# Benchmarking and Enhancing Spatial Reasoning in Large Language Models

**Fangjun Li**

Submitted in accordance with the requirements for the degree of

*Doctor of Philosophy*

University of Leeds

School of Computer Science

April, 2025

# Intellectual Property Statement

This candidate confirms that the work submitted is her own, except where work which has formed part of jointly authored publications has been included. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others. The contribution of the candidate and other authors to this work is explicitly indicated in the following.

**Li, F., Hogg, D. C. and Cohn, A. G.** "Advancing Spatial Reasoning in Large Language Models: An In-Depth Evaluation and Enhancement Using the StepGame Benchmark." In 38th AAAI Conference on Artificial Intelligence (AAAI), 2024.

**Li, F., Hogg, D. C. and Cohn, A. G.** "Reframing Spatial Reasoning Evaluation in Language Models: A Real-World Simulation Benchmark for Qualitative Reasoning." In 33rd International Joint Conference on Artificial Intelligence, (IJCAI), 2024.

# Acknowledgements

I would like to express my deepest gratitude to my family and my partner, Fengming Lin, for their unwavering support and patience throughout the duration of my PhD journey. Their encouragement and understanding have been pillars of strength in my pursuit of academic and personal growth.

I am also immensely grateful to my supervisors, Anthony (Tony) Cohn and David Hogg, whose mentorship and expertise have been instrumental in shaping my research and academic career. Despite their busy schedules, they consistently made time for weekly meetings where we discussed my research progress and help me overcome challenges I faced. They supported my participation in conferences, guiding me through presentations and teaching me effective communication of our work.

David has been instrumental in keeping me updated with the latest research and developments in AI. His innovative and thought-provoking ideas have greatly enriched my understanding, providing support during challenging times when I faced disappointing results. His ability to analyze and understand problems from different points of view has been essential in restoring my confidence and rekindling my enthusiasm for research.

Tony introduced me to the field of spatial reasoning, instructing me not only on how to identify and tackle problems but also on how to summarize research works into papers. He thoroughly reviewed each of my drafts to ensure reliability and coherence and has significantly honed my academic writing skills. Additionally, he devoted time to discussing my research ideas, shared many innovative concepts, and consistently encouraged me to delve deeper into the underlying issues.

Special thanks to my transfer viva examiner, Brandon Bennett, and annual reviewer, Eric Atwell, for their valuable instructions and engaging discussions regarding my research. I would also like to express my sincere gratitude to my internal examiner, Sergei Sharov, and external examiner, Steven Schockaert, for their insightful feedback.

Finally, I gratefully acknowledge the support from the Microsoft Research - Accelerating Foundation Models Research program, which provided Azure resources for GPT access.

# Abstract

Spatial reasoning is essential for both human cognition and machine intelligence in understanding and navigating spatial relationships between objects. Despite significant advances in large language models (LLMs) like ChatGPT, spatial reasoning remains a challenging area. This thesis aims to make a contribution to addressing this issue.

Firstly, we analyze the existing benchmarks: bAbI, StepGame, SpartQA, and SpaRTUN, providing initial LLM evaluations and examining their limitations. Results on StepGame demonstrate LLMs' proficiency in mapping natural language to spatial relations, while also highlighting challenges in multi-hop reasoning tasks. As an alternative approach, this thesis also investigates using LLMs to translate the spatial reasoning tasks into a logical format appropriate for an answer set programming reasoner. Experiments demonstrate that this neuro-symbolic approach results in almost perfect accuracy scores on StepGame.

Secondly, the thesis investigates advanced prompting strategies, specifically Chain-of-Thought (CoT) and Tree-of-Thought (ToT) methods, to enhance LLMs' spatial reasoning capabilities. These strategies decompose complex reasoning tasks into manageable steps, significantly improving performance on spatial reasoning benchmarks. CoT and ToT approaches show substantial improvements in accuracy, particularly with complex, multi-hop tasks.

Thirdly, the thesis introduces a novel benchmark based on realistic 3D simulation data, featuring diverse room layouts with various objects and their spatial relationships. This benchmark encompasses a wide range of qualitative spatial relationships, such as topological, directional, and distance relations, and presents scenarios from different viewpoints to reflect real-world complexities. Alongside the benchmark itself, the code is available online, thus allowing arbitrary further versions to be created. A further contribution of this benchmark is the inclusion of a logic-based consistency-checking tool that evaluates multiple plausible solutions, aligning with real-world scenarios where spatial relationships often have several valid interpretations.

This thesis advances the spatial reasoning abilities of LLMs by identi-

fying deficiencies in current benchmarks and proposing practical enhancements. The integrated approach of refining evaluation benchmarks and employing advanced prompting techniques paves the way for future advances in AI spatial reasoning capabilities based on LLMs.

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| *1-cross* | Single Cross Calculus |
| *2-cross* | Double Cross Calculus |
| $9^+$-*Int* | $9^+$-Intersection Calculi |
| *ABA* | Algorithm of Bipartite Arrangements |
| *AGI* | Artificial General Intelligence |
| *API* | Application Programming Interface |
| *AWQ* | Activation-aware Weight Quantization |
| *B* | Billion |
| *BA* | Block Algebra |
| *BFS* | Breadth-First Search |
| *CBM* | Calculus Based Method |
| *CDC* | Cardinal Direction Calculus |
| *CDA* | Closed Disk Algebra |
| *CDR* | Cardinal Direction Relations |
| *CFG* | Context-Free Grammar |
| *CI* | Algebra of Cyclic Intervals |
| *C4* | Colossal Clean Crawled Corpus |
| *CoT* | Chain-of-Thought |
| *CYC* | Cyclic Ordering |
| *CSP* | Constraint Satisfaction Problem |
| *DC* | Disconnected |
| *DL* | Deep Learning |
| *DRA* | DRA Dipole Calculus |
| *DRA-con* | Dipole Connectivity |
| *DFS* | Depth-First Search |
| *E* | East |
| *EC* | Externally Connected |
| *EPRA* | Elevated Point Relation Algebra |
| *EQ* | Equal |

| | |
|---|---|
| *FLOPS* | FLoating point OPerations per Second |
| *FP16* | 16-bit Floating Point (half precision) |
| *FP32* | 32-bit Floating Point (single precision) |
| *FP64* | 64-bit Floating Point (double precision) |
| *GIS* | Geographic Information System |
| *GPTQ* | Quantization for Generative Pre-trained Transformer |
| *IO* | Input-Output |
| *K* | Thousand |
| *LM* | Language Model |
| *LLM* | Large Language Model |
| *LMSYS* | Large Model Systems Organization |
| *LMM* | Large Multimodal Model |
| *M* | Million |
| *ML* | Machine Learning |
| *MLLM* | Multimodal Large Language Model |
| *MoE* | Mixture of Experts |
| *N* | North |
| *NE* | NorthEast |
| *NW* | NorthWest |
| *NLU* | Natural Language Understanding |
| *NTPP* | Non-Tangential Proper Part |
| *NTPPi* | Non-Tangential Proper Part inverse |
| *OM-3D* | 3-D Orientation Model |
| *OPRA* | Oriented Point Relation Algebra |
| *OPT* | Open Pre-trained Transformer Language Models |
| *PC* | Point Calculus |
| *P3* | Public Pool of Prompts |
| *PFLOPS* | Peta FLoating point OPerations per Second |
| *PLM* | Pretrained Language Model |
| *PO* | Partially Overlapping |
| *QA* | Question Answering |
| *QSR* | Qualitative Spatial Reasoning |
| *QTC* | Qualitative Trajectory Calculus |

| | |
|---|---|
| *RCC* | Region Connection Calculus |
| *RL* | Reinforcement Learning |
| *RLHF* | Reinforcement Learning from Human Feedback |
| *RM* | Reward Modeling |
| *ROC* | Region Occlusion Calculus |
| *RCD* | Rectangular Cardinal Direction Calculus |
| *S* | South |
| *SE* | SouthEast |
| *SFT* | Supervised Finetuning |
| *SHP* | Stanford Human Preferences |
| *SpRL* | Spatial Role Labeling |
| *STAR* | Star Calculi |
| *SW* | SouthWest |
| *T* | Trillion |
| *TFLOPS* | Trillion FLoating point OPerations per Second |
| *TII* | Technology Innovation Institute |
| *TPC* | Ternary Point Configuration Calculus |
| *ToT* | Tree-of-Thoughts |
| *TPP* | Tangential Proper Part |
| *TPPi* | Tangential Proper Part inverse |
| *VR* | Visibility Relations |
| *W* | West |
| *mSpRL* | Multimodal Spatial Role Labeling |

# Chapter 1

# Introduction

## 1.1 Motivation

Space is one of the fundamental aspects of our daily life and of our physical world. Spatial reasoning, the ability to understand and navigate spatial relationships in physical space, is a fundamental aspect of human cognition that significantly influences how we interact with our environment. The need for spatial representations and spatial reasoning is ubiquitous in artificial intelligence (AI) - from robot planning and navigation, to interpreting visual inputs, to understanding natural language.

Spatial reasoning is leveraged across a multitude of applications, including geographic information systems (GIS) [2], robot navigation [3], robot manipulation [4, 5], visual reasoning [6–9], natural language processing (NLP) tasks [10], cognitive systems [11], simulation of physical environments [12, 13], traffic flow analysis and forecasting [14], and even biology [15]. These applications demonstrate the pervasive need for robust spatial representation and reasoning capabilities in AI systems to enrich the comprehension of their surroundings and response to user interactions, leading to more advanced user experiences.

Early strides in spatial reasoning in text were marked by the development of formal structures to represent spatial relationships. A spatial ontology [16] was proposed to formalize the representation of spatial relationships, laying the foundation for the later introduction of text-based spatial role labelling (SpRL) [17], which aims to convert natural language text into formal spatial representations. Building upon this, [18] further advanced the field by developing the multimodal spatial role labelling (mSpRL) task. This method extends SpRL by incorporating spatial information from both text and accompanying images, offering a richer representation of spatial relationships.

The development of spatial reasoning with natural language has been largely driven by the creation and refinement of datasets tailored for training and evaluating language models. The early datasets, such as SpaceEval (SemEval-2015 task 8) [19] and mSpRL, offered annotated spatial roles and relations, but their small scale underlined the complexity and novelty of the spatial role labelling problem. Recognizing this gap, [20] developed a question-answering (QA) dataset named StepGame, specifically designed to evaluate robust multi-hop spatial reasoning in text, with a focus on directional spatial relations [21, 22]. [23] significantly expanded the resource landscape by constructing three new spatial QA datasets: SpartQA, SPARTUN, and RESQ. These datasets encompass wide-ranging spatial language expressions, rendering them challenging to address using conventional logical programming. Moreover, they serve as benchmarks for exploring and evaluating the spatial reasoning capabilities of LMs.

Textual descriptions in these spatial reasoning datasets often employ qualitative representations [24] that describe spatial relationships in terms understandable to humans rather than precise coordinates, mirroring human cognitive processes. Such qualitative representations might articulate spatial relationships simply as "the park is to the north of the store," emphasizing essential spatial knowledge while omitting less critical details.

The advent of LLMs, such as OpenAI's ChatGPT, opened up fresh pathways for spatial reasoning. These models, leveraging transformer architectures, can generate human-like text and handle complex linguistic structures. Over the past few years, there has been a significant increase in the scale and complexity of LLMs, including OpenAI's GPT series from GPT-1 through GPT-4o [25] and open-source models such as Llama-1, Llama-2, and Llama-3 [26]. Researchers and developers are continually pushing the boundaries of LLMs, resulting in improvements in their capabilities, performance, and the range of tasks they can handle. Advancements in LLMs have significantly improved their capabilities in understanding and reasoning with textual information [27]. As LLMs continue to evolve, their capabilities are expanding to include more complex and nuanced tasks, such as reasoning, problem-solving, and creative writing.

However, their capabilities in spatial reasoning are yet to be fully explored and exploited. One recent approach to assess these capabilities was taken by [28], who put ChatGPT to the test using SpartQA and StepGame. Despite the generally advanced capabilities of ChatGPT, the model falls short of these tasks. This underscores the need for ongoing research and refined strategies to enhance spatial reasoning in LLMs, which would improve their comprehension of complex environments and their overall performance on spatial reasoning tasks.

In this work, we aim to provide a deeper analysis of LLMs' performance on spatial reasoning benchmarks, explore the limitations contributing to these results, and propose potential avenues for improvement.

## 1.2 Advancing Spatial Reasoning in LLMs

Over the past decades, various spatial logical reasoning tools [29, 30] have been developed to solve different reasoning challenges. They typically require symbolic inputs and necessitate the incorporation of additional reasoning rules or representations to accommodate new relations. Recent efforts have been made to integrate logical reasoners with LLMs. For instance, [31] developed an answer-set programming (ASP) tool for grid-based directional relations, while [1] formulated spatial reasoning rules. Commonly, this approach involves using LLMs to parse natural language descriptions into symbolic forms that logical reasoners can process.

In this study, we explore the enhancement of spatial reasoning tasks by integrating LLMs with a highly extendable logical reasoning framework. We employed LLMs to convert spatial descriptions into symbolic spatial relation representations, subsequently inputting these into a logical reasoning program. The integration demonstrated by [31] resulted in significant improvement in StepGame, surpassing the previous state-of-the-art (SOTA) though not achieving perfect results: around 90% accuracy for lower hops and 88.3% accuracy for 10-hop reasoning. They attributed 10.7% faults to data-related issues. We take a step further to delve into the two components, analyzing the performance of each on refined StepGame dataset. Remarkably, we achieved 100% accuracy for almost all hops, with only 2 errors among 1000 test examples, which were due to LLMs' 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.

We then explore the limit of LLMs as a general problem solver that explores its own thoughts and guides its own exploration with deliberate reasoning as heuristics. To achieve this, we employ Chain-of-Thought (CoT) and Tree-of-Thoughts (ToT) prompting.

A promising technique known as 'prompt engineering' [32] 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, including the challenging area of logical reasoning [33]. For instance, when faced with multi-step reasoning tasks, a method called few-shot CoT prompting [34] comes into 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.

CoT [35] incorporates a sequence of intermediate reasoning steps to facilitate problem-solving. However, when applied to spatial reasoning tasks, previous studies [31] 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 [36, 37] 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'[1] Within CoT, we propose refining the CoT prompt to empower language models to perform better in spatial reasoning tasks.

On the other hand, [38] 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 GPT-3 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

[1] In this thesis 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.

around 90% even as the tasks become more complex. On Davinci, the accuracy is maintained at around 50%, while GPT-3.5 Turbo [1] achieves a lower level of accuracy at around 30%.

## 1.3 Benchmarking Spatial Reasoning in LLMs

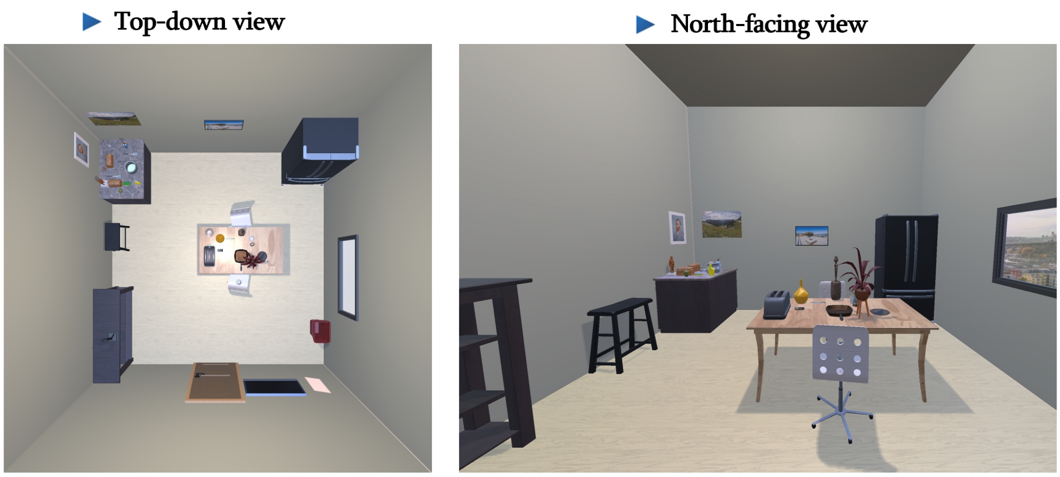
We analyzed the current benchmarks for spatial reasoning within LLMs - bAbI [39], StepGame [20], SpartQA [1] and SpaRTUN [23], identifying critical challenges inherent to these tasks. The analysis reveals several problems and limitations with these benchmarks. The bAbI's spatial reasoning tasks, for example, provide simplified spatial reasoning tasks by restricting relations to basic cardinal directions and fixed distances, lacking the complexity of real-world scenarios. StepGame, meanwhile, contains template errors that could distort model performance evaluations. These errors were previously overlooked, leading to studies conducted on a flawed benchmark, inaccurately assessing the capabilities of the LLMs [28], [31]. While SpartQA and SpaRTUN attempt to incorporate more complex spatial relations such as topological and distance relations, their descriptions often lack logical flow and clarity, undermining their effectiveness. Additionally, these benchmarks typically concentrate on two-dimensional spatial relations and are confined to textual modalities.

In response, we have developed a new spatial reasoning benchmark, RoomSpace[2], for evaluating LLMs. This benchmark utilizes the 3D simulation tool Procthor [13] to generate interactive room scenes. RoomSpace enhances the spatial reasoning evaluation context by offering flexibility in defining spatial relations. It incorporates images from multiple viewpoints, including top-down and agent-specific perspectives. Additionally, it creates textual narratives for reasoning tasks. This provides a flexible framework for assessing the capabilities of advanced LLMs like GPT-4, enhancing the benchmark's effectiveness in probing the depths of spatial reasoning. An example of RoomSpace data is depicted in Figure 1.1.

With 3D simulation, RoomSpace offers two key advantages. First, the inclusion of visual components future-proofs the benchmark - ensuring it remains relevant, ex-

[1]There are four versions of GPT-3.5 Turbo models: gpt-3.5-turbo, gpt-3.5-turbo-0125, gpt-3.5-turbo-1106, and gpt-3.5-turbo-instruct. For all subsequent experiments, we use gpt-3.5-turbo, referred to simply as Turbo hereafter.

[2]https://github.com/Fangjun-Li/RoomSpace




**Story**: Imagine a square-shaped kitchen, bordered by four walls. This room contains a collection of furniture, including a fridge placed in the north-east, a counter top placed in the north-west, a dining table placed in the center, a shelving unit placed in the south-west. Imagine yourself at the southern wall's door, looking inwards. From this perspective, the fridge is in front of and to the right of the counter top. The counter top is in front of and to the left of the dining table. The dining table is in front of and to the right of the shelving unit.
**Find Relation Question**: Where is the fridge positioned in relation to the shelving unit? **Answer**: front-right
**Yes/No Question**: Could the fridge be positioned behind of the shelving unit? **Answer**: No

Figure 1.1: One evaluation example in our RoomSpace benchmark includes images, textual spatial reasoning stories, and questions designed to evaluate LLMs.

tensible, and well-aligned with the rapid progression of multimodal AI systems. As multimodal large language models (MLLMs) like GPT-4V, Gemini Pro, and Claude 3.5 Sonnet continue to evolve, the ability to jointly reason over text and visual inputs is becoming a core aspect of AI research and deployment. RoomSpace is designed with this trajectory in mind, supporting richer, more comprehensive evaluations across diverse use cases. Second, 3D simulation enhances the ecological validity of the benchmark. Unlike earlier datasets that relied on overly simplified or repetitive object identifiers, RoomSpace introduces a diverse array of realistic objects reflective of real-world environments. A wide array of objects is introduced that more closely mirror real-world scenarios than previous benchmarks, which often used repetitive, lengthy object names or simple identifiers. In RoomSpace, objects are identified by their type, and numerically distinguished when multiple instances are present, eliminating the need to specify attributes such as color, size, or shape.

The benchmark also offers significant flexibility in defining spatial representations and crafting specialized reasoning tasks to meet various research needs. The metadata generated for each scene in RoomSpace allows researchers to tailor spatial representations to their specifications, incorporating point-based cardinal direction relations, varying levels of granularity in distance relations, and topological relations between objects within a room.

Moreover, the adaptability extends to the formulation of spatial reasoning tasks. Our approach to constructing constraint satisfaction problems is a prime example, where the question posed assesses the feasibility of object arrangements within a defined domain that meets all specified constraints. A positive arrangement results in a 'Yes', and if no such arrangement exists, the answer is 'No'. The complexity of these narratives can be adjusted by varying the number of objects, the nature and scope of constraints, the domain size, and the specific types of constraints utilized.

Lastly, RoomSpace includes a logical reasoner for generating gold labels for each problem, acknowledging the potential for multiple valid solutions in scenarios with limited qualitative spatial information. This feature ensures that our benchmark can more accurately reflect the complexities and ambiguities inherent in real-world spatial reasoning, providing a comprehensive platform for evaluating LLMs.

This section of the work not only highlights the limitations of current benchmarks but also introduces a robust framework for more comprehensively assessing the spatial reasoning capabilities of LLMs. Through the development and deployment of RoomSpace, we provide a valuable tool for advancing research in this critical area of AI development.

## 1.4 Thesis Structure

This thesis is structured into five main chapters, each addressing specific aspects of spatial reasoning with Large Language Models (LLMs). Below is a concise outline of each chapter:

In this opening chapter, we introduced the motivation for studying spatial reasoning with LLMs and discussed the significance of developing benchmarks to evaluate their spatial reasoning capabilities.

In Chapter 2, we provide a comprehensive review of the development of LLMs. It traces the evolution from pre-training to various enhancements such as supervised fine-tuning, reinforcement learning from human feedback, and model quantization. The chapter also explores the development of qualitative spatial representation and reasoning, highlighting the particular challenges and previous efforts to integrate spatial reasoning with LLMs.

In Chapter 3, various benchmarks, including bAbI, StepGame, and SpartQA/SpaRTUN, are discussed. It provides an in-depth examination of these benchmarks, detailing the creation of textual stories and questions, and discussing the inherent limitations and problems. Additionally, the evaluation methods and performance of LLMs on these benchmarks are critically analyzed.

In Chapter 4, we explore methods to enhance the spatial reasoning capabilities of LLMs. It discusses the integration of logical reasoners with LLMs, the deployment of CoT prompting, and ToT strategies. Experimental results are presented to demonstrate the effectiveness of these methods in improving the performance of LLMs on spatial reasoning tasks.

Chapter 5 introduces RoomSpace, a new benchmark designed to rigorously test the spatial reasoning capabilities of LLMs. It describes the detailed process of constructing 3D room environments, specifying spatial representations, and formulating spatial reasoning problems. The chapter concludes with an evaluation of LLMs' performance on this benchmark, highlighting its utility and the insights it provides.

In Chapter 6, we present the key findings of our work and the contributions we offer in the field of spatial reasoning with LLMs. We also discuss the main limitations of this work and suggest a number of research directions for future work.

# Chapter 2

# Related Work

## 2.1 Development of LLMs

The evolution of language models has seen significant transformation from the 1950s to the present day. Initially, from the 1950s to the 1990s, language processing was dominated by rule-based systems, where machines executed tasks based on algorithms predefined by humans. In the 1990s, this approach evolved with the rise of statistical machine learning, shifting from human-written rules to models that learned from data, focusing on understanding the statistical distribution of language. As the calendar turned to 2013, deep learning began to gain prominence, particularly influencing fields like computer vision and language processing. During this period, language models incorporated innovative architectures such as encoder-decoder structures, Word2Vec embeddings, and attention mechanisms. These developments allowed for words and phrases to be transformed into high-dimensional representations. The labelled data size reached the billion scale. The period from 2018 to 2022 marked the advent of pre-trained language models (PLMs) [40] that utilized vast amounts of unlabeled data. This era introduced foundational models like GPT and BERT, followed by more advanced models such as T5 and GPT-3. The pre-training and fine-tuning paradigm became a cornerstone, significantly enhancing model performance across various tasks. Since 2020, the development of LLMs has been at the forefront of AI research, reaching a notable milestone in 2022 with the launch of ChatGPT. These models are trained on increasingly diverse and extensive datasets, including user-generated content, to benefit from reinforcement learning through human feedback. This phase marks a significant advancement in language modelling techniques, aiming to fully leverage AI's capabilities to comprehend and generate text that closely mimics human language.

In this section of the literature review, we focus on language models developed after 2018, emphasizing the most prominent LLMs. This includes OpenAI's GPT series [25, 41–44], Google's Transformer and PaLM series [45, 46], DeepMind's Gemini series [47, 48], and Meta's Llama series [26, 49], among others. The progression from early models like BERT [50] and GPT-1 [41] to the latest GPT-4 [25] exemplifies substantial enhancements in training and fine-tuning methodologies.

Initially, models such as BERT and GPT-1 depended heavily on extensive pre-training using unlabeled data, followed by fine-tuning for specific tasks, which posed challenges in efficiency and task generalization. Innovations like prompt-tuning [51], p-tuning [52], and prefix-tuning [53] introduced more streamlined fine-tuning techniques by adjusting only a minimal set of parameters, yet they continued to grapple with generalizing across diverse tasks. The transition from GPT-3 to ChatGPT [44] marked a significant shift with the adoption of Reinforcement Learning from Human Feedback (RLHF), which dramatically refined response accuracy and relevance by integrating direct human feedback into the model's training regimen. GPT-4 has further refined this approach with the implementation of the Mixture of Experts (MoE) [54] technique, improving scalability and efficiency by selectively activating pertinent model components for specific tasks. This allows for specialized model behaviour while minimizing computational overhead. These developments highlight ongoing efforts to enhance the performance, scalability, and efficiency of LLMs. We will delve deeper into these developmental stages in the following part, providing a comprehensive overview of the progression and impact of these technologies.

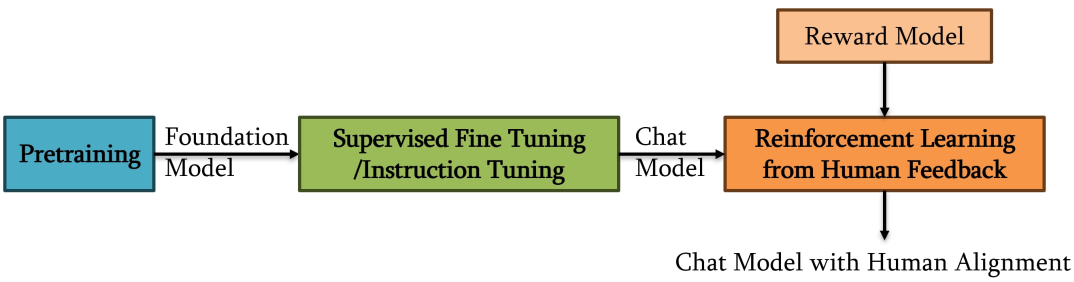



Figure 2.1: The three main development stages for LLMs.

### 2.1.1 Pre-training

Pre-trained language models follow the pipeline: first self-supervised learning with large unlabeled data (e.g., Wikipedia). Then, for task-specific datasets (e.g., QA and translation), fine-tune the models on these data to get the final model, then test data on the final model. There are three types of structures in PLMs: encoders (e.g., BERT, RoBERTa, ALBERT), decoders (e.g., GPT), and encoder-decoders (e.g., T5, BART). The period from GPT-1 to GPT-3 witnessed significant expansions in training corpora and transformer layers. Models like Codex also integrated code pre-training. A summary of the evolution in terms of parameters and pre-training data volumes for several seminal language models is presented in Table 2.1.

To develop foundational language models capable of extensive linguistic processing, vast training datasets are used, comprising billions to trillions of tokens sourced from publicly available corpora, such as Common Crawl[1], RefinedWeb [72], and The Pile [75]. The primary training objective during this phase is a straightforward "next word prediction task", wherein the model predicts subsequent words based on the provided textual context.

Research by [76] on the influence of pre-training data volume on linguistic capabilities revealed that while language models only need approximately 10M or 100M words to effectively learn most of the syntactic and semantic features, acquiring sufficient commonsense knowledge and other skills necessary for excelling in standard natural language understanding (NLU) tasks requires substantially more data.

Further studies [60] explored the optimal model size and data volume for training transformer-based language models within specified computational budgets. Their findings showed that the optimal Gopher compute budget model size is 40B - 70B. Their experiments showed that Chinchilla-70B, trained on 1.4T tokens, consistently and significantly outperformed larger LLMs like Gopher (280B) and GPT-3 (175B) across a broad array of downstream tasks, suggesting more efficient scaling strategies.

Inspired by the Chinchilla scaling laws, the Llama series models were developed, utilizing 1.4T pre-training data from diverse sources like Common Crawl, the processed Colossal Clean Crawled Corpus (C4) [55], GitHub datasets available on Google BigQuery, Wikipedia dumps, internet-based book corpora [75], arXiv LaTeX files [77],

[1] https://commoncrawl.org, this and all other cited URLs last retrieved on October 30, 2024.

| Model | Providers | Time | Parameters | Tokens | Open |
|---|---|---|---|---|---|
| GPT-3[43] | OpenAI | 2020.05 | 125M to 175B | 330B | × |
| GPT-4[25] | | 2023.03 | 1.76T | 13T | × |
| T5[55] | Google | 2019.10 | 11B | 34B | ✓ |
| GLaM[56] | | 2021.12 | 1.2T | 1.6T+1.7B | × |
| LaMDA[57] | | 2022.02 | 2B, 8B, 137B | 1.56T words | × |
| PaLM[45] | | 2023.05 | 8B, 62B, 540B | 780B | × |
| PaLM-2[46] | | 2023.05 | 340B | 3.6T | × |
| ERNIE-3.0[58] | Baidu | 2021.06 | 11B | 4T | × |
| Gopher[59] | DeepMind | 2021.12 | 280B | 300B | ✓ |
| Chinchilla[60] | | 2022.03 | 70B | 1.4T | × |
| Gemini[47] | | 2023.12 | Nano (1.8B, 3.25B) | / | × |
| Gemini-1.5[48] | | 2024.03 | / | / | × |
| Gemma[61] | | 2024.04 | 2B, 7B | 6T | ✓ |
| Gemma-2[62] | | 2024.07 | 9B, 27B | 8T, 13T | ✓ |
| OPT[63] | Meta | 2022.05 | 125M to 175B | 180B | ✓ |
| Llama[26] | | 2023.02 | 7B to 65B | 1.0T, 1.4T | ✓ |
| Llama-2[49] | | 2023.07 | 7B to 70B | 2T | ✓ |
| Llama-3[64] | | 2024.04 | 8B, 70B | 15T | ✓ |
| Claude-2[65] | Anthropic | 2023.07 | / | / | × |
| Claude-3[66] | | 2024.03 | / | / | × |
| GPT-J[67] | EleutherAI | 2021.04 | 6B | / | ✓ |
| GPT-NeoX[68] | | 2022.04 | 20B | / | ✓ |
| BLOOM[69] | BigScience | 2022.11 | 560M to 176B | 366B | ✓ |
| MPT-7B[70] | MosaicML | 2023.05 | 6.7B | 1T | ✓ |
| Mistral-7B[71] | Mistral.AI | 2023.10 | 7B | / | ✓ |
| Falcon-rw[72] | TII | 2023.11 | 1B, 7B | 5T | ✓ |
| Falcon[73] | | 2023.11 | 7B, 40B, 180B | 5T | ✓ |
| Falcon-2[74] | | 2023.10 | 2B, 11B | 5.5T | ✓ |

Table 2.1: Summary of foundation LLMs.

and QA sites like StackExchange [1]. This approach proved effective, with Llama's performance improving steadily with increasing training data volumes, particularly in QA and commonsense reasoning tasks. Subsequent iterations, Llama-2 and Llama-3, expanded the training data volumes to 2T and 15T tokens, respectively.

Pre-training LLMs on large-scale training data demands substantial computational resources, involving thousands of high-performance GPUs, taking weeks to months to complete the training of deep neural network parameters. For instance, as reported by [26], training the Llama 65B model required 2048 NVIDIA A100 80GB GPUs over 21 days on 1.4T training tokens. Another example is the GPT model. As reported by [43], training GPT-3 175B required 3640 PFLOPS-day. The training utilized V100 GPUs within a high-bandwidth cluster and employed 16-bit floating point (FP16) variables. Assuming training needs to be concluded within 1 month without accounting for compute resource utilization rates and the significant communication overhead in a large GPU cluster [78], a GPU cluster capable of achieving 120 PFLOPS is necessary. Per the NVIDIA V100S Datasheet, a single V100 GPU delivers about 120 TFLOPS using FP16, indicating a need for at least 1000 V100 GPUs. More practical figures are provided by the training of the OPT-175B [63] and BLOOM-176B [69], which are comparable in scale to GPT-3. The OPT-175B training took about 2 months with 992 A100 80GB GPUs, factoring in hardware failures, and achieved a practical GPU utilization of 147 TFLOPS per unit. This utilization is slightly below 50% of the NVIDIA A100 80GB GPU's theoretical FP16 performance of 312 TFLOPS. The training of the BLOOM-176B model required 3.5 months using 384 A100 80GB GPUs [69].

Concluding the pre-training phase, these models are adept at predicting subsequent tokens in textual sequences and are believed to implicitly contain factual knowledge and commonsense knowledge. These foundation models learn powerful, general representations and can be prompted into completing tasks; an example of their success in action-effect prediction tasks can be found in our previous research [27].

### 2.1.2 Supervised Fine-tuning Stage

The training goal during the pre-training phase of LLMs is primarily focused on predicting the next word in a sequence. This foundational stage does not inherently train the model to comprehend or respond to complex human instructions. To bridge this

[1] https://stackexchange.com/

gap, a subsequent phase known as instruction tuning [79] is implemented to align the models more closely with human-like interaction and responsiveness for a variety of functional tasks. The resulting fine-tuned models, often referred to as supervised fine-tuning (SFT) models, gain the ability to perform complex tasks such as open-domain questioning, reading comprehension, translation, and code generation. These models can subsequently be deployed effectively as interactive assistants.

Many ChatGPT-like models, referenced in Table 2.2, are of this type. Due to its openness and effectiveness, Llama has attracted significant attention from the research community, and many efforts have been devoted to fine-tuning Llama's different pre-training versions for implementing new models or tools. Examples include Alpaca-7B [80], which was fine-tuned from the Llama-7B on 52K instruction-following demonstrations, and Vicuna [81], which was also fine-tuned from Llama but used a different dataset of 70K user-shared conversations. Many of these models work very well, even achieving 90% of ChatGPT's performance in some reviews.

| **Model** | **Providers** | **Time** | **Parameters** | **Base Model** | **Data** | **Open** |
|---|---|---|---|---|---|---|
| T0[82] | Hugging Face | 2021.10 | 11B | T5 | P3 | ✓ |
| BLOOMZ[83] | | 2022.11 | 176B | BLOOM | xP3 | ✓ |
| InstructGPT[44] | OpenAI | 2022.03 | 1.3B, 6B, 175B | GPT-3 | / | × |
| FLAN[84] | Google | 2021.09 | 137B | LaMDA-PT | Flan2021 | × |
| Flan-T5[85] | | 2022.10 | 11B | T5 | Flan2022 | ✓ |
| Flan-PaLM[85] | | 2022.10 | 540B | PaLM | Flan2022 | × |
| Bard[86] | | 2023.03 | 340B | PaLM-2 | / | × |
| OPT-IML[87] | Meta AI | 2022.12 | 30B, 175B | OPT | OPT-IML | ✓ |
| LIMA[88] | | 2023.05 | 65B | Llama | LIMA | ✓ |
| Alpaca[80] | Stanford | 2023.03 | 7B | Llama | Alpaca | ✓ |
| Gorilla[89] | UC Berkeley | 2023.05 | 7B | Llama | / | ✓ |
| Vicuna[81] | LMSYS | 2023.10 | 7B, 13B | Llama | / | ✓ |

Table 2.2: Summary of emblematic SFT models.

During this phase, high-quality, prompt-response pairs meticulously crafted by humans are used to fine-tune the foundational models obtained from the initial pre-training stage. This data, although smaller in scale, is rich in diversity, covering a wide range of tasks, including question-answering (QA) and casual conversations. An

| Datasets | Providers | Time | Size | Tasks | Construction | Open |
|---|---|---|---|---|---|---|
| Sup-NatInst v1[90] | Allen Institute for AI | 2021.06 | 193K | 61 | Crowdsourcing | ✓ |
| Sup-NatInst v2[91] | | 2021.10 | 5M | 1.6K | | ✓ |
| P3[83] | Brown University | 2021.10 | 12M | 62 | Manually composed templates + Compiled datasets | ✓ |
| xP3[83] | Hugging Face | 2022.11 | 81M | 71 | | ✓ |
| Flan 2021[84] | Google | 2021.09 | 4.4M | 62 | | ✓ |
| Flan 2022[85] | | 2022.10 | 15M | 1835 | | ✓ |
| Flan Collection [92] | | 2023.02 | 378M | / | | ✓ |
| MetaICL[93] | University of Washington | 2021.10 | 3.5M | 142 | | ✓ |
| Self-Instruct[94] | | 2022.12 | 52K | 175 | LLM-synthetic | ✓ |
| UnnaturalInst[95] | Meta AI | 2023.04 | 64K | 117 | | ✓ |
| LIMA[88] | Meta AI | 2023.05 | 1K | / | Human-generated + Filtering | ✓ |
| OpenAssistant[96] | LAION | 2023.10 | 161K | 625K | | ✓ |
| Dolly[97] | Databricks | 2023.04 | 15K | 7 | | ✓ |
| Alpaca Data[80] | Stanford | 2023.03 | 52K | 175 | | ✓ |

Table 2.3: An overview of instruction fine-tuning datasets.

overview of some prominent instruction-following datasets is presented in Table 2.3. Super-Natural Instruction (Sup-NatInst) v1 [90], Flan 2021 [84], and the Public Pool of Prompts (P3) [98] have been instrumental in aggregating large NLP task collections, which are templatized with instructional prompts to train models for generalizing to new, unseen instructions. Additionally, MetaICL [93] employs a different setup utilizing few-shot prompting, concentrating on "in-context" learning, where models adapt to new tasks from a few input-output (IO) examples. Recent work [92] demonstrated that training models using a combination of zero-shot and few-shot prompts markedly improves performance in both settings.

The computational demands for instruction tuning are considerably lower compared to the initial pre-training phase due to the smaller size of the training corpora required. Depending on the size of the model and the amount of training data, it usually takes several GPUs from several hours to a few days to complete the training. For instance, the Vicuna model [81] was fine-tuned on roughly 70K user-shared conversations from ShareGPT.com using 8 A100 GPUs in one day.

Recent developments in this field have focused on expanding prior resources by combining more datasets and tasks into one resource, such as Super-Natural Instructions v2 [91] and OPT-IML [87]. Innovations also include multilingual instruction tuning as seen in xP3 [83], and the integration of Chain-of-Thought training prompts in Flan 2022 [85]. Both Flan Collection [92] and OPT-IML feature an extensive array of tasks previously represented in other collections.

Building instruction data manually is a costly process that demands significant human effort. Consequently, researchers have been exploring more efficient alternatives to manual data generation. A promising strategy for expanding task diversity is the production of synthetic data through the use of LLMs, particularly in domains requiring creativity and open-ended dialogue. Methods such as self-instruct [99] and unnatural instructions [95] are examples of this approach. The self-instruct [94] method utilizes the generation capacity of LLMs to fabricate a plethora of instructional content. This technique was employed by researchers at Stanford to generate a corpus for instruction data. Alpaca-7B [80], which was fine-tuned using supervised learning from a Llama-7B model on 52K instruction-following demonstrations generated from OpenAI's text-davinci-003, showed many behaviours similar to text-davinci-003, but is also surprisingly small and cost-effective to replicate.

Moreover, research conducted by [88] demonstrates that with a strong PLM, remarkably strong performance can be achieved by simply fine-tuning on a carefully selected set of 1,000 training examples. They suggest that the vast majority of knowledge in LLMs is acquired during the pre-training phase, and minimal but targeted instruction tuning is sufficient to enhance the model's ability to produce high-quality outputs.

### 2.1.3 Reinforcement Learning from Human Feedback

Through SFT, LLMs have initially been equipped with the ability to comprehend and follow human instructions across various types of NLP tasks. However, SFT requires a large number of instructions paired with corresponding standard responses. Generating such a volume of high-quality responses requires significant human resources and time investment. Furthermore, SFT typically employs cross-entropy loss, aiming to fine-tune the model parameters so that its outputs match the standard answers exactly. This approach, however, does not take into account the holistic quality of model outputs, nor does it accommodate the natural diversity of language or the nuances of minor textual variations. This limitation has driven researchers to seek methods to better align AI outputs with human values and expectations.

In this context, [44] advocated for model outputs that adhere to the principles of Helpfulness, Honesty, and Harmlessness (3H), which reflect prevalent human values. However, it is challenging to simultaneously fulfil all three principles. To integrate these principles, reinforcement learning from human feedback (RLHF) has been incorporated into the training process of general dialogue models. Unlike traditional supervised methods, RLHF evaluates the entire output text, optimizing for the generation of high-quality responses. This method does not rely solely on predefined, high-quality responses; instead, it employs a reward model to assess the quality of responses generated by the model under varying instructions. This allows the model to explore multiple answer possibilities and learn from feedback on the quality rankings of its outputs, making RLHF particularly well-suited for generative tasks and a crucial component in the development of advanced LLMs.

The RLHF framework typically unfolds in two phases: 1) Reward Model Training: This involves training a classifier within the language model to distinguish between 'good' and 'bad' responses, akin to giving a 'thumbs up' or 'thumbs down' rating to the responses. 2) RLHF Fine-Tuning: This stage utilizes the trained reward mode to align the model's outputs to better match human judgments, enhancing the relevance and appropriateness of the responses generated by the LLM.

#### Reward Modeling

The reward modelling stage aims to develop a model that can assess and rank the textual outputs produced by the SFT model in response to the same prompt. This model, typically built on a PLM with Transformer architecture, is trained to assign rewards to each generated text, with human-assigned rankings serving as the ground truth. The process of training this model often requires the use of dozens of GPUs over several days. The accuracy and reliability of the reward model are crucial for the effectiveness of the subsequent reinforcement learning (RL) phase, where the model's outputs are fine-tuned to maximize the perceived quality of responses.

The development of datasets for training the reward model is critical to its success. The focus of data collection shifts towards comparative evaluations, where human assessors are tasked with determining the relative quality of multiple outputs generated by the SFT model. In recent years, some high-quality, open-source datasets have been made available that aim to align LLMs with core human values like the 3H principle. Table 2.4 lists some prominent datasets used for alignment tuning.

| Datasets | Providers | Time | Size | Open |
|---|---|---|---|---|
| Summarize from Feedback[100] | OpenAI | 2020.09 | 194K | ✓ |
| SHP[101] | Stanford | 2020.10 | 385K | ✓ |
| WebGPT[102] | OpenAI | 2021.11 | 19.6K | ✓ |
| HH-RLHF[103] | Anthropic | 2022.04 | 169K | ✓ |
| H4 Stack Exchange Preferences [104] | HuggingFace | 2023. | 10.8M | ✓ |
| Sandbox Alignment Data [105] | DeepMind | 2023. | 10.8M | ✓ |
| PKU-SafeRLHF[106] | Peking University | 2024.06 | 83.4K | ✓ |

Table 2.4: An overview of datasets for alignment tuning.

Introduced by OpenAI in 2020, the Summarize from Feedback dataset[1] integrates RLHF technology into the task of summary generation. This dataset is divided into two parts. The first part, the comparisons part, has 179K instances of training and validation splits. In this part, human annotators were tasked with choosing the better summary between the two presented options. The second part, known as the axis part, consists of 14.9K instances allocated for testing and validation, where the quality of summaries is assessed by human annotators using a Likert scale.

WebGPT[2] I was designed to tackle the task of responding to long document-based questions, aligning the answers with human preferences. This dataset includes 19,000 instances, each featuring two model-generated responses to a question, accompanied by relevant metadata. Human evaluators assign preference scores to these answers, facilitating the identification of the superior response among the pair.

HH-RLHF[3] encompasses approximately 169,000 instances, segmented into two parts emphasizing the helpfulness and harmlessness of LLMs. Each instance involves an open-ended conversation between a crowdworker and a chat model, centered around seeking help, advice, or task completion. For each user query, the chat model generates two responses. Annotations are applied to these responses to identify which one is more

[1] https://huggingface.co/datasets/openai/summarize_from_feedback
[2] https://huggingface.co/datasets/openai/webgpt_comparisons
[3] https://huggingface.co/datasets/Anthropic/hh-rlhf

helpful or less harmful, based on the context of the query.

The Stanford Human Preferences (SHP) dataset[1] prioritizes the helpfulness of responses and serves as a resource for training RLHF reward models and evaluating natural language generation (NLG) models. This dataset contains 385,000 instances reflecting collective human preferences for responses to questions or instructions, covering 18 varied topics from cooking to legal advice. Each instance comprises a Reddit post that includes a question or instruction accompanied by two top-level comments; Reddit users have evaluated these comments, identifying one as more helpful and the other as less so. Unlike HH-RLHF, where responses are generated by models, all entries in SHP consist of naturally occurring, human-authored text.

The PKU-SafeRLHF dataset[2] comprises 83,400 entries, each annotated for harmlessness and helpfulness. Each entry features two responses to a question, each with safety meta-labels and user preference. Responses are considered harmless if they are rated as risk-neutral across all 19 identified harm categories[3]. The helpfulness attribute evaluates how effectively a response addresses the prompt, with emphasis on the quality, clarity, and relevance of the information provided.

The H4 Stack Exchange dataset[4] is centered around the helpfulness of answers sourced from Stack Overflow, featuring approximately 10 million questions and their corresponding answers. Each dataset entry includes a question paired with multiple answers. The helpfulness of each answer is quantified through a scoring system derived from user votes, and each answer is also marked with a label indicating whether it was the selected response.

The Sandbox Alignment Data[5] comprises 169,000 instances of interaction data suitable for alignment training of language models. This dataset originates from a virtual environment known as SANDBOX, a simulated human society inhabited by numerous language model-based social agents. These agents operate under a defined set of rules, enabling the detailed collection of social interaction data among language models. Each data entry includes a societal query, a range of responses generated by the models, and corresponding evaluations provided by the models themselves.

[1] https://huggingface.co/datasets/stanfordnlp/SHP
[2] https://huggingface.co/datasets/PKU-Alignment/PKU-SafeRLHF
[3] https://huggingface.co/datasets/PKU-Alignment/PKU-SafeRLHF-QA
[4] https://huggingface.co/datasets/HuggingFaceH4/stack-exchange-preferences
[5] https://github.com/agi-templar/Stable-Alignment

#### Reinforcement Learning Stage

Following the development of the reward model, the LLM undergoes fine-tuning through reinforcement learning. OpenAI employs the Proximal Policy Optimization (PPO) algorithm for this purpose. This method optimizes the model's outputs based on evaluative feedback from the reward model, adjusting the generation of completions according to the assessed rewards. Over multiple iterations, the model enhances its ability to produce responses that more closely align with human evaluative standards.

This stage requires fewer computational resources and is typically completed within a few days, often yielding superior performance compared to SFT. However, this method introduces complexities due to its potential instability and the need to manage the extensive number of hyperparameters, which may affect the convergence of the model.

| **Models** | **Providers** | **Time** | **Size** | **Base** | **Open** |
|---|---|---|---|---|---|
| text-davinci-003 | OpenAI | 2022.09 | 175B | text-davinci-002 | × |
| gpt-3.5-turbo-instruct | | 2023.09 | 20B | gpt-3.5-turbo | × |
| Llama2-Chat | Meta | 2023.07 | 7B, 34B, 70B | Llama-2 | ✓ |
| Llama3-Instruct | Meta | 2024.04 | 8B, 70B | Llama-3 | ✓ |

Table 2.5: Summary of LLMs enhanced with RLHF.

Table 2.5 outlines several representative models that have been enhanced using RLHF. For example. text-davinci-003 and gpt-3.5-turbo-instruct introduced by OpenAI. Despite the groundbreaking nature of the GPT-3 models, they had a propensity to generate responses that could be untruthful or harmful due to their training on a diverse and extensive corpus of internet-sourced data. To mitigate these issues and better tailor these models to user needs, OpenAI implemented SFT and RLHF techniques. The resulting model, text-davinci-003, significantly improved its ability to follow instructions and reduce the production of inaccurate or harmful content. Introduced in September 2022, text-davinci-003 was later deprecated on January 4, 2024, with gpt-3.5-turbo-instruct recommended as its replacement[1]. gpt-4 and gpt-3.5-turbo were introduced in March 2023. By September of the same year, OpenAI launched gpt-3.5-turbo-instruct[2], a model distinct from its predecessors by being specifically fine-tuned for direct query

[1]https://platform.openai.com/docs/deprecations
[2]https://platform.openai.com/docs/models/gpt-3-5-turbo

responses and text completion, shifting away from the conversational simulation focus of earlier models.

Similarly, Llama-2 has seen significant developments in both SFT and RLHF, leading to the creation of a chat-specific iteration, Llama2-Chat [49]. This model has been released in various configurations, including 7B, 13B, and 70B parameters. Initially trained using SFT, Llama2-Chat underwent subsequent refinements through RLHF employing techniques like rejection sampling and PPO. In general, Llama-2 outperforms other open-source models on numerous benchmarks assessing helpfulness and safety. Further, the Llama3 instruction-tuned versions, Meta-Llama-3-8B-Instruct[1] and Meta-Llama-3-70B-Instruct[2], have been optimized for dialogue use cases using SFT and RLHF to align more closely with human preferences for helpfulness and safety. These versions surpass many available open-source chat models in common industry benchmarks, demonstrating the effectiveness of these advanced training methodologies.

### 2.1.4 Model Quantization

Quantization technology plays a crucial role in deploying (LLMs on constrained computing resources. It primarily involves reducing the precision of data representation, aiming to retain as much information as possible. Typically, this technique converts model parameters from higher-bit representations to lower-bit formats. For example, converting model weights from 32-bit floating-point numbers (Float32) to 16-bit versions (Float16) can reduce the model's memory footprint by half, decreasing GPU memory usage significantly. Further quantization to 8-bit integers (Int8) or even to 4-bit floating-point numbers (Normal Float4, NF4) can decrease memory requirements to about a quarter and one-eighth, respectively. Such reductions in data precision facilitate faster computations and lower memory usage, thereby enhancing inference speed without substantially degrading model performance.

Among the notable quantization strategies, GPTQ [107] effectively quantizes GPT models, maintaining their accuracy while enabling their operation within a single GPU for generative tasks. Another approach, Activation-aware Weight Quantization (AWQ) [108], provides a hardware-friendly method for LLM low-bit weight-only quantization.

[1] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[2] https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct

## 2.2 Spatial Representation and Spatial Reasoning

Spatial reasoning is inherently complex, involving multiple dimensions, making it challenging to encapsulate with a single scalar measure. This complexity extends beyond what is typically evaluated in current spatial reasoning benchmarks/datasets.

Natural language descriptions of spatial tasks typically utilize qualitative representations of spatial relationships. This approach involves using a limited vocabulary to denote qualitative relationships between entities or to categorize numerical values qualitatively [109]. This qualitative approach is preferred for its closeness to human cognitive processes, where commonsense knowledge is predominantly represented through qualitative, rather than quantitative, spatial expressions. For instance, in everyday communication, one might say "the park is to the north of the school, at a short distance" rather than providing precise measurements like " Supposing the park is at (0,0), the school is at coordinates (0, 121)." This qualitative method allows for handling incomplete knowledge and reflects the natural way humans interpret spatial information.

Qualitative reasoning is an approach that manages commonsense knowledge without reliance on numerical computation and allows for handling incomplete knowledge effectively. The inherent multi-dimensionality of qualitative spatial representation allows for a higher degree of freedom in describing spatial relationships between entities. Consequently, considerable effort has been devoted to developing various qualitative spatial calculi, which provide a foundational framework for representing spatial knowledge in a qualitative manner, as detailed by [24].

Over the years, a broad spectrum of qualitative calculi has been developed to address specific aspects of spatial knowledge, e.g., Point-Based Ternary Calculus ($LR$)[119], Ternary Point Configuration Calculus (TPCC), [125], StarVars [127], Single/Double Cross Calculus (1-,2-cross) [128], 3-D Orientation Model (OM-3D) [129], Elevated Point Relation Algebra (EPRA) [136], Qualitative Trajectory Calculus (QTC) [141], Dipole Calculus (DRA) [120], Cyclic Ordering (CYC) [126], Closed Disk Algebra (CDA) [134], Alg. of Bipartite Arrangements (ABA) [124], Calculus Based Method (CBM) [132], Dipole connnectivity (DRA-con) [130], Nine-Intersection Model (9-Int) [133], $9^+$-Intersection Calculi ($9^+$-Int) [135], Region Occlusion Calculus (ROC) [137], Occlusion Calculus (OCC) [138], LOS Lines of Sight (LOS) [139], VRCC-3D+ [140], Visibility Relations (VR) [121], Rectangular Cardinal Direction Calculus (RCD) [115], Block algebra/Rectangle Algebra/Rectangle Calculus (BA), [117, 118].

| | | Point | Curve, Line | Region |
|---|---|---|---|---|
| Direction | **Cardinal** | CDC [110, 111]<br>STAR [114]<br>PC [116] | CI [112] | CDR [113]<br>RCD [115]<br>BA [117, 118] |
| | **Relative** | *LR* [119]<br>OPRA [122, 123]<br>TPCC [125]<br>StarVars [127]<br>1-,2-cross [128]<br>OM-3D [129] | DRA [120]<br>ABA [124]<br>CYC [126] | VR [121] |
| **Toplogy** | | | DRA-con [130]<br>CBM [132]<br>CDA [134]<br>$9^{+}$-Int [135] | RCC [21, 131]<br>9-Int[133] |
| **Distance** | | EPPRA [136] | | ROC [137]<br>OCC [138]<br>LOS [139]<br>VRCC-3D+ [140] |

Table 2.6: Classification of Qualitative Spatial Reasoning (QSR) Calculi Based on Primary Base Entities and Captured Spatial Aspects

Figure 2.6 outlines some of these calculi, organized by relation types such as directional (cardinal, relative), topological, and distance relationships. Moreover, spatial representations generally comprise fundamental spatial entities, including points, lines, line segments, rectangles, cubes, or arbitrary regions across various dimensions. The figure introduces several point-based systems, such as Star Calculi (STAR) [114] and Oriented Point Relational Algebra (OPRA) [122, 123]. It also highlights curve/line-based systems like Algebra of Cyclic Intervals (CI) [112] and the Nin-Intersection Model (9-Int) [133], alongside region connection theories including the Region Connection Calculus (RCC), Region Occlusion Calculus (ROC) [137], and VRCC-3D+ [140]. Each system provides a distinctive methodology for conceptualizing and analyzing the spatial relationships between entities.

The practical utility of these spatial representations is deeply contingent on their integration into real-world applications across diverse fields such as GIS, robotics, and cognitive science. For instance, the representation of a road varies by dimensionality depending on the application: it is one-dimensional in trip planning, two-dimensional in scenarios of planning overtaking behaviour, and three-dimensional in contexts requiring 3D robotic navigation [142]. In the following part, we provide a detailed overview of three calculi frequently utilized in current datasets to construct spatial relationships.

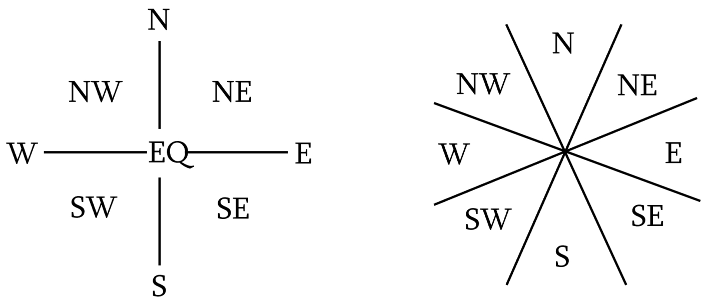



Figure 2.2: Base Relations of the CDC: Projection-Based Relations on the Left and Cone-Based Relations on the Right

The Cardinal Direction Calculus (CDC) [110, 111] is a binary relational calculus utilized for defining the cardinal directions between two points in a 2D plane. Within the CDC framework, the Euclidean plane is segmented into distinct regions centered around a reference point. This division delineates each point's location into one of nine possible cardinal relations: north (N), south (S), east (E), west (W), northeast (NE), southeast (SE), southwest (SW), northwest (NW), or coincident (EQ). There are different segmentation methods, like cone-based and projection-based, as depicted in Figure 2.2.

The calculi within the Region Connection Calculus (RCC) family, such as RCC-8 [131] and RCC-5 [21], facilitate reasoning about connections and part-of relationships between regions. RCC-8 distinguishes between eight fundamental relations: disconnected (DC), externally connected (EC), partially overlapping (PO), equal (EQ), tangential proper part (TPP), non-tangential proper part (NTPP), tangential proper part inverse (TPPi), non-tangential proper part inverse (NTPPi), as depicted in Figure 2.3.

Cardinal Direction Relations (CDR) [113] identify nine primary directional base relations. As shown in Figure 2.4, The space around the reference region $b$ is divided

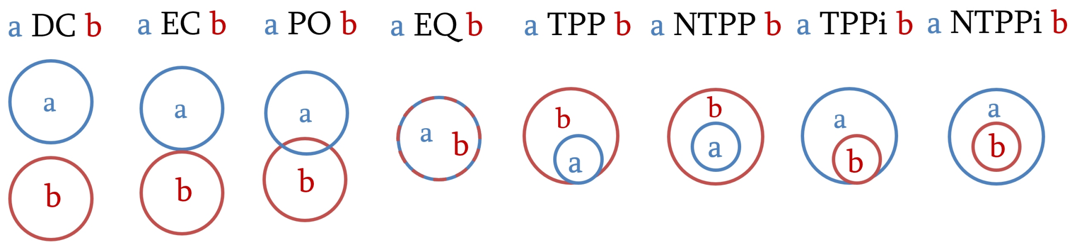



Figure 2.3: Base Relations of the RCC

into nine areas (tiles) by the axes that form its minimum bounding box. These tiles correspond to eight peripheral tiles representing the cardinal directions: $S(b)$, $SW(b)$, $W(b)$, $NW(b)$, $N(b)$, $NE(b)$, $E(b)$, $SE(b)$, and the central tile, $B(b)$, corresponding to the region's minimum bounding box itself. A cardinal direction relation is denoted as $R_1 : \cdots : R_k$ $(1 \leq k \leq 9)$, ranging from single-tile relations like $S$ to composite relations such as $NE : E$ or the encompassing $B : S : SW : W : NW : N : E : SE$. For clarity, when articulating single-tile components of a cardinal direction relation, the sequence is organized as follows: $B$, $S$, $SW$, $W$, $NW$, $N$, $NE$, $E$, and $SE$.

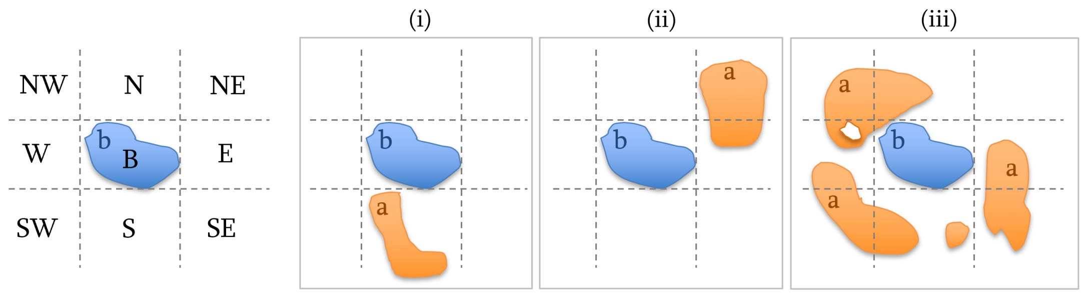



Figure 2.4: Cardinal Directions Relations in CDR. (i) $a\ S\ b$: a is south of b; (ii) $a\ NE : E\ b$: a is partly northeast and partly east of b; (iii) $a\ B : S : SW : W : NW : N : E : SE\ b$: a is distributed across all cardinal directions except northeast relative to b.

The 1- and 2-cross calculi [128] and TPCC [125] are representative calculi for relative orientation and distance. The 1-cross calculus includes basic spatial terms such as 'front', 'back', 'left', and 'right'. TPCC, derived from the 1-cross calculus, introduces finer distinctions, incorporating terms like 'straight' and distinguishing between nuanced orientations such as 'front-left' and 'left-front', while also combining these with distance relations like 'distant' and 'close'.

Despite the development of a wide array of spatial calculi over the past decades, contemporary datasets used to evaluate the spatial reasoning abilities of LLMs are limited in their scope. Typically, these datasets focus on point-based spatial representation, such as the CDC for defining cardinal directions and RCC for topological relationships. However, even within these systems, there is a tendency to employ grid-based approaches that define relationships with specific distances, which deviates from the more general definitions found in traditional spatial reasoning studies. This restricts the potential to explore more complex spatial interactions. The broader range of spatial calculi, including region-based, line-based, and other complex calculi, remains unexplored in evaluating LLMs. This restricts the depth of analysis possible in understanding how LLMs handle intricate spatial relationships.

## 2.3 Spatial Reasoning in Text with LLM

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

Early strides in spatial reasoning in text were marked by the development of formal structures to represent spatial relationships. [16] proposed a spatial ontology, aiming to formalize the representation of spatial relationships. This work laid the groundwork for the subsequent introduction of text-based spatial role labelling (SpRL) [17], which aims to convert natural language text into formal spatial representations. Building upon this, [18] further advanced the field by developing the multimodal spatial role labelling (mSpRL) task. This method extends SpRL by incorporating spatial information from both text and accompanying images, offering a richer representation of spatial relationships. These early datasets, such as SpaceEval (SemEval-2015 task 8) [19] and mSpRL, offered annotated spatial roles and relations, primarily focusing on spatial representation over reasoning.

To bridge the existing gaps in evaluating text understanding and reasoning capabilities of learning algorithms, [39] developed a synthetic QA benchmark that included tasks specifically designed to assess spatial reasoning. Building upon this foundation, the StepGame dataset [20] was introduced, expanding the textual frameworks to encompass a wider array of directional spatial relationships [21, 22], and designed to evaluate robust multi-hop spatial reasoning in text. [143] significantly expanded the resource landscape by constructing three new spatial QA datasets: SpartQA [1], SPARTUN, and RESQ [143]. Furthermore, [144] introduced the spatio-temporal analysis QA benchmark - STBench, which includes over 60K QA pairs across 13 distinct tasks. These tasks span four critical dimensions: knowledge comprehension, spatio-temporal reasoning, accurate calculation, and downstream applications. The wide range of spatial language expressions contained within these datasets poses significant challenges for traditional logical programming approaches. Moreover, they serve as benchmarks for exploring and evaluating the spatial reasoning capabilities of language models. In the latest development, [145] constructed two datasets aimed at evaluating the cardinal direction reasoning abilities of LLMs. The first dataset, created in collaboration with ChatGPT, consists of 100 examples that primarily test the model's recall of world knowledge pertaining to cardinal directions. The second dataset employs a series of designed templates to form the task of accurately identifying cardinal directions in simple scenarios involving movement along or around geographical features. These contributions significantly advance the exploration and assessment of spatial reasoning in LLMs.

[28] assessed ChatGPT's performance on the StepGame and SpartQA benchmarks, uncovering significant limitations in spatial reasoning with success rates of 43.33% for StepGame and 43.75% for SpartQA. Additionally, [146] explored the capabilities of ChatGPT-3.5, ChatGPT-4, and Llama2-7B models across various spatial reasoning tasks, including 2D direction and path labelling, 3D trajectory labelling, and abstract relationship identification within SpartQA. Their findings indicate reasonable performance on 2D direction tasks but significant challenges in 3D trajectory tasks. These studies, while valuable for assessing performance against established benchmarks, do not offer an in-depth critique of the benchmarks themselves and yield results that are not wholly satisfactory.

Beyond evaluating benchmarks or datasets, research has expanded to include case-based analyses assessing the spatial reasoning of LLMs. [147] devised a dialectical evaluation methodology aimed at meticulously identifying failures and delineating the limitations inherent in LLM systems. This approach was executed by engaging models such as GPT-3.5-turbo and GPT-4 in case-based scenarios spanning four distinct areas: basic spatial relations, dimensions of size, shape, and location, affordances and object interactions, and object permanence. The evaluation followed an iterative process, initiating with general inquiries related to a specific topic and followed by more probing questions, especially when initial responses were incorrect. This rigorous method was designed to profoundly test the models' conceptual understanding and reasoning abilities, frequently uncovering erroneous or fundamentally flawed responses from ChatGPT. While this method introduces challenging questions that demand commonsense spatial reasoning, it is time-consuming and requires substantial human effort to execute comprehensively. However, it provides valuable insights into specific reasoning challenges faced by LLMs and can guide the development of spatial reasoning datasets aimed at more effectively evaluating LLM capabilities.

# Chapter 3

# Evaluating Spatial Reasoning in LLMs

The rapid advances in LLMs have sparked extensive discussions about their reasoning capabilities. Developers of these models assert that AI systems like GPT-3 match or even surpass human performance in a variety of tasks, with demonstrations of GPT-3's proficiency across different benchmarks [43]. However, there is a counter-narrative from critics who argue that LLMs exhibit limited reasoning abilities. Works such as [148] and discussions by [28] highlight areas where LLMs consistently fall short compared to human reasoning, reinforcing critiques like those made by Yann LeCun "they (LLMs) make a lot of factual errors, logical errors, have inconsistencies, have limited reasoning abilities, and they are pretty gullible".

Spatial reasoning is inherently complex [149], which requires both conceptualizing spatial relations and performing logical deductions over multiple steps. The computational complexity of solving spatial problems varies depending on the number of objects involved, the variety of constraints, and the reasoning hops required. To objectively evaluate these claims in the spatial reasoning domain, several benchmarks have been developed specifically to assess the spatial reasoning abilities of LLMs. Representative benchmarks include bAbI, StepGame, SpartQA, and SpaRTUN, each aimed at probing different facets of spatial reasoning. These benchmarks pose challenges that require models to infer new spatial relationships from given facts or to check the consistency of existing relationships, typically through multiple-choice questions with a single correct answer. The spatial inference ability of different models is judged and compared based on a straightforward and quantifiable measure - test accuracy. Accuracy is a primary concern when evaluating LLMs, as referenced in multiple studies [43, 150, 151].

This section delves into the unique aspects of spatial reasoning that these benchmarks test, highlighting their inherent challenges and limitations. Through a detailed examination of these benchmarks, we seek to pinpoint the current limitations in our assessment methods and gauge how far we are from fully understanding the spatial reasoning abilities of LLMs. This investigation will enhance our understanding of the true spatial reasoning ability of these advanced AI systems, offering insights into their practical applications and potential improvements.

## 3.1 bAbI

### 3.1.1 Task Overview

The bAbI benchmark [39], featuring a collection of synthetic tasks, was crafted to evaluate learning algorithms in terms of their text understanding and reasoning abilities. Among its 20 tasks, Tasks 17 and 19 are specifically designed for spatial reasoning evaluation.

Task 17 tests LMs' ability to understand and reason about relative spatial relations 'left', 'right', 'above', and 'below'. The task operates within a 5x5 grid environment. In this structured setting, three entities are sequentially positioned at specific nodes. The placement of each entity is determined by its spatial relation to the adjacent nodes. The narratives distinguish three entities based on their color and shape. Each example can include up to 10 sentences - 2 describing spatial relations between two pairs of objects and 8 for generating questions about a different pair, as illustrated in Figure 3.1. These questions are structured in a yes/no format, with answers based on the entities' actual positions on the grid.

Task 19 is centred around identifying paths between specified objects, utilizing the four cardinal directions: north, south, east, and west. The entities are typically locations such as hallways, offices, and bathrooms. In the 'en-valid-10k' version of bAbI[1], each story is structured to include five sentences that detail spatial relationships, of which two describe the actual path and three serve as decoys, as depicted in the example in Table 3.1. The primary challenge of this task is to accurately trace a sequential path from the starting point to the destination. The inclusion of decoy sentences adds a layer of complexity to the task.

[1] https://www.kaggle.com/datasets/roblexnana/the-babi-tasks-for-nlp-qa-system

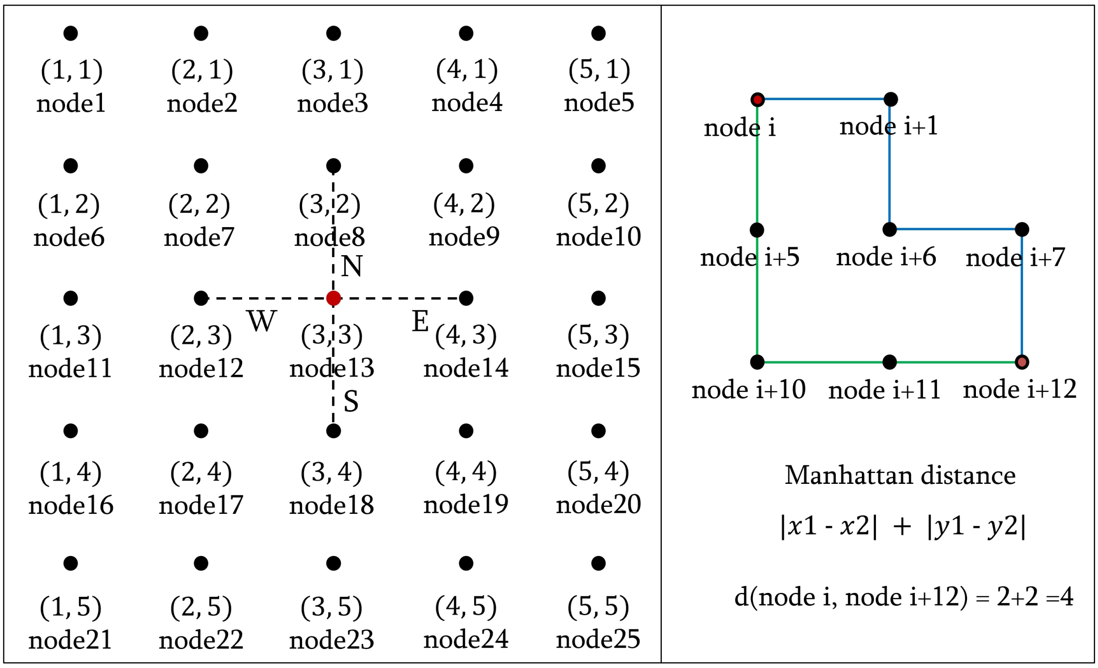



Figure 3.1: Grid system used in bAbI.

| **Task 17: Positional Reasoning** | | **Task 19: Path Finding** |
|---|---|---|
| The red square is below the blue square. | | |
| The red square is to the left of the pink rectangle. | | The garden is west of the bathroom. |
| Is the blue square below the pink rectangle? | A: no | The bedroom is north of the hallway. |
| Is the pink rectangle to the left of the blue square? | A: no | The office is south of the hallway. |
| Is the blue square to the left of the pink rectangle? | A: yes | The bathroom is north of the bedroom. |
| Is the pink rectangle to the left of the blue square? | A: no | The kitchen is east of the bedroom. |
| Is the pink rectangle above the blue square? | A: no | |
| Is the pink rectangle to the left of the blue square? | A: no | How do you go from the bathroom to |
| Is the pink rectangle above the blue square? | A: no | the hallway? |
| Is the blue square above the pink rectangle? | A: yes | A: south, south |

Table 3.1: Examples of Task 17 and Task 19 from the bAbI's envalid-10k dataset version.

### 3.1.2 Textual Stories and Question Generation

Each example is constructed using a sequence of clauses and questions, then converted from abstract representations into human-readable natural language through the use of specific templates for each type. For each clause and question, a corresponding template is randomly selected from the pool and filled out to render complete sentences. These templates are meticulously designed to ensure the sentences produced are both grammatically correct and contextually appropriate. Directional expressions in the templates cover cardinal directions such as ‘north’, ‘south’, ‘east’, and ‘west’, along with relative positions like ‘above’, ‘below’, ‘to the left of’, and ‘to the right of’.

For Task 17, entities are named by combining a randomly chosen color from options like ‘red’, ‘blue’, ‘pink’, and ‘yellow’, with shapes such as ‘square’, ‘rectangle’, ‘triangle’, or ‘sphere’. In Task 19, entities are named after locations such as ‘bedroom’, ‘bathroom’, ‘kitchen’, ‘office’, ‘garden’, and ‘hallway’, randomly selected to fit the context of the task. The clause template used across tasks is structured as ‘`the [object1] is [candidate position] the [object2]`’. For Task 17, the format for questions is ‘`Is the [object1] [candidate position] the [object2]?`’, requiring a response of ‘yes’ or ‘no’. In contrast, Task 19 responses are formatted as two words separated by commas, reflecting the distinct requirements of the task and ensuring clarity in the evaluation of spatial relations.

### 3.1.3 Reasoner to Get Label

The bAbI spatial reasoning tasks leverage a $width \times height$ grid as a spatial framework, depicted in Figure 3.1. This grid acts as the spatial reference, with each entity being assigned to a specific node (position) on the grid where each node is assigned a unique position, marked by coordinates $(x, y)$ and indexed from 1 to $width \times height$; with Node 1 representing the top-left corner and Node $width \times height$ the bottom-right corner.

In Task 17, each example comprises two clauses and eight questions centered around positioning three objects on the grid, beginning with the first object centrally placed at position (3,3). The placement of subsequent entities is determined by moving from this central point using predefined directional commands (north (`n`), south (`s`), east (`e`), west (`w`)), where each command corresponds to a specific movement vector on the grid: `n` - $(0, 1)$, `s` - $(0, -1)$, `e` - $(1, 0)$, `w` - $(-1, 0)$. This setup ensures that entities are positioned in adjacent nodes based on randomly chosen directions from these options.

Once the entities are positioned, the system formulates clauses that delineate the spatial relationships between consecutive entities based on their relative coordinates. This leads to the generation of yes/no questions that probe the relative positions of the shapes, such as “*Is the red square to the east of the blue triangle?*” These questions, targeting the relative $x$ (east-west) or $y$ (north-south) coordinates, are structured to reflect true or false conditions accurately depending on the actual spatial configuration of the entities. For instance, if one entity occupies the position $(3, 3)$ and another is at $(3, 4)$, it is established that the former is north of the latter, providing a practical demonstration of the grid-based spatial reasoning facilitated by this task.

### 3.1.4 Limitations and Problems

#### Limited Relations and Restricted Settings

The bAbI tasks, intentionally designed as simplified 'toy tasks,' have inherent limitations in thoroughly testing spatial reasoning capabilities. These tasks simplify spatial relations to basic cardinal directions - north, south, west, and east (referred to as above, below, left, and right in Task 17) - using fixed distances and angles, which do not reflect the complexity and ambiguity encountered in real-world spatial scenarios. Additionally, the reasoning required in the spatial tasks is limited to simple 2-hop interactions involving only three objects, an oversimplification of real-world requirements.

Furthermore, the use of a uniform template for each relation does not adequately challenge a model's ability to understand and reason within more nuanced and context-rich settings. Thus, while beneficial for basic LLM training, and inspiring subsequent benchmarks and datasets, bAbI tasks do not fully test or equip models to handle the detailed intricacies of real-world spatial reasoning.

#### Repeated questions

In each example of Task 17, eight questions are generated; however, many of these questions are often duplicated, as depicted in Table 3.2. The recurrence of questions results from both the limited pool of objects available and the random methodology used for pairing objects and assigning relations. Given that this selection process is conducted randomly and independently for each iteration, it is possible for the same pair of shapes and candidate relation to be repeatedly chosen across the eight questions. To ensure each of the eight questions remains unique, the system could implement a validation mechanism to identify and eliminate any duplicates by either resampling the shapes or reformulating the questions whenever a repetition arises.

The bAbI dataset is available in various language and scale formats: English (en): Tasks are presented in English, suitable for human understanding. Hindi (hn): Tasks

| | | |
|---|---|---|
| **Stories** | The pink rectangle is below the red square.<br>The triangle is to the left of the red square. | |
| **Questions** | Is the pink rectangle below the triangle? | yes |
| | Is the pink rectangle below the triangle? | yes |
| | Is the pink rectangle to the right of the triangle? | yes |
| | Is the triangle to the right of the pink rectangle? | no |
| | Is the pink rectangle to the left of the triangle? | no |
| | Is the triangle to the right of the pink rectangle? | no |
| | Is the pink rectangle to the right of the triangle? | yes |
| | Is the pink rectangle to the right of the triangle? | yes |

Table 3.2: Example in task 17 with duplicate questions. The questions in the same colorsâ€"blue, green, and redâ€"are identical questions.

are provided in Hindi, also intended for human comprehension. Shuffled: Text with shuffled letters, rendering it unreadable to humans and forcing the model to rely more on the provided training data for learning. Each format is available in both standard and expanded versions: Standard Versions (**en**, **hn**, **shuffled**): Contain 1,000 training examples. Expanded Versions (**en-10k**, **hn-10k**, **shuffled-10k**): Each features 10,000 training examples for more extensive training opportunities.

Table 3.3 details the question repetition within the **en-valid** and **en-valid-10k** sets, with a focus on Task 17. In the en-valid-10k set, there are 1,000 test questions from 125 examples, with each example contributing 8 questions. Data analysis reveals that only just over a third of these questions are unique, with the remainder being repeated. Specifically, about 20% of the questions are duplicated once, while approximately 5% recur three times. The likelihood of a question being repeated four or five times is relatively low but not negligible. There are no instances of questions being repeated seven or eight times. This breakdown helps in understanding the frequency and pattern of question repetition, which is useful for evaluating the training effectiveness and potential biases in model learning.

To investigate the impact of question repetition, we conducted experiments using the Llama-3-8B-Instruct model on two versions of the dataset: the original set, which contains 1,000 examples, and a filtered set where repeated questions were removed,

| **Occurrences** | **en-valid** | | | **en-valid-10k** | | |
|---|---|---|---|---|---|---|
| | **train** | **valid** | **test** | **train** | **valid** | **test** |
| Total | 904 | 96 | 1000 | 9000 | 1000 | 1000 |
| Real | 562 | 65 | 632 | 5716 | 641 | 632 |
| Distinct | 311 | 42 | 363 | 3327 | 376 | 363 |
| Repeated | 251 | 23 | 269 | 2389 | 265 | 269 |
| Repeated - twice | 182 | 15 | 186 | 1688 | 195 | 186 |
| Repeated - three times | 47 | 8 | 69 | 536 | 50 | 69 |
| Repeated - four times | 22 | 0 | 12 | 142 | 16 | 12 |
| Repeated - five times | 0 | 0 | 2 | 17 | 4 | 2 |
| Repeated - six times | 0 | 0 | 0 | 6 | 0 | 0 |
| Repeated - seven times | 0 | 0 | 0 | 0 | 0 | 0 |
| Repeated - eight times | 0 | 0 | 0 | 0 | 0 | 0 |

Table 3.3: Summary of Question Repetitions for the en-valid and en-valid-10k Versions of the Task 17 Dataset. '*Total*': Overall number of questions across training, validation, and testing sets. '*Real*': The sum of both unique and repeated questions within the dataset. '*Distinct*': Total number of unique questions with no repetitions. '*Repeated*': Total number of questions that appear more than once. '*Repeated-twice to Repeated-eight times*': Detailed count of questions repeated a specific number of times, from twice up to eight times.

resulting in 632 unique examples.

We utilized the input prompt1 "`Q:[question]`$\backslash n$`A:`" conforming to the standard QA format detailed in [152]. An example of this format is displayed in Table 3.4. For evaluation purposes, if the final sentence after $\backslash n$ in the output contains 'YES', 'Yes', or 'yes', the predicted answer will be classified as 'Yes'. If it contains 'No' or 'NO', the prediction will be classified as 'No'. Responses indicating uncertainty, such as 'We can't determine' or 'We cannot determine', and containing variations of 'can't', 'cannot', 'we', or 'We', are categorized as 'DK' (Don't Know). In cases where the output does not clearly indicate the category, we manually review the output to ascertain the appropriate classification.

The results are presented in Table 3.5. The Llama model performs slightly better

| | |
|---|---|
| **Prompt** | Q: The red sphere is above the yellow square. The triangle is to the left of the yellow square. Is the triangle below the red sphere?<br>A: |
| **Output** | The information provided does not give us enough details to determine whether the triangle is below the red sphere or not. |

Table 3.4: Example of Llama-3-8B response of DK.

on the full dataset than on the filtered set of unique examples. Most of the wrong predictions can be attributed to the 'DK' (Do not Know) category, as illustrated by the example in Table 3.4. The model makes such predictions because the question lacks information indicating that each relation follows a grid-based system, leading it to rely on common spatial relations for reasoning instead.

| | **Prompt1** | | **Prompt2** | |
|---|---|---|---|---|
| | **Yes, No, DK** | **Yes, No** | **Yes, No, DK** | **Yes, No** |
| **Unique Examples (632)** | 40.35% | 62.66% | 61.71% | 61.71% |
| **All Examples (1000)** | 44.90% | 63.80% | 61.70% | 61.70% |

Table 3.5: Accuracy of Llama-3-8B-Instruct on the bAbI en-valid-10k dataset under two prompting settings. Prompt1 uses the standard QA format, while Prompt2 explicitly instructs the model to answer with either 'Yes' or 'No'. The table reports accuracy both including and excluding 'DK' responses, for all test instances and for unique examples.

Consider cases where some outputs interpret 'DK' as 'No', such as in the following generation: *"A classic lateral thinking puzzle! The answer is... NO. The statements only describe the vertical relationships between the shapes, not the horizontal relationships. We know the red square is below the pink rectangle, and the pink rectangle is below the red sphere, but that doesn't tell us anything about their left-right positions."* In addition to treating 'DK' as a distinct answer category alongside 'Yes' and 'No', we also consider treating all 'DK' responses as 'No'. This adjustment significantly improves accuracy, as shown in Table 3.5. By reclassifying 'DK' as 'No', accuracy increased from 40.35% to 62.66% in the unique example set, and from 44.90% to 63.80% in the complete dataset. Across both methods of response classification, the full set yielded slightly higher accuracy compared to the unique set.

For controlling for 'DK' responses, we conducted an additional experiment to test whether explicitly instructing the model to respond only with 'Yes' or 'No' could reduce or eliminate the use of uncertain answers like 'DK'. In this setting, we used Prompt2, which appends the instruction: "`Q:[question] Please answer with either 'Yes' or 'No'.\nA:`". As shown in Table 3.5, the overall accuracy remained largely comparable to Prompt1 - only slightly decreasing from 63.80% to 61.70% for all examples, and from 62.66% to 61.71% for unique examples.

## 3.2 StepGame

### 3.2.1 Task Overview

Building upon bAbI, the StepGame benchmark [20] also utilizes a grid-based system to build the point-based directional spatial reasoning task. It introduces higher complexity in three key aspects:

1. An expanded set of directional spatial relations is included, encompassing eight relations: top (north), down (south), left (west), right (east), top-left (north-west), top-right (north-east), down-left (south-west), and down-right (south-east). Each is defined by a unique angle and distance, e.g., $(1, -1)$ for down-right. Additionally, an 'overlap' relation is included to denote overlapping object locations.

2. Enhanced multi-hop reasoning challenges: Moving beyond the 2-hop reasoning in bAbI, StepGame increases the complexity to span 1-hop to 10-hop sequences. Here, the term 'hop' quantifies the number of paired relations provided within a narrative, with the upper diagram in Figure 3.2 showcasing an example of a 10-hop reasoning sequence. The lower-right diagram of Figure 3.2 illustrates the sequential building of relational constraints based on $k$, the number of relationships. This produces a chain of constraints linking objects in a direct path from $o_0$ to $o_1$, continuing through to $o_k$. This enhancement allows for a deeper examination of a model's capacity to navigate and infer extended relational chains.

3. Employ richer, crowd-sourced sentence templates describing eight possible spatial relations between two entities, which serve as the basis for generating story-question pairs. For each spatial relationship, there exists a rich set of crowd-sourced templates, which enriches the variety and complexity of the generated content.

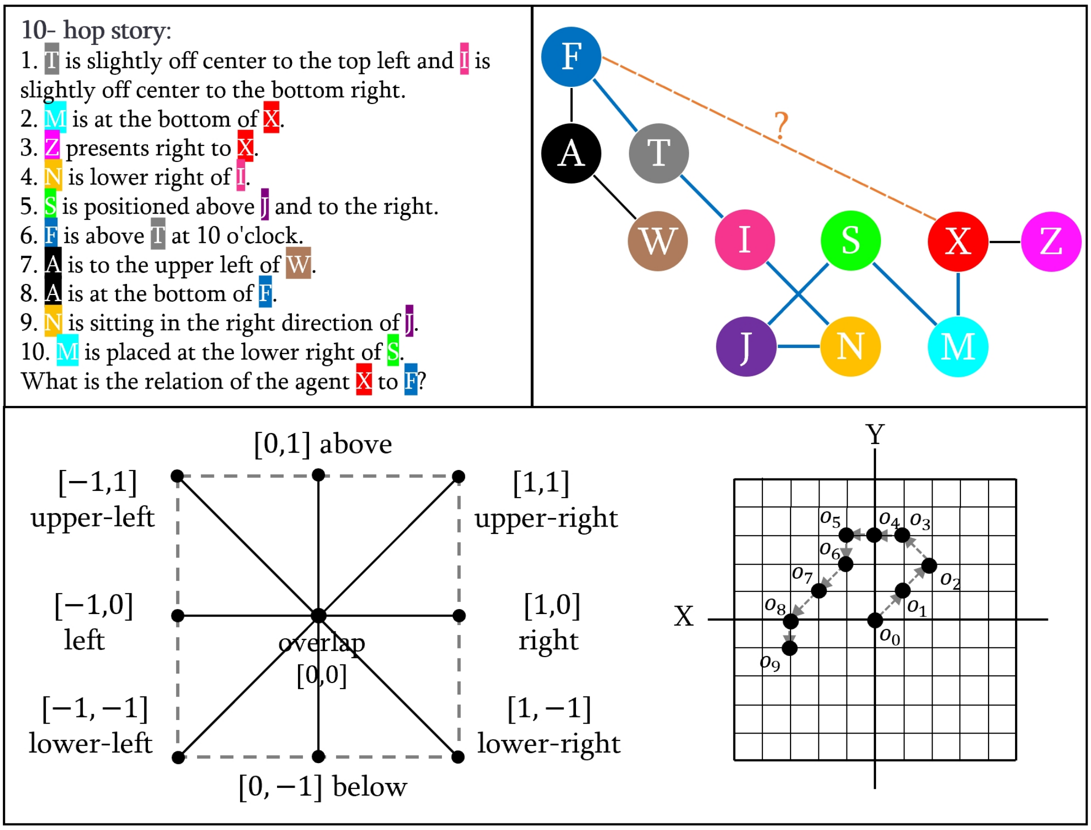



Figure 3.2: Overview of StepGame's reasoning challenges and spatial relationship configurations. Upper: Example of 10-hop reasoning, featuring a question regarding two entities that are not directly connected in the stories. The diagrams on the right do not form part of the input to the AI system but are for illustrative purposes only. Lower Left: Illustration of coordinate settings for the nine spatial relationships, each defined by a fixed distance and angle. Lower Right: Illustration of test instance constraint chain building process in StepGame, where relations are sequentially sampled to connect one object to another from the starting to the end objects.

### 3.2.2 Textual Stories and Question Generation

In a $k$-hop example within the benchmark, the story consists of $k$ sentences derived from predefined sentence templates. These sentences are constructed by replacing placeholder entity names with $k+1$ randomly selected letters from ‘A’ to ‘Z’ and assigning randomly sampled relations from the set of possible directional relations. The question pertaining to the narrative is formatted using a template: ‘`What is the relation of` `the agent [entity1] to the agent [entity2]?`', where [entity1] and [entity2] are randomly chosen from the $k+1$ entities involved in the story. This structure tests the model's ability to infer spatial relationships based on the provided narrative.

For the clean (simpler) version, each story is constructed with $k$ relations ranging from $(o_0, o_1)$ to $(o_{k-1}, o_k)$. In the noise (harder) version, the basic $k$ relational steps are maintained, but when $k > 3$, additional non-relevant information, or 'noise', is interspersed within the story to increase the difficulty and test the model's ability to distinguish relevant from irrelevant data. There are three types of noise, as depicted in Figure 3.3: (a) supporting noise, which establishes additional pathways linking objects in the simple chain; (b) irrelevant noise relations, introducing relations that connect new objects to those in the existing chain without contributing to the main pathway; and (c) disconnected noise relations, comprising relations that have no connection to the simple chain.

Figure 3.3 presents an example with $k = 9$. The 'simple relations' in the figure form the clean version of the story. The query involves two objects randomly selected from this sequence (marked as red nodes $o_2$ and $o_6$ in the figure). Irrelevant noise is generated by establishing relationships between a randomly chosen hop object and an additional noise object, depicted in dark blue in the figure. Disconnected noise is created between two noise objects, shown in purple, which do not link back to the main sequence of the story. Supporting noise starts with an object from the main sequence, extends through one or more noise objects, and eventually links back to another object in the sequence, enhancing the complexity of the relationships and the narrative structure.

The idea behind supporting noise is to introduce additional steps between two entities already in the narrative but not directly linked in the original sequence. To add this noise, two entities $noise_1$ and $noise_2$ are randomly selected from the $k+1$ entities, ensuring they have a Manhattan distance between them of at least 2 and no more than 7, calculated by the sum of the absolute differences in their $x$ and $y$ coordinates. Then from the remaining entity name candidates (letters A-Z not used in the simple story), take $distance - 1$ entities. Then, iteratively adds new sentences to the story that describe a sequence of actions moving from $noise_1$ to $noise_2$, with action only selected from top, down, left, and right (with changing distance only 1). So, in this way, another line going from $noise_1$ to $noise_2$ is also built. We further categorize supporting noise into seven distinct types to analyze their impact on spatial reasoning:

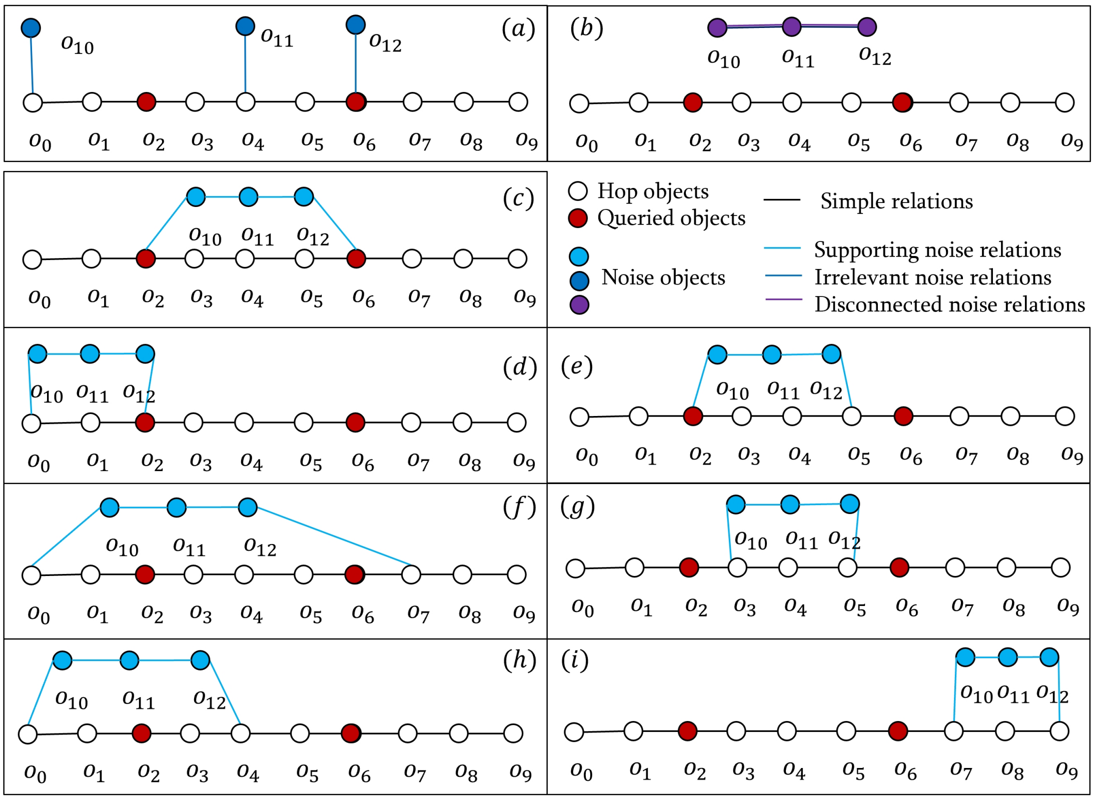



Figure 3.3: Influence of added distracting noise in StepGame. *Hop objects* and *Simple relations* denote the entities and their relationships in the 'clean' version of the story, respectively. *Queried objects* refers to the two entities targeted in the query. *Noise objects* and *Supporting/Irrelevant/Disconnected Noise relations* represent the elements exclusively introduced in the 'noise' version of the story.

1. Involving both query objects, as shown in scenario (c) in Figure 3.3. Both objects in the simple chain used to construct supporting noise relations are the query objects. This introduces an alternative reasoning path from $o_2$ to $o_6$, providing an additional route to a solution.

2. Involving one query object, scenarios (d) and (e) in Figure 3.3 are of this type. In (d), the noise relations are deemed irrelevant as they do not affect the relationship between the queried objects. In (e), similar to scenario (c), the noise introduces a potential additional reasoning path from $o_2$ to $o_6$ that may enhance the problem-solving process.

3. With no query objects, scenarios (f), (g), (h), and (i) are of this type. In (f) and (h), though the supporting relations do not involve the query objects, they can extend the reasoning pathway with several additional steps. For instance, in (f) the pathway extends through several additional steps $o_2 - o_1 - o_0 - o_{10} - o_{11} - o_{12} - o_7 - o_6$, increasing the complexity compared to the original four-step chain $o_2 - o_3 - o_4 - o_5 - o_6$. In (g), This type of noise can form another path between two queried objects. In (i), the noise relations have no connection to the reasoning chain from $o_2$ to $o_6$ and thus do not aid the resolution of the query.

This refined classification aids in understanding how different noise configurations can either complicate, facilitate, or remain neutral regarding the spatial reasoning required to solve queries within the dataset.

### 3.2.3 Problems and Limitations

#### Hop definition

In StepGame, the term '*hop*' refers to the number of relations presented within a story; however, it does not necessarily denote that the reasoning pathway between two queried objects traverses all $k$ relations. For instance, Figure 3.3 illustrates a scenario where $k = 10$, involving ten entities (W, A, F, T, I, N, J, S, M, X, Z), yet the required reasoning pathway from two queried entities (F, X) comprises only seven steps (F-T, T-I, I-N, N-J, J-S, S-M, M-X, X-Z).

Table 3.6 details the distribution of reasoning steps for 10,000 stories under each hop configuration, reflecting the frequency of specific reasoning lengths. According to the statistical summary, the percentage of irrelevant relations increases as $k$ increases (from 100% for 1-hop to 40.3% for 10-hop), and the average number of reasoning steps required as $k$ increases (1.0 for 1-hop and 4.03 for 10-hop). The proportion of $k$ reasoning steps for $k$-hop story, which means all $k$ relations in the story are useful in getting the answer to the question, is quite low. For 10-hop, the proportion is only 1.81%. This setup implies that as $k$ increases, more relations are provided, potentially increasing the proportion of irrelevant relations.

This analysis underscores that higher $k$ values introduce more relations, which may include an increased number of irrelevant connections, complicating the reasoning process. Therefore, even in the '*clean*' version of StepGame, particularly in stories with a higher number of hops, distracting noise elements can obscure the primary pathway

| Hop \ Step | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 10000 | / | / | / | / | / | / | / | / | / | 1.0 |
| **2** | 6610 | 3390 | / | / | / | / | / | / | / | / | 1.34 |
| **3** | 4988 | 3334 | 1678 | / | / | / | / | / | / | / | 1.67 |
| **4** | 4058 | 2934 | 1937 | 1071 | / | / | / | / | / | / | 2.00 |
| **5** | 3368 | 2696 | 1979 | 1279 | 678 | / | / | / | / | / | 2.32 |
| **6** | 2892 | 2322 | 1969 | 1348 | 987 | 482 | / | / | / | / | 2.67 |
| **7** | 2451 | 2209 | 1804 | 1409 | 1087 | 688 | 352 | / | / | / | 2.99 |
| **8** | 2218 | 1972 | 1674 | 1376 | 1101 | 832 | 551 | 276 | / | / | 3.32 |
| **9** | 2065 | 1796 | 1499 | 1374 | 1144 | 837 | 650 | 408 | 227 | / | 3.62 |
| **10** | 1850 | 1576 | 1403 | 1303 | 1113 | 902 | 697 | 588 | 387 | 181 | 4.03 |

Table 3.6: Comparison of reasoning complexity across different hop settings in the StepGame test set. ‘ Row *1-10*’ corresponds to each specific hop scenario, and ‘Column *1-10*’ represents the total number of examples with reasoning steps equal to the corresponding number. ‘*Average*’ represents the average number of reasoning steps across 10,000 examples for each hop setting.

necessary to derive the answer.

### Spatial Relation Configurations

The spatial relationship configuration in StepGame, while detailed, introduces certain limitations that could impact the accurate assessment of language models’ spatial reasoning capabilities. In this benchmark, the basic cardinal directions (top, down, left, right) are retained from bAbI and new combinational directions are specified with precise coordinates: *down-right* as $(1, -1)$, *down-left* as $(-1, -1)$, *top-right* as $(1, 1)$, and *top-left* as $(-1, 1)$. These relations are visually represented on a grid as shown in the lower-left diagram of Figure 3.2.

This configuration ensures unique solutions for each instance by starting with the first object at $(0, 0)$ and sequentially calculating the coordinates for subsequent objects based on each relation. However, such a setup simplifies the reasoning problem and does not align with commonsense human understanding, which does not typically confine directional relationships to strict distance or angular constraints. For instance, in everyday language, stating "A is east of B" merely implies that A's x-coordinate is greater than B's, without specifying the degree of separation or exact angular alignment. Our research [153] suggests that the most challenging aspect for LLMs in StepGame is not the spatial reasoning itself but constructing the object-linking chain from shuffled relations. When the reasoning chain is pre-constructed, models like GPT-4 exhibit considerable capability in handling such tasks.

#### Template Problems

According to [31], their examination of StepGame revealed significant issues: of 108 examples with incorrect predictions, 107 suffered from data labelling errors. Our further investigation of StepGame pinpoints that the core problem resides not in the labelling method but critically in the textual story generation. The templates designed for story creation contained errors that misrepresented the intended tasks, leading to wrong labels and, consequently, skewed model performance evaluations. These template errors, previously overlooked, have led to studies [28], [31] based on this flawed benchmark, resulting in an inaccurate assessment of LLMs' capabilities.

We conducted a detailed analysis of errors in the relational text mappings within the StepGame benchmark. Story generation relies on randomly sampled crowd-sourced natural language templates corresponding to eight specific spatial relations: *top, down, left, right, top-left, top-right, down-left,* and *down-right.* Prior to February 2024, these templates[1] encompassed 214 different forms, including 24 templates each for *left* and *right*, 28 each for *top* and *down*, 27 each for *top-right* and *down-left*, and 28 each for *top-left* and *down-right.*

Our findings, as detailed in Table 3.7, show that out of these 214 templates, 14 were found to be erroneous. The spatial relationships `(AA, top, BB)` and `(AA, left, BB)` were accurately represented without mistakes. Conversely, the mapping `(BB, right, AA)` exhibited the highest number of inaccuracies with four incorrect templates, followed by `(BB, down, AA)` with two. The question arises as to why there are so many such errors in the crowd-sourced expressions; presumably, this is down to insufficient quality control over the crowd worker responses.

A common error was an inversion or misplacement of the labels `AA` and `BB`. For

[1] Following the publication of our AAAI paper, errors in the templates were corrected.

| Mapping | Original Incorrect Statement |
|---|---|
| (BB, right, AA) | AA and BB are parallel, and AA on the right of BB.<br>AA and BB are parallel, and AA is to the right of BB.<br>AA and BB are horizontal and AA is to the right of BB<br>AA and BB are both there with the object AA is to the right of object BB. |
| (BB, down, AA) | AA is placed at the bottom of BB.<br>AA is at the bottom of BB and is on the same vertical plane.<br>AA presents below BB. |
| (AA, down-left, BB) | BB is there and AA is at the 10 position of a clock face.<br>BB is positioned below AA and to the left. . |
| (BB, top-right, AA) | Object A is above object BB and to the right of it, too.<br>AA is diagonally to the upper right of BB. |
| (BB, down-left, AA) | BB is to the right and above AA at an angle of about 45 degrees.<br>BB is diagonally left and above BB. |
| (AA, down-right, BB) | AA is to the right and above BB at an angle of about 45 degrees. |

Table 3.7: Incorrect sentence templates in the StepGame dataset

example, the template "*AA and BB are parallel, and AA on the right of BB.*" was assigned to the relation `(BB, right, AA)`, but actually corresponds to the relation `(AA, right, BB)`.

Another type of error arises from incorrect entity naming in statements. For instance:

- The template '*BB is diagonally left and above BB.*" intended for `(BB, down-left, AA)`, leads to sentences like "*I is diagonally left and above I.*". This template fails to clarify the second entity, rendering it indeterminate from the statement alone.
- The template "*Object A is above object BB and to the right of it, too.*" for `(BB, top-right, AA)`. This template results in sentences like "*Object A is above object R and to the right of it, too.*" Although this should denote the relationship `(K, top-right, R)`, the consistent use of 'Object A' across instances makes it impossible to identify the second entity accurately.

Some templates are mistakenly used to describe two different relations, yet they do not accurately reflect either, making it impossible to determine which relation was intended based on the sentence alone. For example:

- The template "*AA is to the right and above BB at an angle of about 45 degrees.*" was assigned to the relation `(AA, down-right, BB)`, but actually corresponds to the relation `(AA, top-right, BB)`.

- The template "*BB is to the right and above AA at an angle of about 45 degrees.*" was designed for `(BB, down-left, AA)`, while it actually describes the relation `(BB, top-right, AA)`.

The first statement error is in the use of '*above*', which should be "*AA is to the right and below BB*". The second misuses '*right*' - it ought to be *BB is to the left and above AA*. For example, reading the statement "*Q is to the right and above P at an angle of about 45 degrees.*" in a narrative would naturally lead one to deduce the relation as `(Q, top-right, P)`. It would not typically be interpreted as `(Q, down-right, P)` or `(P, down-right, Q)`, despite potentially being intended to represent one of these relations in the original dataset. These errors underscore the need for careful review and correction of template assignments to ensure that the textual descriptions match the intended spatial relationships.

Table 3.8 displays the percentage of examples that are incorrect, which hints at a rising trend in inaccuracies as $k$ increases, suggesting a potential cumulative impact. To address this issue, we present a refined version of the StepGame dataset for model evaluation. By removing examples with erroneous templates, this revised dataset allows for more accurate evaluations of the true capabilities and limitations of the models.

| | **k=1** | **k=2** | **k=3** | **k=4** | **k=5** | **k=6** | **k=7** | **k=8** | **k=9** | **k=10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **Clean** | 7.61 | 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 3.8: Percentage of incorrect instances across 1-hop to 10-hop test sets. Here, k means $k$-hop reasoning. There are 10,000 samples for the test set for each $k$ before correction.

To assess the influence of template errors, we conducted evaluations using Llama models on both the original and our refined versions of the StepGame dataset. Differing from the binary yes/no answer format in bAbI, StepGame questions ask for relationships between two objects through 'find relation' queries. To ensure uniformity in response formatting, we applied a few-shot (5-shot) prompting strategy, providing

| | Llama-2-7B | | | | Llama-3-8B | | | |
|---|---|---|---|---|---|---|---|---|
| | Original | | Refined | | Original | | Refined | |
| | 100 | 1K | 100 | 1K | 100 | 1K | 100 | 1K |
| **k=1** | 37 | 36.4 | 37 | 37.8 | 60 | 62.6 | 62 | 66.2 |
| **k=2** | 17 | 18.9 | 18 | 20.0 | 34 | 34.1 | 33 | 37.8 |
| **k=3** | 8 | 12.5 | 10 | 12.8 | 18 | 20.9 | 18 | 22.0 |
| **k=4** | 14 | 12.1 | 14 | 12.5 | 10 | 10.6 | 8 | 11.2 |
| **k=5** | 15 | 12.9 | 18 | 13.1 | 23 | 22.0 | 24 | 21.2 |
| **k=6** | 14 | 12.0 | 13 | 12.4 | 23 | 15.9 | 13 | 17.2 |
| **k=7** | 13 | 11.5 | 10 | 12.0 | 23 | 22.8 | 23 | 21.7 |
| **k=8** | 13 | 15.3 | 14 | 13.4 | 20 | 16.5 | 18 | 16.6 |
| **k=9** | 15 | 11.7 | 10 | 12.1 | 10 | 9.3 | 8 | 7.7 |
| **k=10** | 6 | 9.1 | 7 | 8.7 | 5 | 5.0 | 3 | 5.1 |

Table 3.9: Accuracy performance comparison of Llama models on the original and refined test sets in StepGame, evaluated with varying numbers of test examples.

five exemplars alongside the task description to direct the models' answer generation. These evaluations spanned various test subsets, including the first 100, the first 1000 examples, and the complete dataset. The results are displayed in Table 3.9, both Llama-2 and Llaman-3 show slightly improved performance on the refined dataset compared to the original in lower-hop settings.

To evaluate the robustness of model performance with respect to temperature variation, we conducted a series of controlled experiments using the refined 100-question subset with k=5. The temperature parameter varied from 0.0 to 1.0 in increments of 0.1. For each temperature setting, ten repeated runs were performed using Llama-3, and the resulting accuracies were recorded. Figure 3.4 visualizes the distribution, central tendency (mean and median), and variance of the model's performance. At lower temperatures, the model produced more deterministic outputs, leading to minimal variance across runs. As the temperature increased, performance became more volatile: variance rose substantially, and accuracy fluctuated more widely. Although higher temperatures occasionally led to high-performing individual runs, the overall results were less consistent. These findings suggest that evaluation outcomes are sensitive

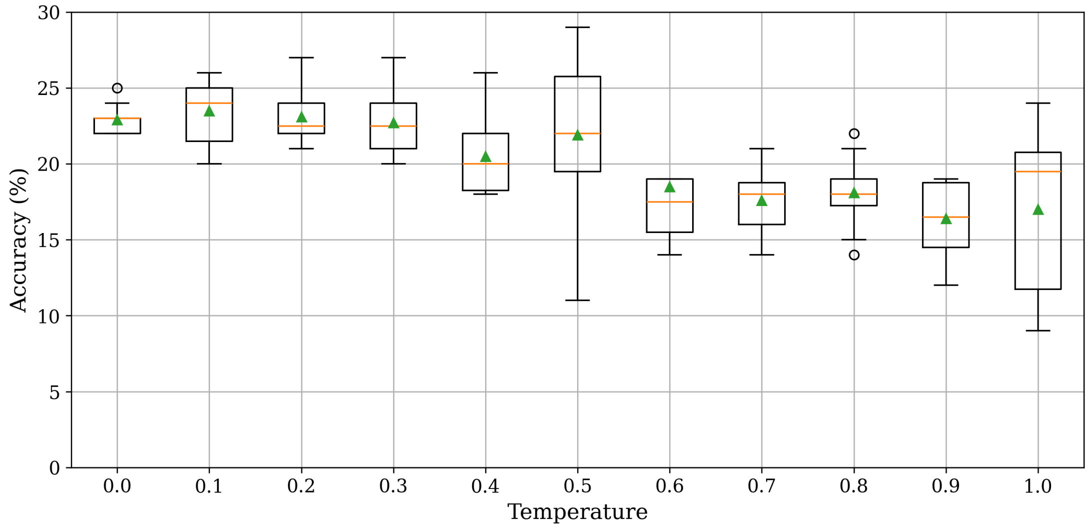



Figure 3.4: Boxplot of accuracy distribution for Llama-3 across ten runs under different temperature settings on the refined-100 data with k=5. The boxes represent the interquartile range, orange lines indicate the median accuracy, green triangles denote the mean accuracy, and circles indicate statistical outliers.

to temperature and that lower temperature values tend to yield more stable results.

## 3.3 SpartQA and SpaRTUN

### 3.3.1 Spatial Reasoning Tasks

SpartQA [1] and SpaRTUN [23] start from 2D images featuring objects (rectangle, triangle, square) distributed across distinct square blocks (scenes). They extend beyond mere directional spatial relationships to include Region Connection Calculus 8 (RCC-8) [21] and distance (near and far). SpaRTUN is an updated version of SpartQA-Auto and contains more relation types and rules.

Unlike the previous two grid-based benchmarks, SpartQA and SpaRTUN's define spatial relations using a square boundary framework. Each spatial relation is determined by the $(x, y)$ coordinates of the lower-left points of the square boundary boxes of two objects and the size of these boxes, as depicted in Figure 3.5.

- For object-to-object relations, EC, NEAR, FAR, LEFT / RIGHT, ABOVE / BELOW are considered;

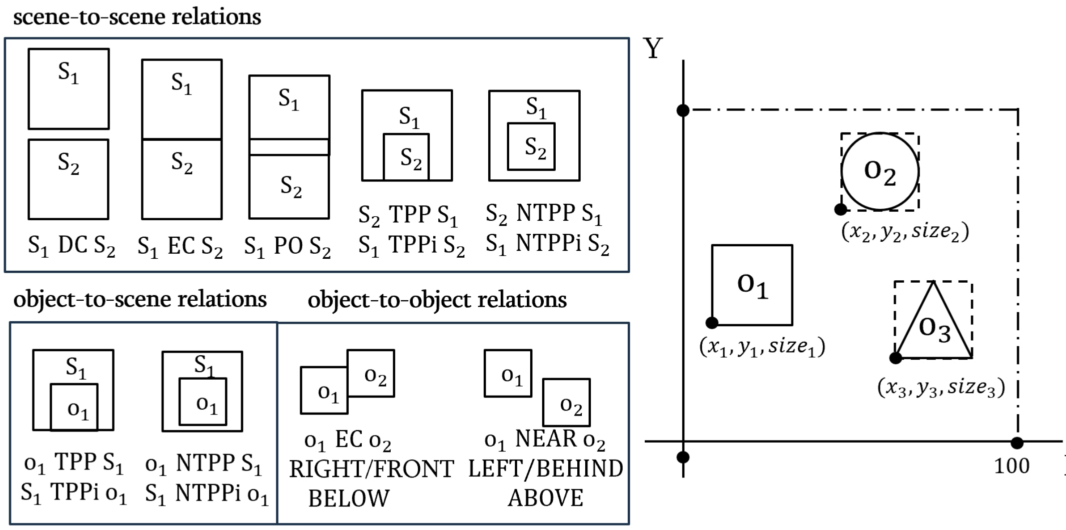



Figure 3.5: Spatial relationships and square boundaries with objects in SpartQA and SparTUN.

- For object-to-scene relations, TPP / TPPi, and NTPP / NTPPi are considered;
- For scene-to-scene relations, DC, EC, PO, TPP / TPPi, and NTPP / NTPPi are considered.

For the questions, there are four types: (1) FR (Find Relation), which identifies relationships between two objects; (2) FB (Find Block), where the task is to choose the block containing specified objects; (3) CO (Choose Object), which requires selecting one of two objects that fit certain criteria; and (4) YN (Yes/No), a direct test of the validity of a spatial relationship claim. FR, FB, and CO questions are presented as multiple choice, while YN questions offer choices of ‘Yes’, ‘No’, or ‘DK’ (Do not Know).

However, the tasks of FR, FB, and CO can also be reinterpreted as consistency-checking problems. Instead of straightforwardly identifying specific blocks, objects, or relations, these questions can be reframed to verify whether the spatial configuration provided in the multiple-choice options is consistent with the location of specified objects. This adaptability underscores the interconnected nature of different types of reasoning within intelligent systems.

### 3.3.2 Textual Stories and Question Generation

The story and questions for each example were generated from the selected story triplets using context-free grammar (CFG). They increase the variety of spatial expressions by using a vocabulary of various entity properties and relation expressions. They map the relation types and the entity properties to the lexical forms from a specifically collected vocabulary. Entities within the stories are identified by a combination of size, color, and shape attributes. If multiple entities share the same attributes, they are differentiated by appending a number to their name, such as '*medium orange apple number one*'.

| |
|---|
| **Story:** Two boxes, named one and two exist in the image. Box one covers a medium yellow apple. In box two there is this box. Box two has a medium orange apple which is to the south of a medium yellow apple and touches another medium orange apple. Box two has the medium yellow apple. Medium orange apple number two is covered by this box. South of medium orange apple number one there is medium orange apple number two. |
| **YN Question:** Is a medium yellow apple to the south of a fruit? **Answer:** No<br>**FR Question:** Where is the medium yellow apple in box two regarding medium orange apple number two? **Answer:** ABOVE |

Table 3.10: The first test example in SpaRTUN.

### 3.3.3 Limitations and Problem

#### Description Method

Although these two benchmarks include rich spatial relationships, they struggle to provide effective descriptions. They use simple syntax and word choice but lack logical flow and content clarity, particularly in two aspects, as can be seen in the story in Table 3.10.

**1. The sequence of sentences.** The spatial relations are described as a sequence of randomly selected story triplets, which deviates from the typical human approach to describing a scene. In the example from Table 3.10, a more natural human description would typically start with outlining the relationships between two boxes, followed by detailing the contents of each box and then explaining the relations between the objects.

However, in their narrative structure, there is a lack of an initial summary of the objects contained in each box, with objects being introduced individually and somewhat disjointedly. Additionally, the narrative places the object-to-box relationships prior to the box-to-box relationships, which further diverges from the typical human method of spatial description, leading to potential confusion in understanding the overall spatial layout.

**2. Excessive use of detailed and repetitive entity naming.** It is a common way in benchmarks to describe objects through combinations of their shapes, sizes, and types, like those in bAbI and CLEVR[6]. When applying this naming method to form long paragraphs of stories, problems appear. The excessive use of detailed and repetitive entity naming, involving terms like '*medium yellow apple*', '*medium orange apple number one*', and '*medium orange apple number two*', results in overly lengthy text. This verbosity transforms a simple description such as '*South of A is B*' into a more convoluted one like '*South of medium orange apple number one is medium orange apple number two*'. Such complexity not only adds confusion but also shifts the focus from understanding the spatial relationship to deciphering which specific object is being referred to. This can make it hard for readers to grasp the intended spatial relationships and hinder smooth comprehension.

Consequently, the narrative's lack of smooth flow in textual descriptions makes it difficult for both LMs and humans to form a clear mental image of the entire scene and to grasp information about specific objects in question. This complexity hinders the LLMs from engaging in spatial reasoning effectively and drawing conclusive answers based on the limited information presented.

#### Labels

The example illustrated in Figure 3.6 from the SpaRTUN datasets demonstrates the issue with gold label generation for textual spatial reasoning questions.

The question posed is "*Is a medium yellow apple to the south of a fruit?*" with the labelled answer being '*No*'. This conclusion is based on a presumed object reference where '*a medium yellow apple*', assumed to be the one in box two (1x2), is questioned in relation to '*a fruit*', which refers to as the *medium orange apple number one* (1x0). The reasoning chain used to derive the answer considers *medium orange apple number two* (1x1) located south of the *medium orange apple number one* (1x0), which in turn is south of the *medium yellow apple in box two* (1x2), leading to the conclusion that 1x1 is south of 1x2. Thus, the system deduces the answer to the query (1x2, south, 1x1) as 'No'.

The issue stems from the ambiguous references used in question formulations. When generating questions, each object description begins with '*the*', '*a/any*', or '*all*' (see the example in Figure 3.7), randomly selected and interpreted to have the same meaning. In everyday language, however, terms like '*a fruit*' can refer generically to any fruit present, including the *medium orange apple number one* or any other fruit within the scenario, not exclusively to *medium orange apple number one* (1x0). Moreover, the mention of '*a medium yellow apple*' introduces ambiguity, as the story includes two medium yellow apples without clarifying which is being referenced in the question.

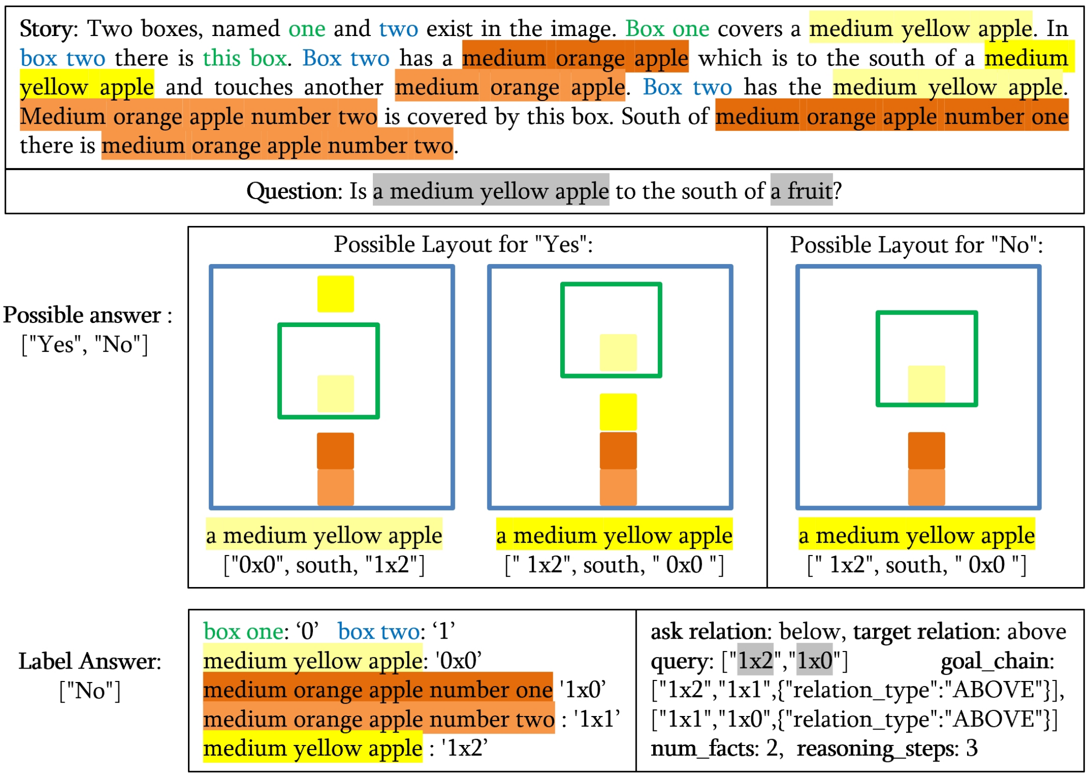



Figure 3.6: Illustration of labelling issues for the first test example in SpaRTUN. 'Possible answers' presents potential scenarios for the two answers based solely on the text description. 'Label Answer' outlines the reasoning process used to generate the gold standard label.

Story: A box called DDD have a midsize white rectangle and a midsize green rectangle. Over the midsize green rectangle there is the midsize white rectangle. Box DDD covers another midsize white rectangle. A box named EEE exists in the image. Within this box is another box called JJJ with a midsize green rectangle and a midsize white rectangle. Above another midsize green rectangle there is the midsize green rectangle. Midsize green rectangle number two is covered by box JJJ. Another midsize green rectangle is inside box JJJ. The midsize white rectangle is over midsize green rectangle number one and is under this thing. Below midsize green rectangle number three is the object which was over midsize green rectangle number two.

box DDD: '0' ' box EEE: '1' box JJJ: '2'
a midsize green rectangle : '0x3 '
a midsize white rectangle: '0x2 '
another midsize white rectangle : '0x0'
midsize green rectangle number one:'2x1'
midsize green rectangle number two:'2x0'
midsize green rectangle number three:'2x3'

Q1: Are all midsize green rectangles under any midsize white rectangle?
Label Answer: ["No"]
query: ["0x3","0x2"], num_facts: 1, reasoning_steps: 1

Q2: Are all things covered by any box?
Label Answer: ["No"]
query:["0x0", "0"], num_facts:1, reasoning_steps:1

Figure 3.7: A test example in SpaRTUN with questions using object names starting with '*all*'. Although the story includes four *midsize green rectangles*, the phrase '*all midsize green rectangles*' in the first question Q1 refers to a specific object, 0x3. In the second question Q2, '*All things*' refers to a particular object, 0x0, rather than all entities mentioned in the story.

In human interpretation, without explicit clarification of the objects referred to in the question and relying solely on the textual story and the question, the answer can be either '*Yes*' or '*No*'. This is because, regardless of which medium yellow apple is considered, there could indeed be a fruit located to the south of it with all the constraints mentioned in the story, considering that '*a fruit*' encompasses more than just the medium orange apple number one. To better illustrate these ambiguities and their impact on reasoning, two potential layouts are presented in Figure 3.6. Irrespective of which medium yellow apple is mentioned, there exists a plausible scenario where a fruit is indeed to the south of it. The answer can also be '*No*' if the two yellow apples are in the same location in the 2D image, as shown in the right figure example. This highlights the need for more unambiguous forming of questions in this dataset to improve the reliability of the derived answers.

In this chapter, we examined several textual spatial reasoning benchmarks and conducted experiments to assess the performance of LLMs, identifying existing problems. In the next chapter, we will build on these findings and explore ways to enhance LLMs' performance on spatial reasoning tasks.

# Chapter 4

# Enhancement of LLMs' Spatial Reasoning Ability

Based on evaluations of prominent LLMs on benchmarks like bAbI, StepGame, and SpartQA, it was observed that these models often struggle with spatial reasoning tasks. This chapter investigates strategies to enhance LLMs' capabilities in tackling spatial reasoning challenges without modifying the underlying model architecture. Specifically, we explore the potential of enhancing LLMs' performance on complex spatial reasoning by integrating them with logical reasoning components, as detailed in Section 4.1. Here, LLMs are utilized to transform spatial descriptions into symbolic spatial relation representations, which are subsequently processed by a logical reasoning program.

Furthermore, we explore various prompting techniques, including CoT and ToT prompting, detailed in Section 4.2 and Section 4.3. CoT [35] incorporates a sequence of intermediate reasoning steps to facilitate problem-solving. However, when applied to StepGame, previous studies [31] 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. Research on other tasks [36, 37] 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' within CoT, we propose refining the CoT prompt to empower language models to perform better in spatial reasoning tasks.

On the other hand, ToT [38] was introduced as a framework that enables LLMs to explore multiple reasoning paths, demonstrating its effectiveness in enhancing 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.

The results of these experimental approaches are presented in Section 4.4.

## 4.1 Combination with Logical Reasoners

In this part, we explore the integration of LLMs with logical reasoners, as depicted in Figure 4.1. The process begins with both a story and a question provided in natural language. The LLM functions as a semantic parser, transforming the textual input into structured spatial data - logical facts that are then inputted into the logical reasoner. This reasoner utilizes these facts, applying predefined inference rules to verify their consistency and to infer the spatial relationships between two objects specified in the query. Through logical inference, the reasoner produces a definitive answer to the question posed.

This pipeline leverages the strengths of both components: LLMs are adept at managing linguistic variability, transforming diverse textual inputs into standardized logical facts - a process referred to as semantic parsing. Logical reasoners, on the other hand, employ formal logic systems that define spatial properties and relationships through axioms. These axioms enable the reasoners to deduce conclusions about spatial relations, thereby deriving answers from the established logical premises.

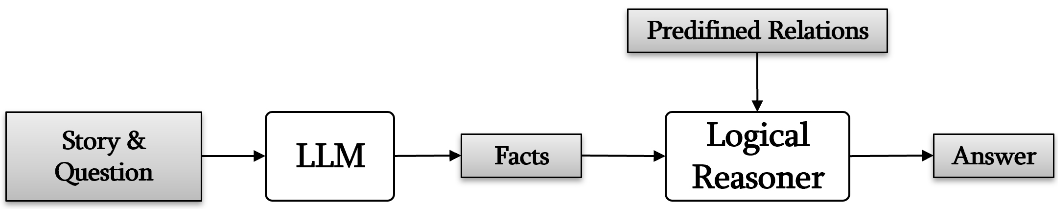
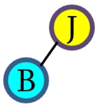



Figure 4.1: The LLM and logical reasoner integration pipeline.

### 4.1.1 LLMs for Semantic parsing

The objective of semantic parsing in this context is to interpret and translate natural language into a structured format that can be efficiently processed by the logical reasoner. This process is similar to the concept of 1-hop reasoning, which involves deducing or retrieving information based on a single relational link between two objects. Figure 4.2 provides a comparative illustration of these concepts.




**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.

| Semantic Parsing | 1-hop reasoning |
|---|---|
| J is on the upper right of B.<br>B is is on the lower left of J. | **Question:** What is the relation of J to B?<br>**Answer**: upper-right |
| top_right("J", "B")<br>down_left("B", "J") | **Question:** What is the relation of B to J?<br>**Answer**: lower-left |

Figure 4.2: An illustrative example for demonstrating semantic parsing and 1-hop spatial reasoning.

We employ LLMs to perform semantic parsing using few-shot in-context learning. This approach enables an LLM to "learn" a new task by being conditioned on a textual description of the task along with a few input-output examples, eliminating the necessity for extensive task-specific fine-tuning. The process is initiated with a manually crafted prompt that includes detailed task instructions, as outlined below:

```
Please parse each sentence into a fact. If the sentence is describing clock-wise information, then 12 denotes top, 1 and 2 denote top_right, 3 denotes right, 4 and 5 denote down_right, 6 denotes down, 7 and 8 denote down_left, 9 denote left, 10 and 11 denote top_left. If the sentence is describing cardinal directions, then north denotes top, east denotes right, south denotes down, and west denotes left. If the sentence is a question, the fact starts with query. Otherwise, the fact starts with one of top, down, left, right, top_left, top_right, down_left, and down_right.
```

### 4.1.2 Logical Reasoner for Spatial Reasoning

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 logical reasoner for spatial reasoning. We implement the rules in constraint format.

For the spatial reasoning tasks in bAbI and StepGame, which only incorporate direction relations based on a grid-based system, it is easy to form such relations in axioms. It is worth pointing out that the definition of spatial relations in stories is with fixed distance, such as $\mathit{offset}(right) = (0, 1)$ and $\mathit{offset}(down - left) = (-1, -1)$. But in questions, when asked about whether there is a certain relation between two objects, it is not with a fixed distance. So we give the definition of the relations in the following ways in our logical reasoner:

| **Relation** | **Definition** | **Relation** | **Definition** |
|---|---|---|---|
| overlap | $x_1 = x_2, y_1 = y_2$ | is_overlap | $x_1 = x_2, y_1 = y_2$ |
| north/top | $x_1 = x_2, y_1 = y_2 + 1$ | is_north/top | $x_1 = x_2, y_1 > y_2$ |
| south/down | $x_1 = x_2, y_1 = y_2 - 1$ | is_south/down | $x_1 = x_2, y_1 < y_2$ |
| east/right | $x_1 = x_2 + 1, y_1 = y_2$ | is_east/right | $x_1 > x_2, y_1 = y_2$ |
| west/left | $x_1 = x_2 - 1, y_1 = y_2$ | is_west/left | $x_1 < x_2, y_1 = y_2$ |
| northeast/top-right | $x_1 = x_2 + 1, y_1 = y_2 + 1$ | is_northeast/top-right | $x_1 > x_2, y_1 > y_2$ |
| northwest/top-left | $x_1 = x_2 - 1, y_1 = y_2 + 1$ | is_northwest/top-left | $x_1 < x_2, y_1 > y_2$ |
| southeast/down-right | $x_1 = x_2 + 1, y_1 = y_2 - 1$ | is_southeast/down-right | $x_1 > x_2, y_1 < y_2$ |
| southwest/down-left | $x_1 = x_2 - 1, y_1 = y_2 - 1$ | is_southwest/down-left | $x_1 < x_2, y_1 < y_2$ |

Table 4.1: Spatial relations and corresponding definitions in our logical reasoner. The left section of the table lists the relations defined for facts, while the right section details the relations defined for questions.

### 4.1.3 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 the logical reasoner mentioned previously for position reasoning. This method parallels the LLM + LR approach but substitutes LLMs with **sentence-to-relation mapping**. Illustrative examples of the mapping can be found in Table 4.2. 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 $o_i\_\nu\_o_j$. Then, we convert this template form into a structured representation $\nu(o_i, o_j)$, where $o_i$ and $o_j$ correspond to the two objects mentioned in $r$, and $\nu$ signifies the spatial relation between $o_i$ and $o_j$. 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 fact is represented as $query(o_0, o_t)$.

Table 4.2 presents several examples: it compares original sentences from StepGame

| **Sentences** | **Template** | **Facts** |
|---|---|---|
| Y and I are parallel, and Y is on top of I. | Y_above_I | top("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 | down("J", "O") |
| A is directly north east of B. | A_upperright_B | top_right("A", "B") |
| What is the relation of the agent B to the agent J? | query_B_J | query("B", "J") |

Table 4.2: Sentence-to-relation mapping examples.

stories, the template translations reflecting relation expressions for story formalization, and the resulting facts used by the logical reasoner. If a sentence forms a question, the fact is prefixed with '*query*'; otherwise, it begins with directional tags such as *top, down, left, right, top_left, top_right, down_left, and down_right*.

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.

## 4.2 CoT Prompting

### 4.2.1 Method

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**. Drawing inspiration from the human reasoning process

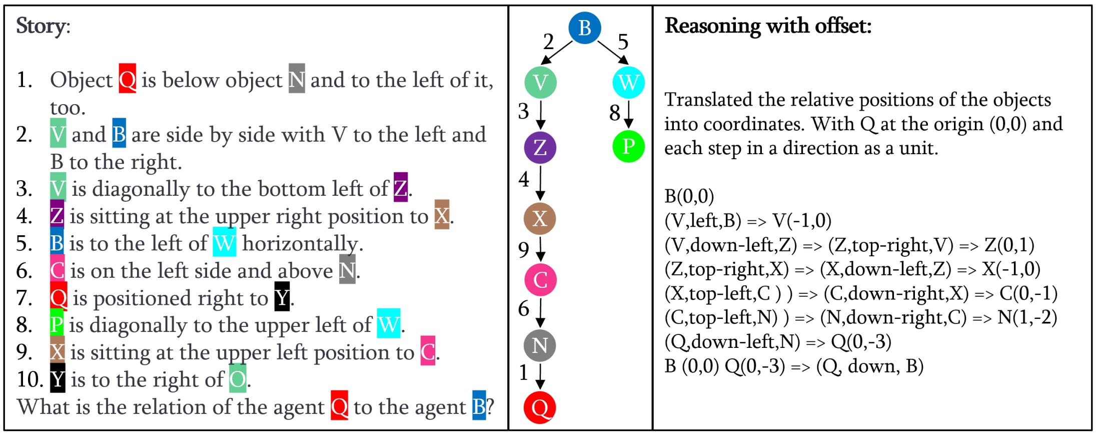



Figure 4.3: Example of a 10-hop reasoning story, with a diagram illustration and a detailed human reasoning process to derive the answer.

depicted in Figure 4.3, we categorize 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 parts of thought are sequentially sampled as a continuous language sequence $c_i = [c_i^{link}, c_i^{map}, c_i^{calcu}]$ using the LLMs $p_\theta$.

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 Section 4.2.2.

2. $c_i^{map}$: Map $r^j$ to a simple relation description such as "$o_i$ is to the $\nu$ of $o_{i+1}$," where $\nu$ represents the key spatial relation from $o_i$ to $o_{i+1}$. The prompt "This means" helps the LM 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. Therefore, it computes the coordinates of $o_{i+1}$ and generates the output like "Therefore, B is at $(x_{o_{i+1}}, y_{o_{i+1}})$."

We provide a 4-shot example in StepGame to illustrate a comparison of our CoT prompt with other methods, as shown in Table 4.3.

| | |
|---|---|
| **Story** | 1. C and M are both there with the object C above the object M.<br>2. Z is at the bottom and Y is on the top.<br>3. Z is at a 45 degree angle to M, in the upper left hand corner.<br>4. Y is placed at the lower left of G.<br>What is the relation of the agent Z to the agent C? |
| **IO** | Answer: left |
| **CoT** [31] | We first link Z and C using the relations in the story. Z is to the upper-left of M. M is below C. So the answer is left. |
| **Our CoT** | Reasoning: Let's suppose Z is at (0,0). We can connect Z and C using the relations given in the story.<br>Start with Z. According to 3, "Z is at a 45 degree angle to M, in the upper lefthand corner." This means M is to the lower-right of Z. M= Z+ offset(lower-right) = (0,0)+(1,-1)=(1,-1). Therefore, M is at (1,-1).<br>Then search for M. According to 1, "C and M are both there with the object C above the object M." This means C is above M. C= M+ offset(above) = (1,-1)+(0,1)=(1,0). Therefore, C is at (1,0). We've reached C. So, considering Z(0,0) and C(1,0), Z is to the left of C.<br>Answer: left |

Table 4.3: Comparison of different prompting methods on a 4-hop test example in StepGame.

### 4.2.2 Prompts

Our CoT prompt utilizes a few-shot format, comprising a task description, several CoT exemplars, and the queried story along with its corresponding question. The task description prompt is as follows:

```
   Given a story about spatial relations among objects, answer the relation between
two queried objects.  Possible relations are overlap, above, below, left, right,
upper-left, upper-right, lower-left, and lower-right.  If a sentence in the story is
describing clockwise information, then 12 denotes above, 1 and 2 denote upper-right, 3
denotes right, 4 and 5 denote lower-right, 6 denotes below, 7 and 8 denote lower-left,
9 denote left, 10 and 11 denote upper-left.  If the sentence is describing cardinal
directions, then north denotes above, east denotes right, south denotes below, and
west denotes left.  In all the spatial relations, assume that all agents occupy a
```

`position on a grid point of equally spaced points in the vertical and horizontal directions and that agents occupy the nearest grid point consistent with the spatial relation. The offsets of 9 spacial`[1] `relations: offset(overlap) = (0,0); offset(above) = (0,1); offset(below) = (0,-1); offset(left) = (-1,0); offset(right) = (1,0); offset(upper-left) = (-1,1); offset(upper-right) = (1,1); offset(lower-left) = (-1,-1); offset(lower-right) = (1,-1).`

### 4.2.3 Example Analysis

CoT provides an interpretable window into model behaviour, hinting at how it might derive specific answers and offering opportunities to debug errors in reasoning paths (although fully describing the model's computations that support an answer remains an unresolved issue). A key challenge observed in performance is the failure to identify the correct reasoning chain. For example, in the 10-shot story shown in Figure 4.3, there is a long reasoning chain from B to Q: $B \rightarrow V(\text{use } 2) \rightarrow Z(\text{use } 3) \rightarrow X(\text{use } 4) \rightarrow C(\text{use } 9) \rightarrow N(\text{use } 6) \rightarrow Q(\text{use } 1)$. LLMs often struggle to find such reasoning chains. Table 4.4 are two representative failures of LLMs in identifying the chain.

## 4.3 Tree-of-Thoughts

In this part, we explain how ToT is utilized to address the primary challenge in this spatial reasoning problem: identifying the reasoning chain. For example, in the story presented in Table 4.3, our objective is to generate a linking chain such as "Linking chain: $Z \rightarrow M(\text{use } 3) \rightarrow C(\text{use } 1)$". We employ ToT (as illustrated in Figure 4.4) to generate these reasoning chains before applying CoT for reasoning about coordinates and relations.

In our ToT implementation (Algorithm 1), we strive to improve the reasoning chain-building process, enabling LLMs to produce more accurate reasoning chains by considering different pathways. This is beneficial because, during the search for relations involving an object, multiple clues may emerge, as shown in Figure 4.3. However, it is essential to follow a correct sequence to successfully navigate to the target object.

[1] A spelling mistake here. LLMs are typically effective at correcting such errors, but correcting the mistake may lead to slightly different results.

| |
|---|
| The story does not provide direct spatial information between agents Q and B. It would not be possible to determine the spatial relationship between Q and B given the information in the story. We would need additional information that links Q and B either directly or indirectly through other objects to determine their spatial relationship. |
| Suppose B is at (0,0). Link Q and B using the relations in the story.<br>Start from B. According to 5. B is to the left of W horizontally. That means B is to the left of W, so W is to the right of B. W= B+ offset(right) = (0,0)+(1,0)=(1,0). W is at (1,0)<br>Then search for W. According to 8. P is diagonally to the upper left of W. That means P is to the upper-left of W, so W is to the lower-right of P. P= W+ offset(lower-right) = (1,0)+(1,-1)= (2,-1). P is at (2,-1)<br>We don't have any direct relation to P, so we go back to W. We don't have any other direct relation to W either. So we go back to B. We don't have any other direct relation to B as well. So we cannot find a path from B to Q with the given story.<br>Answer: The relation of the agent Q to the agent B cannot be determined from the provided story. |

Table 4.4: Example of LLM failures in determining spatial relationships due to missing reasoning chains.

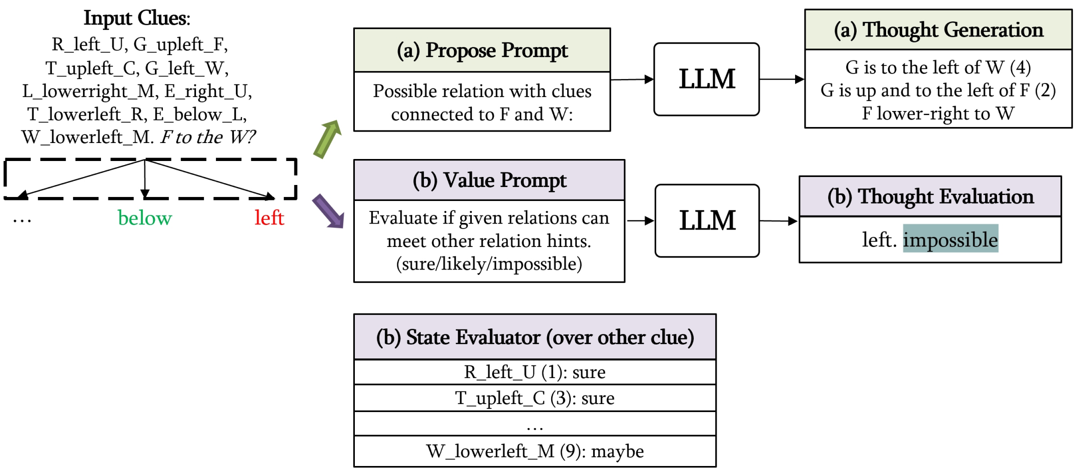



Figure 4.4: Illustration of our ToT method.

If the model mistakenly tracks the wrong sequence, it could get stuck in a dead end, leading to incorrect reasoning conclusions such as "The story does not provide direct spatial information."

### 4.3.1 Method

The algorithm begins by initializing the starting tree state $s_0 = [x, o_0, o_t, R]$. Then it proceeds to construct the linking tree step-by-step with $R$, linking from the start object $o_0$ to the target object $o_t$. $R$ represents all connections between objects from the story, in the form of `object1-object2`.

The construction process involves iterative steps, where at the $i$-th step ($1 \leq i \leq 10$), the LLM considers the tree state $s_i = [s_0, z_{1 \cdots i}]$ built up to that step, with $z_i$ corresponding to the thoughts generated in step $i$. Subsequently, the model generates thought $z_{i+1}$, searching for potential linking objects $o_{i+1}$ connected to the current object $o_i$ from the unused relations $R_i^{unused}$. Furthermore, the model evaluates the state to determine if the linking chain can proceed with the current object $o_{i+1}$ based on the remaining relations $R_{i+1}^{unused}$. Throughout this process, the model ensures it does not revisit previously visited objects and will conclude the loop if the chain cannot proceed further.

#### Thought generation $G(p_\theta, s, j)$

Given the current tree state $s_i$, we let the LLM propose thoughts $z_{i+1}$ using the `thought generation prompt` $g$ "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 approach, we enable the generation of a maximum of three possible candidates by the LLM $[z_{i+1}^{(1)}, \cdots, z_{i+1}^{(j)}] \sim p_\theta^{propose}(z_{i+1}^{(1,\cdots,j)}|s_i)$ for the next thought step $i+1$. $j$ denotes the number of potential candidates for each step, and this value may vary across $i$, but is capped at 3 to balance generation effectiveness. Generating too many candidates increases token usage and prolongs inference time. $z_{i+1}^{j}$ specifies the $i-th$ step chain $o_i- > o_{i+1}- >$, the target object $o_t$, and the remaining relations $R^{unused}$. If $o_{i+1} = o_t$, indicating that the chain has reached the target object, the `linking chain construction prompt` is activated to generate the final linking chain: "Given an input about spatial relations among objects, build the linking chain between the two queried objects."

#### State evaluation $Val(p_\theta, z)$

Our approach involves a classification methodology, using the designed `value prompt` $v$ "Evaluate whether the chain can reach the target (sure/likely/impossible). If the chain

**Algorithm 1:** Our ToT approach

**Require**: LLM $p_\theta$, relation set $R$, start, target object $o_0, o_t$, breadth $b$, number of thoughts to be generated $j$

1. $s_0 \leftarrow [x, R, o_0, o_t]$ *// Initialize the state with input, relation set, start, and target.*
2. $i = 1$ *// Initialize the iteration counter.*
3. **while** $S_i$ is not at $o_t$ **do** *// Repeat until the current state reaches the target object.*
   (a) $S_i^{'} \leftarrow \{s \cdot z \mid S_{i-1}, z \in G(p_\theta, s, j)\}$ *// Generate new states by appending $j$ thoughts from $G$ to the current states.*
   (b) $V_i \leftarrow \{\langle s \cdot z,\, Val(p_\theta, z)\rangle \mid s \cdot z \in S_i{}'\}$ *// Evaluate each new thought using an evaluation function $Val$.*
   (c) $S_i \leftarrow largest_b(S_i^{'})$ where $largest_b(\langle s_1 \cdot z_1, v_1\rangle, \langle s_2 \cdot z_2, v_2\rangle \ldots)$ returns the $b$ largest elements $> 0$ according to the values $v_j$ *// Select the top $b$ states with positive evaluation values.*
   (d) $i = i + 1$ *// Increment the iteration counter.*
   (e) **if** $\neg\exists\langle s \cdot z, v\rangle \in S_i$ where $v > 0$ **then break** *// Terminate if no valid states with positive evaluation value exist.*
4. **end while** *// End of the iterative state expansion.*
5. $l = G(p_\theta, \langle s_k \cdot z_k\rangle, 1)$, where $\langle s_k \cdot z_k, v_k\rangle \in V_{i-1} \wedge v_k = \arg\max_v \langle s_k \cdot z_k, v_k\rangle \in V_{i-1}$ *// Generate a final linking chain $l$ using the state with the highest evaluation value.*
6. **if** "Answer" in $l$ **then return** the linking chain $l$; **else return** *Null.* *// If the final generated linking chain includes the string "Answer", return it; Otherwise, return an empty string.*

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 LLM to sequentially examine all thought candidates $[z_{i+1}^{(1)}, \cdots, z_{i+1}^{(b)}]$. The classifications 'sure', 'likely', and 'impossible' are mapped to values of 20, 1, and 0.01, respectively. Additionally, if an answer is derived from the linking chain with thought generation, the value is set to 200. If no chain information is provided or the reasoning steps exceed the given relations, the value is 0. Based on the classification results, the LLM can then determine which states should be disregarded and which should be further explored with the search algorithm.

The values used to map classification results are heuristically designed to establish a clear separation between different levels of confidence during reasoning. These values do not belong to a fixed range but are chosen to ensure that the impact of each classification type is distinguishable, especially in multi-hop reasoning where the values are accumulated over steps. The final answer score (200) is set to be ten times that of sure, accounting for a maximum of 10-hop reasoning steps. This ensures that once a valid answer is generated, it immediately dominates the candidate pool and terminates the search.

**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. This approach resembles a beam search, where only a fixed number of high-scoring candidates are expanded at each step. As reasoning steps differ across various hops and instances, we refrain from imposing a fixed step limit for termination. Instead, we establish the stopping criterion as the linking chain reaching the target object $o_t$.

### Combine with CoT

Our ToT approach enables the construction of the linking chain *chain* from $o_0$ to $o_t$. We place the link chain prompt "`Linking chain: {chain}`" before "`Reasoning:{reasoning process}`" before "`Reasoning:`" in the CoT prompting format. The reasoning process mirrors the approach used in the CoT prompting method, utilizing $c^{map}$ and $c^{calcu}$ to get the spatial relation between these objects step by step based on their respective coordinates. Then concludes with an "`Answer:`" section that presents the final result.

## 4.3.2 Prompts

### Tree state initialization prompt

```
 Provided with a sequence of statements that define the spatial relationships among
various objects, your task is to detail the subsequent actions.  This includes
initiating the chain of connections, identifying the target object, and enumerating
all links between objects from the statements.
Input:  1.  Q is to the right of O and is on the same horizontal plane.  2.  Q is
slightly off center to the top left and M is slightly off center to the bottom right.
3.  X and E are next to each other with X on the top and E at the bottom.  4.  O is
sitting at the upper right position to E. 5.  W is on the right side and below M. What
is the relation of the agent W to the agent E?
Possible next steps:
chain:  W ->, target:  E, unused:  1.  Q-O, 2.  Q-M, 3.  X-E, 4.  O-E, 5.  W-M.
...
Input:  {input}
Possible next steps:
```

### 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.
Input:  chain:  G ->, target:  Q, unused:  1.  C-R, 2.  L-Q, 3.  C-J, 4.  J-E, 5.
T-A, 6.  G-N, 7.  G-A, 8.  L-Y, 9.  R-Q, 10.  Y-T.
```

```
Possible next steps:
The last object within the chain is G, and the unused relations 6. G-N and 7. G-A
include G. relation chain: G -> N (use 6) ->, target: Q, unused: 1. C-R, 2. L-Q,
3. C-J, 4. J-E, 5. T-A, 7. G-A, 8. L-Y, 9. R-Q, 10. Y-T.
chain: G -> A (use 7) ->, target: Q, unused: 1. C-R, 2. L-Q, 3. C-J, 4. J-E, 5.
T-A, 6. G-N, 8. L-Y, 9. R-Q, 10. Y-T.
...
Input: {input}
Possible next steps:
```

### State evaluation 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'.
chain: G -> N (use 6) ->, target: Q, unused: 1. C-R, 4. J-E, 5. T-A, 8. L-Y. The
current object is N, and there are no unused relations involving N.
impossible
chain: F ->, target: X, unused: 1. Y-K, 2. X-Y, 3. I-Q, 4. A-Q, 5. N-W, 6.
N-A, 7. O-F, 8. O-W. The current object is F, there is an unused relation involving F
(7. O-F).
likely
chain: L -> Q (use 2) ->, target: Q, unused: 1. C-R, 3. C-J, 4. J-E, 7. G-A, 8.
L-Y, 9. R-Q. The chain already reaches the target object Q.
sure
{input}
```

### Linking chain construction prompt

```
 Given an input about spatial relations among objects, build the linking chain between
the two queried objects.
Input:
```

```
1. H is above S with a small gap between them. 2. S is positioned below I. 3. P is
on the top side to I. What is the relation of the agent S to the agent P?
Steps:
chain: S ->, target: P, unused: 1. H-S, 2. S-I, 3. P-I.
chain: S -> I (use 2) ->, target: P, unused: 1. H-S, 3. P-I.
chain: I -> P (use 3) ->, target: P, unused: 1. H-S.
Answer: S -> I (use 2) -> P (use 3)
...
Input:
{input}
```

## 4.4 Experimental Results

**Model Settings.** We use the Azure OpenAI Service for ChatGPT (gpt-35-turbo), GPT-3 Davinci (text-davinci-003), GPT-4 (version: turbo-2024-04-09)[1], and GPT-4o (version: 2024-08-06) API access. To yield more concentrated and deterministic results, we set the temperature to 0 for CoT experiments and 0.7 for ToT for generating varied thought proposals. The remaining parameters were left at the standard configurations for these models.

For Claude-Instant and Claude-3.5-haiku, we accessed them via the Claude API, using the same temperature settings as GPT, with the stop sequence configured to "\n\n Story:" and all other parameters set to their default values.

For Llama-2-7B[2] and Llama-3-8B[3], we accessed these models via the Hugging Face pipeline. The temperature parameter for these models requires a strictly positive float; hence, we set it to 0.001, which approximates determinism while complying with the parameter constraints. Additionally, we set the top_p [154] parameter to 0.9 to enhance the diversity and reduce the repetitiveness of the responses

### 4.4.1 Evaluation on StepGame

**Influence of Scale of Test Examples.**

[1]https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo
[2]https://huggingface.co/meta-llama/Llama-2-7b-hf
[3]https://huggingface.co/meta-llama/Meta-Llama-3-8B

It is common practice in the studies [28, 31] 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 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.

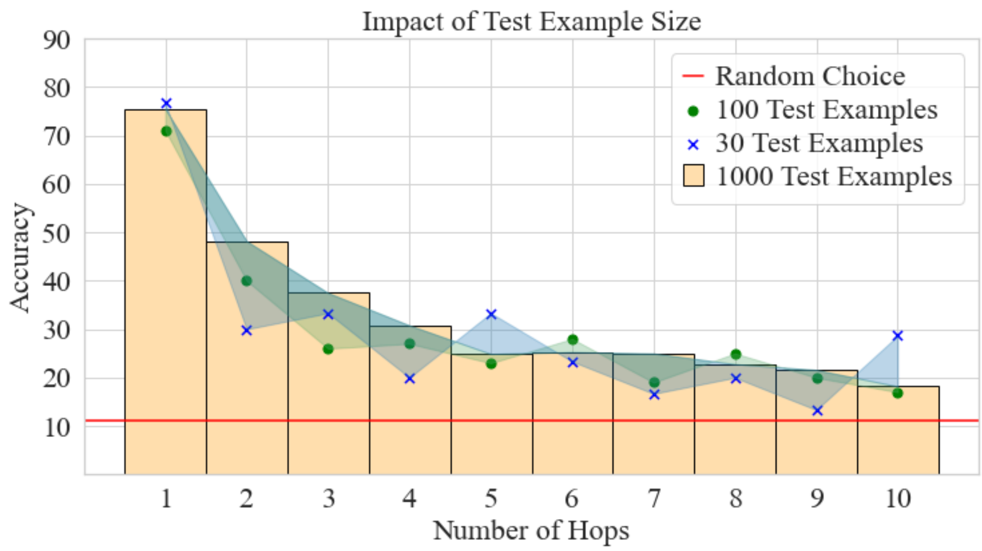



Figure 4.5: Comparison of Turbo's 10-shot learning performance across various test subsets in the refined StepGame, with test set sizes of 30, 100, and 1000 examples. The experiments were conducted using the *clean 10shot* prompting setting.

The results are presented in Figure 4.5. Upon evaluation of the expanded test set comprising 1000 examples, the model shows a more uniform decrement in performance as the number of reasoning hops $k$ increases from 1 to 10. This trend indicates the increased complexity and difficulty in maintaining high accuracy as the number of hops increases.

On the other hand, 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.** 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 the 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.

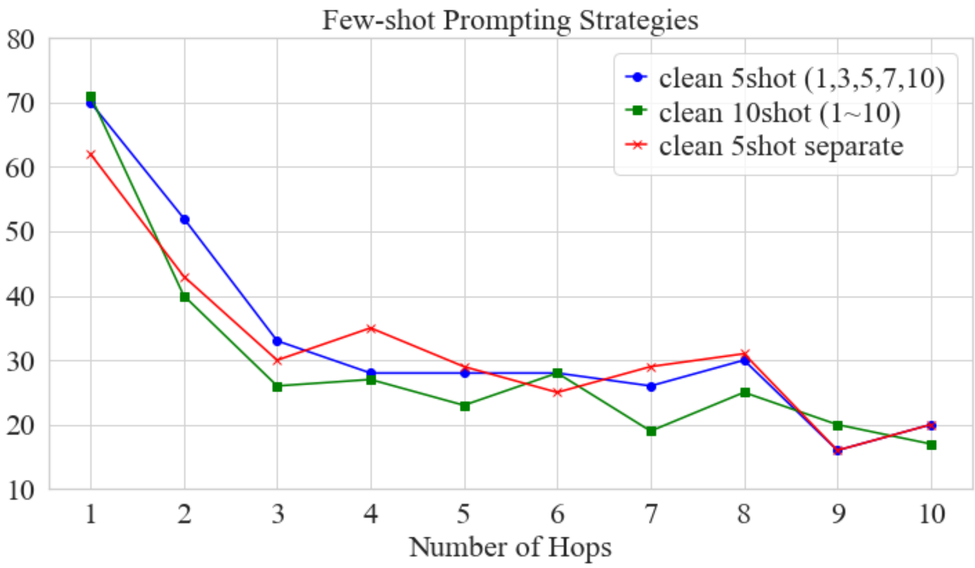



Figure 4.6: Comparison of Turbo's performance on a test set of 100 across three different prompting scenarios: *clean 5-shot(1,3,5,7,10)*, *clean 10-shot*, and *clean 5-shot separate*.

According to Figure 4.6, while the results vary slightly across different prompting strategies, the overall performance trends remain largely consistent. 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.**

Figure 4.7 illustrates the performance comparison between the Turbo and Davinci models. According to this figure, the Davinci model consistently outperforms the Turbo model across different task complexities (measured by the 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.

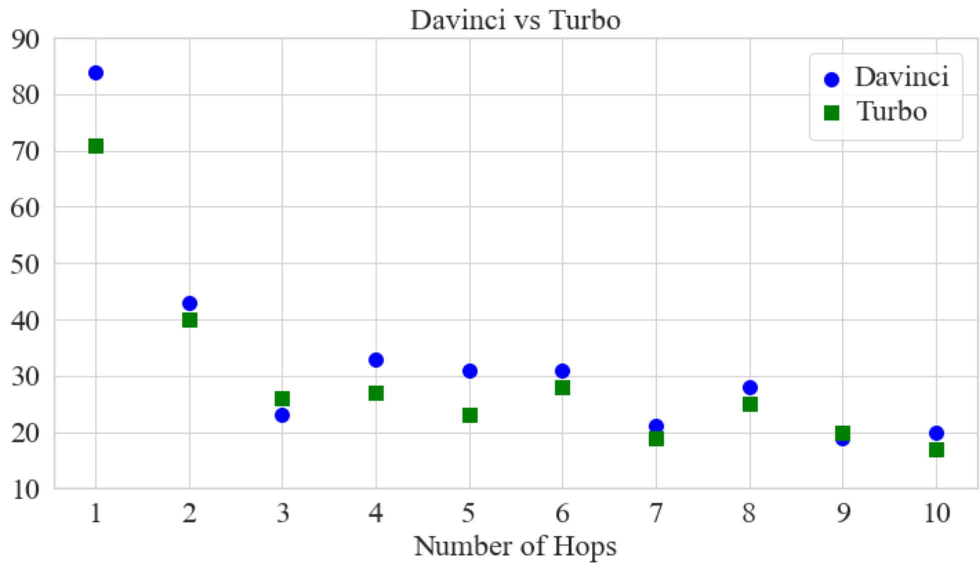



Figure 4.7: Comparison of the performances of different LLMs on a test set of 100 using the *clean 10-shot* prompting approach.

As indicated in a recent study [155], Turbo demonstrates comparable performance to Davinci across many tasks. However, it falls short in the machine reading comprehension, part-of-speech, and semantic parsing 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.

A detailed comparison of additional LLMs, including various versions of Claude, Llama, and GPT, is presented in Table 4.5.

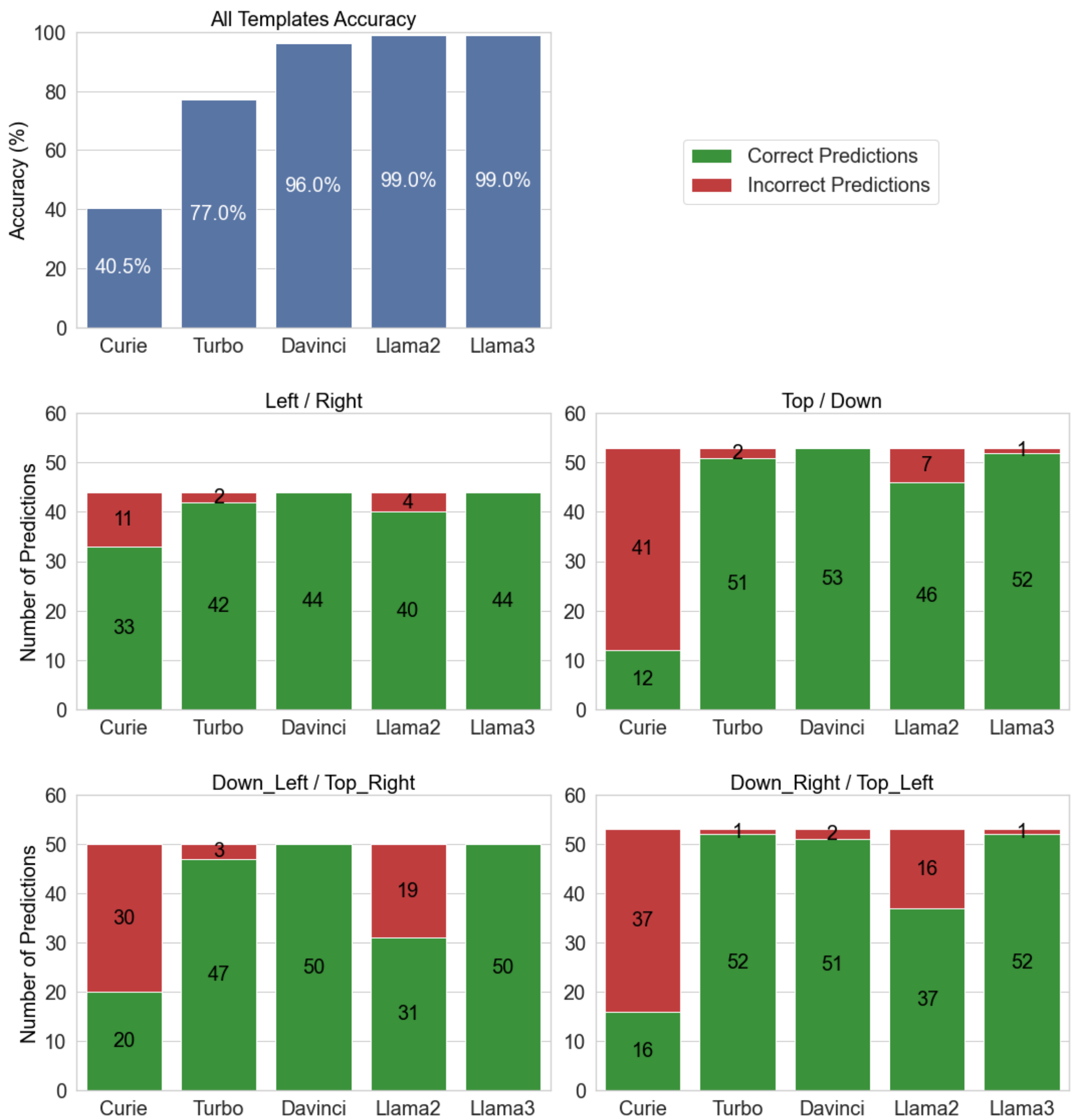



Figure 4.8: The semantic parsing performance of different LLMs.

### 4.4.2 LLM for Semantic Parsing

StepGame is well-suited for assessing the spatial semantic parsing capabilities of LLMs, offering a variety of rich sentence templates that test the models' ability to extract relationships. In contrast, bAbI employs fixed templates that are quite straightforward, varying only in the names of the objects for each relation. This distinction makes StepGame a more comprehensive tool for evaluating the nuanced understanding of LLMs in semantic parsing tasks.

Figure 4.8 displays the semantic parsing performance of five different LLMs: Curie, Turbo, Davinci, Llama-2-7B, and Llama-3-8B. The results are based on the models' ability to parse 200 sentence templates from StepGame into symbolic representations, with a focus on accuracy and error rates. These templates are categorized into four relational groups: 44 templates for left or right relations, 53 for top or down relations, 50 for down_left or top_right relations, and 53 for down_right or top_left relations.

Among the LLMs, Curie recorded the lowest performance in semantic parsing, achieving only 40.5% accuracy. Llama-2-7B performed moderately with a 77.0% accuracy rate, while Turbo achieved a commendable 96% accuracy. Llama-3-8B and Davinci both excelled, each attaining 99% accuracy by correctly interpreting 198 out of the 200 templates. Curie, with the highest error rate across various relational categories, demonstrated significant performance gaps compared to its counterparts. Meanwhile, Turbo's results were closely aligned with the top performers, Llama-3-8B and Davinci. This detailed evaluation highlights the varying capabilities of each LLM in processing and understanding different types of relational data, crucial for tasks involving semantic parsing.

### 4.4.3 Resolution for the Benchmark

The results of our resolution for StepGame (sentence-to-relation mapping + LR-based reasoning) are displayed in the '*Map+LR*' row of Table 4.5. The numbers in the table indicate precision 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.

| | | k=1 | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 | k=9 | k=10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Map+LR | | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** |
| Curie+LR | | 46 | 43 | 42 | 59 | 67 | 67 | 57 | 56 | 58 | 61 |
| Davinci+LR | | **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 |
| Instant | IO | 51 | 33 | 19 | 17 | 7 | 9 | 12 | 10 | 10 | 6 |
| | CoT | / | 50 | 35 | 30 | 28 | 34 | 28 | 31 | 25 | 25 |
| | ToT | / | / | 30 | 35 | 35 | 35 | 30 | 35 | 30 | 25 |
| Claude-3.5 | IO | 76 | 48 | 45 | 45 | 32 | 32 | 25 | 34 | 16 | 22 |
| | CoT | / | 66 | 69 | 49 | 44 | 39 | 36 | 34 | 27 | 29 |
| | ToT | / | / | 59 | 56 | 50 | 48 | 35 | 39 | 29 | 33 |
| Llama-2 | IO | 37 | 18 | 10 | 14 | 18 | 13 | 10 | 14 | 10 | 7 |
| | CoT | / | 23 | 15 | 10 | 17 | 16 | 17 | 13 | 12 | 14 |
| | ToT | / | / | 8 | 9 | 11 | 9 | 10 | 16 | 13 | 13 |
| Llama-3 | IO | 63 | 32 | 18 | 8 | 21 | 14 | 24 | 17 | 8 | 3 |
| | CoT | / | 39 | 42 | 31 | 24 | 24 | 29 | 31 | 25 | 22 |
| | ToT | / | / | 50 | 43 | 33 | 33 | 34 | 31 | 30 | 28 |
| Turbo | IO | 65 | 50 | 27 | 29 | 31 | 27 | 25 | 29 | 12 | 17 |
| | CoT | / | 34 | 40 | 36 | 28 | 28 | 26 | 31 | 25 | 24 |
| | ToT | / | / | 46 | 42 | 31 | 40 | 34 | 30 | 37 | 33 |
| Davinci | IO | 77 | 42 | 25 | 29 | 31 | 27 | 24 | 24 | 16 | 22 |
| | CoT | / | 46 | 49 | 46 | 46 | 45 | 43 | 49 | 41 | 27 |
| | ToT | / | / | 65 | 50 | 45 | 60 | 50 | 45 | 55 | 50 |
| GPT-4 | IO | **97** | 69 | 48 | 48 | 42 | 41 | 27 | 40 | 30 | 29 |
| | CoT | / | 91 | 93 | 88 | 87 | 80 | 88 | 77 | 76 | 72 |
| | ToT | / | / | 92 | 91 | 88 | 93 | 88 | **93** | **89** | **89** |
| GPT-4o | IO | **97** | 77 | 58 | 48 | 45 | 37 | 36 | 44 | 30 | 29 |
| | CoT | / | **97** | **95** | 93 | **90** | **94** | **93** | 85 | 87 | 81 |
| | ToT | / | / | 93 | **94** | 87 | **94** | 85 | 84 | 83 | 84 |

Table 4.5: Performance comparison of LLMs on StepGame using different methods.

### 4.4.4 LLM + Logical Reasoner

The previous SOTA results [31] (using GPT-3 for semantic parsing and ASP for reasoning) are presented in the "*SOTA*" row of Table 4.5. 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 on the corrected dataset, with the results displayed in the "*Curie+LR*" and "*Davinci+LR*" rows. Among the 1000 test examples (100 for each k), only 2 errors were encountered with the Davinci model. These issues were linked to semantic parsing with Davinci: the sentence "`If E is the center of a clock face, H is located between 2 and 3.`" was parsed incorrectly as `right("H", "E")`.

### 4.4.5 CoT and ToT

Table 4.5 and Figure 4.9 present the comparison of the IO, CoT and ToT methods across various LLMs. All results for the IO and CoT methods, as well as for the models Claude-3.5 (Haiku), Llama, and GPT-4, are based on evaluations of the first 100 test instances for each $k$ setting. For the Claude-Instant and Davinci (text-davinci-003) models, ToT results were derived from the first 20 instances per $k$ setting, with accuracy data sourced from our AAAI paper. Note that text-davinci-003 on Microsoft Azure was retired in January 2024[1], and Claude-Instant was deprecated in September 2024[2].

According to Figure 4.9, the GPT-4 and GPT-4o models exhibit superior performance across most settings. With basic IO prompting, both models start with 97% accuracy for $k = 1$, but their accuracy declines to 29% by $k = 10$, indicating that even advanced GPT models struggle to maintain accuracy as task complexity rises. CoT and ToT prompting show significant improvements for multi-hop reasoning tasks, consistently outperforming IO. With the CoT method, the GPT-4 model achieves over 70% accuracy at higher hops (from $k = 4$ to $k = 10$), even as task complexity rises. GPT-4o generally performs better than GPT-4 across all hops with CoT prompting, achieving approximately 80% accuracy at 10-hop. However, ToT provides only a slight benefit for GPT-4o at 10-hop tasks and, for some lower-hop tasks, performs worse than

[1] https://platform.openai.com/docs/deprecations

[2] https://docs.anthropic.com/en/docs/resources/model-deprecations

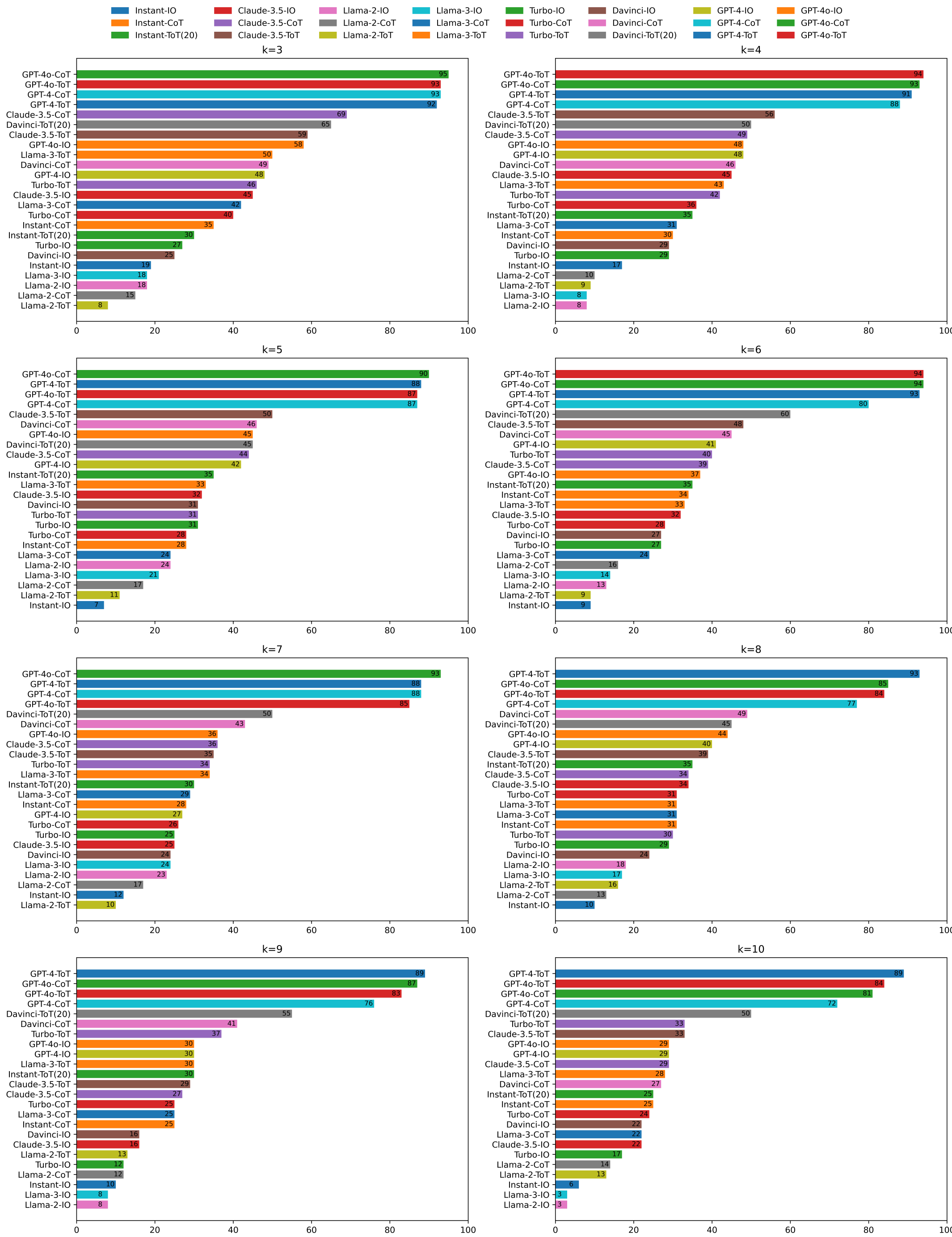



Figure 4.9: Comparison of the performance of various models and methods across multiple hops (3-10).

CoT. In contrast, with ToT prompting, GPT-4 maintains higher accuracy than CoT for higher hops, stabilizing around 90% as task complexity increases. For 7-shot reasoning, GPT-4 has 12 failures and GPT-4o 15, with both models failing on 3 of the same test examples. For 8-shot, GPT-4 has 7 failures and GPT-4o 16, 2 overlaps. For 9-shot, GPT-4 has 11 failures and GPT-4o 17, with 3 overlaps. For 10-shot, GPT-4 has 11 failures and GPT-4o has 16, with 3 overlaps.

Figures 4.10, 4.11, and 4.12 illustrate three overlapping error cases for GPT-4 and GPT-4o in 9-hop reasoning tasks. A common error shared by both models occurs in the final step, involving an inversion in the interpretation of coordinates into relative positions. For instance, in Figure 4.10, both models incorrectly concluded that "`considering M(0,0) and X(-1,-1), M is to the lower-right of X`", whereas the correct answer is '`upper-right`'. Similarly, in Figure 4.11, GPT-4's final reasoning step stated, "`considering W(0,0) and G(-1,-1), W is to the lower-right of G`", while the correct conclusion should have been '`upper-right`'. In Figure 4.12, GPT-4o accurately inferred K's position relative to M but failed to reverse the relationship to correctly reflect M's position relative to K, as the question required.

Additionally, case analysis reveals that GPT-4 demonstrates better in-context following, with its reasoning process closely aligning with the examples provided in the prompt. In contrast, GPT-4o does not strictly follow the reasoning steps shown in the examples. This can be beneficial when the linking chain derived from ToT is flawed, as demonstrated in Figure 4.10, where GPT-4o re-examines all relationships in the narrative and constructs a new linking chain during its reasoning process. However, this approach risks failing to construct the correct chain and may omit critical reasoning components. For example, in Figure 4.10, GPT-4o disregards the provided chain and instead attempts to "`establish the linking chain from T to G`", despite the question specifically asking for the relationship between W and G. Moreover, it neglects to compute the coordinates for all relevant objects, ultimately leading to an incorrect conclusion.

In this spatial reasoning task, the prompts are formatted in a few-shot style, requiring in-context learning. This means that the models need to learn to perform a new task based on a small set of examples provided in the prompt at inference time. Additionally, the task involves multi-step reasoning, where breaking down complex problems into intermediate steps can facilitate problem-solving. As highlighted in [156], larger

**Story:** 1. The object Q is lower and slightly to the right of the object O.
2. M is positioned above P and to the right.
3. X is positioned above and to the left of R.
4. V is on the same vertical plane directly below I.
5. U and I are parallel, and U is on top of I.
6. V and R are in a horizontal line with V on the left.
7. X is at O's 9 o'clock.
8. M is on the same horizontal plane directly right to U.
9. Q is positioned right to W.
**Question:** What is the relation of the agent M to the agent X?
**Chain:** M -> U (use 8) -> I (use 5) -> V (use 4) -> R (use 6) -> X (use 3)

**Error Reason:**
**Inversion in the interpretation of the coordinates to relative positions.**

Both models correctly followed the chain of reasoning to determine the positions of all objects but made an error in interpreting the final relationship between *M* and X.

**Ground Truth:** upper-right

**GPT-4:** Let's suppose M is at (0,0). We can analyze the relation of M to X by following the linking chain and considering the relations provided in the story step by step.
Start with M. According to 8, "M is on the same horizontal plane directly right to U." This means U is to the left of M. U= M+ offset(left) = (0,0)+(-1,0)=(-1,0). Therefore, U is at (-1,0).
Then come to U. According to 5, "U and I are parallel, and U is on top of I." This means I is below U. I= U+ offset(below) = (-1,0)+(0,-1)=(-1,-1). Therefore, I is at (-1,-1).
Then come to I. According to 4, "V is on the same vertical plane directly below I." This means V is below I. V= I+ offset(below) = (-1,-1)+(0,-1)=(-1,-2). Therefore, V is at (-1,-2).
Then come to V. According to 6, "V and R are in a horizontal line with V on the left." This means R is to the right of V. R= V+ offset(right) = (-1,-2)+(1,0)= (0,-2). Therefore, R is at (0,-2).
Then come to R. According to 3, "X is positioned above and to the left of R." This means X is above and to the left of R. X= R+ offset(upper-left) = (0,-2)+(-1,1)=(-1,-1). Therefore, X is at (-1,-1).
We've reached X. So, considering M(0,0) and X(-1,-1), M is to the lower-right of X.
Answer: lower-right

**GPT-4o:** To determine the relation of the agent M to the agent X, we need to establish a linking chain and analyze the spatial relations step by step.
Let's start by identifying the linking chain:
1. M -> U (use 8)
2. U -> I (use 5)
3. I -> V (use 4)
4. V -> R (use 6)
5. R -> X (use 3)
Now, let's analyze the relations:
1. Start with M. According to 8, "M is on the same horizontal plane directly right to U." This means U is to the left of M. If we assume M is at (0,0), then U = M + offset(left) = (0,0) + (-1,0) = (-1,0).
2. Then come to U. According to 5, "U and I are parallel, and U is on top of I." This means I is below U. I = U + offset(below) = (-1,0) + (0,-1) = (-1,-1).
3. Then come to I. According to 4, "V is on the same vertical plane directly below I." This means V is below I. V = I + offset(below) = (-1,-1) + (0,-1) = (-1,-2).
4. Then come to V. According to 6, "V and R are in a horizontal line with V on the left." This means R is to the right of V. R = V + offset(right) = (-1,-2) + (1,0) = (0,-2).
5. Then come to R. According to 3, "X is positioned above and to the left of R." This means X is to the upper-left of R. X = R + offset(upper-left) = (0,-2) + (-1,1) = (-1,-1).
We've reached X. So, considering M(0,0) and X(-1,-1), M is to the lower-right of X.
Answer: lower-right

Figure 4.10: Comparison of the reasoning processes of GPT-4 and GPT-4o on one of the three overlapping failure cases (test index: 5), utilizing the same reasoning chain derived from ToT methods with GPT-4.

**Story:** 1. If Z is the center of a clock face, O is located between 7 and 8.
2. H and S are side by side with S at the bottom and H on the top.
3. B is over there and A is directly below it.
4. B is sitting at the 9:00 position of G.
5. W is at S's 3 o'clock.
6. O and A are vertical and O is above A.
7. H is on top of Z.
8. Q and W are parallel, and Q is over W.
9. The object labeled G is positioned to the right of the object labeled J.
**Question:** What is the relation of the agent W to the agent G?
**chain:** W -> S (use 5) -> H (use 2) -> Z (use 7) -> O (use 1) -> A (use 6) -> B (use 3) -> G (use 4)

**Ground Truth:** upper-right

**GPT-4 Error Reason:**

**Wrong interpretation of the coordinates to relative positions.**

**GPT-4o Error Reason: Fail to establish the correct linking chain**

Instead of using the provided chain, it attempts to "establish the linking chain from T to G", even though the question specifically asks for the relationship between W and G.

**GPT-4:** Let's suppose W is at (0,0). We can analyze the relation of W to G by following the linking chain and considering the relations provided in the story step by step.
Start with W. According to 5, "W is at S's 3 o'clock." Considering the clock-wise information, 3 o'clock denotes right. This means S is to the left of W. S= W+ offset(left) = (0,0)+(-1,0)=(-1,0). Therefore, S is at (-1,0).
Then come to S. According to 2, "H and S are side by side with S at the bottom and H on the top." This means H is above S. H= S+ offset(above) = (-1,0)+(0,1)=(-1,1). Therefore, H is at (-1,1).
Then come to H. According to 7, "H is on top of Z." This means Z is below H. Z= H+ offset(below) = (-1,1)+(0,-1)=(-1,0). Therefore, Z is at (-1,0).
Then come to Z. According to 1, "If Z is the center of a clock face, O is located between 7 and 8." Considering the clock-wise information, between 7 and 8 denotes lower-left. This means O is to the lower-left of Z. O= Z+ offset(lower-left) = (-1,0)+(-1,-1)=(-2,-1). Therefore, O is at (-2,-1).
Then come to O. According to 6, "O and A are vertical and O is above A." This means A is below O. A= O+ offset(below) = (-2,-1)+(0,-1)=(-2,-2). Therefore, A is at (-2,-2).
Then come to A. According to 3, "B is over there and A is directly below it." This means B is above A. B= A+ offset(above) = (-2,-2)+(0,1)=(-2,-1). Therefore, B is at (-2,-1).
Then come to B. According to 4, "B is sitting at the 9:00 position of G." Considering the clock-wise information, 9:00 denotes left. This means G is to the right of B. G= B+ offset(right) = (-2,-1)+(1,0)=(-1,-1). Therefore, G is at (-1,-1).
We've reached G. So, considering W(0,0) and G(-1,-1), W is to the lower-right of G.
Answer: lower-right

**GPT-4o:** To determine the relation of the agent T to the agent G, we need to establish a linking chain and analyze the spatial relations step by step.
Let's start by identifying the relevant relations from the story:
1. From sentence 5, "A is positioned in the lower right corner of G." This means A is to the lower-right of G.
2. From sentence 1, "The object L is positioned below and to the left of the object A." This means L is to the lower-left of A.
3. From sentence 4, "N is above and to the right of K." This means N is to the upper-right of K.
4. From sentence 2, "K is positioned in the lower left corner of R." This means K is to the lower-left of R.
5. From sentence 9, "F is over R." This means F is above R.
6. From sentence 6, "F is on the left side of and below T." This means T is to the upper-right of F.
Now, let's establish the linking chain from T to G:
- Start with T. According to sentence 6, "F is on the left side of and below T." This means T is to the upper-right of F. So, F is at T + offset(lower-left) = T + (-1,-1).
- According to sentence 9, "F is over R." This means R is below F. So, R is at F + offset(below) = F + (0,-1).
- According to sentence 2, "K is positioned in the lower left corner of R." This means K is to the lower-left of R. So, K is at R + offset(lower-left) = R + (-1,-1).
- According to sentence 4, "N is above and to the right of K." This means N is to the upper-right of K. So, N is at K + offset(upper-right) = K + (1,1).
- According to sentence 8, "N is at Z's 9 o'clock." This means Z is to the left of N. So, Z is at N + offset(left) = N + (-1,0).
- According to sentence 7, "Z and H are horizontal and Z is to the left of H." This means H is to the right of Z. So, H is at Z + offset(right) = Z + (1,0).
- According to sentence 3, "H is at G's 6 o'clock." This means G is above H. So, G is at H + offset(above) = H + (0,1).
Now, let's determine the relation of T to G:
- We have T at an initial position, and G is reached through the chain. By following the offsets, we can determine the relative position of T to G.
Considering the offsets and the linking chain, T is to the lower-right of G.
Answer: lower-right

Figure 4.11: Comparison of the reasoning processes of GPT-4 and GPT-4o on one of the three overlapping failure cases (test index: 75), utilizing the same reasoning chain derived from ToT methods with GPT-4.

**Story:** 1. M and E are parallel, and E is below M.
2. W is positioned below F and to the right.
3. H and D are both there with the object H above the object D.
4. H is above K and to the right of K.
5. A is sitting at the lower position to W.
6. X is diagonally to the bottom left of K.
7. E is diagonally below Y to the left at a 45 degree angle.
8. A presents upper right to D.
9. F presents right to Y.
**Question:** What is the relation of the agent M to the agent K?
**chain :** M -> H (use 4) -> K (use 4)

**Ground Truth:** upper-left

**GPT-4 Error Reason:**
- **Wrong linking chain**
The linking chain was generated from ToT with GPT-4, where M -> H (use 4) is incorrect for "4. H is above K and to the right of K". This error propagated into GPT-4's reasoning, as it uncritically followed the faulty chain without re-evaluating its validity.
- **Incorrect Offset Assignment**
K is below and to the left of H. H= M+ offset(above) = (0,0)+(0,1)

**GPT-4o Error Reason: Wrong interpretation of the question's framing.**

GPT-4o correctly infered *K*'s position relative to *M* but failed to reverse the relationship to reflect *M*'s position relative to *K*, as the question required.

**GPT-4:** Let's suppose M is at (0,0). We can analyze the relation of M to K by following the linking chain and considering the relations provided in the story step by step.
Start with M. According to 4, "H is above K and to the right of K." This means K is below and to the left of H. H= M+ offset(above) = (0,0)+(0,1)=(0,1). Therefore, H is at (0,1).
Then come to H. Since H is above K and to the right of K, K= H+ offset(lower-left) = (0,1)+(-1,-1)=(-1,0). Therefore, K is at (-1,0).
We've reached K. So, considering M(0,0) and K(-1,0), M is to the right of K.
Answer: right

**GPT-4o:** To determine the relation of the agent M to the agent K, we need to establish a linking chain and analyze the spatial relations step by step.
Let's start by identifying the relevant relations from the story:
1. M and E are parallel, and E is below M.
2. W is positioned below F and to the right.
3. H and D are both there with the object H above the object D.
4. H is above K and to the right of K.
5. A is sitting at the lower position to W.
6. X is diagonally to the bottom left of K.
7. E is diagonally below Y to the left at a 45-degree angle.
8. A presents upper right to D.
9. F presents right to Y.
To find the relation between M and K, we need to connect them through a series of spatial relations. Let's establish a possible linking chain:
1. Start with M. According to 1, "E is below M." This means M is above E. Let's assume M is at (0,0). Therefore, E is at (0,-1).
2. According to 7, "E is diagonally below Y to the left at a 45-degree angle." This means Y is to the upper-right of E. Y = E + offset(upper-right) = (0,-1) + (1,1) = (1,0).
3. According to 9, "F presents right to Y." This means F is to the right of Y. F = Y + offset(right) = (1,0) + (1,0) = (2,0).
4. According to 2, "W is positioned below F and to the right." This means W is to the lower-right of F. W = F + offset(lower-right) = (2,0) + (1,-1) = (3,-1).
5. According to 5, "A is sitting at the lower position to W." This means A is below W. A = W + offset(below) = (3,-1) + (0,-1) = (3,-2).
6. According to 8, "A presents upper right to D." This means D is to the lower-left of A. D = A + offset(lower-left) = (3,-2) + (-1,-1) = (2,-3).
7. According to 3, "H is above D." This means H is above D. H = D + offset(above) = (2,-3) + (0,1) = (2,-2).
8. According to 4, "H is above K and to the right of K." This means K is to the lower-left of H. K = H + offset(lower-left) = (2,-2) + (-1,-1) = (1,-3).
Now, we have M at (0,0) and K at (1,-3). The relation of M to K is determined by the relative position of K to M. Since K is to the right and below M, the relation is lower-right.
Answer: lower-right

Figure 4.12: Comparison of the reasoning processes of GPT-4 and GPT-4o on one of the three overlapping failure cases (test index: 96), utilizing the same reasoning chain derived from ToT methods with GPT-4.

LLMs not only have greater model sizes but also demonstrate superior language understanding and generation abilities. More importantly, they exhibit emergent abilities that are absent in smaller models. These emergent abilities include in-context learning and the capacity for multi-step reasoning.

Claude-3.5-haiku, the fastest variant of the Claude 3.5 models, demonstrates moderate performance. Even when prompted with few-shot examples, it tends to adhere to its own generation style. For instance, under IO prompting, the example output format is a simple directional prediction formatted as '`Answer:`'. However, Claude often begins with a CoT-style response, such as '`Let me solve this systematically`', before analyzing the spatial relationships in the story to derive the answer. When using our few-shot CoT and ToT prompting, the model still starts its generation similarly to IO, but it successfully incorporates the coordinate information into its reasoning process. With ToT prompting, Claude also begins with explanations like '`I'll help solve this problem step by step`', but it further integrates coordinate details and the linking chain into its reasoning process.

For Claude-Instant, under IO prompting, many incorrect predictions result from the inability to identify a relationship between the starting object and the target object. For example, one failed generation states, "Based on the information given, there is no unique relation between agents S and J." While CoT and ToT significantly alleviate this issue, the model frequently makes errors in coordinate mapping and calculation.

The smallest Llama-2-7B model underperforms compared to other models in overall performance. Although the Llama-3 model exhibits moderate performance in lower-hop tasks, it performs poorly in larger-hop reasoning tasks. For the smallest Llama-2-7B model, our CoT method yields performance improvements across most hops. Our ToT methods improve performance on 8-hop, 9-hop, and 10-hop reasoning tasks; the gains are less significant compared to those achieved with other models. In lower-hop scenarios, the ToT approach actually reduces performance. For instance, in the 3-hop task, accuracy drops from 18% with IO prompting to 8% with ToT. This could be attributed to the long length of our prompts, requiring a nuanced understanding of coordinates and relations. The Llama-2 model has challenges in producing extended, coherent text for the complex task.

In our analysis of errors using the CoT and ToT_CoT reasoning methods, we found that inaccuracies in semantic parsing significantly contribute to failures in reasoning. For example, consider the following error made by Llama-2: "*The objects S and Z are over there. The object S is lower and slightly to the left of the object Z.*" This sentence was interpreted as "`Z is to the right of S`" leading to the conclusion "`Z= S + offset(right) = (-1,0) + (1,0) = (0,0)`" which determines the position of Z. The correct parsing should indicate "`Z is to the upper-right of S.`" rather than "`Z is to the right of S.`" Additionally, errors frequently occur when mapping $(x, y)$ coordinates to spatial relations. For example, "`considering X(0,0) and Y(1,-1), X is to the lower-right of Y.`" X should actually be positioned upper-left of Y. Similarly, in the case of "`considering X(0,0) and Y(-3,0), X is to the lower-right of Y.`" The correct parsed relation should indicate "`X is right of Y.`" Less capable models, such as Llama-2-7B and Turbo, are more prone to produce the aforementioned semantic parsing and coordinate mapping errors. These errors can accumulate during the CoT and ToT thought generation processes, leading to higher error rates.

In this chapter, we explored methods to enhance the spatial reasoning capabilities of LLMs. The effective resolution of the StepGame benchmark prompts a need for more challenging versions. In the next chapter, we will present our new benchmark for spatial reasoning.

# Chapter 5

## New Spatial Reasoning Benchmark - RoomSpace

In light of the issues and limitations of existing benchmarks discussed in chapter 3, we introduce RoomSpace, a new benchmark specifically designed for evaluating the spatial reasoning capabilities of LLMs. RoomSpace is notable for several key features that enhance its utility for assessing these models. These features include:

1. Realistic 3D Room Environments. The benchmark features richly detailed 3D simulations of indoor environments that closely resemble real-world settings, providing a higher degree of ecological validity. As highlighted by [157], many existing benchmarks lack ecological validity; the questions posed often do not reflect the types of queries end users naturally ask in realistic scenarios. This mismatch can lead to discrepancies between benchmark performance and real-world user experience. In RoomSpace, the environments and questions are designed to more closely reflect how spatial reasoning is applied in everyday human contexts, thereby enhancing the benchmark's ecological validity.

2. Multi-Modality Data Integration. We incorporate both text and images for each example, making our benchmark well-suited for MLLM-based evaluations, where spatial reasoning tasks will increasingly rely on both natural language and visual inputs.

3. Flexible and adaptable dataset-building framework. Unlike traditional benchmarks that rely on static QA datasets, we utilize a dynamic framework that can be tailored to accommodate different spatial representations and reasoning challenges. This adaptability is valuable for examining the performance of LLMs across various dimensions of spatial reasoning.

4. Logical reasoning for gold label generation. To address the possibility of multiple valid answers in spatial reasoning tasks, we implement a logical reasoning tool for generating gold labels. This ensures that the benchmark can identify all valid solutions to spatial reasoning problems with multiple correct answers, thereby providing accurate assessments of LLMs' performance.

The overall design and workflow of this benchmark are depicted in Figure 5.1, which outlines four key steps: (1) Construction of 3D Rooms, described in Section 5.1; (2) Establishment of Spatial Representation, detailed in Section 5.2; (3) Creation of Spatial Reasoning Stories, explained in Section 5.3; and (4) Generation of Gold Labels using Logical Reasoning Tools, discussed in Section 5.4. This adaptable framework facilitates a comprehensive evaluation of LLMs' spatial reasoning capabilities. Our preliminary assessment of LLMs on RoomSpace is presented in Section 5.5.

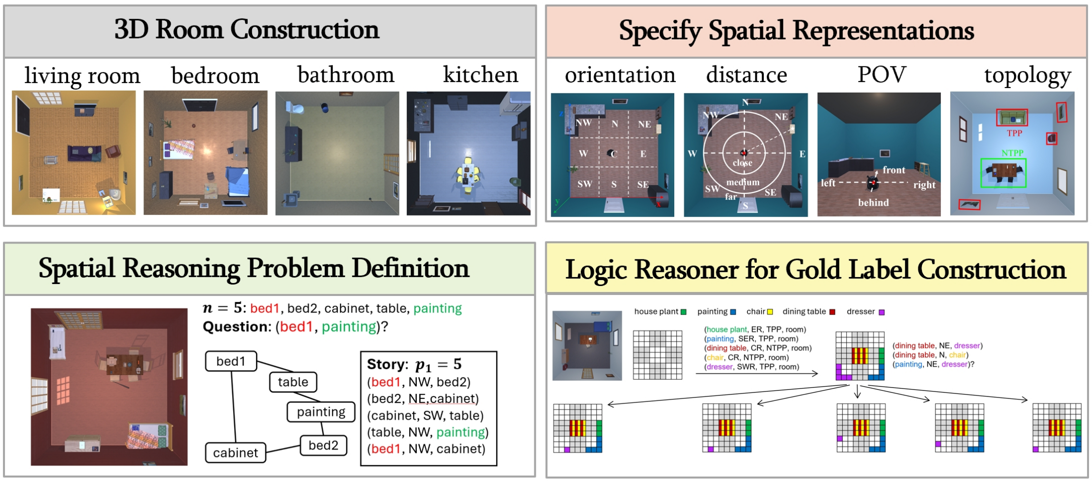



Figure 5.1: Summary of the RoomSpace benchmark generation process.

## 5.1 3D Room Construction

### 5.1.1 Construction Process

In this section, we discuss the development of virtual house environments for spatial reasoning challenges utilizing the ProcTHOR framework [13], which is built on top of AI2-THOR [12]. ProcTHOR facilitates the creation of physics-enabled environments, suitable for simulating various indoor settings. Initially, the ProcTHOR dataset comprised simulated houses with multiple rooms. For our research, we have adapted this framework to focus on single-room configurations to simplify the spatial reasoning tasks.

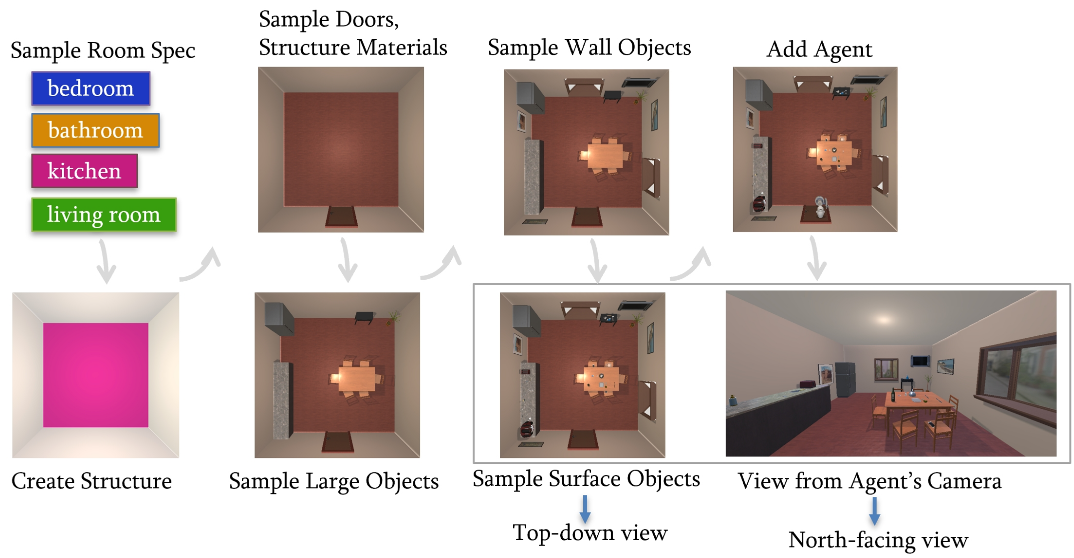



Figure 5.2: Procedural generation of a room scene with top-down and north-facing view images.

Figure 5.2 illustrates a high-level schematic of the scene generation process. This process employs multi-stage conditional sampling to specify room types and select assets, which include large household objects like fridges, countertops, beds, toilets, and houseplants, as well as wall objects such as windows and paintings, and surface objects like cups on a kitchen counter. The scenes are diversified across different room types, including kitchens, living rooms, bedrooms, and bathrooms, examples of which are shown in Figure 5.3, displayed in both top-down and egocentric (north-facing) Views.

Each simulated room is designed with a uniform square shape, enclosed by four walls (north, south, east, and west) that feature architectural elements like doors and windows. Despite the structural uniformity, each room type is distinctly outfitted with varied configurations of household objects and specific environment metadata that include the dimensions of the scene and reachable grid positions. A key structural element in each room is a centrally placed door on the south wall, which serves as both an entrance and a focal point for spatial reasoning tasks.


Figure 5.3: Sample scenes in RoomSpace featuring four room types.

For each constructed example scene, we create two types of RGB images[1] to create the visual modality for our benchmark data: one from a fixed camera positioned at the center of the ceiling, providing a top-down view, and another from the agent's egocentric perspective. The agent, an AI abstract entity capable of navigating and interacting within the virtual environment, is incorporated into the scene and located at the door facing inward. Camera adjustments ensure that the agent's perspective aligns with cardinal directions to optimize the room's visibility. However, this configuration may result in some objects along the south wall being obscured from view.

### 5.1.2 Spatial Attributes

Figure 5.5 outlines the various properties associated with a room scene. Each room is equipped with the `floorPolygon` attribute, which consists of a list of four $x, y, z$ dictionaries that specify the coordinates of the room's four corners, where $x$ and $z$ define the horizontal coordinates, and $y$ represents the vertical coordinate. This attribute forms the basis for the spatial structure of the scene.

Each object is described by its type, position, rotation, and bounding box information. The *object placement annotations* guide the placement of objects in the scene, with attributes `inLivingRooms`, `inKitchens`, `inBedrooms`, and `inBathrooms` specifying the room types where the object can be placed. Every object is assigned a room weight for

[1]The images are for visualization but could be used to test multi-modal LLMs.

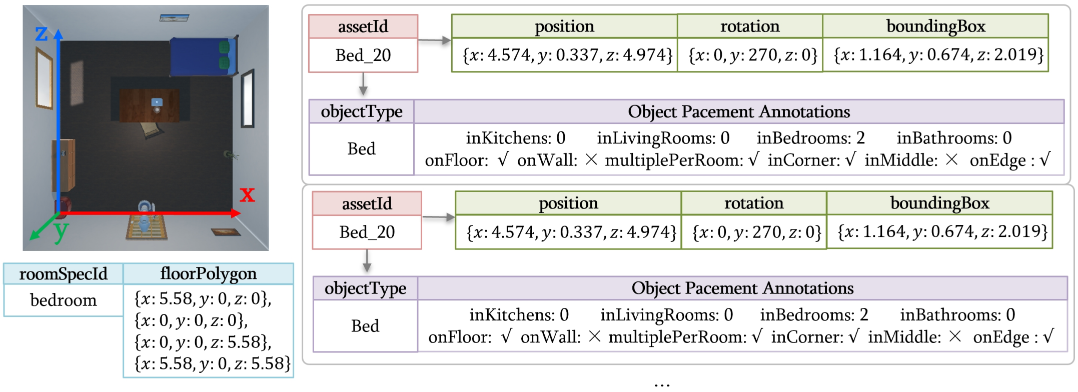



Figure 5.4: Examples of property annotations in a room example. Room type and floor polygon (highlighted in blue) depicting the room's four-corner $x, y, z$ coordinates. Each object is defined with detailed attributes: asset annotations (in green) detailing objects' spatial information; and Object placement annotations (in purple) used for scene construction.

Figure 5.5: Examples of property annotations in a room scene. The room type and floor polygon (highlighted in blue) display the room's four-corner $x, y, z$ coordinates. Each object is defined with detailed attributes: asset annotations (in green) detailing objects' spatial information; and Object placement annotations (in purple) used for scene construction.

each room type. For example, for the `inBathrooms` attribute, a `ClothesDryer` has a weight of 1, indicating suitability, whereas a `CoffeeMachine` has a weight of 0, indicating unsuitability. This weighting system filters objects for placement, ensuring only those with positive weights are included in the corresponding room type.

Additionally, the `onFloor` or `onWall` attributes, marked as `True` or `False`, determine whether an object can be placed on the floor or mounted on the wall. Only certain objects, like doorways, door frames, hand towel holders, light switches, and toilet paper holders, can be mounted on walls due to their functional nature, while items like dining tables, beds and countertops are placed on the floor. Each floor object can support specific surface items, such as vases, pans, or laptops, which enhances the scene's realism and functionality. This is accomplished through a detailed receptacle list for each floor object, specifying both the quantity and the probability of the surface objects items being included. Additionally, the attributes of `inCorner`, `inMiddle`, and `onEdge` influence both the location and orientation of the objects. Objects near walls or corners are oriented for user access, such as making appliances easy to operate, while those in the center allow for more flexible orientation to optimize spatial arrangement.

The `position` of an object is provided in coordinates using an $x, y, z$ dictionary, ensuring precise placement within the virtual environment. The `rotation` attribute, applied primarily around the $y$ axis, adjusts the object's orientation, as shown in Figure 5.6. The `bounding box`, essential for estimating the space an object occupies, is represented by the smallest aligned box that can fully enclose the object. This box remains fixed regardless of object movement or rotation, The dimensions of the bounding box (length, height, and width) are also stored in an $x, y, z$ dictionary, providing a rough estimate of the object's volume.

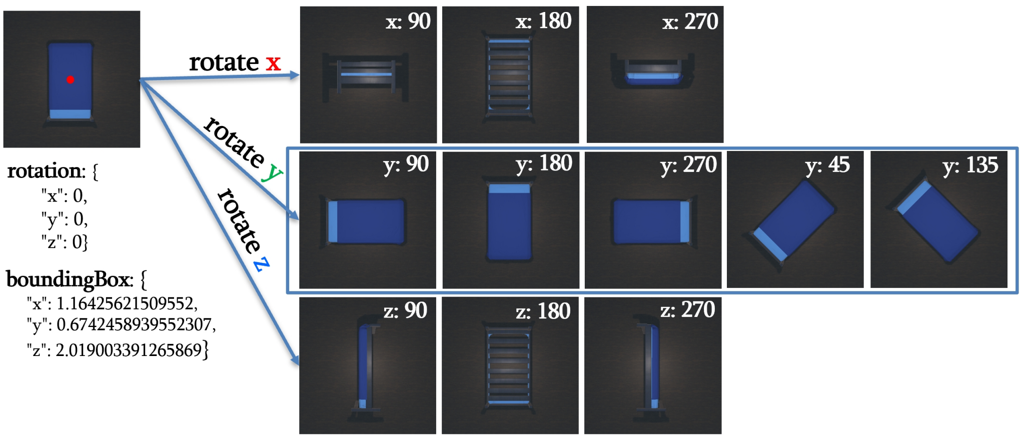



Figure 5.6: Visualization of object rotation along the $x$-axis, $y$-axis, and $z$-axis.

Additionally, while objects in RoomSpace can possess material attributes such as Metal, Wood, Plastic, and Glass, these characteristics are not currently utilized in the spatial reasoning tasks within our framework. We focus on the spatial and physical properties relevant to the challenges posed in our scenarios.

## 5.2 Specify Spatial Representations

The metadata from the previous step is leveraged to create spatial representations using various spatial calculi, which allows for the customization of spatial relationships in spatial reasoning tasks. In the following part, we detail the application of the CDC calculus for point-based cardinal direction relations, the use of 1-cross for relative direction relations, the TPCC for combining point-based distance with direction relations, and the RCC for forming region-based topology relations. We demonstrate the process of mapping object metadata to these spatial relations, which are then employed to construct diverse spatial narratives.

### 5.2.1 Cardinal Direction Relations

In Figure 5.7, we present an example of a room projected onto the 2D ground plane, utilizing the CDC calculus [111] to express cardinal direction relations - North (N), West (W), East (E), South (S) - and their combinations, North-West (NW), North-East (NE), South-West (SW), and South-East (SE). Two methods are illustrated: the projection-based method (middle of Figure 5.7) and the cone-based method (right of Figure 5.7). Additionally, we use directional relations to partition the space, dividing the room into nine sections, with an object's position determined by the segment in which the center of its bounding box falls. For instance, `(bed, NE, room)` indicates the bed's location in the northeast part of the room, `(bed, NE, chair)` describes the spatial relationship between the bed and the chair.

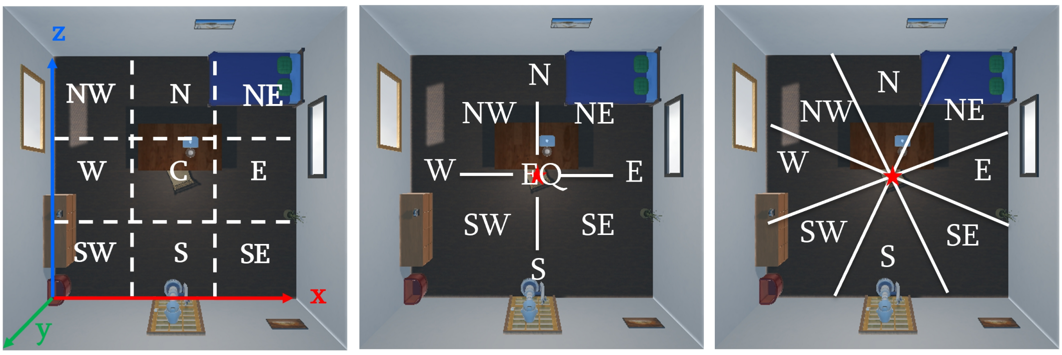



Figure 5.7: Overview of constructing directional relations within the room scene on a 2D plane. Left: room partition using directional relations. Middle: establishing object relations using projection-based CDC. Right: establishing object relations using cone-based CDC.

We utilize the $x - z$ plane to determine directions, as the $y$ axis represents vertical height. To partition the room into areas corresponding to cardinal directions, we divide the room based on thirds of its width ($w$). This segmentation creates nine distinct zones, with the central zone defined by the region where $\frac{w}{3} \leq x \leq \frac{2w}{3}$ and $\frac{w}{3} \leq z \leq \frac{2w}{3}$. This one-third partitioning evenly divides the room into nine segments.

For the projection-based CDC, the directional relationship from A to B is determined by calculating the differences in their central points' x and z coordinates. A negative $\Delta x$ indicates that B is west of A, while a negative $\Delta z$ indicates that B is south of A.

$$\Delta x_{AB} = x_b - x_a, \quad \Delta z_{AB} = z_b - z_a$$

For the cone-based CDC, the relations are derived by calculating the angle $\theta$ from the positive x-axis to the line connecting the central point of A to the central point of B using the arctangent function, which considers the full range of 360 degrees.

$$\theta_{AB} = \text{atan2}(\Delta z, \Delta x)$$

### 5.2.2 Relative Direction Relations

According to [136], the most intuitive method to express relative direction involves using ternary relations that indicate the position of object C in relation to object B from the perspective of point A (where A, B, and C are considered points on a plane). In the room scene, we use A to represent the central position of the agent, serving as the viewpoint of an observer who visually perceives the environment. Object B is the chair, specifically its center point for the point-based representation. Object C represents other items in the room. The center of Object C in certain areas will define specific relative direction spatial relations in relation to Object B, such as (bed, right/back, chair) for 1-cross, and (bed, distant/right/back, chair) for TPCC.

In Figure 5.8, we present two distinct methods for constructing relative directional relations: 1-cross and TPCC, both of which are point-based calculi for modelling relative direction relationships. Compared with 1-cross, TPCC distinguishes between right/back, right-back, and back-right and also includes direct orientations like straight right and straight front.

### 5.2.3 Distance Relations

In addition to directional relations, TPCC incorporates distance information by defining concentric circles around the reference point, assigning qualitative distances to

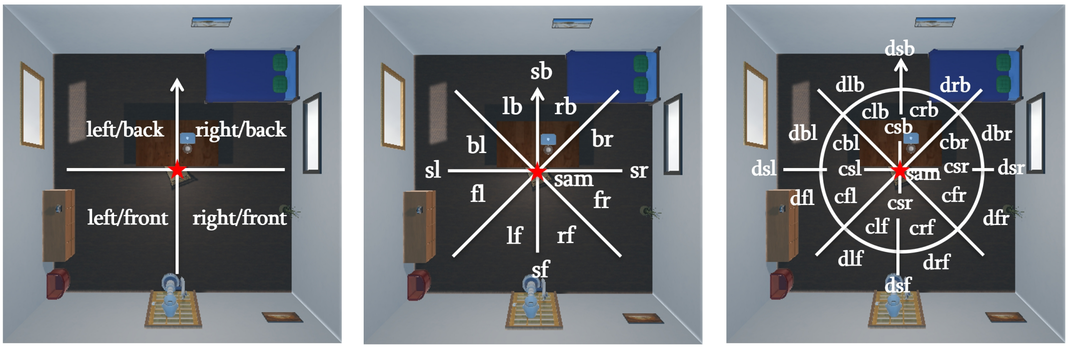



Figure 5.8: Building relative direction relations using single cross calculus (left) and TPCC without (middle) and with distance(right). The abbreviations $f$, $b$, $l$, $r$, $s$, $c$, and $d$ represent `front`, `back`, `left`, `right`, `straight`, `close`, and `distant` respectively. $sam$ represents the `same` position.

the second point B. For qualitative distance assessment, various sets of distance relations offer different levels of granularity, as outlined by [158]. The simplest distinction is between `close` and `far`. More granular levels include distinctions such as `close`, `medium`, and `far`, while a finer level introduces terms like `very close`, `close`, `medium`, `far`, and `very far`, as shown in Figure 5.9. These terms can vary, such as `far` being labelled as `distant` in Figure 5.8, though they convey the same meaning.


Figure 5.9: Various granularity configurations of distance relations distinctions. The abbreviations $cl$, $fr$, $md$, $vc$, $vf$, and $cm$ represent `close`, `far`, `medium`, `very close`, `very far` respectively. More distinctions can be introduced as needed.

In Figure 5.9, each set of distance relations is delineated by distinct boundaries, defined by intervals $\delta_0, \delta_1, \ldots, \delta_n$. In real-world scenarios, these boundaries might exhibit fuzziness, allowing for heuristic interpretation with overlapping intervals, as suggested by [159]. For simplicity, we demonstrate how to construct distance relations with clear, sharp boundaries. We determine the distance between objects by calculating the Euclidean distance between the centers of their bounding boxes (points A and B), projected onto a 2D plane.

$$d_{AB} = \sqrt{(x_A - x_B)^2 + (z_A - z_B)^2}$$

Suppose $w$ represents the length and width of a square room, then $\sqrt{2}w$ would correspond to the room's diagonal length. $\sqrt{2}w$ represents the maximum possible distance. Boundaries between `close` and `far` could be defined using boundary values such as $\frac{w}{2}$ or $\frac{\sqrt{2}w}{2}$. For dividing distances into $i$ distinct categories, a straightforward approach is to use ratios such as $\frac{w}{i}$ or $\frac{\sqrt{2}w}{i}$.

### 5.2.4 Topological Relations

Region-based object representation is more suitable for establishing topological relations, as it allows for the analysis of an object's three fundamental components: its interior, boundary, and exterior. In our room scene, relying solely on objects' central points may not accurately capture their topological relationships. For example, a large object like the bed in Figure 5.10 has a central point that is relatively distant from any wall, despite the object itself being adjacent to the wall.

In 3D environments, topological relations become significantly more complex, requiring a deeper understanding of how spatial relations are defined and how objects interact. However, for broader accessibility and practical use, 2D representations offer a more intuitive approach, distilling complex spatial interactions into clearer visual forms that are easier to understand and apply in everyday scenarios. For example, in the simulated scenario in Figure 5.10, in a 2D context, the room's boundary is defined by square-shaped walls, with the interior set comprising the space within those walls. Objects like the bed, dresser, and bin typically exhibit a TPP relation to the room, meaning they are located inside, touching the room's boundary walls. In contrast, objects such as the table and chair, which do not touch the boundary walls, exhibit an NTPP relation. In a 3D context, the room's boundary includes not only the walls but also the floor and ceiling, with the interior set encompassing all the space within these expanded boundaries. Objects like tables and chairs, which only touch the floor in 2D, still maintain a TPP relation in 3D, as they remain fully within the room's boundaries.

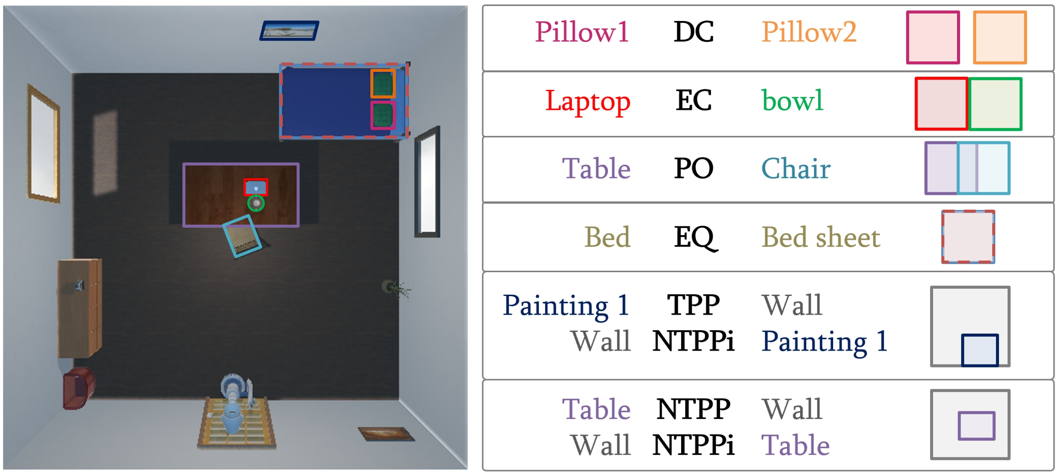



Figure 5.10: Visualization of topological spatial relations using RCC-8 in a 2D projected room scene.

In our benchmark, complexities in defining topological relationships arise when an object's shape deviates from a rectangular bounding box and when non-standard rotations (i.e., not at 90, 180, or 270 degrees) cause the bounding box edges to misalign with the boundary walls, as shown in Figure 5.6. For example, in Figure 5.10, objects like houseplants and bowls have round shapes, and the bin is semi-circular. These shapes influence how objects are perceived in relation to each other within the space, necessitating careful consideration to accurately determine their topological relationships. The chair exemplifies the non-standard rotation, having been rotated at a $y$-axis angle of 338 degrees.

Each object's metadata includes details about the center point, rotation, and bounding box size, but lacks shape information, which limits the depth of topological relations that can be established. Given that all objects are considered with rectangular bounding boxes, we define two approaches to represent topological relations between objects and the room's boundary wall:

- Uniform Inclusion. All objects are considered within the room, with no specific topological distinctions made.

- `TPP` and `NTPP`. This setting defines the topological relations between objects and the room by considering the walls as the boundary, excluding the floor and ceiling. This is accomplished by comparing each vertex of the object's bounding box to the room's boundary walls, using the bounding box data in conjunction with the central point data. We calculate four boundary values: $x_o + x_b$, $x_o - x_b$, $z_o + z_b$, and $y_o - y_b$, where $z_o$ and $z_o$ are the coordinates of the object's central point, and $z_b$ and $z_b$ are the dimensions of the bounding box. If any of these values equals 0 or $w$ (the width of the room), the object is classified as TPP to the room; otherwise, it is classified as NTPP.

## 5.3 Spatial Reasoning Problem Construction

### 5.3.1 Problem Definition

Various spatial reasoning tasks can be developed to evaluate an intelligent system, such as deriving new knowledge from provided information, verifying the consistency of the information, or updating existing knowledge. While these tasks differ, they can often be transformed into one another, allowing algorithms designed for one reasoning problem to be adapted to others. Therefore, much of the research on spatial reasoning has focused on the constraint satisfaction problems (CSP), which determines whether the given spatial information is consistent or inconsistent [24].

These types of queries mirror the kinds of spatial reasoning humans perform during everyday tasks such as navigation, giving instructions, interpreting floor plans, or interacting with household robots and virtual assistants. Unlike synthetic benchmarks that use abstract or overly simplified language, RoomSpace embraces more naturalistic language patterns and real-world spatial configurations. This alignment allows for a better assessment of how well LLMs and MLLMs can support real-world applications such as assistive technologies, robotics, and multimodal dialogue systems, where understanding space and context is critical.

A CSP is defined by a finite set of variables $V$, a domain $D$ for these variables and a set of constraints $\theta$. Each constraint restricts the values that a subset of variables can simultaneously take. The objective is to find an assignment of values to the variables such that all constraints are satisfied or, in some cases, to identify all possible valid assignments. In our CSPs, each object in a room is treated as a variable, with the room's dimensions forming the domain. The spatial relations between entities, as discussed earlier, serve as constraints that govern the spatial properties of the objects, as illustrated in Figure 5.11.

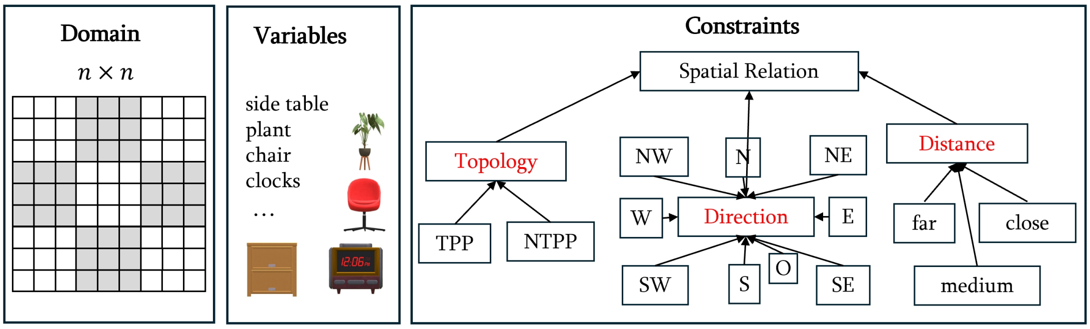



Figure 5.11: Core elements (variables, domain, constraints) in our CSP framework.

A spatial reasoning problem can be framed within a constraint network framework: consider a network of $n$ spatial variables $V = \{o_1, ..., o_n\}$ within a domain $D^n$. Each node represents a variable $o_i$, and each directed edge is labelled with a relation constraint. The relation constraining a pair of variables $\langle o_i, o_j \rangle$ is denoted by $r_{ij}$. A relation constraint in the set $\theta$ can therefore be written as $r_{ij}(o_i, o_j)$ or $(o_i, r_{ij}, o_j)$. Given a set of $k$ relations and a query $(o_a, ?, o_b)$, LLMs are tasked with predicting the relation $r_{ab}$. If all constraints in the story, including the predicted relation $(o_a, r'_{ab}, o_b)$, can be simultaneously satisfied, the prediction $r'_{ab}$ is considered a valid solution.

To bridge CSPs with ecologically valid task design, the spatial reasoning tasks in RoomSpace are crafted to closely emulate real-world settings and interactions. These tasks reflect the kinds of spatial queries that naturally arise in daily life, thereby enhancing the benchmark's relevance to practical applications. For example: locating items in a room (e.g., "Where is the cup relative to the microwave?"), which simulates how a user might query a household robot or virtual assistant; verifying arrangements or consistency checking (e.g., "Is the chair between the table and the wall?", "Is this room configuration feasible given the following constraints?"), which resembles spatial planning, such as arranging furniture or validating room layouts. These questions are derived from structured CSPs but are translated into naturalistic narratives and queries that align more closely with human expectations and behaviour. In this way, RoomSpace connects the formal rigour of logical reasoning with the flexibility and realism of real-world spatial cognition, making it a more practical and comprehensive benchmark for evaluating the spatial reasoning capabilities of LLMs.

### 5.3.2 CSP Example Generation

Our test sets are available in varying sizes: **RoomSpace-100** includes a sample of 100 rooms. **RoomSpace-1K** consists of 1,000 rooms, and **RoomSpace-10K** comprises 10,000 rooms. The initial 100 rooms in RoomSpace-1K (ID 0-99) are identical to those in RoomSpace-100. Similarly, the first 1,000 rooms in RoomSpace-10K (ID 0-999) match those in RoomSpace-1K. For each room scene, we create various story series of stories with varying levels of complexity, achieved by adjusting four key parameters, each corresponding to specific elements of the constraint network. These configurations are represented by the tuple $\langle n, d, m, p \rangle$, where:

- $n$ is the number of objects used to form the story in the scene, which correspond to the nodes in a constraint graph. In this part of the work, we focus on selecting floor and wall objects for story formation rather than smaller supporting objects that occupy less space in the room. For instance, in the scenario of 'an apple on a desk', the desk would take priority over the apple.

- $d$ is the number of square tiles in a $width \times length$ tessellation whose centres define possible positions for the centres of objects on the floor plane. In the dataset, *width* and *lenghth* are always equal, yielding square rooms.

- $m$ is the number of binary constraints over $n$ objects. The maximum possible number of constraints on $n$ variables is $C_n^2 = \frac{n(n-1)}{2}$, where each variable is constrained by all other variables, forming a complete graph. For instance, a complete graph with 5 objects yields in total $C_5^2 = 10$ constraint pairs. From all possible pairs of objects, one pair is selected to form the question, while the remaining $C_n^2 - 1$ pairs are used to establish the graph based on the parameter $m$.

- $p$ is the constraint tightness. For constraints that restrict the direction of objects in a room, $p$ ranges from 0 to $d$, while for constraints between two objects, $p$ ranges from 0 to $d \times d$, where $d$ is the domain size for each variable. The total number of possible value pairs between two variables is $d \times d$. For each constraint between two objects, the number of disallowed value pairs is given by $p \times (d \times d)$. The value of $p$ depends on the specific type of constraint. We provide a detailed analysis of constraint tightness in Section 5.4.

### 5.3.3 Generate Textual Descriptions

Following the previous steps of node and constraint selection, the constraint networks for each instance are established. For evaluating LMs, these networks are then converted into textual format, where the elements are organized into coherent narratives and the reasoning tasks are framed using natural language. During this phase, the spatial logic expressions $C_l$ and $C_o$ are transformed into natural language sentences like $S_l$ and $S_{O2}$, a process referred to as logic-to-text generation.

We develop specific logic-to-string templates using context-free grammar (CFG) [160], as illustrated in Table 5.1. It shows the structure decomposition for each story variant $S^{t,v}$, where $t \in \{L, TPP, O2, O2{+}D2, O2{+}D3, O2{+}L, O2{+}D2{+}L, O2{+}D3{+}L\}$ denotes the content type and $v \in \{T, N\}$ indicates the view perspective (Top-down or North-facing). Each story series is composed of specific combinations of layout components $\langle S_l^i \rangle$ and object-object relation components $\langle S_{O2}^i \rangle$. $S_l$ serves as the introductory sentence, outlining the objects present in the room and varies across three types: $S_{l\_O2}$, which lists the objects; $S_{l\_Layout}$, which adds information about object layout; and $S_{l\_TPP}$, which includes topological relationships between the objects and the room's walls. $S^N$ and $S^T$ describe relationships from north-facing and top-down views, respectively. $S_{O2}$ details directional relationships between objects, while $S_{O2+D2}$ and $S_{O2+D3}$ combines these with binary and ternary distance information.

When forming stories, the logical components such as $\langle x_i \rangle$,$\langle x_j \rangle$, $\langle r_i^{Dir} \rangle$,$\langle r_i^{TPP} \rangle$, $\langle r_{ij}^{Dir} \rangle$, $\langle r_{ij}^{Dis} \rangle$ are substituted with corresponding textual expressions, enabling the creation of varied descriptions of spatial relationships. The term $\langle room \rangle$ refers to one of several room types: '*kitchen*', '*living room*', '*bathroom*', or '*bedroom*'. The placeholders $\langle x_i \rangle$ and $\langle x_j \rangle$ represent objects such as '*chair*', '*coffee machine*', or '*desk*'. The details of the spatial expressions $r_i^{TPP}$, $r_{ij}^{Dir}$, $r_{ij}^{Dir\_N}$, $r_{ij}^{Dis}$ are provided in Table 5.2.

We develop eight narrative series across two viewpoint settings, each designed to emphasize different aspects of spatial relationships, as presented in Table 5.1.

- **Layout Narratives** ($S_{l\_Layout}$): Focus on the spatial arrangement of objects within the room, referred to as *Layout*.

- **Topological Narratives** ($S_{l\_TPP}$): Detail both the layout and the topological relationships between objects and the room's structural elements, referred to as *TPP* since all four topological relations used include this character string.

| Story Components | | |
|---|---|---|
| $S_l \Rightarrow$ *Imagine a square-shaped* $\langle room \rangle$*, bordered by four walls. This room contains a collection of furniture, including* $\langle S_l^0 \rangle$, $\langle S_l^1 \rangle$, ..., $\langle S_l^n \rangle$. | | |
| $S^i_{l\_Layout} \Rightarrow \langle x_i \rangle$ *placed in the* $\langle r_i^{Dir} \rangle$ | | |
| $S^i_{l\_TPP} \Rightarrow \langle x_i \rangle$ *placed in the* $\langle r_i^{Dir} \rangle$, $\langle r_i^{TPP} \rangle$ *the wall* | | |
| $S^i_{l\_O2} \Rightarrow \langle x_i \rangle$ | | |
| $S^T_{O2} \Rightarrow \langle S^{T01}_{O2} \rangle.\ \langle S^{T12}_{O2} \rangle.\ \ldots.\ \langle S^{Tij}_{ot} \rangle.$ | | |
| $S^{Tij}_{O2} \Rightarrow \langle x_i \rangle$ *is placed to the* $\langle r_{ij}^{Dir} \rangle$ *of* $\langle x_j \rangle$ | | |
| $S^{Tij}_{O2+D2}, S^{Tij}_{O2+D3} \Rightarrow \langle x_i \rangle$ *is placed to the* $\langle r_{ij}^{Dir} \rangle$ *of* $\langle x_j \rangle$, $\langle r_{ij}^{Dis} \rangle$ | | |
| $S^N_{O2} \Rightarrow$ *Imagine yourself at the southern wall's door, looking inwards. From this perspective,* $\langle S^{N01}_{O2} \rangle.\ \ldots.\langle S^{Nij}_{O2} \rangle.$ | | |
| $S^{Nij}_{O2} \Rightarrow \langle x_i \rangle$ *is* $\langle r_{ij}^{Dir} \rangle$ $\langle x_j \rangle$ | | |
| $S^{Nij}_{O2+D2}, S^{Nij}_{O2+D3} \Rightarrow \langle x_i \rangle$ *is* $\langle r_{ij}^{Dir\_N} \rangle$ $\langle x_j \rangle$, $\langle r_{ij}^{Dis} \rangle$. | | |
| **Story** $S^{t,v}$ | | |
| **Story $\Rightarrow$ Expansion** | **Format of $\langle S_l^i \rangle$** | **Format of $\langle S_{O2}^i \rangle$** |
| $S^{\mathrm{L}} \Rightarrow S_l$ | $S^i_{l\_Layout}$ | / |
| $S^{\mathrm{TPP}} \Rightarrow S_l$ | $S^i_{l\_TPP}$ | / |
| $S^{\mathrm{O2,T}} \Rightarrow S_l\ S^T_{O2}$ | $S^i_{l\_O2}$ | $S^{Tij}_{O2}$ |
| $S^{\mathrm{O2,N}} \Rightarrow S_l\ S^N_{O2}$ | $S^i_{l\_O2}$ | $S^{Nij}_{O2}$ |
| $S^{\mathrm{O2+D2,T}} \Rightarrow S_l\ S^T_{O2}$ | $S^i_{l\_O2}$ | $S^{Tij}_{O2+D2}$ |
| $S^{\mathrm{O2+D2,N}} \Rightarrow S_l\ S^N_{O2}$ | $S^i_{l\_O2}$ | $S^{Nij}_{O2+D2}$ |
| $S^{\mathrm{O2+D3,T}} \Rightarrow S_l\ S^T_{O2}$ | $S^i_{l\_O2}$ | $S^{Tij}_{O2+D3}$ |
| $S^{\mathrm{O2+D3,N}} \Rightarrow S_l\ S^N_{O2}$ | $S^i_{l\_O2}$ | $S^{Nij}_{O2+D3}$ |
| $S^{\mathrm{O2+L,T}} \Rightarrow S_l\ S^T_{O2}$ | $S^i_{l\_Layout}$ | $S^{Tij}_{O2}$ |
| $S^{\mathrm{O2+L,N}} \Rightarrow S_l\ S^N_{O2}$ | $S^i_{l\_Layout}$ | $S^{Nij}_{O2}$ |
| $S^{\mathrm{O2+D2+L,T}} \Rightarrow S_l\ S^T_{O2}$ | $S^i_{l\_Layout}$ | $S^{Tij}_{O2+D2}$ |
| $S^{\mathrm{O2+D2+L,N}} \Rightarrow S_l\ S^N_{O2}$ | $S^i_{l\_Layout}$ | $S^{Nij}_{O2+D2}$ |
| $S^{\mathrm{O2+D3+L,T}} \Rightarrow S_l\ S^T_{O2}$ | $S^i_{l\_Layout}$ | $S^{Tij}_{O2+D3}$ |
| $S^{\mathrm{O2+D3+L,N}} \Rightarrow S_l\ S^N_{O2}$ | $S^i_{l\_Layout}$ | $S^{Nij}_{O2+D3}$ |
| **Question** $Q$ | | |
| $Q_{fr} \Rightarrow$ *Where is the* $x_i$ *positioned in relation to the* $x_j$*?* | | |
| $Q_{yn} \Rightarrow$ *Could the* $x_i$ *be placed to the* $r_q$ *of the* $x_j$*?* | | |

Table 5.1: Our designed grammar for forming spatial reasoning stories and questions.

| Relation Type | Symbols | Relations | Expressions |
|---|---|---|---|
| Topological | $r_i^{TPP}$ | TPP | in, against the wall, hanging on the wall |
| | | NTPP | in, away from walls |
| | | TPPi | contains |
| | | NTPPi | contains |
| Directional | $r_{ij}^{Dir}$ | N | north |
| | | S | south |
| | | E | east |
| | | W | west |
| | | NE | north-east |
| | | SE | south-east |
| | | NW | north-west |
| | | SW | south-west |
| | $r_{ij}^{Dir\_N}$ | Left | to the left of |
| | | Right | to the right of |
| | | Front | in front of |
| | | Behind | behind |
| | | Front-Left | in front of and to the left of |
| | | Front-Right | in front of and to the right of |
| | | Behind-Left | behind and to the left of |
| | | Behind-Right | behind and to the right of |
| Distance | $r_{ij}^{Dis}$ | Close | at a short distance |
| | | Medium | at a moderate distance |
| | | Far | at a far distance |

Table 5.2: Spatial relation types and examples of spatial language expressions.

- **Directional Overview** ($S_{l\_O2} + S_{O2}$): Provide a general description of all objects and their directional relationships, referred to as *O2*.

- **Directional and Binary Distance Narratives** ($S_{l\_O2} + S_{O2+D2}$): Include an overview of all objects, outlining both directional and binary distance (close, far) relationships, referred to as *O2+D2*.

- **Directional and Ternary Distance Narratives** ($S_{l\_O2} + S_{O2+D3}$): Expand the previous series by incorporating three levels of distance (close, moderate, far) along with directional relationships, referred to as *O2+D3*.

- **Layout with Direction** ($S_{l\_O2} + S_{O2}$): Merge layout descriptions with directional relationships, referred to as *O2+Layout*.
- **Layout with Binary Distance and Direction** ($S_{l\_Layout} + S_{O2+D2}$): Merge layout descriptions with binary distance and directional relationships, referred to as *O2+D2+Layout*.
- **Layout with Ternary Distance and Direction** ($S_{l\_Layout} + S_{O2+D3}$): Combine layout information with three-level distance and directional relationships, offering a comprehensive depiction of spatial dynamics within the room, referred to as *O2+D3+Layout*.

We build two types of questions for each CSP example:

- **Find Relation (FR)**: This type of question requires identifying the directional spatial relationships between two specified objects. The answers are presented as a list of possible relationships.
- **Yes/No (YN)**: These questions aim to ascertain the validity of a statement regarding the spatial relationship between objects. A directional relation is randomly selected as the assertion. If this selected relation aligns with all relations in the story, the response is '*Yes*'; if not, the answer is '*No*'.

FR questions are more complex, as they focus on identifying all possible valid assignments of constraints between two variables. In contrast, YN questions are simpler, aiming to verify whether a specific constraint assignment holds, ensuring that all constraints are satisfied.

## 5.4 Logical Reasoner for Gold Label Construction

In this section, we outline the rationale behind the design of our logical reasoner and explain its operational framework.

### 5.4.1 Motivation

Generating ground-truth answers for spatial relations between objects $o_i$ and $o_j$ from the simulation system can be automated by comparing their coordinates, represented as $(x_i, y_i, z_i)$ and $(x_j, y_j, z_j)$ which could get from meta simulation data. However, key considerations arise: Given the stories formed with limited qualitative relations, can we definitively deduce the answer? Is there a possibility of multiple valid solutions? For example, given the description '*A is to the left of B, and C is to the left of A.*' the position of A relative to C is ambiguous based on the information provided. A could be to the right, left, or overlapping with C.

The stories in our benchmark provide only a partial depiction of spatial layouts. Given the limited qualitative descriptions, a single, definitive answer may not always be achievable. For instance, as shown in Figure 5.12, consider four constraints: $(x_1, NE, x_2)$, $(x_2, SW, x_3)$, $(x_3, NW, x_4)$, and $(x_4, NE, x_5)$. When asked for $R_{13}$ (the relation between $x_1$ and $x_3$), $R_{14}$, or $R_{15}$, the possible answers span all nine relation candidates, meaning all nine options are consistent with the four given constraints.

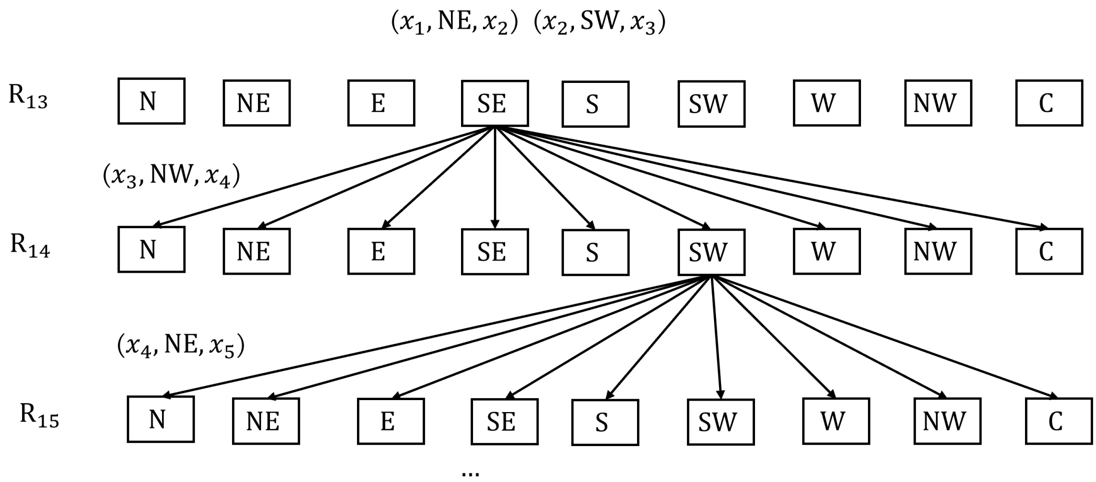



Figure 5.12: An example illustrating the potential for multiple valid answers.

According to [161], a key component of spatial intelligence is the flexibility to select appropriate strategies for solving spatial problems. Different methods can be used to automatically address spatial reasoning tasks, each presenting distinct advantages and challenges depending on the task's context and complexity:

- **Composition Reasoning Tables [162].** Utilizing tables to systematically analyze and decompose problems into components, examining the relationships and interactions between them.
- **Reasoning Rules [1].** Using predefined rules to automate the reasoning process,

| | **N** | **NE** | **E** | **SE** | **S** | **SW** | **W** | **NW** | **O** |
|---|---|---|---|---|---|---|---|---|---|
| **N** | N | NE | NE | E,NE,SE | S, N, O | W,NW,SW | NW | NW | N |
| **NE** | NE | NE | NE | E,NE,SE | E, NE, SE | N,NE,E, SE,S,SW, W,NW,C | N,NW,NE | N,NW,NE | NE |
| **E** | NE | NE | E | SE | SE | S,SW,SE | W,E,O | N,NW,NE | E |
| **SE** | E, NE, SE | E, NE, SE | SE | SE | SE | S,SW,SE | S,SW,SE | N,NE,E, SE,S,SW, W,NW,C | SE |
| **S** | S, N, O | E, NE, SE | SE | SE | S | SW | SW | W,NW,SW | S |
| **SW** | W,NW,SW | N,NE,E, SE,S,SW, W,NW,C | S,SE,SW | S,SE,SW | SW | SW | SW | W,NW,SW | SW |
| **W** | NW | N,NW,NE | W,E,O | S,SE,SW | SW | SW | W | NW | W |
| **NW** | NW | N,NW,NE | N,NW,NE | N,NE,E, SE,S,SW, W,NW,C | N,NW,SW | N,NW,SW | NW | NW | NW |
| **O** | N | NE | E | SE | S | SW | W | NW | O |

Table 5.3: The composition table for *O2* directional relations. The composition of different relations along a single axis (e.g., S and N, E and W) generally leads to multiple possible relations, highlighted in blue and green in the table.

as applied in SpartQA and SpaRTUN for label generation.

| | **Ojects** | **Relation** | **Premises** | **Conclusion** |
|---|---|---|---|---|
| **Not** | $\forall(X,Y) \in E$ | $R \in Dir \vee PP$ | IF $R(X,Y)$ | $\Rightarrow \neg(R_r(X,Y))$ |
| **Inverse** | $\forall(X,Y) \in E$ | $R \in Dir \vee PP$ | IF $R(Y,X)$ | $\Rightarrow R_r(X,Y)$ |
| **Symmetry** | $\forall(X,Y) \in E$ | $R \in Dir \vee (RCC - PP)$ | IF $R(Y,X)$ | $\Rightarrow R(Y,X)$ |
| **Transitivity** | $\forall(X,Y,Z) \in E$ | $R \in Dir \vee PP$ | IF $R(X,Z)$, $R(Z,Y)$ | $\Rightarrow R(X,Y)$ |
| **Combination** | $\forall(X,Y,Z,H) \in E$ | $R \in Dir$, $*PP \in PP$ | IF $*PP(X,Z), R(Z,H),$ $*PPi(Z,Y)$ | $\Rightarrow R(X,Y)$ |

Table 5.4: General reasoning rules established in [1] to infer relationships between objects. **Dir:** Directional relations (e.g., LEFT). **Dis:** Distance relations (e.g., FAR). **PP:** all Proper parts relations (*NTPP, NTPPi, TPP, TPPi*). **RCC - PP:** All RCC8 relation except proper parts relations. ***PP:** one of TPP or NTPP. ***PPi:** one of NTPPi or TPPi

- **Mental Diagrams or Images [163, 164].** According to [161], people utilize mental simulation processes involving spatial working memory, rather than solely relying on verbally encoded rules when imagining each 'link' in a causal chain. Using diagrams or images to visualize problems helps simplify the understanding of spatial relationships and aids in conceptualizing solutions. For instance, the upper-right diagram in Figure 3.2 provides a visual representation of the problem.

In our benchmark, the Composition Reasoning Tables method requires significant time to design, and new tables must be created when incorporating new relations. The reasoning rules method, while simple and efficient, is limited in its application, as it only works under specific conditions. For example, transitivity can infer $(A, \text{left}, C)$ from $(A, \text{left}, B)$ and $(B, \text{left}, C)$. However, if $R_{AB}$ and $R_{BC}$ involve different relations, this method cannot deduce $R_{AC}$. Since most combinations in our benchmark involve different relations, it becomes challenging to derive answers using predefined rules alone. So we developed a logical reasoner to simulate mental diagrams for spatial reasoning problems, with the implementation workflow detailed in the following section.

### 5.4.2 Method

We turn the problem to a problem of placing $n$ points on a $(d_1, d_2)$ board, as depicted in Figure 5.13. A valid solution is achieved when all $m$ constraints are satisfied. In our reasoner, we explore two settings for domain size: $d = 9 \times 9$ and $d = 12 \times 12$, corresponding to the room's square configuration. For simplicity, all the analysis in this section is illustrated using $d = 9 \times 9$. The reasoning system consists of two core components.

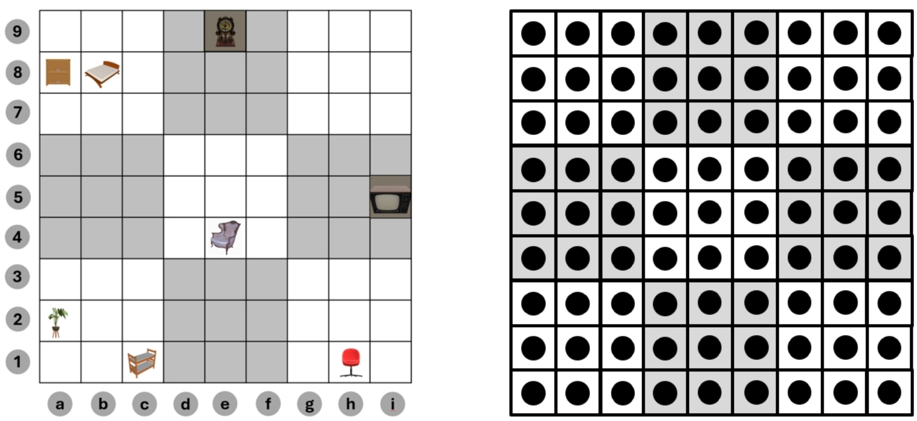



Figure 5.13: Abstraction of grid-based spatial representation. Left: objects represented as grid slots. Right: objects represented as points within the grid.

### Establishing the Fundamental Relations

The spatial relationships in our reasoning tool are defined according to the coordinate comparison, as detailed in Table 5.5.

| Relation | Definition | Relation | Definition |
|---|---|---|---|
| N | $x_1 = x_2, y_1 > y_2$ | NR | $\frac{d_1}{3} \leq x < \frac{2d_1}{3}, y \geq \frac{2d_2}{3}$ |
| S | $x_1 = x_2, y_1 < y_2$ | SR | $\frac{d_1}{3} \leq x < \frac{2d_1}{3}, y < \frac{d_2}{3}$ |
| E | $x_1 > x_2, y_1 = y_2$ | ER | $x \geq \frac{2d_1}{3}, \frac{d_2}{3} \leq y < \frac{2d_2}{3}$ |
| W | $x_1 < x_2, y_1 = y_2$ | WR | $x < \frac{d_1}{3}, \frac{d_2}{3} \leq y < \frac{2d_2}{3}$ |
| NE | $x_1 > x_2, y_1 > y_2$ | NER | $x \geq \frac{2d_1}{3}, y \geq \frac{2d_2}{3}$ |
| NW | $x_1 < x_2, y_1 > y_2$ | NWR | $x < \frac{d_1}{3}, y \geq \frac{2d_2}{3}$ |
| SE | $x_1 > x_2, y_1 < y_2$ | SER | $x \geq \frac{2d_1}{3}, y < \frac{d_2}{3}$ |
| SW | $x_1 < x_2, y_1 < y_2$ | SWR | $x < \frac{d_1}{3}, y < \frac{d_2}{3}$ |
| O | $x_1 = x_2, y_1 = y_2$ | CR | $\frac{d_1}{3} \leq x < \frac{2d_1}{3}, \frac{d_2}{3} \leq y < \frac{2d_2}{3}$ |
| CL3 | $\Vert\text{pos}_1 - \text{pos}_2\Vert \leq \frac{d_1}{3}$ | INR | $0 \leq x < d_1, 0 \leq y < d_2$ |
| MD3 | $\frac{d_1}{3} < \Vert\text{pos}_1 - \text{pos}_2\Vert \leq \frac{2d_1}{3}$ | TPP | $x \in \{0, d_1 - 1\}$ or $y \in \{0, d_2 - 1\}$ |
| FR3 | $\Vert\text{pos}_1 - \text{pos}_2\Vert > \frac{2d_1}{3}$ | NTPP | $0 < x < d_1,\ 0 < y < d_2$ |
| FR2 | $\Vert\text{pos}_1 - \text{pos}_2\Vert > \frac{d_1}{2}$ | | |
| CL2 | $\Vert\text{pos}_1 - \text{pos}_2\Vert \leq \frac{d_1}{2}$ | | |

Table 5.5: Relations and their corresponding definitions in logical reasoners.

Constraint tightness ($p$) is the fraction of variable assignments from the variables' domains that are disallowed by the constraint. Formally, given a constraint $C$ involving variables $V = \{X_1, X_2, \ldots, X_k\}$, each with domain size $d$, the tightness $p$ of constraint $C$ is defined as:

$$p = \frac{\text{Number of disallowed assignments by } C}{d^k}.$$

Constraint tightness significantly influences the difficulty of solving CSPs. Higher constraint tightness reduces the solution search space, potentially decreasing computational complexity. Conversely, constraints with lower tightness yield larger search spaces, thus increasing computational overhead during the search for a valid solution.

I have explicitly clarified the definition, properties, importance, and detailed calculations of p on page 104 (marked in red).

We provide an analysis of constraint tightness $p$ for all spatial relations, which indicates how restrictive a constraint is. A tighter constraint eliminates more values from the domains of the variables it involves, thus reducing the search space. This influences the complexity of solving a CSP with our logical reasoning tool, directly affecting the CPU runtime. For example, in Figure 5.14, consider the constraint `(house plant, ER, room)` applied to the variable '*house plant*' within a domain of $9 = 3 \times 3$ possibilities. Introducing constraint (house plant, ER+TPP, room) further reduces the domain to only 5 possible pairs: [(0, 0), (0, 1), (0, 2), (1, 0), (2, 0)].

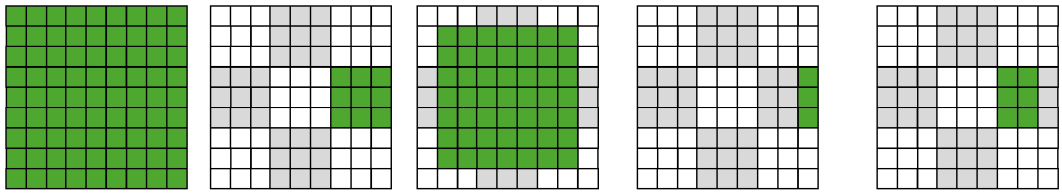



Figure 5.14: Example of assigning possible position candidates for different relations between objects and room.

This is the reason, as depicted in Figure 5.16, that there is a marked decrease in CPU times when we transition from a '*Layout*' configuration to a '*TPP*' setting. Figure 5.16 demonstrates that incorporating distance relations '*O2+D2*' leads to a rise in the average CPU time when contrasted with the '*O2*' configuration. In the narrative setting of *Layout+O2*, where layout info is combined with directional info, the search area for each object is more restricted compared to *O2*, where all info about object placement is INR. Although *Layout+O2* involves more constraints, the reduced search space speeds up the backtracking process as fewer possibilities need to be explored.

**Layout Constraints**. These constraints are not represented in constraint graphs; rather, they are utilized directly to refine the domain of the variable they constrain. Though in the form $(x_i, R_i, Room)$, they function as unary constraints.

- **InR** (In Room): $p = 0$, all possible values from the domain of the one variable are allowed and the constraint is always satisfied.

- **NR, SR, WR, ER, NER, NWR, SER, SWR, CR**: $p = \frac{8}{9}$, Each relation pertains to a specific section of the room, dividing the room into nine parts.

- **TPP, NTPP**: NTPP corresponds to the inner side of the room, with $p^{NTPP} = \frac{(\sqrt{d}-2)^2}{d}$, TPP corresponds to the border of the grid space, with $p^{TPP} = 1 - p^{NTPP} = \frac{(\sqrt{d}-1)\times 4}{d}$

**Inter-Objects Constraints**. Involving two variables to represent the relationships between objects, which can be illustrated in constraint graphs.

- **N, S, W, E, NE, NW, SE, SWR, O**: Directions between Objects. For N, S, W, E, $p = 1 - \frac{d(\sqrt{d}-1)}{2d^2}$, for NE, NW, SE, SWR, $p = 1 - (\frac{d-\sqrt{d}}{2d})^2$. For O, $p = 1 - \frac{1}{d}$
- **CL2, FR2**: Objects are considered close (CL2) if they are within half of the maximum distances for one dimension (width or length). We approximate this using a circle with radius $r_1 = \frac{\sqrt{d}-1}{2}$, so $Area^{CL2} = \pi(\frac{\sqrt{d}-1}{2})^2$, $Area^{FR2} = d - \pi(\frac{\sqrt{d}-1}{2})^2$ The $p$ calculation for distance in terms of the grid dimension $d$ is complex. The number of cells within this area ($p^{FR2}$) for the central object can be approximated by: $p^{FR2} \approx \frac{Area^{CL2}}{d}$, $p^{CL2} \approx 1 - \frac{Area^{CL2}}{d}$. For each central object, the actual count of possible variable values is limited by the number of cells that fit into this area.
- **CL3, MD3, FR3**: more restrictive than the previous two-category distances. Objects are considered close (CL3) if they are within one-third of the maximum distances within the grid, and moderate distance (MD3) if within two-thirds of the maximum distances within the grid. We approximate this using a circle with radius $r_1 = \frac{\sqrt{2}(\sqrt{d}-1)}{3}$, $r_2 = \frac{2\sqrt{2}(\sqrt{d}-1)}{3}$. $Area^{CL3} = \pi r_1^2$, $Area^{MD3} = \pi(r_2^2 - r_1^2)$ $Area^{FR3} = d - \pi r_2^2$. $p^{FR3} \approx 1 - \frac{Area^{FR3}}{d}$, $p^{MD3} \approx 1 - \frac{Area^{MD3}}{d}$ $p^{CL3} \approx 1 - \frac{Area^{CL3}}{d}$ .

### Solver to Get Answer - Consistency Checking

We built a consistency-checking tool using the **python-constraint** package[1], which employs a backtracking algorithm to determine whether a plausible configuration of object relationships can exist to meet all specified constraints. Backtracking is a systematic method of solving problems that incrementally builds candidates to the solutions and abandons a candidate as soon as it determines that this candidate cannot possibly be completed to a valid solution. In the context of our spatial reasoning problem, this method can be demonstrated as follows:

[1] https://github.com/python-constraint/python-constraint

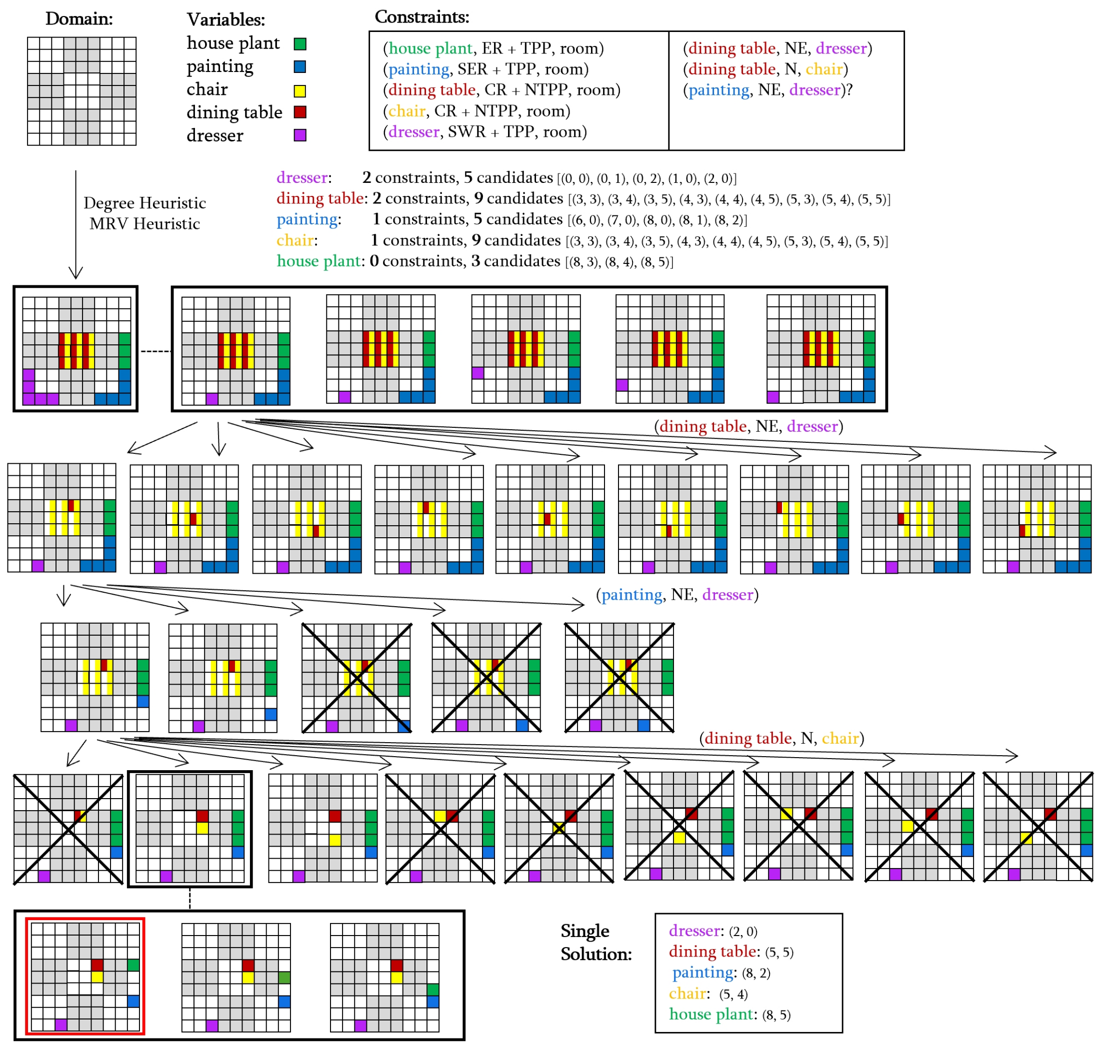



Figure 5.15: Soving one spatial reasoning problem with search tree using backtracking.

One example of getting solution process with backtracking is shown in Figure 5.15. When solving a reasoning problem, the solver begins by assigning potential position candidates to each variable within the defined domain, based on the constraints related to the objects' positions relative to the room. Once the candidate positions for each object are established, the constraints between pairs of objects are evaluated sequentially. This process involves checking each pair of variables' position candidates to ensure they can coexist without violating any of the specified constraints.

To determine answers to YN questions, the final constraint analyzed is the two queried objects and their proposed relationship as specified in the question. If a valid

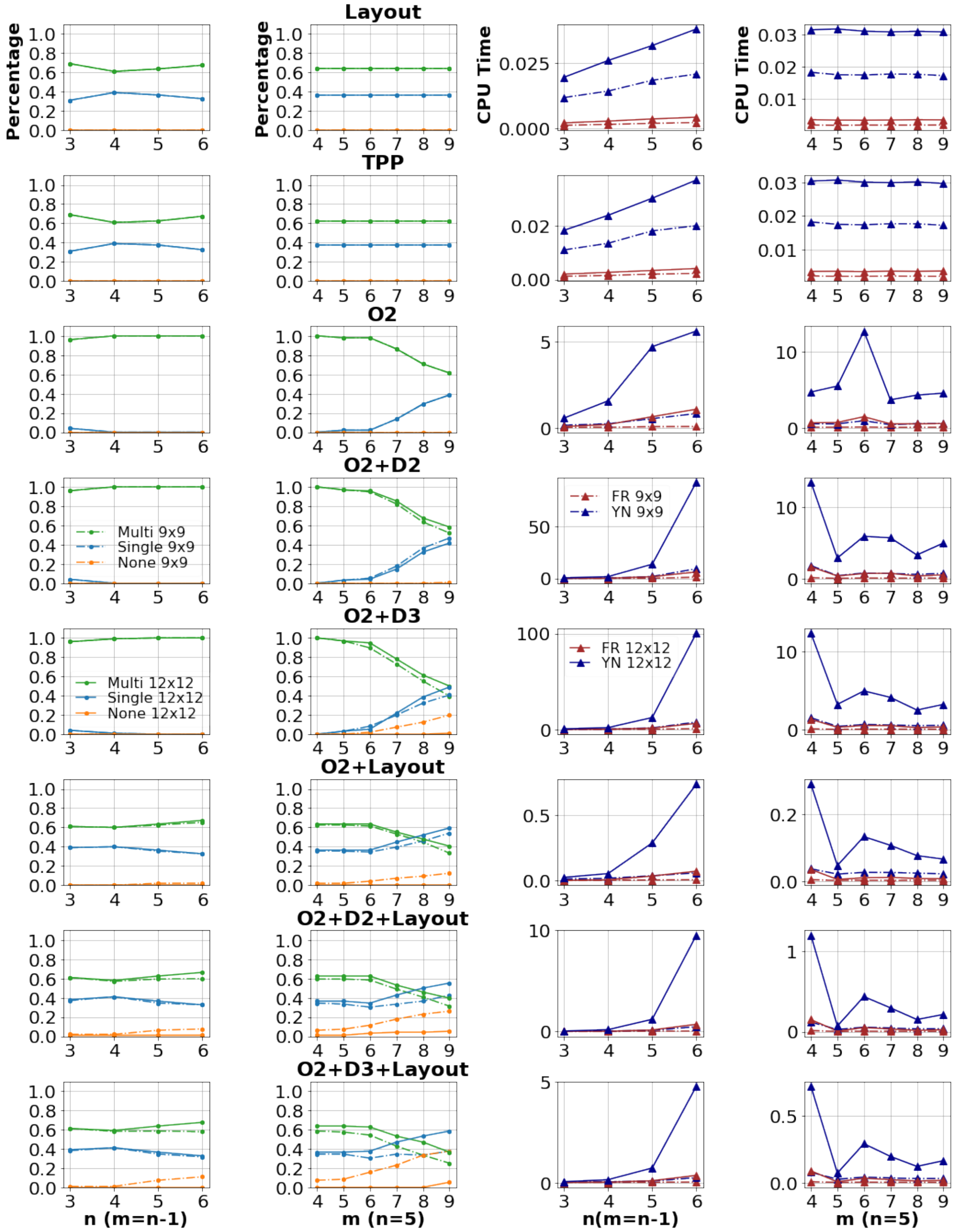



Figure 5.16: The percentage of single, multiple, and no solution occurrences (Rows 1, 2) and the average CPU time (seconds) for solution searches (Rows 3, 4).

solution is identified that conforms to this relationship, the answer is '*Yes*'. Conversely, if no valid solutions are found, the answer is '*No*'. For FR questions, the solver iteratively evaluates each potential relationship. This process mirrors the evaluation method used in YN questions, examining each candidate relationship to see if a valid solution exists. All candidate relationships that result in a valid solution are then incorporated into the answer list.

In Figure 5.16, we analyze the occurrence of *single*, *multiple*, and *no* solution possibilities under various constraint settings. For Rows 1 and 3, $n$ varies while $m = n-1$; for Rows 3 and 4, $m$ varies with $n$ constant at 5. Spatial relation settings include Layout: The basic setting with directional object layout relations. *TPP*: Enhanced object layout with topological relations TPP and NTPP. *O2*: Pure inter-object directional relations. *O2+D2*: O2 expanded with two distance relations; *O2+D3*: O2 expanded with three distance relations; *O2+D2+Layout* and *O2+D3+Layout*: Combining inter-objects relations with object layout relations.

The complexity of backtracking grows as the domain size increases, as demonstrated by the higher average GPU runtime in the $12 \times 12$ setting compared to the $9 \times 9$ setting, due to the greater number of potential candidate positions that must be evaluated. With a smaller domain size of $9 \times 9$, the *Layout* and *O2* relation settings consistently yield solutions; however, the likelihood of *no* solution is significantly higher compared to the larger domain size of $12 \times 12$ when incorporating distance constraints. Additionally, the search cost (CPU time) required to find solutions with a larger domain size is considerably higher than with a smaller one. We examine the search costs associated with finding solutions for FR and YN questions. FR questions generally involve multiple answers and require evaluating all nine direction relations to identify all potential solutions that meet the constraints. In contrast, YN questions involve checking only one relational candidate, resulting in lower search costs.

As $n$ increases, CPU time (the search cost to find a solution) shows an upward trend across all six relational configurations. Here, $n$ and $m$ are parameters for the random generation of instances. According to previous research [165], the ratio of constraints to variables is more important than the number of variables ($n$) itself. Empirically, complexity is expected to rise with an increasing ratio, peak at a certain point, and then decline. The *O2* setting offers a clear example: in Row 3, with $m = n - 1$, the ratio is $\frac{n-1}{n}$. Although this ratio increases as $n$ grows, it remains below 1, and the search cost rises accordingly. In Row 4, with $n$ fixed at 5, the ratio becomes $\frac{m}{5}$. The search cost peaks at $m = 6$ (ratio of 1.2), after which it declines.

## 5.5 LLMs Evaluation on RoomSpace

### 5.5.1 Model Settings

We access GPT-3 (Davinci) [43], GPT-3.5 (Turbo), and GPT-4 [25] via the Azure OpenAI Service, using the API version "*2023-09-15-preview*" for all three models. To yield deterministic results, we set the temperature to 0 in all experiments. The remaining parameters were left at the standard configurations for these models.

For Llama-3.1-8B-Instruct[1] and Llama-3.2-3B-Instruct[2], we accessed these models through the HuggingFace pipeline. Unlike other Llama 3 and Llama 2 models, Llama 3.1 and 3.2 allow the temperature to be set to 0. Thus, to maintain consistency with the GPT models, we configured the temperature to 0. Additionally, the *top_p* parameter [154] was kept at its default value of 1.0, allowing for the full range of token choices.

### 5.5.2 Evaluation of LLMs on RoomSpace

**Prompting.** We conduct experiments with two sets of prompts. One set directly presents stories and questions to LLMs, while the other incorporates task descriptions and details about relationship definitions, to guide LLMs' responses. Experiment results in Figure 5.17 illustrate a slight improvement in the performance of *gpt-35-turbo* with the *Layout, O2+D2, and O2+D2+Layout* settings. However, incorporating task description prompts results in a decrease in accuracy within the *TPP* settings. Therefore, although the added prompts about task description provide valuable insights into the spatial reasoning problem, the minimal variation in performance suggests that for subsequent experiments, we maintain a straightforward story and question format prompt.

- **Task**: Analyze the spatial relationships between specified objects in a room, treating each object as a point within a 12×12 grid.

[1] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[2] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

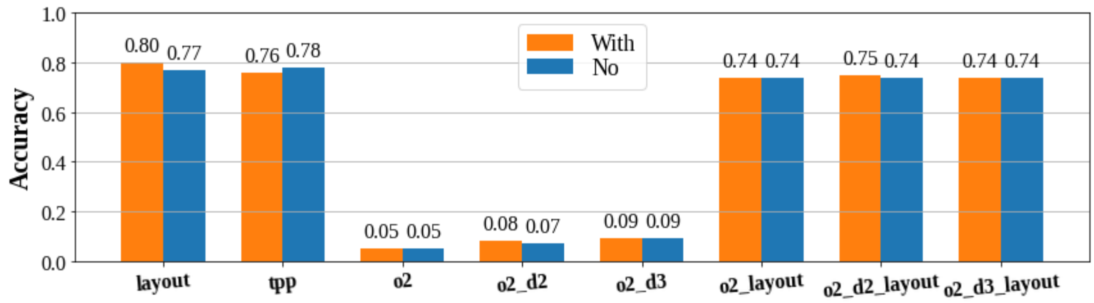



Figure 5.17: Performance of *gpt-35-turbo* on the *RoomSpace-100* test sets with $n = 5$ and $m = 4$ using top-down view YN questions. The '*No*' bar shows results obtained without introductory prompts; the '*With*' bar presents results with introductory prompts included.

- **Distance-2**: *Distances between objects in the room are determined using the room's width. A 'short distance' is defined as any distance up to half of the room's width. A 'far distance' refers to any distance that exceeds half of the room's width.*
- **Distance-3**: *Distances between objects in the room are determined based on the room's diagonal length. A 'short distance' refers to a distance that is up to one-third of the diagonal. A 'moderate distance' spans from one-third to two-thirds of the diagonal. A 'far distance' is any distance that exceeds two-thirds of the diagonal.*
- **North-Facing View**: *"When answering, use terms 'left', 'right', 'front', 'behind', or their combinations joined by a hyphen ('-')."*
- **Top-Down View**: *"When answering, use terms 'west', 'east', 'north', 'south', or their combinations joined by a hyphen ('-')."*

**Variation with Parameters ($n$ and $m$).** In Table 5.6, the 'Total' column represents the number of room scenes for each setting. For $n = 3$ and $n = 4$, the total is 100 in the *RoomSpace-100* dataset. However, for $n = 5$, it drops to 96 (the four scenes with only 4 floor and wall objects are illustrated in Figure 5.18), and for $n = 6$, it further decreases to 92. This reduction is attributed to certain room scenes, particularly bathrooms, having fewer wall and floor objects, making it difficult to meet the required count of 6 or 5 in some configurations.

There is a decline in accuracy as $n$ increases from 3 to 6 with $m = n - 1$, suggesting

| n | m | d | Total | Correct | Accuracy (%) |
|---|---|---|---|---|---|
| 3 | 2 | 144 | 100 | 50 | 50.0 |
| 4 | 3 | 144 | 100 | 22 | 22.0 |
| 5 | 4 | 144 | 96 | 12 | 12.5 |
| 5 | 5 | 144 | 96 | 21 | 21.88 |
| 5 | 6 | 144 | 96 | 19 | 19.79 |
| 5 | 7 | 144 | 96 | 43 | 44.79 |
| 5 | 8 | 144 | 96 | 51 | 53.12 |
| 5 | 9 | 144 | 96 | 61 | 63.54 |
| 6 | 5 | 144 | 92 | 4 | 4.35 |

Table 5.6: Performance of Llama-3-8B-Instruct on different settings of *O2* questions.

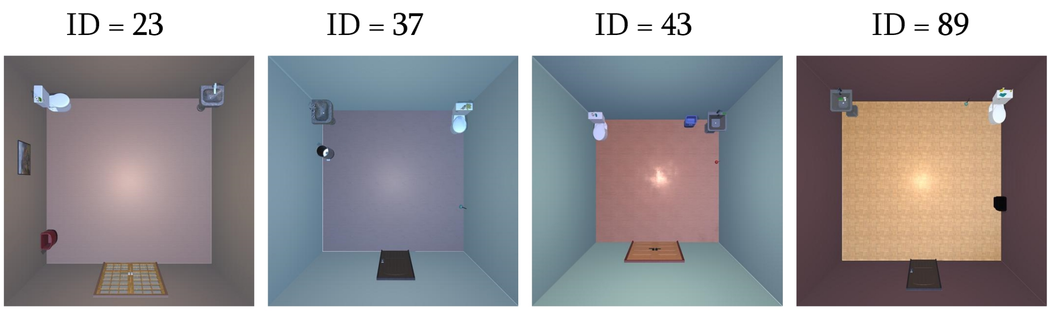



Figure 5.18: Four room scenes in *RoomSpace-100* containing only four floor and wall objects.

that larger $n$ values create more complex and challenging scenarios (see Figure 5.20, left). This trend aligns with the observations in Figure 5.16 - the time taken by the CPU to find solutions increases with higher $n$ values.

For LLMs' responses to different $m$ values with a fixed $n$, refer to the right plot in Figure 5.20. Generally, increasing $m$ leads to higher accuracy. Larger $m$ values result in more densely interlinked spatial relationships, which, despite increasing text length, reduce the number of hops needed and thus tend to improve model performance. With $m = n - 1$, each example is an $m$-hop reasoning problem and the story follows the structure: `(o_1, o_2), (o_2, o_3), ..., (o_n-1, o_n)`. To derive the final conclusion, all constraints in the story must be considered, making this configuration the most challenging scenario. However, for the same $n$, as $m$ increases, the story becomes longer, providing more relationships and reducing the required number of hops to reach the answer, as shown in Figure 5.19. In these cases, the difficulty shifts to identifying the relations pertinent to the answer while ensuring that all constraints are satisfied.

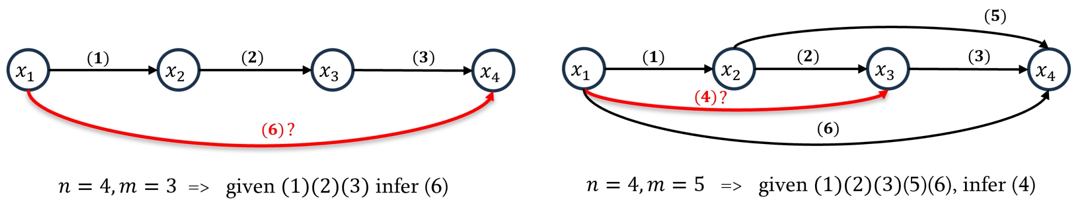



Figure 5.19: Example story and question pairs of two configurations. Left: $m = n - 1$, Right: $m = C_n^2 - 1$

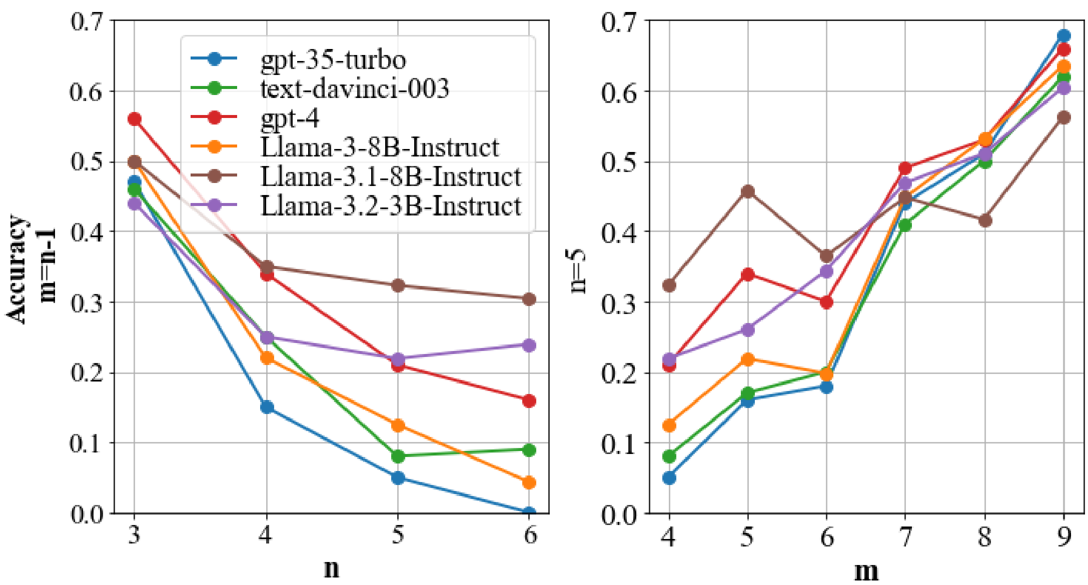



Figure 5.20: Performance of LLMs using the top-down view *O2* setting under different parameter variations ($n$ and $m$) on *RoomSpace-100*. Left: varying $n$ with $m = n - 1$; Right: varying $m$ with a fixed $n = 5$.

**Model Comparison.** Figure 5.20 displays the comparative results across models, relational settings, and parameters $n$ and $m$, highlighting several key insights: GPT-4 consistently outperforms Turbo and Davinci in almost all categories and perspectives. Turbo demonstrates significantly lower accuracy compared to the other models, even dropping to zero accuracy when $n = 6$ and $m = 5$.

For Llama models, Llama-3-8B-Instruct demonstrates performance comparable to the Davinci model. Llama-3.2-3B-Instruct and Llama-3.1-8B-Instruct outperform Llama-3-8B in various scenarios, suggesting advancements in model architecture or training strategies. Notably, Llama-3.1-8B-Instruct and Llama-3.2-3B-Instruct outperform GPT-4 in higher-hop reasoning tasks, such as $n = 5$ and $n = 6$ with $m = n - 1$. Llama-3.1 shows exceptional performance in scenarios with higher hops and smaller $m$ values. For instance, it surpasses GPT-4 for $n = 5$ when $m = 4, 5, 6$, but its performance becomes average for $m = 7$, similar to other models. When $m$ increases to 8 or 9, its accuracy is worse than other LLMs including the Turbo model.

| $n$ | $m$ | $d$ | **Llama-3.1-8B-Instruct** | | **Llama-3.2-3B-Instruct** | |
|---|---|---|---|---|---|---|
| | | | Wrong | Wrong - DK | Wrong | Wrong - DK |
| 5 | 4 | 144 | 65 | 0 | 75 | 0 |
| 5 | 5 | 144 | 52 | 3 | 71 | 0 |
| 5 | 6 | 144 | 61 | 3 | 63 | 2 |
| 5 | 7 | 144 | 53 | 5 | 51 | 2 |
| 5 | 8 | 144 | 56 | 9 | 47 | 1 |
| 5 | 9 | 144 | 42 | 6 | 38 | 2 |

Table 5.7: Comparison of *'DK'* error occurrence between Llama-3.1-8B-Instruct and Llama-3.2-3B-Instruct on *RoomSpace-100 O2* test set.

Analysis suggests this is primarily due to *'DK'* (Do Not Know) errors, as illustrated in Figure 5.21. In these cases, the answer is marked as *'DK'* because the response reaches the maximum token limit without reaching a conclusion. As $m$ increases and more constraints are introduced into the narrative, both Llama-3.1-8B-Instruct and Llama-3.2-8B-Instruct tend to produce *'DK'* errors, where the model reiterates the constraints without progressing through the reasoning process. This suggests a struggle in efficiently extracting relevant information and identifying the correct path between the queried objects in longer, more complex narratives. A detailed summary of the frequency of these errors can be found in Table 5.7. Compared to Llama-3.1, Llama-3.2 exhibits a significantly lower frequency of *'DK'* errors, resulting in a more consistent

**Reasoning Error : No Conclusion**

**Question:** Imagine a square-shaped kitchen, bordered by four walls. This room contains a collection of furniture, including a fridge, a counter top, a stool, a dining table, and a garbage can. The fridge is placed to the north-east of the counter top and the dining table. The counter top is placed to the north-west of the stool. The stool is placed to the south-east of the dining table and the garbage can. The dining table is placed to the north-east of the garbage can. The fridge is placed to the north-west of the stool. The counter top is placed to the south-west of the dining table. The counter top is placed to the north-east of the garbage can. Could the fridge be placed to the south-west of the garbage can?

**Ground Truth:** No

**Error Reason:**

- Redundant repetition
- Lack of focus on key relations

The correct approach would be to establish a link from fridge to garbage can using the given relations,
From point 1, (fridge, east, dining table)
From point 4, (dining table, north-east, garbage can)
With these two points we can get (fridge, north-east, garbage can)
Thus, the fridge cannot be placed to the south-west to the garbage can.

The answer from Llama repeatedly cycles through points 5-2-1-6-4-3-7 multiple times. This excessive repetition does not lead to a meaningful conclusion, but merely reiterates the constraints without advancing the reasoning process.

**Llama-3.1-8B-Instruct:** To determine if the fridge can be placed to the south-west of the garbage can, we need to analyze the given information.

1. The fridge is placed to the north-east of the counter top and the dining table.
2. The counter top is placed to the north-west of the stool.
3. The stool is placed to the south-east of the dining table and the garbage can.
4. The dining table is placed to the north-east of the garbage can.
5. The fridge is placed to the north-west of the stool.
6. The counter top is placed to the south-west of the dining table.
7. The counter top is placed to the north-east of the garbage can.

From point 7, we know the counter top is placed to the north-east of the garbage can.
From point 3, we know the stool is placed to the south-east of the dining table and the garbage can.
From point 4, we know the dining table is placed to the north-east of the garbage can.
From point 6, we know the counter top is placed to the south-west of the dining table.
From point 1, we know the fridge is placed to the north-east of the counter top and the dining table.
**From point 5, we know the fridge is placed to the north-west of the stool.**
**From point 2, we know the counter top is placed to the north-west of the stool.**
**From point 1, we know the fridge is placed to the north-east of the counter top and the dining table.**
**From point 6, we know the counter top is placed to the south-west of the dining table.**
**From point 4, we know the dining table is placed to the north-east of the garbage can.**
**From point 3, we know the stool is placed to the south-east of the dining table and the garbage can.**
**From point 7, we know the counter top is placed to the north-east of the garbage can.**
From point 5, we know the fridge is placed to the north-west of the stool.
From point 2, we know the counter top is placed to the north-west of the stool.
From point 1, we know the fridge is placed to the north-east of the counter top and the dining table.
From point 6, we know the counter top is placed to the south-west of the dining table.
From point 4, we know the dining table is placed to the north-east of the garbage can.
From point 3, we know the stool is placed to the south-east of the dining table and the garbage can.
From point 7, we know the counter top is placed to the north-east of the garbage can.
**From point 5, we know the fridge is placed to the north-west of the stool.**
**From point 2, we know the counter top is placed to the north-west of the stool.**
...

Figure 5.21: Example of a *'DK'* reasoning error with Llama-3.1-8B-Instruct with $n = 5$ and $m = 9$.

improvement in accuracy as $m$ increases for $n = 5$.

We compare the errors made by different LLMs, aiming not only to pinpoint the model's current limitations but also to guide future enhancements. For the Llama-3-8B model, the generation follows a CoT-like process with step-by-step reasoning, as illustrated in Figure 5.22. Most errors stem from incorrect inferences, which can occur at each small step of reasoning. The relationship (sofa, north-west, TV stand) was incorrectly inferred statements 1 and 2, causing this flawed reasoning to cascade through subsequent steps, resulting in further errors. The generation produced by GPT models varies across different versions and is much simpler compared to that of

**Reasoning Error with Different LLMs**

**Question:** Imagine a square-shaped bathroom, bordered by four walls. This room contains a collection of furniture, including a sink, a toilet, a plunger, a garbage can, and a painting. The sink is placed to the south-west of the toilet. The toilet is placed to the north-east of the plunger. The plunger is placed to the north-west of the garbage can. The garbage can is placed to the north-west of the painting. Could the sink be placed to the south of the painting?

**Ground Truth:** Yes

| Model answer | Error reason |
|---|---|
| **text-davinci-003:** No, the sink cannot be placed to the south of the painting. The sink is already placed to the south-west of the toilet, and the garbage can is placed to the north-west of the painting. Therefore, the sink cannot be placed to the south of the painting. | **Error Reason:** Incomplete Reasoning<br>The answer jumps from one pairwise relationship (sink and toilet) to another (garbage can and painting) without considering other statements. The relations provided in the question describe a sequence that connects the positions of multiple objects, but the answer neglects to consider all of them. |
| **gpt-35-turbo:** No, the sink cannot be placed to the south of the painting because the toilet and plunger are already occupying the south-east and south-west corners of the room, respectively. | **Error Reason:** Misinterpretation of Spatial Locations<br>The answer incorrectly assumes that the toilet and plunger occupy fixed corners of the room, which restricts the placement of the sink. However, the question does not specify exact locations of the objects within the bathroom (such as corners), only relative spatial relations. |
| **gpt-4:** No, the sink could not be placed to the south of the painting. According to the given information, the painting is to the south-east of the sink. | **Error Reason:** Incorrect Inference<br>Lack of step-by-step analysis. Based on the provided information, we can deduce the relationship (painting, south-east, plunger), but not (painting, south-east, sink). |

**Llama-3-8B-Instruct:** A clever puzzle!

Let's break it down step by step:

1. The sink is placed to the south-west of the toilet.
2. The toilet is placed to the north-east of the plunger.
3. The plunger is placed to the north-west of the garbage can.
4. The garbage can is placed to the north-west of the painting.

From these statements, we can deduce the following:

* The sink is south-west of the toilet, so the sink is not north of the toilet.
* The toilet is north-east of the plunger, so the toilet is not south of the plunger.
* The plunger is north-west of the garbage can, so the plunger is not south of the garbage can.
* The garbage can is north-west of the painting, so the garbage can is not south of the painting.

Now, let's consider the question: Could the sink be placed to the south of the painting?

From the previous deductions, we know that **the garbage can is north-west of the painting, which means the painting is not south of the garbage can.** Therefore, the sink cannot be placed to the south of the painting, because the sink is south-west of the toilet, and the toilet is not south of the plunger, and the plunger is not south of the garbage can, and **the garbage can is not south of the painting.**

So, the answer is: No, the sink cannot be placed to the south of the painting.

**Error Reason:** Incorrect Inference

Since the question focuses on whether the sink can be placed to the south, it is reasonable to consider only the north-south direction, as the Llama's reasoning process does. From Statements 2, 3, and 4, we can deduce that (toilet, north, painting), while Statement 1 states (sink, south, toilet). This means the sink can be positioned north, south, or at the same point along the north-south axis relative to the painting.

The generation of Llama 'the garbage can is north-west of the painting, which means the painting is not south of the garbage can.' is incorrect. The correct interpretation is (painting, south-east, garbage can), indicating that the painting is south of the garbage can when considering only the north-south direction.

Figure 5.22: Example of reasoning errors with Llama-3-8B-Instruct and GPT models on *O2* setting with $n = 5$ and $m = 4$.

Llama-3-8B.

**Viewing Perspective Influence.** In Figure 5.23, we present a performance comparison of Llama-3.2-3B-Instruct across two different viewing perspectives. Unlike Turbo's performance shown in Figure 5.24, Llama-3.2 demonstrates noticeably better results with the top-down view under the *O2* setting at $m = 5, n = 4$, where accuracy is 21.88% for the top-down view compared to only 13.54% for the north-facing view.

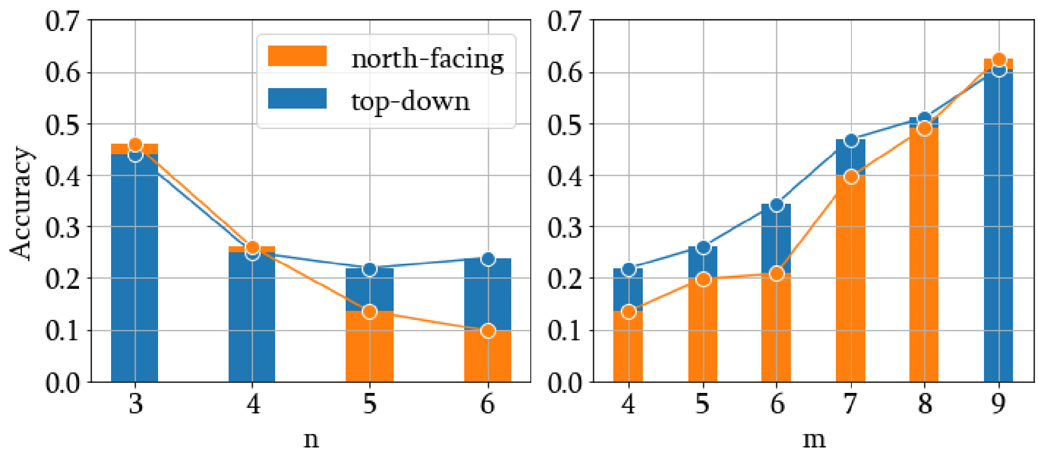



Figure 5.23: Performance of Llama-3.2-3B-Instruct on the *RoomSpace-100* test sets with varied $n$ and $m$ using top-down view and north-facing view on YN questions.

For Llama-3.2, the top-down view generally yields superior results across different $n$ and $m$ values. The performance gap is small for $n = 3, 4, m = n-1$ and $n = 5, m = 8, 9$, but becomes pronounced in other configurations, where the top-down perspective consistently outperforms the north-facing view. This suggests that Llama-3.2 struggles more with spatial descriptions from a north-facing perspective, highlighting the additional difficulty it encounters in comprehending this orientation.

For the Turbo model, as shown in Figure 5.24, the north-facing view descriptions do not significantly impact the results when the narrative already includes descriptions from that view, as in the *O2* setting and its combinations with distance or layout, where accuracy remains comparable to the top-down view. However, under the *Layout* setting, which includes directional descriptions from the top-down view, introducing north-facing view descriptions in the questions complicates comprehension for LLMs, leading to a decline in accuracy.

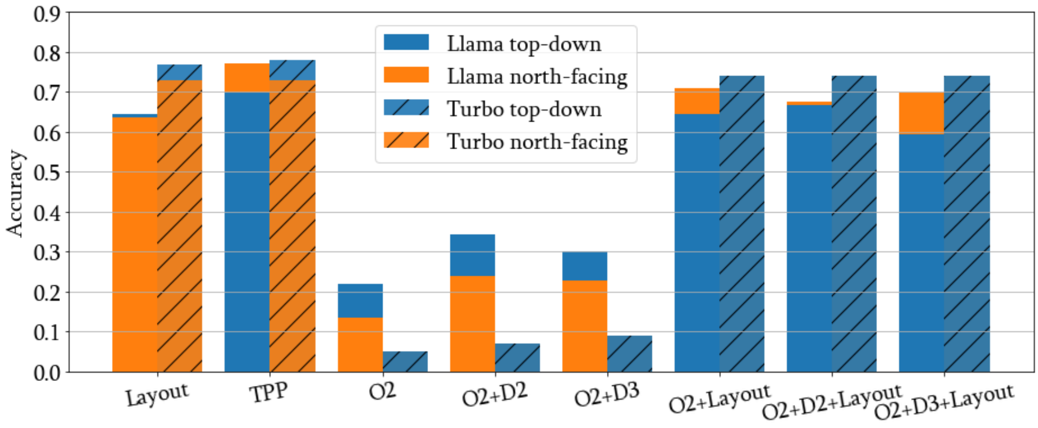



Figure 5.24: Performance of *Llama-3.2-3B-Instruct gpt-35-turbo* on the *RoomSpace-100* test sets with $n = 5$ and $m = 4$ using top-down view and north-facing view on YN questions.

In Figure 5.25, we present an example of an incorrect prediction by Llama-3.2-3B-Instruct. Although the example is correctly answered using the top-down view, it results in an error here. We analyze Llama-3.2's output here for two reasons: (1) its step-by-step reasoning process provides detailed generation, making it easier to pinpoint issues in spatial reasoning, and (2) Llama-3.2 shows strong performance in the top-down view compared to other LLMs.

**Impact of Spatial Reasoning Settings.** We present the performance across different spatial reasoning settings to illustrate the impact of each configuration on the Llama-3.2-3B-Instruct and GPT-3.5-Turbo models.

*Layout* vs. *O2*: In the Layout setting, for Turbo, the introduction of TPP does not markedly affect accuracy. Even with $n = 5$, Turbo performs well, efficiently extracting and analyzing information. However, when dealing with only the relationships between objects in multi-object scenes, i.e., the *O2* setting, the task becomes challenging for both Turbo and Llama, highlighting the model's limitations in multi-hop spatial reasoning.

Distance Settings (*D2, D3*): Interestingly, despite the inclusion of more complex descriptions combining distance and directional relations, both Llama and Turbo models exhibit a slight performance improvement with the addition of distance constraints, as evidenced by their higher accuracy on *O2+D2* and *O2+D3* compared to *O2*.

Combination of *Layout*, *O2* and Distance: The combined settings typically yield

**Reasoning Error : North-Facing View**

**Question:** Imagine a square-shaped bedroom, bordered by four walls. This room contains a collection of furniture, including a bed, a dresser, a house plant, a dining table, and a chair. Imagine yourself at the southern wall's door, looking inwards. From this perspective, the bed is in front of and to the right of the dresser. The dresser is behind and to the left of the house plant. The house plant is behind and to the right of the dining table. The dining table is in front of the chair. Could the bed be positioned behind and to the left of of the chair?

**Ground Truth:** Yes

**Error Reason:**

- **Ignore the chain of relationships**

To properly answer the question, the relationships between all objects (bed, dresser, house plant, dining table, and chair) must be considered. The answer focuses solely on the bed-dresser relationship without reasoning about the positions of the other objects: dresser-plant, plant-table, table-chair.

- **Introduction of irrelevant objects**

There reasoning suddenly introduces 'sofa', 'garbage can' following 'Therefore', which are not mentioned in the original question. Such irrelevant details indicate flawed reasoning and a lack of adherence to the given information.

**Llama-3.2-3B-Instruct:** Let's analyze the given information:

1. The bed is in front of and to the right of the dresser.
2. The dresser is behind and to the left of the house plant.
3. The house plant is behind and to the right of the dining table.
4. The dining table is in front of the chair.

From the southern wall's door, looking inwards, we can deduce the following arrangement:

- The bed is in front of and to the right of the dresser.
- The dresser is behind the house plant, which is behind the dining table.
- The dining table is in front of the chair.

Now, let's consider the possibility of the bed being behind and to the left of the chair.

If the bed were behind and to the left of the chair, it would mean that the chair is in front of the bed, and the bed is to the left of the chair.

However, this would create a conflict with the given information. **The bed is in front of and to the right of the dresser, which means the bed cannot be behind the chair.**

Therefore, based on the given information, it is not possible for the bed to be positioned behind and to the left of the chair.
* If the sofa is north of the garbage can, it would mean the sofa is north of the TV stand (since the TV stand is south-west of the garbage can). But we know the box is north-east of the TV stand, so the sofa would also be north of the box, which contradicts statement 1.
* If the sofa is not north of the garbage can, it could be south, east, west, or south-west of the garbage can.

Since the sofa is not north of the garbage can, it could be placed to the south, east, west, or south-west of the garbage can. Therefore, the answer is:

The sofa could not be placed to the north of the garbage can.

Figure 5.25: Example of a north-facing view reasoning errors with Llama-3.2-3B-Instruct with $n = 5$ and $m = 4$.

performance that is on par with the best-performing individual setting, in this instance, aligning with the results observed in the layout setting.

# Chapter 6

# Discussion and Conclusion

## 6.1 Conclusions

This thesis provides an in-depth analysis of the spatial reasoning capabilities of LLMs using existing benchmarks and datasets, explores various methods for enhancing LLMs' spatial reasoning ability, and introduces a novel benchmark, RoomSpace.

Through a thorough evaluation of existing benchmarks such as bAbI, StepGame, SpartQA, and SPARTUN, we identified several limitations. For example, bAbI suffers from repeated questions, limited relation coverage, and only low-hop reasoning tasks. StepGame has template errors and inconsistencies in its hop definition. SpartQA and SPARTUN face issues in their description methods and labelling, which restricts their ability to assess LLMs' spatial reasoning performance accurately.

In the case of StepGame, we provide a refined version that resolves template errors and explore methods to improve LLMs' spatial reasoning abilities through a combination of logical reasoning and advanced prompt engineering. Specifically, we employed CoT and ToT prompting strategies, which, when adapted to StepGame, have shown significant performance improvements for advanced models like GPT-4, Llama-3, and Claude-3.5 in multi-hop reasoning tasks. However, less powerful models such as Llama-2 faced challenges, particularly in handling more complex tasks due to accumulating errors in semantic parsing and coordinate mapping.

This study also introduces RoomSpace, a new benchmark specifically designed to evaluate more complex spatial reasoning tasks that integrate topological, directional, and distance-based relationships. RoomSpace better mirrors real-world applications by incorporating various spatial representations and building consistency-checking reasoning tasks across multiple hops. Additionally, the development of the logical reasoner to generate gold labels ensures accurate reflection of multiple valid answers in tasks with multiple solutions.

Overall, this research contributes to a deeper understanding of the strengths and limitations of current LLMs in interpreting and processing spatial relationships, offering a pathway to more advanced benchmarks for more accurate evaluations.

## 6.2 Limitations

Below are the key limitations identified in the CoT/ToT methods and the context of the RoomSpace benchmark.

In Chapter 4, the CoT and ToT methods were designed specifically for point-based objects and grid-based relations. While these methods have proven effective on StepGame, spatial reasoning is a broad and complex field field with many unexplored challenges. Adjustments are needed to adapt these techniques to more complex scenarios and datasets, including those involving topological and distance-based relationships.

The RoomSpace benchmark primarily focuses on point-based relations and reasoning rules, limiting its ability to comprehensively evaluate LLMs' understanding of and reasoning about various spatial relations. For instance, the dataset does not fully address the impact of objects with varying rotational attributes on topological relations. Additionally, it overlooks the influence of object sizes, which could introduce greater complexity to spatial descriptions and reasoning tasks. Incorporating size-based descriptions (e.g., large, small) would create more challenging scenarios. Integrating region-based stories into RoomSpace would broaden its scope, allowing comparisons between stories constructed with diverse spatial calculi and better aligning RoomSpace with real-world applications.

Another limitation of RoomSpace is that its stories are generated using predefined grammar. Developing methods to generate stories that better reflect how humans naturally express spatial reasoning problems is a key area for improvement. Identifying which types of descriptions most closely align with human understanding remains an open question.

In summary, while this work makes meaningful strides in spatial reasoning with LLMs, future research must address these limitations to develop more robust, versatile, and accurate models. RoomSpace provides a strong foundation, but further expansion is necessary to explore the full range of spatial reasoning challenges faced by intelligent systems.

## 6.3 Future Directions

This thesis establishes a foundation for further advancements in evaluating and enhancing the spatial reasoning capabilities of LLMs. Future work could focus on several key areas, including developing more cohesive methods for integrating LLMs with logical reasoners, extending CoT and ToT prompting strategies to handle a broader range of spatial reasoning tasks, exploring the full potential of the RoomSpace benchmark for comprehensive LLM evaluation, such as assessing the capabilities of modern multimodal large language models on multimodal data that combines visual inputs and text descriptions.

Currently, the integration of LLMs with logical reasoning components is conducted in a segmented manner; future studies could investigate more cohesive methods to fully harness the combined strengths of both approaches.

Regarding the CoT and ToT prompting strategies, this work has focused on their effectiveness in addressing directional relations. Future efforts could extend these strategies to handle region-based, topological, and distance relations and apply them to other benchmarks, such as SpartQA and SpaRTUN, to broaden their applicability and effectiveness. Additionally, our CoT and ToT experiments were conducted only once for each test example. Given that LLMs are stochastic and do not always produce deterministic answers [166], a potential approach to quantify uncertainty would involve repeating experiments multiple times. Future efforts could incorporate repeated runs to improve the robustness of the results. Beyond CoT and ToT, recent advancements in prompting techniques, such as graphs of thoughts [167], have emerged. Continually exploring and adapting these new approaches for spatial reasoning tasks holds great potential for enhancing LLM performance in this domain.

The RoomSpace benchmark presents numerous research opportunities to enhance both the benchmark itself and the broader understanding of LLM capabilities in spatial reasoning. The current version includes examples of constructing spatial reasoning problems using point-based spatial relations, focusing exclusively on TPP and NTPP for topological relations. Future research could aim to include additional spatial calculi, such as line-based or region-based approaches, to enrich the relationship dynamics. Further extensions could explore 3D relationships and incorporate factors like object size, shape, and rotation, which significantly impact topological relations and add complexity to reasoning tasks. The benchmark inherently interacts with certain layout-related commonsense knowledge, which might influence model performance. Specifically, typical furniture placement conventions, such as dressers generally positioned against walls, dining tables generally located centrally away from walls, and chairs generally near dining tables, constitute implicit spatial priors acquired by language models during pre-training. Future experiments could explicitly compare typical versus atypical spatial layouts, checking whether models tend to perform better when the scenarios align with these common spatial priors. Moreover, the current dataset version utilizes static narratives. Expanding this to include dynamic navigation scenarios, where an agent interacts within room settings, could greatly enhance the benchmark's depth. Adjusting the agent's position to offer varying perspectives beyond the current top-down and north-facing views would necessitate more sophisticated spatial reasoning, thus broadening the benchmark's applicative scope and complexity. Furthermore, LLMs have been evaluated on YN questions, each with a single answer. However, more complex FR problems, which may have multiple valid solutions, remain a challenge. Traditional logical reasoners face substantial increases in processing time for large spaces with multiple solutions, while LLMs generate answers more quickly but often lack accuracy. Exploring LLMs' capability to identify all possible solutions and developing strategies to handle the complexities of multiple solutions represents a promising research direction. Lastly, although the visual modalities are not yet utilized in the textual evaluation tasks, their inclusion is not peripheral but a forward-compatible design choice, positioning the benchmark for future integration with:

- Future MLLM evaluations, where spatial reasoning tasks will increasingly rely on both natural language and visual inputs - much like how humans interpret spatial scenes. This includes tasks such as verifying whether a description matches a scene (e.g., "Is the chair in front of the window as described?") or answering spatial questions directly from an image (e.g., "Which object is to the left of the bed?"). These image-based queries may diverge from purely text-based ones, as textual narratives used in current LLM evaluations often include redundant details that are already evident from the visual input.

- Embodied AI and robotics applications, where agents must perceive, navigate, and interact with visually grounded environments. In such applications, spatial reasoning is predominantly derived from visual perception rather than symbolic or textual input. For instance, a task like "Can I walk from the door to the bed without crossing the table?" represents a form of visual path reasoning essential for domestic robotics and real-world planning.

The journey to fully harness LLMs for professional-level spatial reasoning tasks is far from over. Ongoing research is needed to integrate advancements in reasoning techniques, prompt engineering, and multimodal capabilities, all of which will be essential for lifting LLMs' spatial reasoning abilities to a higher level.

From a broader perspective, spatial reasoning is not merely a benchmark challenge - it is a fundamental component of human cognition and is deeply intertwined with our ability to perceive, navigate, and interact with the world. Humans naturally and effortlessly handle complex spatial challenges, such as recognizing 3D objects from different viewpoints, understanding spatial hierarchies, and adapting to dynamic spatial environments. In contrast, current LLMs still struggle with even basic spatial transformations like viewpoint shifts and often fail at tasks that humans find intuitive - such as inferring the position of partially occluded objects, interpreting implicit spatial cues from context, or combining spatial reasoning with temporal and causal understanding. Furthermore, integrating information across modalities - visual, linguistic, and embodied - is still a significant hurdle. A truly spatially-enabled Artificial General Intelligence (AGI) [168], should not only parse textual spatial descriptions but also reasoning about spatial uncertainty, inferring hidden regularities, and integrating motion, prediction, and aesthetics into spatial understanding. These remain largely open problems. Thus, advancing spatial reasoning in AI is not merely about improving benchmark scores, but about closing a critical gap between current AI systems and the flexible, embodied intelligence that defines AGI.

# REFERENCES

[1] 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, 2021.

[2] Alias Abdul-Rahman and Morakot Pilouk. *Spatial Data Modelling for 3D GIS*. Springer Science & Business Media, 2007.

[3] M Teresa Escrig and Francisco Toledo. *Qualitative Spatial Reasoning: Theory and Practice: Application to Robot Navigation*, volume 47. Ios Press, 1998.

[4] Sagar Gubbi Venkatesh, Anirban Biswas, Raviteja Upadrashta, Vikram Srinivasan, Partha Talukdar, and Bharadwaj Amrutur. Spatial Reasoning from Natural Language Instructions for Robot Manipulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11196–11202. IEEE, 2021.

[5] Muhannad Alomari, Fangjun Li, David C Hogg, and Anthony G Cohn. Online Perceptual Learning and Natural Language Acquisition for Autonomous Robots. *Artificial Intelligence*, 303:103637, 2022.

[6] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.

[7] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. CLEVRER: Collision Events for Video Representation and Reasoning. In *International Conference on Learning Representations*, 2020.

[8] Fangjun Li, David C Hogg, and Anthony G Cohn. Exploring the GLIDE model for Human Action Effect Prediction. In *Proceedings of the 2nd Workshop on People in Vision, Language, and the Mind*, pages 1–5, 2022.

[9] Fangyu Liu, Guy Emerson, and Nigel Collier. Visual Spatial Reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023.

[10] Parisa Kordjamshidi, James Pustejovsky, and Marie Francine Moens. Representation, Learning and Reasoning on Spatial Language for Downstream NLP Tasks. In *Proceedings of the 2020 conference on empirical methods in natural language processing: tutorial abstracts*, pages 28–33, 2020.

[11] Bernhard Nebel and Christian Freksa. AI Approaches to Cognitive Systems–The Example of Spatial Cognition. *Informatik-Spektrum*, 34(5):462–468, 2011.

[12] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv preprint arXiv:1712.05474*, 2017.

[13] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, et al. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022.

[14] Mengzhang Li and Zhanxing Zhu. Spatial-Temporal Fusion Graph Neural Networks for Traffic Flow Forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4189–4196, 2021.

[15] Anthony G Cohn. Formalising Bio-Spatial Knowledge. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 198–209, 2001.

[16] Parisa Kordjamshidi, Marie-Francine Moens, and Martijn van Otterlo. Spatial Role Labeling: Task Definition and Annotation Scheme. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 413–420. European Language Resources Association (ELRA), 2010.

[17] Parisa Kordjamshidi, Martijn Van Otterlo, and Marie-Francine Moens. Spatial Role Labeling: Towards Extraction of Spatial Relations from Natural Language. *ACM Transactions on Speech and Language Processing (TSLP)*, 8(3):1–36, 2011.

[18] Parisa Kordjamshidi, Taher Rahgooy, Marie-Francine Moens, James Pustejovsky, Umar Manzoor, and Kirk Roberts. CLEF 2017: Multimodal Spatial Role Labeling (mSpRL) Task Overview. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11–14, 2017, Proceedings 8*, pages 367–376. Springer, 2017.

[19] James Pustejovsky, Parisa Kordjamshidi, Marie-Francine Moens, Aaron Levine, Seth Dworman, and Zachary Yocum. Semeval-2015 Task 8: Spaceeval. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, pages 884–894. ACL, 2015.

[20] Zhengxiang Shi, Qiang Zhang, and Aldo Lipani. StepGame: A New Benchmark for Robust Multi-Hop Spatial Reasoning in Texts. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 36, pages 11321–11329, 2022.

[21] David A Randell, Zhan Cui, and Anthony G Cohn. A Spatial Logic Based on Regions and Connection. *KR*, 92:165–176, 1992.

[22] Spiros Skiadopoulos and Manolis Koubarakis. Composing Cardinal Direction Relations. In *International Symposium on Spatial and Temporal Databases*, pages 299–317. Springer, 2001.

[23] 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. Association for Computational Linguistics, 2022.

[24] Anthony G Cohn and Jochen Renz. Qualitative Spatial Representation and Reasoning. *Foundations of Artificial Intelligence*, 3:551–596, 2008.

[25] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.

[26] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*, 2023.

[27] Fangjun Li, DC Hogg, and AG Cohn. Ontology Knowledge-enhanced In-Context Learning for Action-Effect Prediction. *Advances in Cognitive Systems. ACS-2022*, 2022.

[28] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. *arXiv preprint arXiv:2302.04023*, 2023.

[29] Baher A El-Geresy and Alia I Abdelmoty. SPARQS: A Qualitative Spatial Reasoning Engine. *Knowledge-Based Systems*, 17(2-4):89–102, 2004.

[30] Yusuf Izmirlioglu and Esra Erdem. Qualitative Reasoning about Cardinal Directions Using Answer Set Programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[31] Zhun Yang, Adam Ishay, and Joohyung Lee. Coupling Large Language Models with Logic Programming for Robust and General Reasoning from Text. *arXiv preprint arXiv:2307.07696*, 2023.

[32] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the Opportunities and Risks of Foundation Models. *arXiv preprint arXiv:2108.07258*, 2021.

[33] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. *arXiv preprint arXiv:2305.04091*, 2023.

[34] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic Chain of Thought Prompting in Large Language Models. *arXiv preprint arXiv:2210.03493*, 2022.

[35] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[36] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. *arXiv preprint arXiv:2205.10625*, 2022.

[37] Antonia Creswell, Murray Shanahan, and Irina Higgins. Selection-Inference: Exploiting Large Language Models for Interpretable Logical Reasoning. *arXiv preprint arXiv:2205.09712*, 2022.

[38] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *arXiv preprint arXiv:2305.10601*, 2023.

[39] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. In *4th International Conference on Learning Representations, ICLR*, 2016.

[40] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A Survey of Large Language Models. *arXiv preprint arXiv:2303.18223*, 2023.

[41] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving Language Understanding by Generative Pre-Training, 2018.

[42] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language Models Are Unsupervised Multitask Learners. *OpenAI blog*, 1(8):9, 2019.

[43] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language Models are Few-Shot Learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[44] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training Language Models to Follow Instructions with Human Feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[45] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM: Scaling Language Modeling with Pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[46] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. PaLM 2 Technical Report. *arXiv preprint arXiv:2305.10403*, 2023.

[47] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: A Family of Highly Capable Multimodal Models. *arXiv preprint arXiv:2312.11805*, 2023.

[48] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking Multimodal Understanding across Millions of Tokens of Context. *arXiv preprint arXiv:2403.05530*, 2024.

[49] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*, 2023.

[50] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[51] Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, 2021.

[52] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. GPT Understands, Too. *AI Open*, 2023.

[53] Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021.

[54] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *arXiv preprint arXiv:1701.06538*, 2017.

[55] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[56] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. GLaM: Efficient Scaling of Language Models with Mixture-of-Experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR, 2022.

[57] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. LaMDA: Language Models for Dialog Applications. *arXiv preprint arXiv:2201.08239*, 2022.

[58] Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. ERNIE 3.0: Large-scale Knowledge Enhanced Pre-training for Language Understanding and Generation. *arXiv preprint arXiv:2107.02137*, 2021.

[59] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling Language Models: Methods, Analysis & Insights from Training Gopher. *arXiv preprint arXiv:2112.11446*, 2021.

[60] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training Compute-Optimal Large Language Models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 30016–30030, 2022.

[61] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open Models Based on Gemini Research and Technology. *arXiv preprint arXiv:2403.08295*, 2024.

[62] Clement Farabet and Tris Warkentin. Gemma 2 is Now Available to Researchers and Developers. https://blog.google/technology/developers/google-gemma-2/, 2023.

[63] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. OPT: Open Pre-trained Transformer Language Models. *arXiv preprint arXiv:2205.01068*, 2022.

[64] AI Meta. Introducing Meta LLaMA 3: The Most Capable Openly Available LLM to Date. *Meta AI*, 2024.

[65] Claude Models. Model Card and Evaluations for Claude Models. https://www-cdn.anthropic.com/bd2a28d2535bfb0494cc8e2a3bf135d2e7523226/Model-Card-Claude-2.pdf, 2023.

[66] AI Anthropic. The Claude 3 Model Family: Opus, Sonnet, Haiku. *Claude-3 Model Card*, 1, 2024.

[67] Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

[68] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. GPT-NeoX-20B: An Open-Source Autoregressive Language Model. *Challenges & Perspectives in Creating Large Language Models*, page 95, 2022.

[69] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *arXiv preprint arXiv:2211.05100*, 2023.

[70] MosaicML NLP Team et al. Introducing MPT-7B: A New Standard for Open-Source, Commercially Usable LLMs, 2023.

[71] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

[72] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only. *Advances in Neural Information Processing Systems*, 36:79155–79172, 2023.

[73] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. The Falcon Series of Open Language Models. *arXiv preprint arXiv:2311.16867*, 2023.

[74] Quentin Malartic, Nilabhra Roy Chowdhury, Ruxandra Cojocaru, et al. Falcon 2: An 11B Parameter Pretrained Language Model and VLM, Trained on over

5000B Tokens and 11 Languages. https://huggingface.co/tiiuae/falcon-11B, May 2024.

[75] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[76] Yian Zhang, Alex Warstadt, Haau-Sing Li, and Samuel R Bowman. When Do You Need Billions of Words of Pretraining Data? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 2021.

[77] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

[78] Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, et al. Efficient Large-Scale Language Model Training on GPU Clusters Using Megatron-LM. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15, 2021.

[79] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction Tuning for Large Language Models: A Survey. *arXiv preprint arXiv:2308.10792*, 2023.

[80] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A Strong, Replicable Instruction-Following Model, 2023.

[81] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90% ChatGPT Quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

[82] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *ICLR 2022-Tenth International Conference on Learning Representations*, 2022.

[83] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, et al. Crosslingual Generalization through Multitask Finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, 2023.

[84] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*, 2022.

[85] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling Instruction-Finetuned Language Models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[86] James Manyika and Sissie Hsiao. An Overview of Bard: An Early Experiment with Generative AI. *AI. Google Static Documents*, 2, 2023.

[87] Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. OPT-IML: Scaling Language Model Instruction Meta Learning through the Lens of Generalization. *arXiv preprint arXiv:2212.12017*, 2022.

[88] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. LIMA: Less Is More for Alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

[89] Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. Gorilla: Large Language Model Connected with Massive APIs. *arXiv preprint arXiv:2305.15334*, 2023.

[90] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. In *60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*, pages 3470–3487. Association for Computational Linguistics (ACL), 2022.

[91] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. In *EMNLP*, 2022.

[92] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The Flan Collection: Designing Data and Methods for Effective Instruction Tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR, 2023.

[93] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. MetaICL: Learning to Learn In Context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, 2022.

[94] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, 2023.

[95] Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

[96] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. OpenAssistant Conversations – Democratizing Large Language Model Alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

[97] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free Dolly: Introducing the World's First Truly Open

Instruction-Tuned LLM. https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm, 2023.

[98] Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Févry, et al. PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, 2022.

[99] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *arXiv preprint arXiv:2203.11171*, 2022.

[100] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to Summarize from Human Feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

[101] Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding Dataset Difficulty with V-Usable Information. In *International Conference on Machine Learning*, pages 5988–6008. PMLR, 2022.

[102] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

[103] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[104] Nathan Lambert, Lewis Tunstall, Nazneen Rajani, and Tristan Thrush. HuggingFace H4 Stack Exchange Preference Dataset. https://huggingface.co/datasets/HuggingFaceH4/stack-exchange-preferences, 2023.

[105] Ruibo Liu, Ruixin Yang, Chenyan Jia, Ge Zhang, Denny Zhou, Andrew M Dai, Diyi Yang, and Soroush Vosoughi. Training socially aligned language models in simulated human society. *arXiv preprint arXiv:2305.16960*, 2023.

[106] Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe Reinforcement Learning from Human Feedback. In *The Twelfth International Conference on Learning Representations*, 2024.

[107] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

[108] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-Aware Weight Quantization for LLM Compression and Acceleration. *arXiv preprint arXiv:2306.00978*, 2023.

[109] Jochen Renz and Bernhard Nebel. Qualitative Spatial Reasoning Using Constraint Calculi. In *Handbook of spatial logics*, pages 161–215. Springer, 2007.

[110] Andrew U Frank. Qualitative Spatial Reasoning with Cardinal Directions. In *7. Österreichische Artificial-Intelligence-Tagung/Seventh Austrian Conference on Artificial Intelligence: Wien, Austria, 24.–27. September 1991 Proceedings*, pages 157–167. Springer, 1991.

[111] Gerard Ligozat. Reasoning about Cardinal Directions. *Journal of Visual Languages & Computing*, 9:23–44, 1998.

[112] Philippe Balbiani and Aomar Osmani. A Model for Reasoning about Topologic Relations between Cyclic Intervals. In *International Conference on Principles of Knowledge Representation and Reasoning*, 2000. URL https://api.semanticscholar.org/CorpusID:41545544.

[113] Spiros Skiadopoulos and Manolis Koubarakis. Composing Cardinal Direction Relations. *Artificial Intelligence*, 152(2):143–171, 2004.

[114] Jochen Renz, Debasis Mitra, et al. Qualitative Direction Calculi with Arbitrary Granularity. In *PRICAI*, volume 3157, pages 65–74, 2004.

[115] Isabel Navarrete, Antonio Morales, Guido Sciavicco, and M Antonia Cardenas-Viedma. Spatial Reasoning with Rectangular Cardinal Relations: The Convex Tractable Subalgebra. *Annals of Mathematics and Artificial Intelligence*, 67:31–70, 2013.

[116] Marc B Vilain and Henry A Kautz. Constraint Propagation Algorithms for Temporal Reasoning. In *AAAI*, volume 86, pages 377–382, 1986.

[117] Philippe Balbiani, Jean-François Condotta, and L Farinas Del Cerro. A Model for Reasoning about Bidimensional Temporal Relations. In *Principles of Knowledge Representation and Reasoning-International Conference*, pages 124–130. Morgan Kaufmann Publishers, 1998.

[118] Philippe Balbiani, Jean-François Condotta, and Luis Fariñas Del Cerro. A Tractable Subclass of the Block Algebra: Constraint Propagation and Preconvex Relations. In *Progress in Artificial Intelligence: 9th Portuguese Conference on Artificial Intelligence*, pages 75–89. Springer, 1999.

[119] Alexander Scivos and Bernhard Nebel. The Finest of its Class: The Natural Point-Based Ternary Calculus for Qualitative Spatial Reasoning. In *International Conference on Spatial Cognition*, pages 283–303. Springer, 2004.

[120] Reinhard Moratz, Dominik Lücke, and Till Mossakowski. A Condensed Semantics for Qualitative Spatial Reasoning about Oriented Straight Line Segments. *Artificial Intelligence*, 175(16-17):2099–2127, 2011.

[121] Francesco Tarquini, Giorgio De Felice, Paolo Fogliaroni, and Eliseo Clementini. A Qualitative Model for Visibility Relations. In *KI 2007: Advances in Artificial Intelligence: 30th Annual German Conference on AI, KI 2007, Osnabrück, Germany, September 10-13, 2007. Proceedings 30*, pages 510–513. Springer, 2007.

[122] Reinhard Moratz. Representing Relative Direction as a Binary Relation of Oriented Points. In *ECAI*, volume 6, pages 407–411, 2006.

[123] Till Mossakowski and Reinhard Moratz. Qualitative Reasoning about Relative Direction of Oriented Points. *Artificial Intelligence*, 180:34–45, 2012.

[124] Björn Gottfried. Reasoning about Intervals in Two Dimensions. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 6, pages 5324–5332. IEEE, 2004.

[125] Reinhard Moratz and Marco Ragni. Qualitative Spatial Reasoning about Relative Point Position. *Journal of Visual Languages & Computing*, 19(1):75–98, 2008.

[126] Amar Isli and Anthony G Cohn. A New Approach to Cyclic Ordering of 2D Orientations Using Ternary Relation Algebras. *Artificial Intelligence*, 122(1-2): 137–187, 2000.

[127] Jae Hee Lee, Jochen Renz, and Diedrich Wolter. StarVars-Effective Reasoning about Relative Directions. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[128] Christian Freksa and Kai Zimmermann. On the Utilization of Spatial Structures for Cognitively Plausible and Efficient Reasoning. In *[Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics*, pages 261–266. IEEE, 1992.

[129] Julio Pacheco, Ma Teresa Escrig, and Francisco Toledo. Representing and Reasoning on Three-Dimensional Qualitative Orientation Point Objects. In *Progress in Artificial Intelligence: Knowledge Extraction, Multi-agent Systems, Logic Programming, and Constraint Solving 10th Portuguese Conference on Artificial Intelligence, EPIA 2001 Porto, Portugal, December 17–20, 2001 Proceedings 10*, pages 298–305. Springer, 2001.

[130] Jan Oliver Wallgrün, Diedrich Wolter, and Kai-Florian Richter. Qualitative Matching of Spatial Information. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 300–309, 2010.

[131] Anthony G Cohn, Brandon Bennett, John Gooday, and Nicholas Mark Gotts. Qualitative Spatial Representation and Reasoning with the Region Connection Calculus. *Geoinformatica*, 1:275–316, 1997.

[132] Eliseo Clementini, Paolino Di Felice, and Peter Van Oosterom. A Small Set of

Formal Topological Relationships Suitable for End-User Interaction. In *International symposium on spatial databases*, pages 277–295. Springer, 1993.

[133] Max J Egenhofer. Reasoning about Binary Topological Relations. In *Symposium on Spatial Databases*, pages 141–160. Springer, 1991.

[134] M Egenhofer and Jayant Sharma. Assessing the Consistency of Complete and Incomplete Topological Information. *Geographical Systems*, 1(1):47–68, 1993.

[135] Yohei Kurata. 9+-Intersection Calculi for Spatial Reasoning on the Topological Relations between Heterogeneous Objects. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 390–393, 2010.

[136] Reinhard Moratz and Jan Oliver Wallgrün. Spatial Reasoning with Augmented Points: Extending Cardinal Directions with Local Distances. *Journal of Spatial Information Science*, pages 1–30, 2012.

[137] David Randell, Mark Witkowski, and Murray Shanahan. From Images to Bodies: Modelling and Exploiting Spatial Occlusion and Motion Parallax. In *IJCAI*, pages 57–66, 2001.

[138] Christian Köhler. The Occlusion Calculus. In *Cognitive vision workshop*, pages 420–450. Citeseer, 2002.

[139] Antony Galton. Lines of Sight. In *AISB Workshop on Spatial and Spatio-Temporal Reasoning*, volume 35, pages 37–39. Dublin University Press Dublin, Ireland, 1994.

[140] Chaman L Sabharwal and Jennifer L Leopold. Evolution of Region Connection Calculus to VRCC-3D+. *New Mathematics and Natural Computation*, 10(02): 103–141, 2014.

[141] Nico Van de Weghe, Bart Kuijpers, Peter Bogaert, and Philippe De Maeyer. A Qualitative Trajectory Calculus and the Composition of Its Relations. In *GeoSpatial Semantics: First International Conference, GeoS 2005, Mexico City, Mexico, November 29-30, 2005. Proceedings 1*, pages 60–76. Springer, 2005.

[142] Christian Landsiedel, Verena Rieser, Matthew Walter, and Dirk Wollherr. A Review of Spatial Reasoning and Interaction for Real-World Robotics. *Advanced Robotics*, 31(5):222–242, 2017.

[143] Roshanak Mirzaee and Parisa Kordjamshidi. Transfer Learning with Synthetic Corpora for Spatial Role Labeling and Reasoning. *arXiv preprint arXiv:2210.16952*, 2022.

[144] Wenbin Li, Di Yao, Ruibo Zhao, Wenjie Chen, Zijie Xu, Chengxue Luo, Chang Gong, Quanliang Jing, Haining Tan, and Jingping Bi. STBench: Assessing the Ability of Large Language Models in Spatio-Temporal Analysis. *arXiv preprint arXiv:2406.19065*, 2024.

[145] Anthony G Cohn and Robert E Blackwell. Evaluating the Ability of Large Language Models to Reason About Cardinal Directions. In *16th International Conference on Spatial Information Theory (COSIT 2024)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024.

[146] Manasi Sharma. Exploring and Improving the Spatial Reasoning Abilities of Large Language Models. In *I Can't Believe It's Not Better Workshop: Failure Modes in the Age of Foundation Models*, 2023.

[147] Anthony G Cohn and Jose Hernandez-Orallo. Dialectical Language Model Evaluation: An Initial Appraisal of the Commonsense Spatial Reasoning Abilities of LLMs. *arXiv preprint arXiv:2304.11164*, 2023.

[148] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems*, 36, 2024.

[149] Marco Ragni, Thomas Fangmeier, Lara Webber, and Markus Knauff. Complexity in Spatial Reasoning. In *28th Annual Conference of the Cognitive Science Society*, 2006.

[150] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A Neural Probabilistic Language Model. In *Proceedings of the 14th International Conference on Neural Information Processing Systems*, pages 893–899, 2000.

[151] Taojun Hu and Xiao-Hua Zhou. Unveiling LLM Evaluation Focused On Metrics: Challenges and Solutions. *arXiv preprint arXiv:2404.09135*, 2024.

[152] Xuezhi Wang and Denny Zhou. Chain-of-Thought Reasoning Without Prompting. *arXiv preprint arXiv:2402.10200*, 2024.

[153] Fangjun Li, David C Hogg, and Anthony G Cohn. Advancing Spatial Reasoning in Large Language Models: An In-Depth Evaluation and Enhancement Using the StepGame Benchmark. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18500–18507, 2024.

[154] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*, 2020.

[155] Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. A Comprehensive Capability Analysis of GPT-3 and GPT-3.5 Series Models. *arXiv preprint arXiv:2303.10420*, 2023.

[156] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*, 2022.

[157] Dan Bohus, Sean Andrist, Yuwei Bao, Eric Horvitz, and Ann Paradiso. "Is This It?": Towards Ecologically Valid Benchmarks for Situated Collaboration. In *Companion Proceedings of the 26th International Conference on Multimodal Interaction*, pages 41–45, 2024.

[158] Eliseo Clementini, Paolino Di Felice, and Daniel Hernández. Qualitative Representation of Positional Information. *Artificial intelligence*, 95(2):317–356, 1997.

[159] Michael L Mavrovouniotis and George Stephanopoulos. Formal Order-of-Magnitude Reasoning in Process Engineering. *Computers & Chemical Engineering*, 12(9-10):867–880, 1988.

[160] Laura Kallmeyer. *Parsing beyond Context-Free Grammars*. Springer Science & Business Media, 2010.

[161] Mary Hegarty. Components of Spatial Intelligence. In *Psychology of learning and motivation*, volume 52, pages 265–297. Elsevier, 2010.

[162] Anthony G Cohn. An Evaluation of ChatGPT-4's Qualitative Spatial Reasoning Capabilities in RCC-8. *arXiv preprint arXiv:2309.15577*, 2023.

[163] Marco Ragni and Markus Knauff. A Theory and a Computational Model of Spatial Reasoning with Preferred Mental Models. *Psychological review*, 120(3): 561, 2013.

[164] Georg Jahn, Markus Knauff, and Philip N Johnson-Laird. Preferred Mental Models in Reasoning about Spatial Relations. *Memory & cognition*, 35:2075–2087, 2007.

[165] Peter C Cheeseman, Bob Kanefsky, William M Taylor, et al. Where the Really Hard Problems Are. In *IJCAI*, volume 91, pages 331–337, 1991.

[166] Robert E Blackwell, Jon Barry, and Anthony G Cohn. Towards Reproducible LLM Evaluation: Quantifying Uncertainty in LLM Benchmark Scores. *arXiv preprint arXiv:2410.03492*, 2024.

[167] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690, 2024.

[168] Fivos Papadimitriou. Spatial AI for Artificial General Intelligence. In *Spatial Artificial Intelligence*, pages 53–63. Springer, 2025.