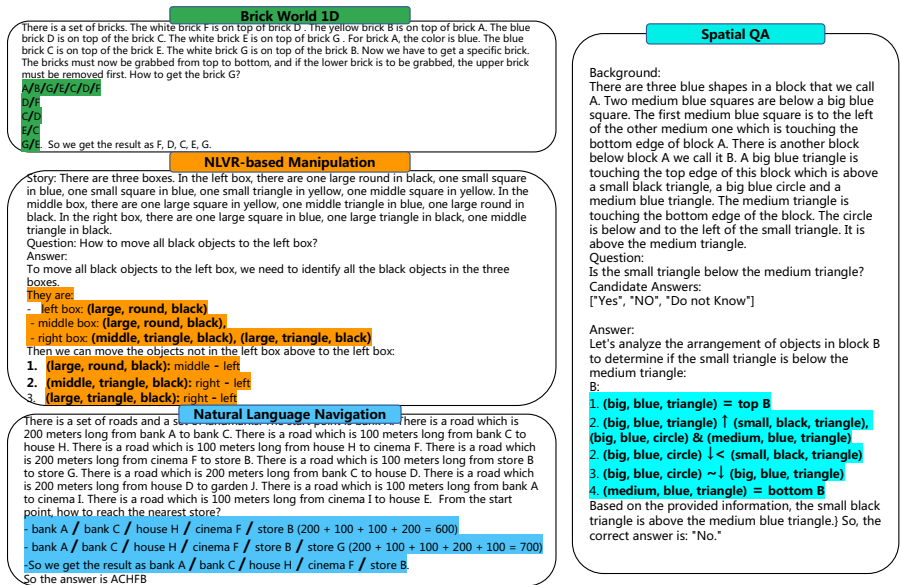


Figure 2: <input, Chain of Symbol, output> example triples for our three proposed tasks: Brick World, NLVR-based Manipulation, and Natural Language Navigation, and SPARTUN dataset (Mirzaee and Kordjamshidi, 2022). Chains of Symbols are highlighted.

contains a series of steps to achieve a final goal. The final test tasks consist of 5,500 evaluation instances, with 4,000 from Brick World, 1,000 from NLVR-based Manipulation, and the remaining 500 from Natural Language Navigation. We use code to generate these instances based on definition of each task.

2.2 BRICK WORLD

Figure 2 demonstrates an instance for Brick World (top), which requires the LLMs to acquire certain bricks by grabbing the bricks sequentially. We explore 1D and 2D scenarios for the Brick Worlds task. Specifically, in the 1D scenario, the relationship between bricks is only vertical. In the 2D scenario, in addition to the vertical relationship, there is also a horizontal relationship, which we express as "in the front of". To explore the characteristics of language understanding from LLMs, we investigate different levels of difficulty in the way of describing virtual scenarios. We describe them in increasing levels of difficulty as below.

- Firstly, we explore labelling bricks from A to Z according to the order of spatial stacking from bottom to top, and the corresponding texts are also described in order from bottom to top, we call this setting "No shuffle".
- Secondly, we shuffle the order of the corresponding natural language description while maintaining the labelling rules in alphabetic order called "Shuffle description".
- Thirdly, we shuffled the order of labelling so that the spatial relationships do not correspond to the alphabetic order anymore, but are still described in the order from bottom to top in the text description, called "Shuffle label".
- Finally, we shuffled both the order of labelling and description. We call it "Shuffle both".

We use colors to represent the bricks, which enriches the information and increases the difficulty of the tasks. For each setting with 1D and 2D, we create 500 evaluation instances. The final evaluation set consists of 4,000 instances.

2.3 NLVR-BASED MANIPULATION

Figure 2 demonstrates an instance for NLVR-based Manipulation (middle). We convert the format of Natural Language Visual Reasoning (NLVR, Suhr et al. (2017)) tasks into a text-based planning

task. Based on the creation rules of synthetic images of NLVR, we create 1,000 natural language descriptions for the virtual spatial environments using Python code. Specifically, for each description, we set three boxes just like NLVR, in the left, middle, and right, and in each box, and there are several objects. Each object has three properties: color, shape, and size. Each description has one related question, the question is about how to move all objects that satisfy a certain condition of one property (such as "all objects in black" or "all rounds") to a specific target box. The ground truth is the set of all objects satisfied with this condition which needs to be moved (not in the target boxes).

2.4 NATURAL LANGUAGE NAVIGATION

Figure 2 demonstrates an instance for Natural Language Navigation (bottom). Inspired by Vision-and-Language navigation (Gu et al., 2022), we create a virtual spatial environment that is similar to a 2D map of navigation tasks but using natural language description only. Specifically, we define a set of landmarks: '*store*', '*bank*', '*house*', '*cinema*', '*garden*', '*school*'. For each description, there are 7 to 10 landmarks. We create 500 evaluation instances using Python code: the relationship between landmarks is a binary tree structure, with a root node which indicates the start point in the virtual scenario, and each node other than the leaf nodes has one or two child nodes, with a distance of 100 meters or 200 meters between them. Each description has one related question which is about how to reach the nearest one specific kind of landmark from the starting point.

2.5 SPATIAL QA

We also evaluate COS on manually annotated existing spatial question answering task, SPARTUN (Mirzaee and Kordjamshidi, 2022), which contains a larger variety of spatial relation types and spatial expressions compared with previous Spatial QA datasets and our three synthetic spatial planning tasks. And the questions in this dataset are manually annotated, which is closer to real-world scenes. The scenarios in this dataset are described in natural languages based on NLVR (Suhr et al., 2017) and SPARTQA (Mirzaee et al., 2021).

3 CHAIN-OF-SYMBOL PROMPTING

We propose Chain-of-Symbol (CoS) prompting for LLMs, which converts the simulated environment with natural language into a condensed symbolic representation that considers spatial relationship. In order to make our constructing method of COS generalizable and reliable, we adopt a three-step procedure in creating the demonstrations of our COS which can be used in any related tasks:

- (i) Automatically prompt the LLMs to generate a CoT demonstration in a zero-shot manner
- (ii) Correct the generated CoT demonstration if there existing errors.
- (iii) Replace the spatial relationships described in natural languages in CoT with random **symbols**, and only keep objects and symbols, remove other descriptions.

We then use the CoS demonstrations to guide the language model in a few-shot manner for prompting LLMs just like CoT (Wei et al., 2022).

Figure 1 depicts an example of a demonstration of CoS produced by models. In this example, we see that both CoT and COS receive the same shared simulated spatial environment in natural language texts. COS depicts a different intermediate thinking process than CoT. The latter represents the environments in a natural language only, while the former use a condensed symbolic representation that considers spatial relationship. Specifically, we use the symbol "/" to represent the spatial relationship "from the top of" here. By doing such a conversion, and removing redundant descriptions, COS effectively improves the model performance as well as reduces the inference costs with LLMs.

Figure 2 depicts examples of CoS demonstration for all three planning tasks we proposed. For NLVR-based Manipulation, we convert natural language descriptions for objects to the format of a triplet such as "(large, round, black)". For Natural Language Navigation, we represent the order of landmarks by using symbol "/" to connect them. For Spatial QA task, we use a set of symbols such as "=", "~" to represent different spatial relationships, and use triplet with "(, ,)" to represent objects and their attributes.

CoS prompting has multiple properties that are attractive as a prompting approach for LLMs:

- First, CoS effectively allows a neater, shorter, and condensed intermediate procedure than CoT. It is more structured than natural languages, hence easier for human annotators to analyze, check and correct the intermediate thinking process for LLMs.
- Second, CoS improves important planning tasks that current LLMs do not tackle well. It provides a better representing method for spatial environments which is easier for LLMs to learn compared with natural language.
- Finally, CoS clearly reduces the amount of text input into the LLMs and output from LLMs. This makes it much cheaper to access LLMs with API/GPU.

4 EXPERIMENTS

In this section, we first introduce our experimental setup in Section 4.1 about the settings of different methods we use, the language models, and the evaluation metrics. Then, in Section 4.2, we report the results of the three spatial planning tasks we proposed. In Section 4.3, we report the results on the SPARTUN dataset.

4.1 EXPERIMENTAL SETUP

We evaluate CoS and CoT on our proposed three spatial planning tasks and the existing SQA dataset, based on various LLMs like ChatGPT(gpt-3.5-turbo) and text-davinci-003. There are three prompts: zero-shot CoT, few-shot CoT, and few-shot CoS (Ours).

Zero-shot Chain-of-Thought Prompting We consider zero-shot CoT as our baseline. The reason is that we have found that our choices of LLMs naturally give their intermediate steps (CoT) in their answers, even without specifically asking them to do so. We also found that asking them to remove the thinking steps obviously degrades the results. Therefore, we allow the LLMs to generate CoT, while we do not put any demonstration to the prompt but give prompts like "Let's think step by step" just as Kojima et al. (2023). For an easier evaluation, we ask the LLMs to output the final results by separating the landmarks with commas.

Chain-of-Thought Prompting This baseline uses a few-shot CoT, in which we encourage LLMs to think step by step, and we use five demonstrations to guide the LLMs in the thinking procedure. Note that the intermediate thinking procedure is represented as natural language text, just like the Standard Prompting. Like in (Wei et al., 2022), we manually crafted five demonstrations for each task to guarantee their correctness. To guarantee the consistency and reliability of the prompts, we follow the format of CoT generated by zeroshot-CoT prompting. We use these fixed five demonstrations for evaluations on each task.

Chain-of-Symbol Prompting As described in Section 3, CoS augments the standard CoT prompting with condensed symbolic representation. While CoT has been shown to give large improvements to LLMs on various tasks (Wei et al., 2022), we argue that using condensed symbolic representations can be an alternative to describing using natural language texts. We manually converted from CoT demonstrations to CoS using the procedure described in Section 3. Five CoS demonstrations of the same examples with CoT are created for each task of Natural Language Planning.

Language Models We use Text-Davinci-003 and ChatGPT(Gpt-3.5-turbo) for the evaluation of all tasks. We set the temperature to 0 for all the experiments throughout this paper.

Evaluation Metrics For planning tasks, we use three evaluation metrics, namely accuracy, precision, and recall. We define accuracy as the success rate in achieving the final goal. We then compute the Longest Common Sequence (LCS) between the ground truth and LLM output sequence to measure their similarity. We compute precision as the ratio of LCS against the length of the LLM output, and we compute recall as the ratio of LCS against the length of the ground truth. For spatial QA task, we only compute accuracy.

Table 1: The results of ChatGPT(gpt-3.5-turbo) on Brick World. We report the results with four settings as described in Section 2.2, under both 1D and 2D scenarios. We adopt $N_s = 5$, where N_s represents the number of demonstrations for CoS and CoT. The best results are bolded. For CoS and CoT, we report the average and the standard deviation from three runs with different sets of demonstrations. **Acc.** represents accuracy, **Pre.** represents precision, and **Rec.** represents recall. zs-CoT represents zero-shot CoT. We report the average number of tokens in the intermediate steps.

Model	No Shuffle			Shuffle Description			Shuffle Label			Shuffle Both			Tok.
	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	
<i>1D Scenario</i>													
zs-CoT	61.0	77.2	71.9	60.4	77.5	77.5	31.8	63.4	59.8	28.2	58.6	55.3	-
CoT	81.0±11.0	87.7±4.5	90.1±2.6	71.5±9.2	90.7±3.6	81.8±7.1	75.1±10.1	88.0±3.6	90.1±0.9	43.0±4.4	71.4±3.3	75.7±1.6	407
CoS	96.6±1.9	98.0±0.8	97.7±0.8	95.9±1.2	97.9±0.6	97.5±0.3	92.6±2.0	97.0±1.3	95.9±1.1	69.7±5.1	86.7±4.2	83.6±1.6	139
<i>2D Scenario</i>													
zs-CoT	32.7	53.8	60.6	14.8	31.9	46.9	13.0	32.0	42.3	9.8	30.4	38.4	-
CoT	25.0±15.6	49.8±9.8	45.0±10.5	21.5±8.2	45.6±5.4	41.2±6.3	21.8±2.3	44.7±5.9	43.2±4.0	14.9±3.4	38.1±2.9	36.4±3.5	546
CoS	60.7±1.9	67.2±1.1	71.3±1.3	33.7±3.2	46.7±0.8	50.0±1.5	23.5±5.0	45.9±0.8	63.0±12.1	28.9±2.3	46.3±1.0	44.4±2.8	341

Table 2: The automatic evaluation results with Text-Davinci-003 on Brick World. We report the results with four settings as described in Section 2.2, under the 1D scenario. We adopt $N_s = 5$, where N_s represents the number of demonstrations for CoS and CoT. The best results are bolded. **Acc.** represents accuracy, **Pre.** represents precision, and **Rec.** represents recall.

Model	No Shuffle			Shuffle Description			Shuffle Label			Shuffle Both		
	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.
<i>1D Scenario</i>												
zs-CoT	51.0	77.6	75.2	51.6	83.2	83.1	43.4	67.4	73.0	21.8	51.4	54.6
CoT	89.0	89.3	89.5	56.2	72.0	57.0	79.4	83.2	80.0	59.6	74.4	60.6
CoS	90.0	90.0	90.0	75.8	90.4	78.6	85.0	86.8	86.6	73.4	87.0	74.8

4.2 RESULTS OF SPATIAL PLANNING TASKS

4.2.1 BRICK WORLD

Table 1 reports the results of CoS against the zs-CoT and CoT on the task of Brick World. First of all, we can see that the complexity increases both from the 1D scenario to the 2D scenario and from the setting of No Shuffle to the setting of Shuffle Both, together with a drop in the performance. ChatGPT with zs-CoT does not perform well, with only 9.8% accuracy on the most difficult setting Shuffle Both under the 2D scenario. Although CoT brings some improvements, the performance for CoT is still not satisfying, with an accuracy of 43.0% which is just below the 50% bar for setting Shuffle Both under the 1D scenario. In contrast, we see that CoS gives very good improvements on this setting (from 28.2% to 69.7%). We found that CoS gives consistent improvements to all the settings on Brick World, clearly surpassing CoT. The largest gain is on the setting of Shuffle Label under the 1D scenario, with 60.8% improvements in accuracy (from 31.8% to 92.6%). We postulate that such improvements come from symbolic spatial representations that are more condensed and easier to be understood by LLMs. Another underlying reason could be the elimination of redundant information, as LLMs can be easily distracted by irrelevant context (Shi et al., 2023).

4.2.2 FURTHER ANALYSIS OF BRICK WORLD

Randomness in the Experiments To investigate the randomness in our experiments, we run multiple trials with three different sets of demonstrations for CoT and CoS. Table 1 reports their means and standard deviations. We see a general trend here that CoS usually reports a lower standard deviation

Table 3: The automatic evaluation results with Text-Davinci-003 and gpt-3.5-turbo on NLVR-based Manipulation. We set $N_s = 5$, where N_s represents the number of demonstrations for prompting with CoS and CoT. The best results are bolded. We report the average and the standard deviation from three runs with different demonstrations. **Acc.** represents accuracy, **Pre.** represents precision, and **Rec.** represents recall (precision and recall are computed with sets in this case).

Model	text-davinci-003			gpt-3.5-turbo			Tokens
	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Tok.
zs-CoT	42.0	56.9	79.9	18.6	26.9	19.7	-
CoT	74.2±5.1	71.2±1.7	85.7±5.3	63.9±1.7	62.6±3.0	80.4±1.7	653
CoS	74.9±3.4	87.9±1.8	86.7±3.0	68.4±2.3	71.2±1.9	82.9±2.1	534

than CoT (for example, a standard deviation of 1.9 for Acc. for No Shuffle under the 1D scenario for CoS, against 11.0 for CoT). This represents that CoS is more stable than CoT on Brick World.

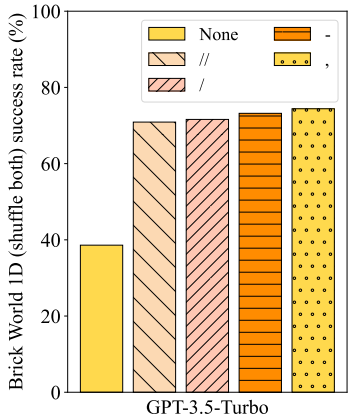


Figure 3: Performance of using different symbols for CoS on Brick World 1D (Shuffle Both) in accuracy.

CoS on the Different Language In addition to the tasks described in English, we also tested CoS on Brick World in Chinese. CoT reports 22.9% accuracy, and CoS reports 39.1% in 1D scenario of Brick World, which demonstrates the robustness of CoS to a language other than English.

Robustness to Different Symbols Figure 3 demonstrates the robustness of using different symbols for CoS. As we can see, using different symbols brings consistent performance, while not having any symbol drastically impacts the results. Among all the symbols, the comma gives the best results. We conclude that CoS is robust to the selection of symbol.

Results on Different Language Models Table 2 reports the results on InstructGPT under the 1D scenario. The experimental results align with our previous conclusion that CoS outperforms CoT obviously on all of the metrics.

Saving Tokens for Prompting One advantage featured by CoS is that it reduces the number of input tokens to be fed into LLMs. For Brick World (1D scenario), CoS reduces the

number of tokens for the intermediate steps from 407 to 139 (Table 1, the numbers are reported from OpenAI Playground¹). This subsequently saves the costs of accessing LLMs via API/GPUs.

4.2.3 NLVR-BASED MANIPULATION

For the task of NLVR-based Manipulation, we adopt almost the same experimental settings as the ones for Brick World, including both baseline settings as well as our choice of language models. The only difference is the evaluation metrics we report. In contrast to Brick World, we compute precision and recall based on the set rather than the Longest Common Sequence.

Main Results Table 3 reports the results of NLVR-based Manipulation with both text-davinci-003 and gpt-3.5-turbo. For this task, we see that both two models give an unsatisfying performance. Gpt-3.5-turbo only has an accuracy of 18.6% on its own, with 61.5 % accuracy when CoT is applied. In comparison, InstructGPT with CoS gives the highest accuracy (71.7%) among all the models. CoS reports a higher performance than CoT and the standard prompting on all of the metrics.

Saving Tokens for Prompting One advantage featured by CoS is that CoS reduces the number of input tokens to be fed into LLMs. Table 3 reported that for NLVR-based Manipulation, CoS reduces the number of tokens for the intermediate steps from 653 to 534, nearly by half of the

¹<https://platform.openai.com/playground>

Table 4: The results with Text-Davinci-003 and gpt-3.5-turbo on Natural Language Navigation. **Acc.** represents accuracy, **Pre.** represents precision, and **Rec.** represents recall. We report the average and the standard deviation from three runs with different demonstrations. We report the number of tokens in the intermediate steps from demonstrations as the last column.

Model	text-davinci-003			gpt-3.5-turbo			Tokens
	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Tok.
zs-CoT	32.5	50.0	64.4	52.8	74.0	79.6	-
CoT	65.6±2.3	83.8±1.7	84.1±1.9	53.6±2.8	76.3±1.1	81.7±0.8	390
CoS	69.4±4.4	85.3±2.6	85.4±3.5	64.1±3.8	81.7±1.3	84.5±0.7	239

Table 5: The automatic evaluation results with GPT-3.5-Turbo and GPT-4 on SPARTUN dataset. We apply CoT with 5 shots, and CoS with 5 shots (Ours) respectively. We report the number of tokens in the intermediate steps from demonstrations as the last column.

Model	GPT-3.5-Turbo	GPT-4	Tokens
CoT-5	47.1	69.8	198
CoS-5	49.4	72.6	167

original intermediate steps (we separate the tokens by space). This subsequently saves the costs of accessing LLMs via API/GPU, which enables easier access to the models.

4.2.4 NATURAL LANGUAGE NAVIGATION

For Natural Language Navigation, we adopt almost the same experimental settings as for Brick World. We adopt $N_s = 3$, where N_s represents the number of demonstrations for CoS and CoT. And we adopt the same evaluation metrics, baseline settings as well as our choice of language models.

Main Results Table 4 reports the results of Natural Language Navigation with both text-davinci-003 and gpt-3.5-turbo. For this task, we see that both two models give an unsatisfying performance. Text-davinci-003 only has an accuracy of 32.5% on its own, with 68.7 % accuracy when CoT is applied. In comparison, CoS gives much better performance. text-davinci-003 with CoS gives the highest score among all the models, which is about 71.7%. CoS reports the best performance on all of the metrics.

Saving Tokens for Prompting One advantage that CoS features is that it reduces the number of input tokens to be fed into LLMs. Table 4 reports that for Natural Language Navigation, CoS reduces the number of tokens for the intermediate steps from 390 to 239. This subsequently saves the costs of accessing LLMs via API/GPUs, which enables easier access to the models.

4.3 SPATIAL QUESTION ANSWERING

We also explore the effectiveness of CoS in a more real-world scenario, by using existing human annotated spatial QA dataset SPARTUN (Mirzaee and Kordjamshidi, 2022). Specifically, we applied both CoS and CoT on GPT-3.5-Turbo and GPT-4. CoS gains better performance and uses fewer tokens compared with CoT. It indicates in real-world scenarios, where both background descriptions and questions are described with more various expressions, CoS is still a better method than conventional CoT described in natural languages. In table 5, we report the results of performance, and both CoT and CoS have 5 shots. And the results show the superior of the CoS compared with using CoT with better performance and fewer tokens. It should be noticed that there are far more types of spatial relationships in SPARTUN dataset than our proposed planning tasks, so the results indicate CoS can gain promising performance even when there are a lot of symbols to represent different spatial relationships.

5 RELATED WORK

Large Language Models Large language models (LLMs) have demonstrated remarkable few-shot learning abilities across various domains (Brown et al., 2020; Srivastava et al., 2022), leading to a paradigm shift in AI to use LLMs as foundational models for language-related tasks, either directly or through fine-tuning (Bommasani et al., 2021; Hu et al., 2022). Srivastava et al. (2022) proposed a benchmark that covers many areas from education to software development. However, the planning task with text environments is overlooked. While less relevant to CoS, a concurrent work converts natural language into executable actions for robots with ChatGPT (Wake et al., 2023). Another very recent concurrent work uses Symbol Tuning that replaces natural language labels with arbitrary symbols to improve in-context learning (Wei et al., 2023).

Chain-of-Thought Reasoning The ability of LLMs (Brown et al., 2020; Srivastava et al., 2022) to perform complex reasoning tasks can be significantly enhanced by using a show known as Chain-of-Thought (CoT) prompting, which involves providing them with intermediate reasoning steps (Nye et al., 2021; Wei et al., 2022). Such a phenomenon also generalizes to the multilingual settings (Shi et al., 2023). Despite the fact that CoT is powerful, there are reports that demonstrate that CoT is not always useful and that integrating CoT degrades the performance on the task of Machine Translation in their experiment. And this is possibly due to the word-by-word translation (Peng et al., 2023).

Spatial Reasoning Spatial reasoning over natural language texts has been an important research direction in the community (Janner et al., 2018; Mirzaee et al., 2021). Janner et al. (2018) proposes to leverage representation learning on a navigation task that requires the agent to move a specific location. Rojowiec et al. (2020) proposes a new task on spatial reasoning that requires the language model to generate natural language instructions for ‘before’ and ‘after’ image pairs. Mirzaee et al. (2021) proposes a new benchmark for spatial question-answering with ‘which’ and ‘what’ questions regarding the environment. In a concurrent work, Tsai et al. (2023) demonstrates that LLMs perform poorly on text-based games with question-answering tasks that require several steps of reasoning.

Navigation and Path Planning Language grounding navigation (Gu et al., 2022) refers to the interdisciplinary task that requires the intelligent agent to perceive the visual environment and guide the user to the goal location through natural language instructions (Nguyen et al., 2019; Chen et al., 2019). Path planning (Panov et al., 2018; Krishna Lakshmanan et al., 2020) refers to the tasks that require the agent to plan its own path to achieve certain goals such as the shortest path or maximizing the cleaning area, typically through the use of reinforcement learning. These areas are highly relevant to the spatial planning tasks we explored and CoS, as the spatial environments can be potentially represented by symbolic representations. We leave the investigations of these application areas to future studies.

6 CONCLUSION

We found that current popular LLMs still lack abilities in complex spatial planning and understanding tasks. To this end, we propose a novel method called CoS (**Chain-of-Symbol Prompting**) that converts spatial relationships described in natural languages to condensed symbolic representations in the chained intermediate thinking steps. CoS is easy to use and does not need additional training on LLMs. Extensive experiments indicate that using few-shot CoS demonstration clearly surpasses the performance of using CoT described in natural languages on all three spatial planning tasks we proposed and the representative spatial QA benchmark with even fewer tokens (down to about 1/3 tokens of the thinking steps with CoT) used in the inputs compared with CoT prompting. The performance gain is strong, by up to 60.8% accuracy (from 31.8% to 92.6%) on Brick World for ChatGPT.

Limitations Refer to the Appendix for the section on Broader Impact. In addition, we only use two models to verify the effectiveness of our method due to the limited time and resources. It would be interesting to apply our method to more models with different sizes to see whether there is an emergent ability of CoS for LLMs. Nevertheless, our choices of foundation models are representative and they are popular LLMs.

REFERENCES

- Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the Opportunities and Risks of Foundation Models. *arXiv e-prints*, art. arXiv:2108.07258, August 2021. doi: 10.48550/arXiv.2108.07258.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.
- Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12530–12539, 2019. doi: 10.1109/CVPR.2019.01282.
- Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Wang. Vision-and-language navigation: A survey of tasks, methods, and future directions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7606–7623, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.524. URL <https://aclanthology.org/2022.acl-long.524>.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.
- Michael Janner, Karthik Narasimhan, and Regina Barzilay. Representation learning for grounded spatial reasoning. *Transactions of the Association for Computational Linguistics*, 6:49–61, 2018. doi: 10.1162/tacl_a_00004. URL <https://aclanthology.org/Q18-1004>.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023.
- Anirudh Krishna Lakshmanan, Rajesh Elara Mohan, Balakrishnan Ramalingam, Anh Vu Le, Prabhar Veerajagadeshwar, Kamlesh Tiwari, and Muhammad Ilyas. Complete coverage path planning using reinforcement learning for tetromino based cleaning and maintenance robot. *Automation in Construction*, 112:103078, 2020. ISSN 0926-5805. doi: <https://doi.org/10.1016/j.autcon.2020.103078>. URL <https://www.sciencedirect.com/science/article/pii/S0926580519305813>.

- Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+p: Empowering large language models with optimal planning proficiency, 2023.
- Roshanak Mirzaee and Parisa Kordjamshidi. Transfer learning with synthetic corpora for spatial role labeling and reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6148–6165, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.413. URL <https://aclanthology.org/2022.emnlp-main.413>.
- Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa 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>.
- Khanh Nguyen, Debadepta Dey, Chris Brockett, and Bill Dolan. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12519–12529, 2019. doi: 10.1109/CVPR.2019.01281.
- Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show Your Work: Scratchpads for Intermediate Computation with Language Models. *arXiv e-prints*, art. arXiv:2112.00114, November 2021. doi: 10.48550/arXiv.2112.00114.
- Aleksandr I. Panov, Konstantin S. Yakovlev, and Roman Suvorov. Grid path planning with deep reinforcement learning: Preliminary results. *Procedia Computer Science*, 123:347–353, 2018. ISSN 1877-0509. doi: <https://doi.org/10.1016/j.procs.2018.01.054>. URL <https://www.sciencedirect.com/science/article/pii/S1877050918300553>. 8th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2017 (Eighth Annual Meeting of the BICA Society), held August 1-6, 2017 in Moscow, Russia.
- Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. Towards Making the Most of ChatGPT for Machine Translation. *arXiv e-prints*, art. arXiv:2303.13780, March 2023.
- Robin Rojowiec, Jana Götze, Philipp Sadler, Henrik Voigt, Sina Zarriß, and David Schlangen. From “before” to “after”: Generating natural language instructions from image pairs in a simple visual domain. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 316–326, Dublin, Ireland, December 2020. Association for Computational Linguistics. URL <https://aclanthology.org/2020.inlg-1.38>.
- Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. Large Language Models Can Be Easily Distracted by Irrelevant Context. *arXiv e-prints*, art. arXiv:2302.00093, January 2023. doi: 10.48550/arXiv.2302.00093.
- Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*, 2023.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan

Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khachabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engfu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Berant, Jörg Froberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqi, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramírez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Śwędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimeo Xu, Mirac Suzgun, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdah Gheini, Mukund Varma T, Nanyun Peng, Nathan Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramón Risco Delgado, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima, Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Mishnerghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan,

- Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Timothy Telleen-Lawton, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. *arXiv e-prints*, art. arXiv:2206.04615, June 2022. doi: 10.48550/arXiv.2206.04615.
- Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. A corpus of natural language for visual reasoning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 217–223, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2034. URL <https://aclanthology.org/P17-2034>.
- Chen Feng Tsai, Xiaochen Zhou, Sierra S. Liu, Jing Li, Mo Yu, and Hongyuan Mei. Can Large Language Models Play Text Games Well? Current State-of-the-Art and Open Questions. *arXiv e-prints*, art. arXiv:2304.02868, April 2023. doi: 10.48550/arXiv.2304.02868.
- Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. Chatgpt empowered long-step robot control in various environments: A case application, 2023.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.
- Jerry Wei, Le Hou, Andrew Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, and Quoc V. Le. Symbol tuning improves in-context learning in language models. *arXiv e-prints*, art. arXiv:2305.08298, May 2023. doi: 10.48550/arXiv.2305.08298.

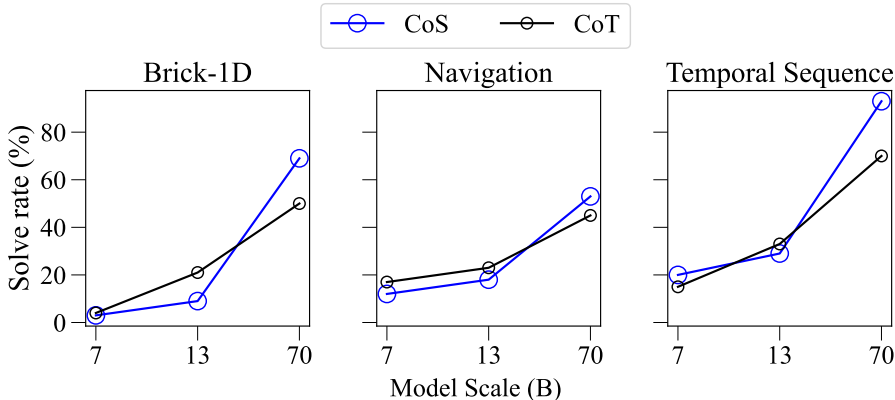


Figure 4: Scaling curve of CoS and CoT of Llama-2 on three tasks.

A BROADER IMPACT

CoS is a prompting technique that is easy to use, which can effectively improve the performance of complex planning with LLMs. It also indicates that future training with LLMs can also be well benefited by employing CoS in the training procedure to further improve LLM’s planning abilities.

B EXTENDED SETTINGS

B.1 NUMBER OF TOKENS

We have mentioned that we used white spacing for calculating the number of tokens in the intermediate thinking steps. This was a typo and in fact, we accurately measures the number of tokens using the OpenAI Playground.² The numbers we reported are correct and there is no need for modification.

B.2 DESIGNING THE INTERMEDIATE STEPS

The intermediate steps we use in the demonstrations for CoT are created and modified from the zero-shot CoT from the LLMs by simply adding “Let’s think step by step” before the answer. We then manually correct the intermediate steps from the outputs of using zero-shot CoT for further improvements. We attempted our best efforts in tuning the baselines, and we report the best results we achieved.

C FEW-SHOT EXEMPLARS

In the remaining of this section, we demonstrate the few-shot exemplars used in the experiments in our study. We demonstrate the exemplars for both CoS and CoT.

²<https://platform.openai.com/playground>

Table 6: Few-shot exemplars for full Chain-of-Symbol prompt for brick 1D.

Question: There is a set of bricks. For brick B, the color is yellow. The yellow brick C is on top of the brick B . The yellow brick A is on top of the brick C . Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick B

Answer:

A//C

C//B

In sum: A//C//B

So we get the result as A, C, B.

Question: There is a set of bricks. The yellow brick A is on top of the brick C . The yellow brick B is on top of the brick A . For the brick C, the color is white. Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick A?

Answer:

B//A

A//C

In sum: B//A//C

So we get the result as B, A.

Question: There is a set of bricks. The blue brick A is on top of the brick E . For the brick C, the color is blue. The blue brick D is on top of the brick B . The white brick E is on top of the brick D . The blue brick B is on top of the brick C . Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick D?

Answer:

A//E

E//D

D//B

B//C

In sum: A//E//D//B//C

So we get the result as A, E, D.

Question: There is a set of bricks. The white brick F is on top of the brick C . The white brick C is on top of the brick B . The yellow brick E is on top of the brick A . For the brick B, the color is white. The white brick D is on top of the brick F . The blue brick A is on top of the brick D . Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick F?

Answer:

E//A

A//D

D//F

F//C

C//B

In sum: E//A//D//F//C//B

So we get the result as E, A, D, F.

Question: There is a set of bricks. The white brick F is on top of the brick D . The yellow brick B is on top of the brick A . The blue brick D is on top of the brick C . The white brick E is on top of the brick G . For the brick A, the color is blue. The blue brick C is on top of the brick E . The white brick G is on top of the brick B . Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick G?

Answer:

A//B//G//E//C//D//F

D//F

C//D

E//C

G//E

So we get the result as F, D, C, E, G.

Table 7: Few-shot exemplars for full CoT prompt for brick 1D.

Question: There is a set of bricks. The yellow brick B is on top of the brick D . For the brick D, the color is white. The yellow brick A is on top of the brick C . The yellow brick E is on top of the brick A . The blue brick C is on top of the brick B . Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick D?

Answer: The bricks from bottom to top is B, D, C, A, E 1. Remove brick E from the top of brick A. 2. Remove brick A from the top of brick C. 3. Remove brick C from the top of brick B. 4. Now brick B is the topmost yellow brick and can be grabbed, but we need to remove it to get to brick D. 5. Remove brick B from the top of brick D. 6. Now brick D is the topmost white brick and can be grabbed. So we get the result as E, A, C, B, D.

Question: There is a set of bricks. For the brick A, the color is blue. The white brick B is on top of the brick C . The blue brick C is on top of the brick A . Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick B?

Answer: The bricks from bottom to top is A, C, B 1. Remove brick C from the top of brick A. 2. Now brick A is the topmost brick and can be grabbed, but we need to remove it to get to brick B. 3. Remove brick A from the top of brick C. 4. Now brick B is the topmost white brick and can be grabbed. So we get the result as C, A, B.

Question: There is a set of bricks. The white brick B is on top of the brick A . The white brick A is on top of the brick G . The white brick G is on top of the brick D . The blue brick D is on top of the brick F . The white brick H is on top of the brick B . For the brick C, the color is yellow. The white brick E is on top of the brick C . The white brick F is on top of the brick E . Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick F?

Answer: The bricks from bottom to top is C, E, F, D, G, A, B, H 1. Remove brick H from the top of brick B. 2. Remove brick B from the top of brick A. 3. Remove brick A from the top of brick G. 4. Remove brick G from the top of brick D. 5. Remove brick D from the top of brick F. 6. Now brick F is the topmost white brick and can be grabbed. So we get the result as H, B, A, G, D, F.

Question: There is a set of bricks. The white brick B is on top of the brick C . The yellow brick D is on top of the brick A . For the brick A, the color is yellow. The yellow brick E is on top of the brick B . The white brick C is on top of the brick D . Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick D?

Answer: The bricks from bottom to top is A, D, C, B, E 1. Remove brick E from the top of brick B. 2. Remove brick B from the top of brick C. 3. Now brick C is the topmost white brick and can be grabbed, but we need to remove it to get to brick D. 4. Remove brick C from the top of brick D. 5. Now brick D is the topmost yellow brick and can be grabbed. So we get the result as E, B, C, D.

Question: There is a set of bricks. The yellow brick C is on top of the brick B . The white brick B is on top of the brick A . For the brick A, the color is white. Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick B?

Answer: The bricks from bottom to top is A, B, C 1. Now brick C is the topmost brick and can be grabbed, but we need to remove it to get to brick B. 2. Now brick B is the topmost white brick and can be grabbed. So we get the result as C, B.

Table 8: Few-shot exemplars for full CoS prompt for brick 2D.

Question: There is a set of bricks. There is a brick A. The brick B is in front of the brick A. The brick C is in front of the brick B. The brick D is on top of the brick B. The brick E is on top of the brick A. The brick F is on top of the brick C. The brick G is on top of the brick D. The brick H is on top of the brick G. Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick A?

Answer:

We can get
A//E,
C//F,
B//D//G//H.

So we get the result as E, A.

Question: There is a set of bricks. For the brick A. The yellow brick B is on top of the brick A. The blue brick C is on top of the brick B. The blue brick D is on top of the brick C. Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick B?

Answer:

We can get
A//B//C//D.

So we get the result as D, C, B.

Question: There is a set of bricks. For the brick A, the color is blue. The yellow brick B is in front of the brick A. The yellow brick C is in front of the brick B. The white brick D is on top of the brick C. The white brick E is on top of the brick D. The yellow brick F is on top of the brick E. Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick C?

Answer:

We can get
C->B->A,
C//D//E//F.

So we get the result as F, E, D, C.

Question: There is a set of bricks. The brick P is on top of the brick R. The brick J is on top of the brick B. The brick D is on top of the brick P. The brick R is on top of the brick H. The brick K is in front of the brick M. The brick B is on top of the brick D. For the brick M, the color is blue. The brick C is on top of the brick J. The brick H is in front of the brick K. Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick M?

Answer:

We can get
H//R//P//D//B//J//C,
M->K->H.

So we get the result as M directly.

Question: There is a set of bricks. The brick K is on top of the brick F. The brick M is in front of the brick F. The brick N is on top of the brick K. For the brick O, the color is blue. The brick G is on top of the brick A. The brick F is in front of the brick I. The brick I is in front of the brick O. The brick A is on top of the brick N. Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick O?

Answer:

We can get
F//K//N//A//G,
F->I->O.

So we get the result as O directly.

Table 9: Few-shot exemplars for full CoT prompt for brick 2D.

Question: There are a set of bricks. There is a brick A. The brick B is in front of the brick A. The brick C is in front of the brick B. The brick D is on top of the brick B. The brick E is on top of the brick A. The brick F is on top of the brick C. The brick G is on top of the brick D. The brick H is on top of the brick G. Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick A?

Answer: Let's think step by step:

1. To get brick A, we find E is on top of A.
 2. We find E is on the top.
 - 3 We need to remove brick E first, as it is on top of brick A.
 4. Brick A is now accessible and can be grabbed.
- So we get the result as E, A.

Question: There are a set of bricks. For the brick A. The yellow brick B is on top of the brick A. The blue brick C is on top of the brick B. The blue brick D is on top of the brick C. Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick B?

Answer: Let's think step by step: 1. To get brick B, we find C is on top of B 2. We find D is on top of C 3. We find D is on the top 4. We need to remove brick D, as it is on top of brick C. 5. We need to remove brick C, as it is on top of brick B. 6. Brick B is now accessible and can be grabbed. So we get the result as D, C, B.

Question: There are a set of bricks. The brick P is on top of the brick R. The brick J is on top of the brick B. The brick D is on top of the brick P. The brick R is on top of the brick H. The brick K is in front of the brick M. The brick B is on top of the brick D. For the brick M, the color is blue. The brick C is on top of the brick J. The brick H is in front of the brick K. Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick M?

Answer: Let's think step by step: 1. To get brick M, we find there is no brick on top of brick M, So we get the result as M directly.

Question: There are a set of bricks. The brick K is on top of the brick F. The brick M is in front of the brick F. The brick N is on top of the brick K. For the brick O, the color is blue. The brick G is on top of the brick A. The brick F is in front of the brick I. The brick I is in front of the brick O. The brick A is on top of the brick N. Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick O?

Answer: Let's think step by step: 1. To get brick O, we find there is no brick on top of brick O, So we get the result as O directly.

Question: There are a set of bricks. For the brick A, the color is blue. The yellow brick B is in front of the brick A. The yellow brick C is in front of the brick B. The white brick D is on top of the brick C. The white brick E is on top of the brick D. The yellow brick F is on top of the brick E. Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick is to be grabbed, the upper brick must be removed first. How to get brick C?

Answer: Let's think step by step: 1. To get brick C, we find C is in front of B.
2. We find D is on top of C.
3. We find E is on top of D.
4. We find F is on top of E.
5. We need to remove brick F, as it is on top of brick E.
6. We need to remove brick E, as it is on top of brick D.
7. We need to remove brick D, as it is on top of brick C.
8. Brick C is now accessible and can be grabbed.
So we get the result as F, E, D, C.

Table 10: Few-shot exemplars for full CoS prompt for Natural Language Navigation

Question: There is a set of roads and a set of landmarks. The start point is bank A. There is a road which is 200 meters long from bank A to bank C. There is a road which is 100 meters long from bank C to house H. There is a road which is 100 meters long from house H to cinema F. There is a road which is 200 meters long from cinema F to store B. There is a road which is 100 meters long from store B to store G. There is a road which is 200 meters long from bank C to house D. There is a road which is 200 meters long from house D to garden J. There is a road which is 100 meters long from bank A to cinema I. There is a road which is 100 meters long from cinema I to house E. From the start point, how to reach the nearest store?

Answer: There are two stores on the map, store B and store G.

- bank A / bank C / house H / cinema F / store B ($200 + 100 + 100 + 200 = 600$)

- bank A / bank C / house H / cinema F / store B / store G ($200 + 100 + 100 + 200 + 100 = 700$)

So we get the result as bank A / bank C / house H / cinema F / store B.

So the answer is ACHFB

Question: There is a set of roads and a set of landmarks. The start point is bank H. There is a road which is 100 meters long from bank H to store E. There is a road which is 200 meters long from store E to bank C. There is a road which is 100 meters long from bank C to house A. There is a road which is 100 meters long from house A to house F. There is a road which is 200 meters long from bank C to garden I. There is a road which is 200 meters long from store E to cinema G. There is a road which is 200 meters long from cinema G to school J. There is a road which is 200 meters long from bank H to school D. There is a road which is 100 meters long from school D to store B. From the start point, how to reach the nearest school?

Answer: There are two schools on the map, school J and school D. - bank H / store E / bank C / garden I / cinema G / school J ($200 + 200 + 200 = 600$)

- bank H / school D (200)

We get the result as bank H / school D.

So the answer is HD

Question: There is a set of roads and a set of landmarks. The start point is garden B. There is a road which is 200 meters long from garden B to cinema D. There is a road which is 200 meters long from cinema D to school J. There is a road which is 100 meters long from garden B to school I. There is a road which is 100 meters long from school I to house C. There is a road which is 100 meters long from house C to bank F. There is a road which is 100 meters long from house C to cinema G. There is a road which is 200 meters long from cinema G to store H. There is a road which is 100 meters long from store H to cinema E. There is a road which is 200 meters long from cinema E to bank A. From the start point, how to reach the nearest school?

Answer: There are two schools on the map, school J and school I.

- garden B / cinema D / school J ($200+200 = 400$)

- garden B / school I (100)

We get the result as garden B / school I (100).

So the answer is BI

Table 11: Few-shot exemplars for full CoT prompt for Natural Language Navigation

Question: There is a set of roads and a set of landmarks. The start point is bank A. There is a road which is 200 meters long from bank A to bank C. There is a road which is 100 meters long from bank C to house H. There is a road which is 100 meters long from house H to cinema F. There is a road which is 200 meters long from cinema F to store B. There is a road which is 100 meters long from store B to store G. There is a road which is 200 meters long from bank C to house D. There is a road which is 200 meters long from house D to garden J. There is a road which is 100 meters long from bank A to cinema I. There is a road which is 100 meters long from cinema I to house E. From the start point, how to reach the nearest store?

Answer: Let's think step by step 1. Start at bank A. 2. There are two stores on the map, store B and store G. 3. The road from bank A to store B is from bank A to bank C to house H to cinema F to store B (200 meters + 100 meters + 100 meters + 200 meters = 600 meters) 4. The road from bank A to store B is shorter than the road from bank A to store G. 5. Therefore, the nearest store from the start point (bank A) is store B. So the answer is ACHFB.

Question: There is a set of roads and a set of landmarks. The start point is bank H. There is a road which is 100 meters long from bank H to store E. There is a road which is 200 meters long from store E to bank C. There is a road which is 100 meters long from bank C to house A. There is a road which is 100 meters long from house A to house F. There is a road which is 200 meters long from bank C to garden I. There is a road which is 200 meters long from store E to cinema G. There is a road which is 200 meters long from cinema G to school J. There is a road which is 200 meters long from bank H to school D. There is a road which is 100 meters long from school D to store B. From the start point, how to reach the nearest school?

Answer: Let's think step by step 1. Start at bank H. 2. There are two schools on the map, school J and school D. 3. The road from bank H to school J is from bank H to store E to cinema G to school J (200 meters + 200 meters + 200 meters = 600 meters). 4. The road from bank H to school D is from bank H to school D (200 meters). 5. Therefore, the nearest school from the start point (bank H) is school D. 6. Follow the road from bank H to school D (200 meters). So the answer is HD

Question: There is a set of roads and a set of landmarks. The start point is garden B. There is a road which is 200 meters long from garden B to cinema D. There is a road which is 200 meters long from cinema D to school J. There is a road which is 100 meters long from garden B to school I. There is a road which is 100 meters long from school I to house C. There is a road which is 100 meters long from house C to bank F. There is a road which is 100 meters long from house C to cinema G. There is a road which is 200 meters long from cinema G to store H. There is a road which is 100 meters long from store H to cinema E. There is a road which is 200 meters long from cinema E to bank A. From the start point, how to reach the nearest school?

Answer: Let's think step by step 1. Start at garden B. 2. There are two schools on the map, school J and school I. 3. The road from garden B to school J is from garden B to cinema D to school J (200 meters + 200 meters = 400 meters). 4. The road from garden B to school I is from garden B to school I (100 meters). 5. Therefore, the nearest school from the start point (garden B) is school I. 6. Follow the road from garden B to school I (100 meters). So the answer is BI

Table 12: Few-shot exemplars for full CoS prompt for NLVR-based Manipulation.

Story: There are three boxes. In the left box, there are one middle square in yellow, one middle square in black, one small square in blue, one middle square in blue. In the middle box, there are one large square in blue, one middle square in blue, one middle square in black, one large triangle in black, one middle round in blue, one small square in yellow. In the right box, there are one large round in blue, one small triangle in yellow, one large triangle in blue. Question: How to move all rounds to the middle box?

Answer: To move all rounds to the middle box, we need to identify all the round objects in the three boxes. They are:

- middle box: (middle, round, blue)
- right box: (large, round, blue)

Then we can move the objects not in the middle box above to the middle box one by one:

- (large, round, blue) middle - right

Story: There are three boxes. In the left box, there are one small round in blue, one small round in blue, one large round in black, one large square in blue, one small round in yellow, one small round in yellow. In the middle box, there are one large square in blue, one middle triangle in yellow. In the right box, there are one large round in black, one large round in blue, one middle triangle in black, one middle triangle in black. Question: How to move all squares to the right box?

Answer: To move all square objects to the right box, we need to identify all the square objects in the three boxes. They are:

- left box: (large, square, blue)
- middle box: (large, square, blue)

Then we can move the objects not in the right box above to the right box:

- (large, square, blue) left - right box
- (large, square, blue) middle - right box

Story: There are three boxes. In the left box, there are one large square in blue, one middle square in blue, one small round in blue, one middle triangle in blue, one middle round in yellow, one large square in yellow. In the middle box, there are one small round in yellow, one middle square in blue, one small triangle in black, one small square in black, one small triangle in yellow, one large round in black. In the right box, there are one small square in yellow, one small triangle in yellow, one middle triangle in black, one large round in yellow, one middle square in blue, one large square in yellow. Question: How to move all black objects to the right box?

Answer: To move all black objects to the right box, we need to identify all the black objects in the three boxes. They are:

- middle box:
(small, triangle, black),
(small, square, black),
(large, round, black)

Then we can move the objects not in the right box above to the right box:

- (small, triangle, black) middle - right
- (small, square, black) middle - right
- (large, round, black) middle - right

Story: There are three boxes. In the left box, there are one middle square in yellow, one middle square in black, one small square in blue, one middle square in blue. In the middle box, there are one large square in blue, one middle square in blue, one middle square in black, one large triangle in black, one middle round in blue, one small square in yellow. In the right box, there are one large round in blue, one small triangle in yellow, one large triangle in blue. Question: How to move all rounds to the middle box?

Answer: To move all rounds to the middle box, we need to identify all the round objects in the three boxes. They are:

- middle box: (middle, round, blue)
- right box: (large, round, blue)

Then we can move the objects not in the middle box above to the middle box:

- (large, round, blue) right - middle

Story: There are three boxes. In the left box, there are one large round in black, one small square in blue, one small square in blue, one small triangle in yellow, one middle square in yellow. In the middle box, there are one large square in yellow, one middle triangle in blue, one large round in black. In the right box, there are one large square in blue, one large triangle in black, one middle triangle in black. Question: How to move all black objects to the left box?

Answer: To move all black objects to the left box, we need to identify all the black objects in the three boxes. They are:

- left box: (large, round, black)
- middle box: (large, round, black),
- right box: (middle, triangle, black), (large, triangle, black)

Then we can move the objects not in the left box above to the left box:

1. (large, round, black) middle - left
2. (middle, triangle, black) right - left
3. (large, triangle, black) right - left

Table 13: Few-shot exemplars for full CoT prompt for NLVR-based Manipulation.

Story: There are three boxes. In the left box, there are one middle square in yellow, one middle square in black, one small square in blue, one middle square in blue. In the middle box, there are one large square in blue, one middle square in blue, one middle square in black, one large triangle in black, one middle round in blue, one small square in yellow. In the right box, there are one large round in blue, one small triangle in yellow, one large triangle in blue. Question:How to move all rounds to the middle box?

Answer: To move all rounds to the middle box, we need to identify all the round objects in the three boxes. They are:

- One middle round in blue in the middle box - One large round in blue in the right box

Then we can move the boxes above not in the middle box to the middle box one by one:

1. Move the large round in blue from the right box to the middle box.

Now all the rounds are in the middle box.

Story: There are three boxes. In the left box, there are one small round in blue, one small round in blue, one large round in black, one large square in blue, one small round in yellow, one small round in yellow. In the middle box, there are one large square in blue, one middle triangle in yellow. In the right box, there are one large round in black, one large round in blue, one middle triangle in black, one middle triangle in black. Question:How to move all squares to the right box?

Answer: To move all square objects to the right box, we need to identify all the square objects in the three boxes. They are:

- One large square in blue in the left box - One large square in blue in the middle box

Then we can move the box above with the square object to the right box:

1. Move the large square in blue from the left box to the right box. 2. Move the large square in blue from the middle box to the right box.

Now all the square objects are in the right box.

Story: There are three boxes. In the left box, there are one large square in blue, one middle square in blue, one small round in blue, one middle triangle in blue, one middle round in yellow, one large square in yellow. In the middle box, there are one small round in yellow, one middle square in blue, one small triangle in black, one small square in black, one small triangle in yellow, one large round in black. In the right box, there are one small square in yellow, one small triangle in yellow, one middle triangle in black, one large round in yellow, one middle square in blue, one large square in yellow. Question:How to move all black objects to the right box?

Answer: To move all black objects to the right box, we need to identify all the black objects in the three boxes. They are:

- One small triangle in black in the middle box - One middle triangle in black in the middle box - One large round in black in the middle box

Then we can move the boxes above not in the right box to the right box one by one: 1. Move the small triangle in black from the middle box to the right box. 2. Move the middle triangle in black from the middle box to the right box. 3. Move the large round in black from the middle box to the right box.

Now all the black objects are in the right box.

Story: There are three boxes. In the left box, there are one middle square in yellow, one middle square in black, one small square in blue, one middle square in blue. In the middle box, there are one large square in blue, one middle square in blue, one middle square in black, one large triangle in black, one middle round in blue, one small square in yellow. In the right box, there are one large round in blue, one small triangle in yellow, one large triangle in blue. Question:How to move all rounds to the middle box?

Answer: To move all rounds to the middle box, we need to identify all the round objects in the three boxes. They are:

- One middle round in blue in the middle box - One large round in blue in the right box

Then we can move the boxes above not in the middle box to the middle box one by one:

1. Move the large round in blue from the right box to the middle box.

Now all the rounds are in the middle box.

Story: There are three boxes. In the left box, there are one large round in black, one small square in blue, one small square in blue, one small triangle in yellow, one middle square in yellow. In the middle box, there are one large square in yellow, one middle triangle in blue, one large round in black. In the right box, there are one large square in blue, one large triangle in black, one middle triangle in black. Question:How to move all black objects to the left box?

Answer: To move all black objects to the left box, we need to identify all the black objects in the three boxes. They are:

- One large round in black in the left box - One large round in black in the middle box - One large triangle in black in the right box - One middle triangle in black in the right box

Then we can move the boxes above not in the left box to the left box one by one:

1. Move the large round in black from the middle box to the left box. 2. Move the large triangle in black from the right box to the left box. 3. Move the middle triangle in black from the right box to the left box.

Now all the black objects are in the left box.

Table 14: Few-shot exemplars for full CoT prompt for SPARTQA.

Example 1:

Background: There are three blocks called A, B, and C. In block A, there is a small blue square far above a small blue circle. There is a small black triangle far to the right of both small blue objects. The black triangle is near and above a medium blue square. The medium blue square is touching the right edge of A and is near and above a medium black circle. Near and to the right of the medium black circle is a medium yellow triangle. The yellow triangle is to the left of the small blue circle. In B, which is to the right of A, there is a large blue square near and to the left of a large yellow triangle. There is also a small yellow triangle far below the square. In C, which is to the right of B, there is a small yellow circle near and above a small yellow triangle. The small yellow triangle is touching the top of a medium black triangle. The black triangle is touching the right edge of C. There is a large blue circle far below the black triangle.

Question: Are all yellow objects in C above a blue thing?

Candidate Answers: ["Yes", "No", "Do not Know"]

Answer:

Let's analyze the information provided step by step: In block C: 1. There is a small yellow circle near and above a small yellow triangle. 2. The small yellow triangle is touching the top of a medium black triangle. 3. There is a large blue circle far below the black triangle. Based on the provided information, the small yellow circle and the small yellow triangle in Block C are indeed above a blue thing, which is the large blue circle. So, the correct answer is: "Yes".

Example 2:

Background: There are three blocks called A, B, and C. A is to the left of B and B is to the left of C. In A, there is a small black triangle near and to the left of a large blue circle. The large blue circle is near and to the left of a small blue circle. The small blue square is far below a medium yellow square. In B, there is a small yellow circle. In C, there is a medium black triangle that is near and to the left of a small yellow circle. The small yellow circle is near and to the left of a blue triangle. There is a large yellow square touching the bottom of the small yellow circle. Near and below the large yellow square is a large black square.

Question: What is the relation between the large black thing and the small yellow thing?

Candidate Answers: ["left", "right", "above", "below", "near to", "far from", "touching", "Do Not Know"]

Answer:

Let's analyze the information provided step by step: In block C: 1. There is a medium black triangle that is near and to the left of a small yellow circle. 2. The small yellow circle is near and to the left of a blue triangle. 3. There is a large yellow square touching the bottom of the small yellow circle. 4. Near and below the large yellow square is a large black square. Now, let's determine the relation between the "large black thing" (large black square) and the "small yellow thing" (small yellow circle): 1. The large black square is in block C, and it is described as being "near and below" the large yellow square. 2. The small yellow circle is also in block C, and it is described as being "near and to the left of" the medium black triangle. Based on the provided information, the large black square is below the large yellow square and the small yellow circle is near to the medium black triangle. So, the correct answer: "below" and "near to".

Example 3:

Background: There are three blocks called A, B, and C. A is to the right of B and B is to the right of C. In A, there is a medium black circle that is far to the right of a medium blue square. The medium blue square is near and to the right of a small yellow circle. Far above the blue square is a medium yellow triangle. Near above the medium triangle is a small blue circle. In B, there is a large blue square that is touching the right edge of B. There is a medium black triangle that is touching the left side of the blue square. Near and above the medium black triangle is a large blue circle. Near and above the large blue circle is a large yellow triangle. In C, there is a small yellow triangle near and to the right of a medium black circle. The yellow triangle is touching the right edge of C. Far to the left of the medium black circle is a small blue square.

Question: What block does not have any objects touching the edge of it?

Candidate Answers: ["A", "B", "C"]

Answer:

Let's analyze the information provided step by step: In block A: 1. There is a medium black circle, which is far to the right of a medium blue square. 2. The medium blue square is near and to the right of a small yellow circle. 3. Far above the blue square is a medium yellow triangle. 4. Near above the medium triangle is a small blue circle. In block B: 1. There is a large blue square that is touching the right edge of B. 2. There is a medium black triangle that is touching the left side of the blue square. 3. Near and above the medium black triangle is a large blue circle. 4. Near and above the large blue circle is a large yellow triangle. In block C: 1. There is a small yellow triangle near and to the right of a medium black circle. 2. The yellow triangle is touching the right edge of C. 3. Far to the left of the medium black circle is a small blue square. Based on the provided information, it's Block A, which does not have any objects touching its edge. So, the correct answer is: "A".

Example 4:

Background: There are three blocks called A, B, and C. A is to the left of B and B is to the left of C. In A, there is a large black triangle far above a medium black triangle. The medium black triangle is near and above a large yellow circle. Near and to the right of the large circle is a small blue circle. In B, there is a medium blue circle near and to the left of a small black square. Far below the blue circle is a small yellow circle. In C, there is a small blue square.

Question: What object is near the large yellow thing, the medium black triangle or the small blue circle the medium black triangle or the small blue circle?

Candidate Answers: ["the medium black triangle", "the small blue circle", "both of them", "none of them"]

Answer:

Let's analyze the information provided step by step: In block A: 1. There is a large black triangle far above a medium black triangle. 2. The medium black triangle is near and above a large yellow circle. 3. Near and to the right of the large circle is a small blue circle. In block B: 1. There is a medium blue circle near and to the left of a small black square. 2. Far below the blue circle is a small yellow circle. In block C: 1. There is a small blue square. Now, let's analyze the position of the large yellow circle: 1. The large yellow circle is in Block A and is near and above the medium black triangle. 2. The small blue circle is also in Block A and is near and to the right of the large yellow circle. Based on the provided information, both the medium black triangle and the small blue circle are near the large yellow circle in Block A. So, the correct answer is: "both of them".

Example 5:

Background: There are three blue shapes in a block that we called A. Two medium blue squares are below a big blue square. First medium blue square is to the left of the other medium one which is touching the bottom edge of block A. There is another block below the block A we call it B. A big blue triangle is touching the top edge of this block which is above a small black triangle, a big blue circle and a medium blue triangle. The medium triangle is touching the bottom edge of the block. The circle is below and to the left of the small triangle. It is above the medium triangle.

Question: Is the small triangle below the medium triangle?

Candidate Answers: ["Yes", "No", "Do not Know"]

Answer:

Let's analyze the information provided step by step: In block B: 1. A big blue triangle is touching the top edge of this block which is above a small black triangle, a big blue circle and a medium blue triangle. 2. The medium triangle is touching the bottom edge of the block. 3. The circle is below and to the left of the small triangle. It is above the medium triangle. Based on the provided information, the small black triangle is above the medium blue triangle. So, the correct answer is: "No".

Table 15: Few-shot exemplars for full CoS prompt for SPARTQA.

Example 1:

Background: There are three blocks called A, B, and C. In block A, there is a small blue square far above a small blue circle. There is a small black triangle far to the right of both small blue objects. The black triangle is near and above a medium blue square. The medium blue square is touching the right edge of A and is near and above a medium black circle. Near and to the right of the medium black circle is a medium yellow triangle. The yellow triangle is to the left of the small blue circle. In B, which is to the right of A, there is a large blue square near and to the left of a large yellow triangle. There is also a small yellow triangle far below the square. In C, which is to the right of B. There is a small yellow circle near and above a small yellow triangle. The small yellow triangle is touching the top of a medium black triangle. The black triangle is touching the right edge of C. There is a large blue circle far below the black triangle.

Question: Are all yellow objects in C above a blue thing?

Candidate Answers:["Yes", "NO", "Do not Know"]

Answer:

Let's analyze the information provided step by step to determine if all yellow objects in Block C are above a blue thing: C: 1. (small, yellow, circle) \uparrow (small, yellow, triangle) 2. (small, yellow, triangle) = top of (medium, black, triangle) 3. (large, blue, circle) $\infty \downarrow$ (medium, black, triangle). Based on the provided information, C: (small, yellow, circle) \uparrow (large, blue, circle) and (small, yellow, triangle) \uparrow (large, blue, circle). So, the correct answer is: "Yes".

Example 2:

Background: There are three blocks called A, B, and C. A is to the left of B and B is to the left of C. In A, there is a small black triangle near and to the left of a large blue circle. The large blue circle is near and to the left of a small blue circle. The small blue square is far below a medium yellow square. In B, there is a small yellow circle. In C, there is a medium black triangle that is near and to the left of a small yellow circle. The small yellow circle is near and to the left of a blue triangle. There is a large yellow square touching the bottom of the small yellow circle. Near and below the large yellow square is a large black square.

Question: What is the relation between the large black thing and the small yellow thing?

Candidate Answers:["left", "right", "above", "below", "near to", "far from", "touching", "Do Not Know"]

Answer:

Let's break down the relationships between the objects step by step: A: 1. (small, black, triangle) $<$ (large, blue, circle) 2. (large, blue, circle) $<$ (small, blue, circle) 3. (small, blue, square) $\infty \downarrow$ (medium, yellow, square) B: 1. (small, yellow, circle) C: 1. (medium, black, triangle) $<$ (small, yellow, circle) 2. (small, yellow, circle) $<$ (blue, triangle) 3. (large, yellow, square) = \downarrow (small, yellow, circle) 4. (large, black, square) \downarrow (large, yellow, square) Now, let's determine the relation between the (large, black, square) and the (small, yellow, circle): 1. C: (large, black, square) \downarrow (large, yellow, square). Therefore, (large, black, square) \downarrow (large, yellow, square) 2. C: (small, yellow, circle) $<$ (medium, black, triangle). Therefore, (small, yellow, circle) (medium, black, triangle) So, the correct answer: "below" and "near to".

Example 3:

Background: There are three blocks called A, B, and C. A is to the right of B and B is to the right of C. In A, there is a medium black circle that is far to the right of a medium blue square. The medium blue square is near and to the right of a small yellow circle. Far above the blue square is a medium yellow triangle. Near the medium triangle is a small blue circle. In B, there is a large blue square that is touching the right edge of B. There is a medium black triangle that is touching the left side of the blue square. Near and above the medium black triangle is a large blue circle. Near and above the large blue circle is a large yellow triangle. In C, there is a small yellow triangle near and to the right of a medium black circle. The yellow triangle is touching the right edge of C. Far to the left of the medium black circle is a small blue square.

Question: What block does not have any objects touching the edge of it?

Candidate Answers:["A", "B", "C"]

Answer:

Let's analyze each block step by step to determine which one does not have any objects touching its edge: A: 1. (medium, black, circle) $\infty \gg$ (medium, blue, square) 2. (medium, blue, square) $>$ (small, yellow, circle). 3. (medium, yellow, triangle) $\infty \uparrow$ (blue, square). 4. (small, blue, circle) \uparrow (medium, triangle). B: 1. (large, blue, square) \Rightarrow edge of B. 2. (medium, black, triangle) \Rightarrow (blue, square). 3. (large, blue, circle) \uparrow (medium, black, triangle). 4. (large, yellow, triangle) \uparrow (large, blue, circle). C: 1. (small, yellow, triangle) $>$ (medium, black, circle). 2. (yellow, triangle) \Rightarrow edge of C. 3. (small, blue, square) $\infty <$ (medium, black, circle). Based on the provided information, let's determine which block does not have any objects touching its edge. It's Block B. So, the correct answer is: "A".

Example 4:

Background: There are three blocks called A, B, and C. A is to the left of B and B is to the left of C. In A, there is a large black triangle far above a medium black triangle. The medium black triangle is near and above a large yellow circle. Near and to the right of the large circle is a small blue circle. In B, there is a medium blue circle near and to the left of a small black square. Far below the blue circle is a small yellow circle. In C, there is a small blue square.

Question: What object is near the large yellow thing, the medium black triangle the small blue circle the medium black triangle or the small blue circle?

Candidate Answers:["the medium black triangle", "the small blue circle", "both of them", "none of them"]

Answer:

Let's break down the information step by step to determine which object is near the large yellow circle, the medium black triangle, or the small blue circle: A: 1. (large, black, triangle) $\infty \uparrow$ (medium, black, triangle). 2. (medium, black, triangle) \uparrow (large, yellow, circle). 3. (small, blue, circle) $>$ (large, circle). B: 1. (medium, blue, circle) $<$ (small, black, square). 2. (small, yellow, circle) $\infty \downarrow$ (blue, circle). C: 1. there is a small blue square. Now, let's analyze the position of the (large, yellow, circle): 1. A: (large, yellow, circle) \uparrow (medium, black, triangle). 2. A: (small, blue, circle) $>$ (large, yellow, circle). Based on the provided information, A: (medium, black, triangle) (large, yellow, circle) and (small, blue, circle) (large, yellow, circle). So, the correct answer is: "both of them".

Example 5:

Background: There are three blue shapes in a block that we call A. Two medium blue squares are below a big blue square. The first medium blue square is to the left of the other medium one which is touching the bottom edge of block A. There is another block below block A we call it B. A big blue triangle is touching the top edge of this block which is above a small black triangle, a big blue circle and a medium blue triangle. The medium triangle is touching the bottom edge of the block. The circle is below and to the left of the small triangle. It is above the medium triangle.

Question: Is the small triangle below the medium triangle?

Candidate Answers:["Yes", "NO", "Do not Know"]

Answer:

Let's analyze the arrangement of objects in block B to determine if the small triangle is below the medium triangle: B: 1. (big, blue, triangle) = top edge of B 2. (big, blue, triangle) \uparrow (small, black, triangle), (big, blue, circle) and (medium, blue, triangle) 2. (big, blue, circle) $\downarrow <$ (small, black, triangle) 3. (big, blue, circle) \downarrow (big, blue, triangle) 4. (medium, blue, triangle) = bottom edge B Based on the provided information, the small black triangle is above the medium blue triangle. So, the correct answer is: "No."

Table 16: Comparison of CoS(above) and CoT(below) prompts for BBH’s Temporal Sequence task.

Example 1:

Q: Today, Emily went to the museum. Between what times could they have gone?

We know that:

Emily woke up at 1pm.

Elizabeth saw Emily reading at the library from 2pm to 4pm.

Jessica saw Emily watching a movie at the theater from 4pm to 5pm.

Leslie saw Emily waiting at the airport from 5pm to 6pm.

William saw Emily buying clothes at the mall from 6pm to 7pm.

The museum was closed after 7pm.

Between what times could Emily have gone to the museum?

Options:

(A) 1pm to 2pm

(B) 6pm to 7pm

(C) 5pm to 6pm

(D) 2pm to 4pm

A: Let’s think step by step.

1pm/woke-up.

1pm-2pm//free.

2pm-4pm/reading.

4pm-5pm/watching

5pm-6pm/waiting.

6pm-7pm/buying

7pm/closure.

=> 1pm-2pm//free, So the answer is (A).

Q: Today, Emily went to the museum. Between what times could they have gone?

We know that:

Emily woke up at 1pm.

Elizabeth saw Emily reading at the library from 2pm to 4pm.

Jessica saw Emily watching a movie at the theater from 4pm to 5pm.

Leslie saw Emily waiting at the airport from 5pm to 6pm.

William saw Emily buying clothes at the mall from 6pm to 7pm.

The museum was closed after 7pm.

Between what times could Emily have gone to the museum?

Options:

(A) 1pm to 2pm

(B) 6pm to 7pm

(C) 5pm to 6pm

(D) 2pm to 4pm

A: Let’s think step by step.

Wake-up time: 1pm.

1pm-2pm: free.

2pm-4pm: reading at the library.

4pm-5pm: watching a movie at the theater.

5pm-6pm: waiting at the airport.

6pm-7pm: buying clothes at the mall.

The museum closure time: 7pm.

The only time when Emily could have gone to the museum was 1pm to 2pm. So the answer is (A).