Advancing Spatial Reasoning in Large Language Models: An In-Depth Evaluation and Enhancement Using the StepGame Benchmark

Fangjun Li[1], **David C. Hogg**[1], **Anthony G. Cohn**[1,2]
[1]School of Computing, University of Leeds, Leeds, UK
[2]Alan Turing Institute, UK
scfli@leeds.ac.uk, d.c.hogg@leeds.ac.uk, a.g.cohn@leeds.ac.uk

Abstract

Artificial intelligence (AI) has made remarkable progress across various domains, with large language models like ChatGPT gaining substantial attention for their human-like text-generation capabilities. Despite these achievements, spatial reasoning remains a significant challenge for these models. Benchmarks like StepGame evaluate AI spatial reasoning, where ChatGPT has shown unsatisfactory performance. However, the presence of template errors in the benchmark has an impact on the evaluation results. Thus there is potential for ChatGPT to perform better if these template errors are addressed, leading to more accurate assessments of its spatial reasoning capabilities. In this study, we refine the StepGame benchmark, providing a more accurate dataset for model evaluation. We analyze GPT's spatial reasoning performance on the rectified benchmark, identifying proficiency in mapping natural language text to spatial relations but limitations in multi-hop reasoning. We provide a flawless solution to the benchmark by combining template-to-relation mapping with logic-based reasoning. This combination demonstrates proficiency in performing qualitative reasoning on StepGame without encountering any errors. We then address the limitations of GPT models in spatial reasoning. We deploy Chain-of-thought and Tree-of-thoughts prompting strategies, offering insights into GPT's "cognitive process", and achieving remarkable improvements in accuracy. Our investigation not only sheds light on model deficiencies but also proposes enhancements, contributing to the advancement of AI with more robust spatial reasoning capabilities.

Introduction

Spatial reasoning, the ability to understand and navigate relationships in physical space, is a fundamental aspect of human cognition that significantly influences interactions with the environment. Enhancing spatial reasoning in AI models has the potential to enrich their comprehension of their surroundings and response to user interactions, leading to more advanced and immersive user experiences (Alomari et al. 2022). In recent years, AI has revolutionized numerous domains, from healthcare to finance to entertainment. Notably, OpenAI's large language models (LLMs), such as ChatGPT and GPT-4 (OpenAI 2023), have gained significant attention for their human-like text generation capabilities. However, despite their impressive abilities, LLMs encounter challenges in many logical reasoning aspects crucial for human communication, particularly spatial reasoning (Bang et al. 2023; Cohn and Hernandez-Orallo 2023).

One approach to evaluating spatial reasoning in an AI system is to use synthetic benchmarks such as StepGame (Shi, Zhang, and Lipani 2022) and SpartQA (Mirzaee and Rajaby 2021). Unfortunately, models like ChatGPT have shown unsatisfactory performance on these benchmarks. Improving the spatial reasoning capabilities of LLMs remains a primary focus to enhance their overall performance and understanding of complex environments.

Whilst examining the StepGame benchmark we discovered that it contains template errors that distort model performance evaluations. These errors were previously overlooked, leading to studies conducted on a flawed benchmark, inaccurately assessing the capabilities of the LLMs (Bang et al. 2023; Yang, Ishay, and Lee 2023). To rectify this issue, we present a more accurate version of the StepGame dataset for model evaluation, ensuring precise assessments of the models' true capabilities and limitations[1].

We then conducted evaluation tests on the rectified benchmark across various test subsets, few-shot sets, and models. We observed that larger GPT models demonstrate proficiency in mapping natural language text to spatial relations. However, they struggle with multi-hop spatial reasoning.

Our goal is not merely to critique, but also to propose potential improvements. To this end, we provide a flawless solution to the benchmark, and explore different approaches to enhance the spatial reasoning ability of LLMs.

The solution we propose for the benchmark entails combining template-based sentence-to-relation mapping with logic-based spatial reasoning. The logical reasoner used in this approach comes from (Yang, Ishay, and Lee 2023), where they integrated GPT-3 for the task. GPT-3 was employed to parse spatial descriptions into symbolic spatial relation representations, which were then passed to the logical program for spatial reasoning. This fusion resulted in significant improvement in StepGame, achieving state-of-the-art (SOTA) but not perfect results: around 90% accu-

[1]The data associated with this paper are openly available from the University of Leeds Data Repository. https://doi.org/10.5518/1468. All code is available at https://github.com/Fangjun-Li/SpatialLM-StepGame.

racy for lower hops and 88.3% accuracy for 10-hop reasoning. They attributed 10.7% faults to data-related issues. With our aforementioned work on rectifying the benchmark, We take a step further to delve into the two components, analyzing the performance of each on our filtered version of the dataset. Remarkably, we achieved 100% accuracy for almost all hops, with only 2 errors among 1000 test examples, which were due to GPT-3's incorrect semantic parsing. Building on this, we replaced the GPT-3 parser with our sentence-to-relation mapping method and combined it with the ASP reasoner, showcasing proficiency in performing qualitative reasoning without encountering any errors, thus demonstrating a method to achieve a perfect score on the corrected benchmark.

Neither our solution or the SOTA utilize LLMs for the actual spatial reasoning functionality. Thus, we proceed to enhance GPT's capabilities as a native spatial reasoner. To achieve this, we employ Chain-of-Thoughts (CoT) and Tree-of-Thoughts (ToT) prompting strategies.

CoT (Wei et al. 2022) incorporates a sequence of intermediate reasoning steps to facilitate problem-solving. However, when applied to StepGame, previous studies (Yang, Ishay, and Lee 2023) have shown that CoT does not consistently improve performance and may even reduce accuracy in complex k-hop reasoning tasks. This observation is attributed to the higher probability of errors occurring in lengthy CoT processes. On the other hand, research on other tasks (Zhou et al. 2022; Creswell, Shanahan, and Higgins 2022) has demonstrated that breaking down complex problems into simpler subproblems and solving them sequentially can be beneficial. Given the ambiguity in the decomposition of "thoughts"[2] within CoT, we propose refining the CoT prompt to empower language models to perform better in spatial reasoning tasks.

On the other hand, (Yao et al. 2023) introduced ToT, a framework enabling LLMs to explore multiple reasoning paths, and they demonstrated its effectiveness in improving problem-solving capabilities across tasks like the game of 24, creative writing, and mini crosswords. In our work, we customize the ToT approach for object-linking chain building, a crucial subproblem in addressing spatial reasoning benchmarks.

Our customized CoT method showcases its advantages more prominently in larger models such as GPT-4 and Davinci, maintaining accuracy even as the tasks become more complex. Our ToT approach demonstrates its strengths on the three GPT models: on the largest model, GPT-4, we are able to maintain an accuracy of around 90% even as the tasks become more complex. On Davinci, the accuracy is maintained at around 50%, while Turbo achieves a lower level of accuracy at around 30%.

By identifying current deficiencies and proposing enhancements, we aim to contribute to the ongoing discourse in AI development, pushing the boundaries of what LLMs can achieve. Ultimately, our investigation can pave the way for the development of advanced, intuitive, and user-friendly AI systems with robust spatial reasoning capabilities.

Related Work

The field of spatial reasoning in language with artificial intelligence has evolved through sustained efforts over time, with significant advancements achieved through both traditional methods and modern LLMs.

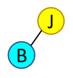
Early strides in spatial reasoning in language were marked by the development of formal structures to represent spatial relationships. (Kordjamshidi, Moens, and van Otterlo 2010) proposed a spatial ontology to formalize the representation of spatial relations. This work laid the groundwork for the subsequent introduction of text-based spatial role labeling (Kordjamshidi, Van Otterlo, and Moens 2011), which aims to convert text into formal spatial representations.

Then comes synthetic tasks designed to evaluate the text understanding and spatial reasoning capabilities of learning algorithms. The positional reasoning task (Task 17) in the bAbI dataset (Weston et al. 2015) is for spatial reasoning and requires models to reason using one or two sentences, which makes this task comparatively simple. (Shi, Zhang, and Lipani 2022) advanced this field by creating the StepGame benchmark to evaluate multi-hop spatial reasoning in text, with richer variety in spatial relation descriptions. Both of these datasets emphasize directional spatial relations (Cohn and Hazarika 2001; Skiadopoulos and Koubarakis 2001; Cohn and Renz 2008; Chen et al. 2015). Three spatial QA datasets: SpartQA(Mirzaee and Rajaby 2021), SPARTUN, and RESQ (Mirzaee and Kordjamshidi 2022) expanded the resource landscape by encompassing wider-ranging spatial language expressions, posing challenges for traditional logical programming, and are important benchmarks for evaluating LLMs' spatial reasoning capabilities.

Concurrently, the advent of LLMs such as OpenAI's ChatGPT has opened up fresh pathways for spatial reasoning. These models, leveraging transformer architectures, can generate human-like text and handle complex linguistic structures. However, while these models are indeed impressive, their capabilities in spatial reasoning are yet to be fully explored and exploited. One recent approach to assess these capabilities was taken by (Bang et al. 2023), who put ChatGPT to the test using SpartQA and StepGame. Despite the generally advanced capabilities of ChatGPT, the model showed shortcomings in these tasks, signaling a need for further enhancements in the realm of spatial reasoning.

A promising technique known as 'prompt engineering' (Bommasani et al. 2021) has been making its mark recently. This approach involves crafting specific prompts to guide the responses of the models, leading to outputs that are more contextually apt and insightful. This method demonstrates significant potential in enhancing the capabilities of LLMs like ChatGPT in various domains (Li, Hogg, and Cohn 2022), including the challenging area of logical reasoning (Wang et al. 2023). For instance, when faced with multi-step reasoning tasks, a method called 'few-shot chain-of-thought' (CoT) prompting (Zhang et al. 2022) comes into

[2]In this paper we use the word 'thoughts' in the same way as is now being used in the literature on CoT and ToT, whilst noting that these are not thoughts in the human sense but rather generated coherent units of text , serving as intermediate steps in a problem solving setting, and without wishing to ascribe an anthropomorphic meaning to the word.



Story: J is diagonally above B to the right at a 45 degree.
Clock-position: If B is the center of a clock face, J is located between 2 and 3.
Compass-position: J is north east of B.

	Relation extraction	1-hop reasoning
	J is on the upper right of B./ B is is on the lower left of J.	**Question:** What is the relation of the agent J to the agent B? **Answer**: upper-right
Semantic Parsing	top_right("J", "B") down_left("B", "J")	**Question:** What is the relation of the agent B to the agent J? **Answer**: lower-left

Figure 1: An illustrative example for demonstrating relation extraction and 1-hop spatial reasoning.

10- hop story:
1. T is slightly off center to the top left and I is slightly off center to the bottom right.
2. M is at the bottom of X.
3. Z presents right to X.
4. N is lower right of I.
5. S is positioned above J and to the right.
6. F is above T at 10 o'clock.
7. A is to the upper left of W.
8. A is at the bottom of F.
9. N is sitting in the right direction of J.
10. M is placed at the lower right of S.

What is the relation of the agent X to the agent F?

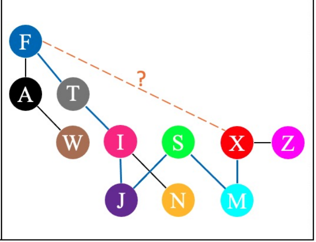


Figure 2: Example of 10-hop reasoning, featuring a question regarding two entities that are not directly connected in the stories. The diagrams on the right do not form part of the input to the AI system but are for illustrative purposes only.

play. These demonstrations enable LLMs to explicitly generate reasoning steps, thereby improving their accuracy in reasoning tasks. This technique involves a handful of manually curated step-by-step reasoning demonstrations.

As we review these developments, it is clear that while significant progress has been made, challenges remain in both traditional and LLM approaches to spatial reasoning. The limitations of models like ChatGPT indicate the need for continued research and enhancement strategies. This paper aims to contribute to this by examining these limitations more closely and proposing potential avenues for improvement. We aim to explore the limit of GPT as a general problem solver that explores its own thoughts and guides its own exploration with deliberate reasoning as heuristics.

The StepGame Benchmark for Evaluating Spatial Reasoning

In this paper, we focus on StepGame, in line with other studies that evaluate ChatGPT's spatial reasoning proficiency using StepGame and SpartQA. StepGame comprises story-question pairs in natural language, The objective is to answer questions regarding the spatial relations between two specified entities. The StepGame benchmark contains two sets of data: a *clean* set where there are precisely k facts given for any given k-hop instance, and a *noise* set where there are more than k facts given, and the extra facts are distracting.

Spatial Reasoning Types

- **1-Hop Spatial Reasoning**. In 1-hop reasoning, we are given a relation description between two entities and are asked about the spatial relation from one entity to the other. 1-hop relation reasoning and relation extraction can be considered similar processes. As exemplified in Figure 1, consider the story where *J is diagonally above B to the right at a 45-degree angle*. The question is *What is the spatial relation of agent J to agent B?*. This is similar to relation extraction. However, if we change the question to *What is the spatial relation of agent B to agent J?*, it needs a reverse reasoning process *top_right("J", "B")* $\rightarrow$ *down_left("B", "J")*. Both expressions are correct representations for relation extraction.
- **Multi-Hop Spatial Reasoning**. Figure 2 provides one example of 10-hop reasoning, which is from the 'clean' set. The questions ask about the relation between two objects, either directly or indirectly connected. Multi-hop reasoning adds more complexity to the problem, as it involves a greater number of provided relations. To solve the problem, one needs to identify useful relations and then proceed with relation inference step by step.

Problems with the Dataset

Eight spatial relations (*top, down, left, right, top-left, top-right, down-left, and down-right*) are utilized for the story generation of StepGame. These relations are expressed through sentences in natural language. All sentences/statements are based on a crowd-sourced template base[3]. Each "story" is accompanied by a question that seeks to identify the relations between two objects, and it is labeled according to the intended relations at the time of story creation, rather than the actual sentences used. A template is considered to contain an error if the meaning conveyed by the sentence does not align with the relationship that was intended to be expressed during the creation of stories and labels.

Table 1 presents a detailed enumeration of errors in the relation-to-sentence mappings identified in StepGame. Out of the 214 templates examined, 14 were found to contain errors. Of the eight different relationship mappings available, only $o_1\_above\_o_2$ and $o_1\_left\_o_2$ are devoid of mistakes. The question arises as to why there are so many errors in the crowd-sourced expressions; presumably this is down to insufficient quality control over the crowdworker reponses.

For each k value, the StepGame dataset includes 10,000 test samples. Table 2 displays the percentage of examples containing sentences derived from incorrect templates, which hints at a rising trend in inaccuracies as k increases, suggesting a potential cumulative impact.

Among these 14 incorrect templates, four cannot be remedied in existing StepGame benchmark examples.

- $o_1\_upperright\_o_2$: Object A is above object o_1.
- $o_1\_upperleft\_o_2$: o_1 is diagonally left and above o_1.
- $o_1\_lowerright\_o_2 \mid o_1\_upperleft\_o_2 \mid o_1\_upperright\_o_2$: o_1 is to the right and above o_2 at an angle of about 45 degrees.

[3]https://github.com/ZhengxiangShi/StepGame/blob/main/Code/template.py

Relation	Original Incorrect Template
right	o_2 and o_1 are parallel, and o_2 on the right of o_1. o_2 and o_1 are parallel, and o_2 is to the right of o_1. o_2 and o_1 are horizontal and o_2 is to the right of o_1. o_2 and o_1 are both there with the object o_2 is to the right of object o_1.
below	o_2 is placed at the bottom of o_1. o_2 is at the bottom of o_1 and is on the same vertical plane. o_2 presents below o_1.
lowerleft	o_2 is there and o_1 is at the 10 position of a clock face. o_2 is positioned below o_1 and to the left.
upperright	Object A is above object o_1 and to the right of it, too. o_2 is diagonally to the upper right of o_1.
lowerright	o_1 is to the right and above o_2 at an angle of about 45 degrees.
upperleft	o_1 is to the right and above o_2 at an angle of about 45 degrees. o_1 is diagonally left and above o_1.

Table 1: Incorrect sentence templates in StepGame. The Relation column signifies relation for $o_1\_relation\_o_2$.

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
Clean	7.64	15.03	20.87	26.39	32.54	37.66	41.71	47.20	51.50	**54.29**
Noise	20.43	30.19	34.59	48.18	57.13	61.14	63.60	69.45	72.84	**74.21**

Table 2: Percentage of incorrect instances out of all instances over k=1–10 test sets.

- $o_1\_lowerleft\_o_2 \mid o_1\_upperleft\_o_2$: o_2 is there and o_1 is at the 10 position of a clock face.

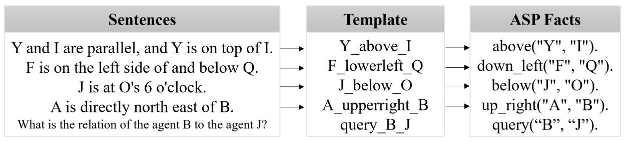
The first and second templates are irreparable because it is impossible to identify what o_2 is when sentences are formed using them. The third and fourth templates cannot be corrected since they were applied to multiple spatial relations, although each accurately represents just one. For example, for the sentence 'Q is to the right and above P at an angle of about 45 degrees', three mapping relations exist: $Q\_upperright\_P$, $Q\_lowerright\_P$, and $Q\_upperleft\_P$. Although this sentence expresses the meaning $Q\_upperright\_P$, it is uncertain which candidate was used for the label. For such templates, a unique correction could not be chosen, necessitating the removal of the sentences that use these template from the dataset.

Methods

Solution for the Corrected Benchmark

Our error-free approach is entirely logic-based, without the use of LLMs. We begin by performing template-based sentence-to-relation mapping, akin to semantic parsing. Then, we employ ASP for logical reasoning, utilizing the ASP reasoner introduced by (Yang, Ishay, and Lee 2023). These two components operate independently:

- **Sentence-to-Relation Mapping**. When presented with a natural language relation description r, we first identify the template used in r through a comparison with the template base. This template is symbolized as $o_0\_\nu\_o_1$. Then, we convert this template form into a structured representation $\nu(o_0, o_1)$, where o_0 and o_1 correspond to the two objects mentioned in r, and ν signifies the spatial relation between o_0 and o_1. Specifically, for questions inquiring about relations from the start object o_0 to the target object o_t, the template is $query\_o_0\_o_t$, and the corresponding ASP fact is represented as $query(o_0, o_t)$. Illustrative examples of this process can be found in Figure 3.
- **Logical Reasoning with ASP**. The logical facts $\nu(o_0, o_1)$, generated through semantic parsing for all relations in the story R, are used as input to the ASP module for spatial reasoning. The ASP module was implemented using Clingo and includes rules specifically tailored for StepGame. These rules transform StepGame into a qualitative spatial reasoning problem in a 2D grid space. These rules incorporate offsets for 9 spatial relations, such as *offset*$(right) = (0, 1)$ and *offset*$(lower\text{-}left) = (-1, -1)$. The main rule in the ASP module calculates the location of o_0 to o_1 by adding the offsets $\nu(o_0, o_1)$.



Sentences	Template	ASP Facts
Y and I are parallel, and Y is on top of I.	Y_above_I	above("Y", "I").
F is on the left side of and below Q.	F_lowerleft_Q	down_left("F", "Q").
J is at O's 6 o'clock.	J_below_O	below("J", "O").
A is directly north east of B.	A_upperright_B	up_right("A", "B").
What is the relation of the agent B to the agent J?	query_B_J	query("B", "J").

Figure 3: Sentence-to-Relation Mapping Examples.

While this approach offers a solution to the StepGame benchmark challenge, it does require prior familiarity with the templates and mandates updates to the template base when confronted with new stories employing novel templates. In contrast, an LLM approach holds the potential to flexibly adjust to unfamiliar templates. Additionally, the method's dependence on customized rules within the logical program constitutes another aspect to be mindful of.

Chain-of-Thought (CoT) Prompting

We devised a customized CoT for the spatial reasoning task. The core idea of CoT is to introduce a chain of thoughts $c_1, \ldots, c_i, \ldots, c_n$ to bridge input x and output y, where i represents i-th step. In our customized CoT for StepGame, x consists of the task description, few-shot examples, relation story, and question, while y represents the answer regarding the relations between the queried objects (from the start object o_i to the target object o_t). Each thought c_i is to identify direct spatial connections between objects (o_i and o_{i+1}). We take CoT a step further by decomposing each step of thought c_i to explore the potential advantages of incorporating a coherent and detailed reasoning process.

Thought Categorisation. We categorise the thought into three types: link establishment thoughts c^{link}, relation mapping thoughts c^{map}, and coordinate calculation thoughts c^{calcu}. At each reasoning step, these three types of thought are sequentially sampled as a continuous language sequence $c_i = [c_i^{link}, c_i^{map}, c_i^{calcu}]$ using the LLM.

1. c_i^{link}: Guide the LLM to examine all relations in the story ($R = [r^1, \ldots, r^j, \ldots, r^k]$) and select r^j for the i-th step

for k-hop reasoning, ensuring it directly describes the relation with o_i and has not been used in any previous step. For the start object ($i = 0$), we use the prompt "Start with o_0. According to" and for the middle objects ($i \geq 1$), we use the prompt "Then search for o_i. According to". Full details of the prompts can be found in the Appendix[4].

2. c_i^{map}: Map r^j to a simple relation description such as "o_i is to the ν of o_{i+1}," where ν represents the key spatial relation from o_i to o_{i+1}. The prompt "This means" helps the LLM perform this mapping.
3. c_i^{calcu}: Use r^j to calculate the coordinates of o_{i+1}. We set o_o at (0,0), and each spatial relation is assigned an offset to determine the positions of the objects. The prompt "$o_{i+1} = o_i + \mathit{offset}(r^j) = (x_{o_i}, y_{o_i}) + (x_\nu, y_\nu) = (x_{o_{i+1}}, y_{o_{i+1}})$" instructs the LLM on the calculation process. It computes the coordinates of o_{i+1} and generates the output like "Therefore, B is at $(x_{o_{i+1}}, y_{o_{i+1}})$."

Tree-of-Thoughts (ToT) Prompting

Algorithm 1 is designed to enhance the reasoning chain-building process, allowing LLMs to consider different pathways. This is useful because during the search for relations with an object, distracting connections may arise, as shown in Figure 2. However, it is essential to follow a correct sequence to successfully reach the target object. If an LLM mistakenly tracks an incorrect sequence, it could get stuck in a dead end leading to incorrect reasoning conclusions such as "The story does not provide direct spatial information."

The algorithm initiates by prompting the LLM to set up the initial tree state, denoted as S_0, using the input x, which comprises a story and a question. S_0 is in the form "chain: o_0 −>, target: o_t, unused: R". R represents all connections between objects in the story, in the form of $object1$-$object2$. Then it proceeds to construct a linking chain from o_0 to o_t in iterative steps, wherein for the i-th step ($1 \leq i \leq 10$), the LLM considers the tree state S_{i-1} built up to that step. If no state s in S_{i-1} reaches o_t, the LLM is prompted to generate j candidate thoughts for each s in the current set of states, S_i ($j = 2$ in this paper). G prompts the LLM to search for a potential object o_i connected to the current object o_{i-1} from the unused relations R_{i-1}^{unused}. A check is made ($CheckExtn(c)$)) to see if the proposal made is a real candidate extension. For all candidate thoughts, V prompts the LLM to evaluate the state to determine if the chain can proceed with o_i and the updated R_{i-1}^{unused} to reach o_t. The top-rated b tree states in $S_i^{'}$ are selected as S_i. When there is a state s_f which reaches o_t, the L will be prompted with the linking chain construction prompt (Appendix D.4) to form the final links l.

- **Thought Generation** $G(s, j)$. Given a tree state s, we let the LLM propose j thoughts using the thought generation prompt "Use relations listed in unused relations to enumerate all potential expansions of the chain by considering unused relations that exhibit a direct link to the last object within the chain." In our experiment, we set $j = 2$, meaning that we instruct the LLM to generate content twice for each state $s \in S_{i-1}$.

[4]The ArXiv version of this paper includes the Appendix containing prompting examples.

Algorithm 1: Our ToT Approach

```
Require: LLM, input $x$
1: $S_0 \leftarrow Init(x)$
2: $i \leftarrow 1$
3: while no $s_f \in S_{i-1}$ has arrived at $o_t$ do
4:     $S_i^{'} \leftarrow \{s{\cdot}c | c \in G(s,j) \wedge ChainExtn(c) \wedge s \in S_{i-1}\}$
5:     if $S_i^{'} = \emptyset$ then return failure
6:     $S_i \leftarrow select(b, \{\langle s, y\rangle | s \in S_i' \wedge y = \Sigma_1^n \sigma(V(s))\})$
7:     $i = i + 1$
8: end while
9: return $Link(s_f)$
```

- **State Evaluation** $V(s)$. Our approach involves a classification methodology, using the designed value prompt "Evaluate whether the chain can reach the target (sure/-likely/impossible). If the chain has already reached the target, it's 'sure'. If the unused relations include the current object, it's 'likely'. If there are no unused relations that include the current object, it's 'impossible'." This prompt guides the LLM to sequentially examine all newly generated states $s \in S_i^{'}$ n times – using the stochasticity of the LLM with a non zero temperature to increase the reliability of the scoring. The three types of outputs - 'sure', 'likely', and 'impossible' - are converted into numerical scores using a function $\sigma()$ to facilitate the selection process among all newly generated states.
- **Search Algorithm** The choice between utilizing breadth-first search (BFS) or depth-first search (DFS) depends on the tree structure. In the StepGame benchmark, the tree depth is limited ($depth \leq 10$), and the number of thought candidates k for each step is also limited ($width \leq 3$ in most cases). However, a deeper search does not necessarily guarantee better results. In certain scenarios, o_0 and o_t may be directly connected in one relation statement, allowing for shorter linking chains between them, which is preferable. Therefore, we opt for BFS to maintain all promising states. We set the breadth width $b = 3$, maintaining the three most promising linking-chain states per step. The criterion for stopping searching is set when the linking chain arrives at the target object.

Our ToT approach is used to construct the reasoning chain from o_0 to o_t. Subsequently, the spatial relation between these objects is computed following the previous CoT prompting method, with the use of c^{map} and c^{calcu}.

Experimental Design

Model Settings

We use the Azure OpenAI Service for ChatGPT (3.5-Turbo) and GPT-3 (Davinci), and GPT-4 API access. To yield more concentrated and deterministic results, we set the temperature to 0 in CoT experiments. In ToT experiments, we follow

(Yao et al. 2023), setting the temperature to 0.7 for generating varied thought proposals. The remaining parameters were left at the standard configurations for these models.

Different Test Subsets It is common practice in the studies cited (Bang et al. 2023), (Yang, Ishay, and Lee 2023) to use a subset of 30 or 100 test examples from the full set of 10,000 for each k value. While this method helps in conserving token usage, it could potentially introduce biases or inaccurate estimations of the model performance.

We examine the effect of the number of test examples. Specifically, we wanted to determine whether evaluating on a limited number of test examples could introduce inaccuracies. To achieve this, we conducted tests on a clean, filtered test set for k-hop reasoning ($k \in [1, 10]$), thereby covering a range of task complexities. Tests were carried out on 30, 100, and 1000 test examples to assess the impact of the number of test examples on the evaluation.

Different Few-Shot Sets We created three different few-shot prompting sets to evaluate the influence of input examples in prompts.

- *clean 5shot(1,3,5,7,10)*: Create a prompt consisting of five examples, with one example each from tasks requiring 1-hop, 3-hop, 5-hop, 7-hop, and 10-hop reasoning.
- *clean 10shot*: Formulate a prompt using ten examples, each one derived from a distinct k-hop task in clean set.
- *clean 5shot separate*: Construct a prompt for each k-hop reasoning task, utilizing five examples from the corresponding k-hop training set as few-shot examples.

Experimental Results

Evaluation Results

Influence of Scale of Test Examples We employ the *clean 10shot* prompting setting. The results are presented in the left subplot of Figure 4. Upon evaluation of the expanded test set comprising 1000 examples, the model shows a uniform decrement in performance as k increases from 1 to 10. This trend indicates the increased complexity as the number of hops increases. With smaller test sets of 100 or 30 examples, the trend is less consistent, and there are occasional increases in performance at certain hop levels. The variance in performance, particularly for the 30-example test set, may indeed be larger. This could be due to the smaller sample size providing less comprehensive coverage of the potential range of tasks, leading to more fluctuations in performance. This indicates larger test sets can provide a more stable and reliable indicator of a model's performance across different complexity levels (i.e., number of hops).

Influence of Prompting Examples The middle subplot in figure 4 indicates that the choice of prompting strategy can impact the model's ability to handle tasks of varying complexity. Similar to the previous data, all prompting strategies show a trend of decreasing accuracy as the number of hops increases. This trend is consistent and suggests that the complexity of the tasks grows with the number of hops.

The performances of the three methods are close. While differences exist at specific hop levels, no single method consistently outperforms the others across all hop levels. Interestingly, clean 5shot (1,3,5,7,10) performs better than clean 10shot (1∼10) at almost every hop level. This suggests that selecting examples from a wider range of hop levels (1,3,5,7,10) can be more beneficial than having an example from each hop level from 1 to 10.

Influence of Models As indicated in a recent study (Ye et al. 2023), Turbo demonstrates comparable performance to Davinci across many tasks. However, it falls short in the machine reading comprehension, part-of-speech, and relation extraction tasks, potentially owing to its smaller model size. The StepGame spatial reasoning task requires the comprehension of sequential spatial connections and the ability to draw deductions from them. According to the right subplot of Figure 4, the Davinci model generally outperforms the Turbo model across varying levels of task complexity (number of hops). The differences in performance between the two models are more significant at lower complexity levels, but they appear to converge as the complexity increases.

	left/ right	above /below	lower_left/ upper_right	lower_right/ upper_left
total	44	53	50	53
text-curie-001	11	41	30	37
text-davinci-003	**0**	**0**	**0**	2
gpt-3.5-turbo	2	2	3	**1**

Table 3: The relation extraction performance of GPT. The numbers in rows 2-4 are incorrect predictions numbers.

Results of the Improved Methods

Resolution for the Benchmark The results of our resolution (sentence-to-relation mapping + ASP-based reasoning) are displayed in the 'Map+ASP' row of Table 4. The numbers in the table indicate accuracy scores, with higher values indicating better performance. This demonstrates the proficiency achieved in spatial relation mapping and multi-hop spatial reasoning, all without encountering any errors.

GPT for Relation Extraction + ASP for Reasoning We analyze the performance of GPT in the relation extraction subtask, as outlined in Table 3. Curie has the highest number of wrong predictions across different relations, Davinci and Turbo show better performance.

The state-of-the-art results achieved by (Yang, Ishay, and Lee 2023) (using GPT-3 for semantic parsing and ASP for reasoning) are presented in the "SOTA" row of Table 4. They achieve approximately 90% accuracy for lower hops and 88.3% accuracy for 10-hop reasoning. They attribute 10.7% of the inaccuracies to data-related concerns.

We provide an evaluation of their approach onthe corrected dataset, with the results displayed in the "Curie+ASP" and "Davinci+ASP" rows. Among the 1000 test examples (100 for each k), only 2 errors were encountered with Davinci. caused by semantic parsing: the sentence "If E is the center of a clock face, H is located between 2 and 3." was parsed incorrectly as $right(\text{“}H\text{”}, \text{“}E\text{”})$, but

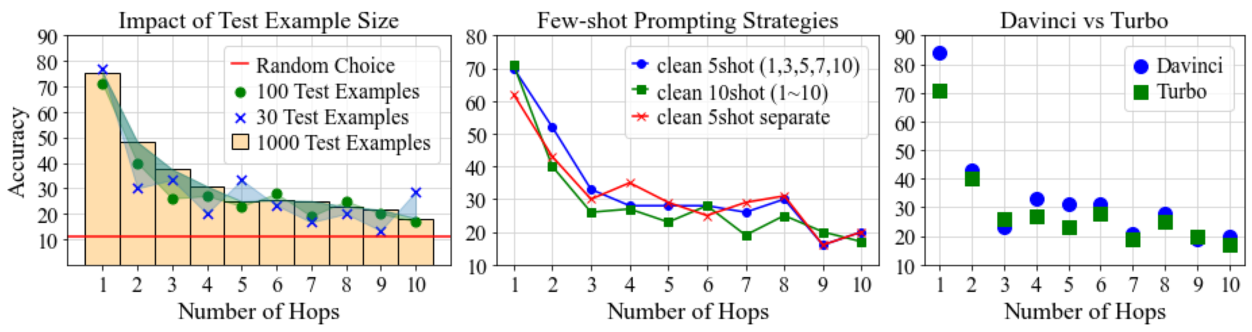


Figure 4: Accuracy comparison for varying numbers of hops (1-10) on the clean test set. On the left, we show the performance variation of the Turbo model with *10shot* prompting over different test set sizes (30, 100, and 1000 examples). The middle section illustrates the performance of the Turbo model under three distinct prompting settings: *5shot(1,3,5,7,10)*, *10shot*, and *5shot separate*. The right portion showcases the performance of two models - Davinci and Turbo - using *10shot* prompting.

		k=1	**k=2**	**k=3**	**k=4**	**k=5**	**k=6**	**k=7**	**k=8**	**k=9**	**k=10**
Map+ASP		**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Curie+ASP		46	43	42	59	67	67	57	56	58	61
Davinci+ASP		**100**	**100**	99	**100**	**100**	99	**100**	**100**	**100**	**100**
SOTA		92.6	89.9	89.1	93.8	92.9	91.6	91.2	90.4	89.0	88.3
Turbo	base	62	43	30	35	29	25	29	31	16	20
	CoT	/	34	40	36	28	28	26	31	25	24
	ToT_CoT	/	/	35	35	25	45	15	40	40	35
Davinci	base	77	42	21	26	25	30	23	23	22	22
	CoT	/	48	53	46	46	48	40	45	41	32
	ToT_CoT	/	/	65	50	45	60	50	50	55	50
GPT-4	base	100	70	55	45	40	25	40	35	35	25
	CoT	/	**80**	75	**95**	85	85	**90**	80	60	65
	ToT_CoT	/	/	**85**	85	**90**	**90**	85	**90**	**100**	**95**

Table 4: Accuracy comparison of GPT models on revised StepGame using different methods.

supposed to be $up\_right(“H”, “E”)$.

CoT and ToT The experimental results in Table 4 involving GPT-4 and ToT are based on a test set comprising 20 instances considering token usage, while for Davinci and Turbo, we used a larger test set of 100 samples. The results for the base and CoT methods were obtained using the *5shot separate* prompting on the *clean* set. All the ToT_CoT results presented in the table involve the use of GPT-4 for building the linking chain, followed by the application of Turbo, Davinci, and GPT-4 for CoT reasoning with the constructed linking chain. The GPT-4 model exhibits superior performance across nearly all settings. With the basic input-output prompt, despite starting at 100% accuracy for $k = 1$, its accuracy dips to 25% for $k = 10$, indicating that even the most powerful GPT model struggles to maintain accuracy as task complexity rises. Humans would probably find this challenging too.

With the implementation of our CoT and ToT approach, the GPT-4 model demonstrates significant performance enhancements for more complex tasks (ranging from $k = 2$ to $k = 10$). Our ToT and CoT method considerably enhances the performance of the Davinci and GPT-4, particularly in larger hops. For the Turbo model, although our CoT method brings improvements as k increases, the gains are not as profound as those observed with the Davinci and GPT-4. This could be attributed to the long length of our prompts, requiring a nuanced understanding of coordinates and relations.

Conclusion

This paper has introduced a revised version of the StepGame benchmark, correcting template errors that distort model performance evaluations, leading to a more accurate evaluation of the spatial reasoning capabilities of AI systems attempting the challenge. We highlight Davinci and Turbo's abilities in mapping texts to spatial relations and their limitations in multi-hop spatial reasoning. Our solution combines template-to-relation mapping with logic-based reasoning, effectively addressing challenges in this task. We also enhance LLMs' spatial reasoning ability through prompt engineering, using CoT and ToT strategies.

This paper focuses on StepGame; future studies could extend our findings to other benchmarks. Our methods are suitable for adaptation to various 2D grid-based directional spatial tasks, such as the bAbI (task 17). This adaptation would involve customizing the template for the ASP-based solution and modifying task descriptions and few-shot examples for CoT and ToT approaches. For tasks that require a combination of directional, topological, and distance reasoning, like SpartQA, it would be necessary to integrate additional rules and ontology into both the ASP program and the prompts to LLMs for effective solution development.

The effective resolution of the StepGame benchmark prompts a need for more challenging versions. While having a well-defined set of spatial relations converted into natural language using a set of templates is appealing, it leads to controlled natural language which is more amenable to special purpose reasoning. Finding a way to generate more naturalistic problem statements automatically would therefore be highly desirable. Additionally, the current independent use of LLMs and logic programs suggests a potential research direction towards integrating these tools for more comprehensive and cohesive problem-solving strategies.

Acknowledgments

This work has been partially supported by Microsoft Research - Accelerating Foundation Models Research program, with the provision of Azure resources to access GPT. This work was also partially supported by the Turing's Defence and Security programme through a partnership with the UK government in accordance with the framework agreement between GCHQ and The Alan Turing Institute.

Author Contributions

AGC and DCH proposed the initial line of work. FL designed the actual implementation, performed all the evaluations, and wrote the initial paper draft. DCH and AGC supervised FL. All authors contributed to subsequent paper revisions.

References

Alomari, M.; Li, F.; Hogg, D. C.; and Cohn, A. G. 2022. Online perceptual learning and natural language acquisition for autonomous robots. *Artificial Intelligence*, 303: 103637.

Bang, Y.; Cahyawijaya, S.; Lee, N.; Dai, W.; Su, D.; Wilie, B.; Lovenia, H.; Ji, Z.; Yu, T.; Chung, W.; et al. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Bommasani, R.; Hudson, D. A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M. S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Chen, J.; Cohn, A. G.; Liu, D.; Wang, S.; Ouyang, J.; and Yu, Q. 2015. A survey of qualitative spatial representations. *The Knowledge Engineering Review*, 30(1): 106–136.

Cohn, A. G.; and Hazarika, S. M. 2001. Qualitative spatial representation and reasoning: An overview. *Fundamenta informaticae*, 46(1-2): 1–29.

Cohn, A. G.; and Hernandez-Orallo, J. 2023. Dialectical language model evaluation: An initial appraisal of the commonsense spatial reasoning abilities of LLMs. *arXiv preprint arXiv:2304.11164*.

Cohn, A. G.; and Renz, J. 2008. Qualitative spatial representation and reasoning. *Foundations of Artificial Intelligence*, 3: 551–596.

Creswell, A.; Shanahan, M.; and Higgins, I. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*.

Kordjamshidi, P.; Moens, M.-F.; and van Otterlo, M. 2010. Spatial role labeling: Task definition and annotation scheme. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, 413–420. European Language Resources Association (ELRA).

Kordjamshidi, P.; Van Otterlo, M.; and Moens, M.-F. 2011. Spatial role labeling: Towards extraction of spatial relations from natural language. *ACM Transactions on Speech and Language Processing (TSLP)*, 8(3): 1–36.

Li, F.; Hogg, D. C.; and Cohn, A. G. 2022. Ontology Knowledge-enhanced In-Context Learning for Action-Effect Prediction. In *Advances in Cognitive Systems*. ACS-2022.

Mirzaee, R.; and Kordjamshidi, P. 2022. Transfer Learning with Synthetic Corpora for Spatial Role Labeling and Reasoning. *arXiv preprint arXiv:2210.16952*.

Mirzaee, R.; and Rajaby, H. 2021. SpartQA: A Textual Question Answering Benchmark for Spatial Reasoning. In *The 2021 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-2021)*.

OpenAI. 2023. GPT-4 Technical Report. *ArXiv*, abs/2303.08774.

Shi, Z.; Zhang, Q.; and Lipani, A. 2022. Stepgame: A new benchmark for robust multi-hop spatial reasoning in texts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11321–11329.

Skiadopoulos, S.; and Koubarakis, M. 2001. Composing cardinal direction relations. In *International Symposium on Spatial and Temporal Databases*, 299–317. Springer.

Wang, L.; Xu, W.; Lan, Y.; Hu, Z.; Lan, Y.; Lee, R. K.-W.; and Lim, E.-P. 2023. Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. *arXiv preprint arXiv:2305.04091*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Chi, E.; Le, Q.; and Zhou, D. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Weston, J.; Bordes, A.; Chopra, S.; Rush, A. M.; Van Merriënboer, B.; Joulin, A.; and Mikolov, T. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Yang, Z.; Ishay, A.; and Lee, J. 2023. Coupling Large Language Models with Logic Programming for Robust and General Reasoning from Text. *arXiv preprint arXiv:2307.07696*.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T. L.; Cao, Y.; and Narasimhan, K. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Ye, J.; Chen, X.; Xu, N.; Zu, C.; Shao, Z.; Liu, S.; Cui, Y.; Zhou, Z.; Gong, C.; Shen, Y.; et al. 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*.

Zhang, Z.; Zhang, A.; Li, M.; and Smola, A. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q.; et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.