# Learning to Draw ASCII Improves Spatial Reasoning in Language Models

**Anonymous ACL submission**

## Abstract

Human spatial cognition involves both perceiving spatial configurations and reconstructing them through sketching or mental imagery. Although perceiving is often intuitive, reconstruction is substantially more demanding. To facilitate this process, humans frequently rely on sketching as a "read-write loop" to visualize ideas and anchor their reasoning when solving complex spatial problems. In this work, we investigate whether Large Language Models (LLMs) exhibit a similar dependency when processing spatial topology. We introduce TEXT2SPACE, a dataset that pairs natural language descriptions with ground-truth ASCII representations and QA pairs, using ASCII as a diagnostic probe to decouple representation failures from reasoning failures. Our evaluation reveals a pronounced "Read-Write Asymmetry": LLMs interpret ASCII representations effectively but struggle to reconstruct them, and this generation bottleneck propagates to the subsequent reasoning phase, producing flawed intermediate representations that distort subsequent reasoning. Training models on Text↔ASCII alignment significantly improves performance on text-only spatial reasoning tasks, even without generating ASCII at inference time. Our findings show that enabling LLMs to "learn to draw" not only allows them to better leverage sketching as a reasoning strategy but also fosters deeper internalization of spatial understanding.

## 1 Introduction

Spatial reasoning (Byrne and Johnson-Laird, 1989; Newcombe and Huttenlocher, 2000) is fundamental to human cognition (Malanchini et al., 2020). As language models (LLMs) are increasingly applied to spatial reasoning tasks, evaluating their capabilities has become a critical priority (Li et al., 2024; Cohn and Hernandez-Orallo, 2023; Wang et al., 2024; Liu et al., 2023; Cheng et al., 2024).

However, existing benchmarks are limited to evaluating final answers (Yamada et al., 2023; Liu et al., 2023). When a model fails, the absence of interpretability makes it unclear whether the error stems from an inaccurate internal representation or from insufficient reasoning over an otherwise correct representation. This lack of observability into intermediate spatial representations during the reasoning process creates a diagnostic gap, hindering the identification of effective ways to achieve faithful spatial reasoning in LLMs (Singh et al., 2024; Parcalabescu and Frank, 2024; Jacovi and Goldberg, 2020; Creswell and Shanahan, 2022).

We address this gap by introducing ASCII representations that encode two dimensional structure as explicit text based grids, reducing the ambiguity and variability inherent in sequential natural language descriptions (Piantadosi et al., 2012). ASCII diagrams offer an explicit and verifiable way to represent spatial layouts, allowing us to examine the intermediate representation a model produces and directly assess its correctness. To support this analysis, we introduce TEXT2SPACE, a benchmark of diverse spatial configurations paired with natural language descriptions, images, and three styles of ASCII grids. Each instance includes a ground truth ASCII representation, enabling evaluation of intermediate spatial representations during the reasoning process and improving interpretability in spatial reasoning tasks. The dataset further provides fine-grained annotations, including spatial complexity, terminology, and transitive inference requirements, to support systematic analysis.

Using TEXT2SPACE, we contribute three analyses. First, we **benchmark** language models on ASCII comprehension, generation, and spatial reasoning. Second, we examine the **reasoning process** in spatial tasks involving ASCII representations, assessing the extent to which these representations support model reasoning. Finally, we investigate whether enhancing models' understanding

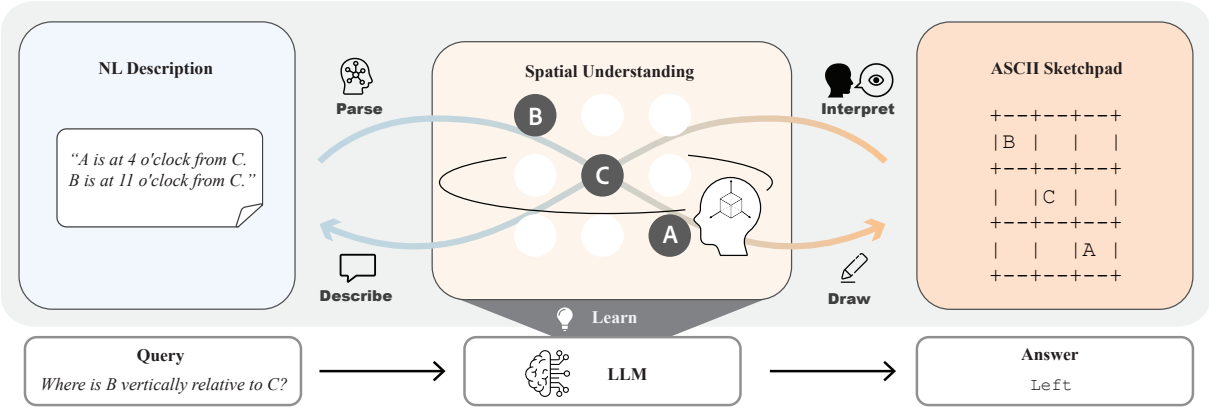


Figure 1: Overview of the study design. Inspired by empirical cognitive human strategies, we utilize Natural Language (NL) and ASCII representations as independent reasoning sources. This bidirectional cycle connects the two modalities via Parsing, Drawing, Interpreting, and Describing steps. We find that training LLMs to mimic this human-like, sketch-based reasoning process significantly improves accuracy on complex spatial reasoning tasks.

of spatial relations through ASCII-based learning **generalizes** to improved performance on purely linguistic spatial reasoning tasks.

Our evaluation yields three principal findings: **Finding 1: Comprehension exceeds generation.** Models can interpret spatial information from ASCII diagrams more effectively than they can produce such diagrams from text. This suggests that constructing a structured spatial representation from abstract descriptions is a more demanding cognitive process than reading and understanding one. **Finding 2: Flawed representations impair reasoning.** Most models achieve higher accuracy when answering directly from text without generating ASCII diagrams. Errors in constructing the intermediate spatial representation can propagate to the final answer. This highlights that reasoning performance is fundamentally constrained by the quality of the underlying representation. **Finding 3: Spatial knowledge generalizes from learning with ASCII.** Fine-tuning on ASCII-based tasks significantly enhances spatial reasoning from natural language descriptions. This indicates that models can acquire abstract spatial representations rather than merely memorizing ASCII syntax. The knowledge learned in the ASCII domain generalizes to broader spatial reasoning tasks.

We present three contributions: (1) a benchmark for evaluating spatial reasoning using ASCII as a reference; (2) a framework for assessing through the construction and interpretation of ASCII spatial representation; and (3) empirical evidence showing that spatial knowledge acquired from ASCII representations generalizes effectively to a wide range of spatial reasoning tasks.

## 2 Related Works

**Linguistic Spatial Reasoning Benchmarks.** Early work on spatial language processing focused on extracting formal representations from natural language text (Kordjamshidi et al., 2011). Subsequent benchmarks like bAbI (Weston et al., 2015) introduced positional reasoning, while StepGame (Shi et al., 2022) and SpartQA (Mirzaee et al., 2021) advanced the field with more complicated reasoning and broader linguistic coverage. However, a critical limitation remains across these benchmarks: they rely exclusively on final-answer accuracy. This "black-box" evaluation obscures the underlying inference process, making it impossible to distinguish between failures in representation and failures in reasoning. Our work addresses this methodological gap by introducing ground-truth intermediate representations, enabling precise diagnosis of spatial reasoning errors.

While integrating visual modalities (VLMs) offers a potential remedy for spatial understanding, recent studies indicate that visual perception introduces its own challenges (Liu et al., 2023, 2024; Song et al., 2024; Kamath et al., 2023) and spatial reasoning (Chen et al., 2024; Zhang et al., 2025a; Ogezi and Shi, 2025). In contrast, our work decouples spatial reasoning from visual perception by utilizing ASCII grids. These grids serve a dual purpose: they act not only as a reasoning source within the context but also as a verifiable representation of all spatial relations.

**ASCII and Spatial Representation.** The utility of intermediate reasoning steps is well-established through Chain-of-Thought (CoT) prompting (Wei

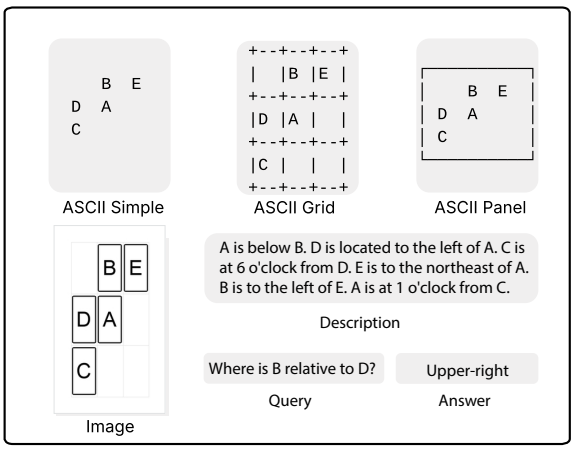


Figure 2: An example in our TEXT2SPACE dataset: we have natural language description, Q-A pairs, three types of ASCII, and a rendered image.

et al., 2022) and its extensions, such as Tree-of-Thoughts (Yao et al., 2023) and latent reasoning (Hao et al., 2024). Beyond purely linguistic traces, prior work has shown that symbolic, text-native spatial representations can effectively support spatial reasoning in language models, including grid-based worlds and discrete relational layouts (Côté et al., 2018). Visualization-of-Thought (Wu et al., 2024; Li et al., 2025) draws on human mental imagery to generate symbolic or pixel-level visualizations for spatial reasoning. We propose a text-native alternative using ASCII to visualize spatial representations. Unlike natural language reasoning chains, which are hard to evaluate, ASCII creates a fully verifiable, spatially explicit "mental map" directly within the language model.

**Generalization of Spatial Reasoning** Recent research demonstrates that enhancing specific capabilities in LLMs often yields transfer benefits to other domains. For instance, Shao et al. (2024) indicate that training on mathematics improves general reasoning benchmarks, not just quantitative tasks. Additionally, in the field of spatial understanding, numerous studies have shown that a robust grasp of space benefits visual reasoning tasks (Wei et al., 2025; Han et al., 2025; Wang et al., 2025; Hu et al., 2025). Complementing these findings, our approach emphasizes building spatial cognition via ASCII generation and comprehension. We demonstrate that improvements in these seemingly non-reasoning tasks effectively generalize to text-based reasoning benchmarks.

## 3 TEXT2SPACE Dataset

We introduce TEXT2SPACE, a spatial reasoning QA dataset with text-based graph (ASCII) representations designed to validate the reasoning process in LLMs. This dataset evaluates how well LLMs interpret, align, and reason about spatial relationships. It presents diverse spatial configurations across four formats: ASCII art, natural language, rendered images, and Query-Answer pairs. An example is shown in Fig 2.

### 3.1 Data Generation

The dataset is constructed from connected spatial graphs where entities are linked via eight directional relationships: *above, below, left, right, upper-left, upper-right, lower-left, lower-right*. All relations adhere to a unified coordinate system, serving as the ground truth for both ASCII diagrams and rendered images.

Instances are generated using a breadth-first search (BFS) procedure to guarantee connectivity. We enforce strict validation rules: self-loops and duplicate edges are prohibited, and any configuration conflicting with existing spatial constraints is rejected. These safeguards ensure every instance represents a valid, internally consistent topology. A detailed data generation process is presented in Appendix F.

We vary complexity from simple two-component structures to intricate graphs with up to eight components and twelve relational statements, capturing configurations that demand transitive reasoning across multiple inference steps.

Each instance includes natural language descriptions synthesized from the spatial graph, with a wide range of synonyms used to ensure linguistic diversity. Corresponding queries are designed so that the target object pairs admit unique inferability, allowing only a single, unambiguous spatial relation to be inferred.

### 3.2 Multi-Format Representations

We derive both ASCII and image representations from the same underlying structure to facilitate multimodal evaluation and cross-format consistency.

**ASCII Visualization:** We include three rendering styles to test robustness against formatting variations. The *simple format* uses minimal spacing without borders for clarity. The *grid format* introduces structure using +, -, and | characters. The

*panel format* employs box-drawing characters for a visually distinct layout.

**Image Representation:** The spatial arrangement is also rendered in a standardized image format with configurable grid lines and directional arrows, serving as both a multimodal evaluation target and a human-readable reference. Further dataset details and data examples are provided in Appendix A.

## 4 Experiments

We split TEXT2SPACE into training and test sets, with the test set consisting of 1,000 samples used in the following experiments. We choose the ASCII *grid format* throughout our experiments; details of this format are provided in Appendix B. **Benchmarking:** How effectively do LLMs comprehend, generate, and utilize ASCII-based graph representations for spatial reasoning? **Internal Spatial Processing:** Does an ASCII intermediate step encourage structured, human-like reasoning, and how does this affect downstream performance? **Learning Spatial Relations with ASCII:** Does training LLMs to convert natural language descriptions into ASCII structures enhance spatial reasoning, and does this benefit transfer to pure natural language tasks?

### 4.1 Models

We evaluate two model categories. **Instruct models**: including LLAMA-3-70B-INSTRUCT (AI@Meta, 2024), QWEN2.5-7B-INSTRUCT (Yang et al., 2024; Team, 2024), QWEN3-30B-A3B-INSTRUCT-2507, QWEN3-235B-A22B-INSTRUCT-2507 (Team, 2025), GPT-4.1 (OpenAI, 2025). **Thinking model:** GPT-OSS-20B (OpenAI, 2025). Unless otherwise specified, we use default settings and apply 5-shot prompting (Brown et al., 2020) to ensure consistency.

### 4.2 Task Setups

Since ASCII is a unique representation that encodes two-dimensional spatial structure within a one-dimensional token sequence, we first evaluate language models' ability to understand and generate ASCII representations. Specifically, we assess two complementary capabilities: generating an ASCII representation from a textual description (*Desc→ASCII*) and generating a natural-language description from an ASCII representation (*ASCII→Desc*).

We then move to spatial reasoning tasks, denoted by $\mathcal{R}$, in which the model is given a reasoning source and asked to answer queries about spatial relationships. $\mathcal{R}_{\text{Desc}}$ refers to spatial reasoning based solely on textual descriptions, while $\mathcal{R}_{\text{ASCII}}$ refers to reasoning based solely on ASCII representations.

Beyond reasoning from a single reference, we investigate the role of ASCII as an intermediate sketching process alongside descriptions, which follows a workflow sequence of *Desc* → *ASCII* → *Answer*. In $\mathcal{R}_{\text{Desc:(Ans, ASCII)}}$, the model is given a description and first generates the spatial reasoning answer, and then produces an ASCII representation. Conversely, in $\mathcal{R}_{\text{Desc:(ASCII, Ans)}}$, the model first generates an ASCII representation and then uses it to answer the query. For these settings, we consider both *one-turn generation*, where the model produces the ASCII and the answer within a single response, and *two-turn generation*, where the second output is elicited via a follow-up prompt. In this sense, ASCII functions as a form of self-explanation (Huang et al., 2023; Madsen et al., 2024), and the two generation orders mirror established paradigms: making a prediction and then explaining it, or generating an explanation and using it to make a prediction (Camburu et al., 2018).

Finally, we evaluate $\mathcal{R}_{\text{Desc+ASCII}}$. In this phase, we substitute the model-generated ASCII with the ground-truth ASCII before the model answers the query. This replacement strategy simulates an ideal reasoning process, allowing us to assess the upper-bound performance of the model when guided by an ideal spatial representation.

### 4.3 Evaluation Methods

Our evaluation protocol targets three key aspects of the model's performance: **QA Correctness:** For the spatial reasoning QA tasks, we validate the prediction against a predefined set of relation aliases. **Representation Alignment:** For the *ASCII↔Desc* conversion tasks, we measure the correspondence between converted representations. For *ASCII→Desc*, the ASCII diagram is treated as the reference. We evaluate, across all generated descriptions, the percentage that can be correctly inferred from the given ASCII representation. Conversely, for *Desc→ASCII*, we evaluate the generated ASCII diagrams, $\mathcal{R}_{\text{Desc:(Ans, ASCII)}}$ and $\mathcal{R}_{\text{Desc:(ASCII, Ans)}}$, by taking the description as the reference and measuring the proportion of the description that is correctly reflected in the ASCII output. **Reasoning Consistency:** To analyze internal

coherence across the sketching tasks, we introduce two complementary metrics. Consistency: Measures the alignment between the generated ASCII and the generated Answer. This determines if the model's intermediate output supports its final prediction. Consistency Accuracy: Measures answer accuracy strictly on instances where the ASCII and Answer are consistent. By controlling for internal coherence, this isolates the model's reasoning capability, helping distinguish whether errors arise from failures in constructing spatial representations or from limitations in reasoning over them.

### 4.4 Learning Spatial Relations with ASCII

We propose that ASCII representations function as a structured reasoning intermediate: a verifiable "cognitive map" that explicates spatial relationships during model reasoning. We hypothesize that the active *process* of constructing such verifiable maps (Lightman et al., 2023; Zhang et al., 2025b) consolidates spatial logic during training, leading to improved performance on spatial reasoning tasks even when explicit ASCII representations are not produced at inference time.

To investigate the distinct roles of understanding versus generating ASCII representations, we fine-tune the models with three configurations: **Learning to Draw** (*Desc→ASCII*), which forces the model to actively construct spatial layouts from text (performing "mental mapping"); **Learning to Read** (*ASCII→Desc*), which focuses on parsing and verbalizing ASCII representations; and **Bidirectional Alignment** (*Desc↔ASCII*), a combined approach training on both tasks simultaneously.

We fine-tuned QWEN3-30B-A3B on 4x NVIDIA A40 GPUs using the SFTTrainer from the HuggingFace TRL library (von Werra et al., 2020). We employed Low-Rank Adaptation (LoRA) (Hu et al., 2021; Mangrulkar et al., 2022) for parameter-efficient training, utilizing a split of 4,000 training and 500 validation examples. We adhered to default library configurations for all other settings; full training details are provided in Appendix C.

We evaluate performance on the Spatial Reasoning Task ($\mathcal{R}_{\text{Desc}}$) where models predict spatial relations solely from natural language descriptions without explicit ASCII input. We specifically investigate whether improving the ability to *construct* spatial layouts (via ASCII) enhances reasoning transfer on $\mathcal{R}_{\text{Desc}}$ without direct supervision, and whether these gains generalize to external benchmarks, such as STEPGAME (Shi et al., 2022), under a zero-shot prompting setting. Our hypothesis is that the "Learning to Draw" process fosters a generalized spatial understanding.

| Model | *Desc → ASCII* | *ASCII → Desc* |
|---|---|---|
| QWEN2.5-7B | 16.1 | 38.4 |
| LLAMA-3-70B | 43.9 | 57.4 |
| GPT-OSS-20B | 44.0 | 58.1 |
| GPT-4.1 | 53.6 | 72.5 |
| QWEN3-235B | 46.0 | 59.7 |
| QWEN3-30B | 29.2 | 55.8 |
| QWEN3-30B $FT_{D\to A}$ | 67.9 | 53.5 |
| QWEN3-30B $FT_{D\leftarrow A}$ | 24.1 | 66.5 |
| QWEN3-30B $FT_{D\leftrightarrow A}$ | 64.6 | 68.4 |

Table 1: Performance on bidirectional mapping between ASCII and description. Reported scores are the average percentage of correctly aligned representations across all test samples for mapping textual descriptions to ASCII (Desc→ASCII) and for identifying descriptions from ASCII (ASCII→Desc).

## 5 Results and Discussion

### 5.1 Benchmarking

**ASCII Understanding and Generation** Table 1 presents the results for the *Desc↔ASCII* conversion tasks. We identify a consistent asymmetry across all models, where performance on the interpretation task (*ASCII→Desc*) significantly exceeds that of the generation task (*Desc→ASCII*). This suggests that while models are relatively proficient at **"reading"** or identifying patterns in ASCII, they struggle significantly more with **"drawing"** it correctly. Even the reasoning model GPT-OSS-20B fails to construct accurate layouts in over 50% of cases, despite robust interpretative success. We attribute this asymmetry to the inherent difficulty of grounding abstract semantic relations into concrete 2D layouts, a generative process demanding global planning capabilities that standard pre-training does not explicitly cultivate.

**Spatial Reasoning Performance** Table 2 reports accuracy scores in spatial reasoning tasks ($\mathcal{R}$) under different settings. While GPT-OSS-20B leads in overall performance and QWEN2.5-7B trails, a consistent pattern emerges across all models regarding input format.

When ASCII serves as a provided input signal, it significantly boosts performance. Most models reason more accurately from ASCII alone ($\mathcal{R}_{\text{ASCII}}$) than from text descriptions ($\mathcal{R}_{\text{Desc}}$), and the combination of both ($\mathcal{R}_{\text{Desc+ASCII}}$) yields the highest accuracy. This demonstrates that as an input, ASCII

| Model | $\mathcal{R}_{\text{Desc}}$ | $\mathcal{R}_{\text{ASCII}}$ | $\mathcal{R}_{\text{Desc(Ans, ASCII)}}$ | | $\mathcal{R}_{\text{Desc(ASCII, Ans)}}$ | | $\mathcal{R}_{\text{Desc+ASCII}}$ |
|---|---|---|---|---|---|---|---|
| | Ans | Ans | Ans | ASCII | Ans | ASCII | Ans |
| *One-turn generation* | | | | | | | |
| QWEN2.5-7B | 42.0 | 39.9 | 43.6 | 12.5 | 36.7 | 12.0 | **55.5** |
| LLAMA-3-70B | 71.7 | 75.3 | 71.1 | 42.2 | 66.9 | 41.7 | **78.4** |
| GPT-4.1 | 85.8 | 90.1 | 86.5 | 50.6 | 83.8 | 50.5 | **92.8** |
| GPT-OSS-20B | 93.3 | 92.7 | 94.3 | 38.9 | 91.9 | 42.8 | **97.0** |
| QWEN3-235B-A22B | 84.8 | 89.0 | 84.2 | 45.3 | 82.3 | 45.4 | **90.1** |
| QWEN3-30B-A3B | 75.0 | 83.2 | 71.1 | 27.9 | 66.6 | 27.5 | **83.7** |
| QWEN3-30B-A3B $\text{FT}_{\text{DESC}\rightarrow\text{ASCII}}$ | 82.2 | 78.6 | 82.5 | 64.2 | 79.1 | 64.7 | **84.9** |
| QWEN3-30B-A3B $\text{FT}_{\text{DESC}\leftarrow\text{ASCII}}$ | 73.7 | 87.4 | 68.0 | 22.0 | 70.2 | 20.0 | **87.2** |
| QWEN3-30B-A3B $\text{FT}_{\text{DESC}\leftrightarrow\text{ASCII}}$ | 82.7 | 81.3 | 82.9 | 60.2 | 83.1 | 61.0 | **86.2** |
| *Two-turn generation* | | | | | | | |
| QWEN2.5-7B | | | 43.6 | 13.5 | 42.1 | 13.7 | **52.3** |
| LLAMA-3-70B | | | 70.9 | 42.8 | 69.0 | 42.3 | **78.9** |
| GPT-4.1 | | | 85.9 | 50.8 | 81.0 | 50.6 | **93.0** |
| GPT-OSS-20B | | | 93.1 | 45.0 | 87.7 | 47.9 | **94.3** |
| QWEN3-235B-A22B | | | 85.6 | 45.1 | 77.8 | 45.3 | **89.0** |
| QWEN3-30B-A3B | | | 74.3 | 27.4 | 69.3 | 26.4 | **82.6** |
| QWEN3-30B-A3B $\text{FT}_{\text{DESC}\rightarrow\text{ASCII}}$ | | | 80.7 | 65.1 | 81.3 | 64.7 | **83.7** |
| QWEN3-30B-A3B $\text{FT}_{\text{DESC}\leftarrow\text{ASCII}}$ | | | 70.1 | 22.1 | 65.2 | 20.2 | **87.7** |
| QWEN3-30B-A3B $\text{FT}_{\text{DESC}\leftrightarrow\text{ASCII}}$ | | | 83.1 | 61.4 | 82.7 | 61.8 | **87.5** |

Table 2: Spatial reasoning results across reasoning sources and settings. The $\mathcal{R}_{\text{Desc(Ans, ASCII)}}$ and $\mathcal{R}_{\text{Desc(ASCII, Ans)}}$ representations denote generation orders in which the final answer precedes or follows the ASCII representation. *"One-turn"* refers joint generation, while *"two-turn"* indicates sequential generation in separate inference calls. In the two-turn $\mathcal{R}_{\text{Desc+ASCII}}$ setting, gold-standard ASCII is provided. Answer accuracy is determined by label matching; ASCII accuracy is measured as the proportion of correctly depicted relations in the generated ASCII output. For each language model, we **bold the best-performing reasoning source** and underline the second best.

serves as a critical signal that resolves spatial ambiguities in natural language.

When ASCII is placed as an intermediate generation step ("sketching"), the results are not showing significant improvement. In both *one-turn* and *two-turn* settings, generating the layout prior to prediction ($\mathcal{R}_{\text{Desc(ASCII, Ans)}}$) consistently yields lower accuracy than generating the answer first ($\mathcal{R}_{\text{Desc(Ans, ASCII)}}$). Notably, this approach degrades performance below the text-only baseline ($\mathcal{R}_{\text{Desc}}$). Unlike textual CoT (Wei et al., 2022), where intermediate steps are difficult to verify, ASCII allows us to directly observe that flawed spatial construction propagates errors to subsequence reasoning. The benefit of explicit intermediates depends critically on their accuracy.

The specific drop in answer accuracy implies that **generated ASCII representations are insufficiently accurate to support spatial reasoning**, serving instead as misleading information. This mirrors prior findings suggesting a trade-off between generating supporting artifacts and task accuracy (Camburu et al., 2018; Huang et al., 2023). However, unlike the unstructured text in previous studies, our intermediate artifact is verifiable. This distinction allows us to pinpoint the source of degradation rather than treating the performance loss as a black-box effect.

Finally, the performance degradation caused by generating ASCII first persists in the *two-turn* generation setting, and does not appear to be amplified relative to the *one-turn* case. This pattern motivates a deeper interpretability analysis. If generating ASCII does not improve performance and can even degrade it, does the resulting representation actually support the final spatial reasoning answer, or is it largely unrelated to the model's prediction?

### 5.2 Consistency Analysis in Spatial Reasoning

Table 3 shows the alignment between ASCII representations and textual answers across both output orders and turn settings. We report three key metrics: (1) Answer Accuracy measures overall task performance; (2) Consistency quantifies the logical alignment between the generated ASCII representation and the textual answer; and (3) Consistent Accuracy evaluates correctness specifically on instances where the representation and answer correspond.

In the *one-turn* generation setting, producing the

ASCII representation before the answer does not improve consistency. Similar consistency across output orders suggests that the model does not clearly use the ASCII as a reasoning scaffold, with both outputs likely reflecting the same underlying representation.

In the *two-turn* generation setting, we observed an improved consistency rate, where the model first generates either the answer or the ASCII representation, and is then prompted to produce the other. The consistency gap between $\mathcal{R}_{\text{Desc:(Ans, ASCII)}}$ and $\mathcal{R}_{\text{Desc:(ASCII, Ans)}}$ indicates that the response to the follow-up query is strongly influenced by the previously generated output. Despite this notable improvement in consistency, answer accuracy increases only marginally. This limited gain may be partly due to the suboptimal quality of the generated ASCII representations, which correlates with lower answer accuracy (Table 2).

Nevertheless, consistency remains a valuable indicator, particularly given that it operates without requiring ground-truth annotations for either the spatial layout or the final answer. While prior work has leveraged Natural Language Inference (NLI) to assess reasoning via consistency (Wang et al., 2022; Lyu et al., 2023; Kumar and Talukdar, 2020), our approach benefits from the explicit verifiability of the generated artifact. Unlike unstructured text, ASCII representations enforce strict spatial constraints, allowing us to objectively assess the alignment between the representation and the answer. Although faulty representations can occur, we observe that accuracy on consistent instances significantly exceeds the accuracy over all instances. This demonstrates that the internal coherence between the verifiable layout and the text serves as a powerful unsupervised signal for reliability, enabling the identification of high-quality answers even in the absence of external labels.

From a forward-looking perspective, our results shows **"ASCII as a Verifier"** as a promising direction for post-CoT reasoning research. The structural constraints imposed by ASCII representations suggest that cross-representation consistency may provide a more reliable signal of reasoning validity than text-only self-consistency. Future work may explore integrating such structured verification signals into reasoning pipelines, both as an error-detection mechanism and as a guiding objective for improving intermediate representation construction.

| **Model** | $\mathcal{R}_{\text{Desc:(Ans, ASCII)}}$ | | | $\mathcal{R}_{\text{Desc:(ASCII, Ans)}}$ | | |
|---|---|---|---|---|---|---|
| | Ans | Cons | C_Acc | Ans | Cons | C_Acc |
| *One-turn generation* | | | | | | |
| QWEN2.5-7B | 43.6 | **23.0** | 65.2 | 36.7 | 20.5 | 64.4 |
| LLAMA-3-70B | 71.1 | 65.2 | 87.6 | 66.9 | **65.4** | 83.8 |
| GPT-OSS-20B | 94.3 | **75.9** | 97.0 | 91.9 | 75.5 | 96.9 |
| QWEN3-235B | 84.2 | 67.8 | 95.0 | 82.3 | **72.5** | 89.9 |
| QWEN3-30B | 71.1 | 46.1 | 88.5 | 66.6 | **46.3** | 82.5 |
| QWEN3-30B $\text{FT}_{\text{D}\rightarrow\text{A}}$ | 82.5 | **73.2** | 97.5 | 79.1 | 71.9 | 96.5 |
| QWEN3-30B $\text{FT}_{\text{D}\leftarrow\text{A}}$ | 68.0 | 39.2 | 84.9 | 70.2 | **46.4** | 81.2 |
| QWEN3-30B $\text{FT}_{\text{D}\leftrightarrow\text{A}}$ | 82.9 | **73.7** | 97.4 | 83.1 | 72.9 | 96.6 |
| *Two-turn generation* | | | | | | |
| QWEN2.5-7B | 43.6 | 24.4 | 69.3 | 42.1 | **31.1** | 56.9 |
| LLAMA-3-70B | 70.9 | 63.9 | 89.9 | 69.0 | **71.3** | 81.9 |
| GPT-OSS-20B | 93.1 | 71.8 | 96.7 | 87.7 | **77.2** | 91.6 |
| QWEN3-235B | 85.6 | 70.6 | 93.2 | 77.8 | **78.4** | 84.1 |
| QWEN3-30B | 74.3 | 50.8 | 88.8 | 69.3 | **58.6** | 75.7 |
| QWEN3-30B $\text{FT}_{\text{D}\rightarrow\text{A}}$ | 80.7 | 74.1 | 98.1 | 81.3 | **74.9** | 96.7 |
| QWEN3-30B $\text{FT}_{\text{D}\leftarrow\text{A}}$ | 70.1 | 42.8 | 85.7 | 65.2 | **55.7** | 69.7 |
| QWEN3-30B $\text{FT}_{\text{D}\leftrightarrow\text{A}}$ | 83.1 | 73.8 | 96.9 | 82.7 | **76.1** | 94.2 |

Table 3: Answer accuracy (Ans), consistency between the generated ASCII and the generated answer (Cons), and accuracy on consistent instances across different settings (C_Acc). For each model, the higher Cons value is in **bold**.

| **Model** | **STEPGAME** |
|---|---|
| QWEN3-30B-A3B | 41.1 |
| QWEN3-30B-A3B $\text{FT}_{\text{DESC}\rightarrow\text{ASCII}}$ | **44.9** |
| QWEN3-30B-A3B $\text{FT}_{\text{DESC}\leftarrow\text{ASCII}}$ | 39.4 |
| QWEN3-30B-A3B $\text{FT}_{\text{DESC}\leftrightarrow\text{ASCII}}$ | 40.4 |

Table 4: STEPGAME results under zero-shot prompting. The best performing system is shown in **bold**.

### 5.3 Learning Spatial Relations with ASCII

Table 1 details the performance of conversion tasks after unidirectional and bidirectional fine-tuning. Specifically, *Desc→ASCII* accuracy increases from 29.2% to 67.9%, whereas *Desc←ASCII* improves from 55.8% to 66.5%. This asymmetry improvement supports our earlier claim that the primary bottleneck lies in ASCII representation *generation* rather than ASCII *understanding*. Moreover, this gap between generation and understanding further widens after fine-tuning on a unidirectional task: as performance on the targeted task improves significantly, the non-targeted task inevitably degrades slightly. In contrast, joint fine-tuning on both ASCII generation and understanding (*Desc↔ASCII*) leads to a more balanced capability, yielding an improvement to 64.6% in ASCII generation and an improvement to 68.4% in ASCII understanding.

We examine whether balanced or targeted fine-tuning strategies yield better spatial capability in

QWEN3-30B. Table 2 reports the results on the unseen spatial reasoning task ($\mathcal{R}_{\text{Desc}}$) alongside the supervised objectives to probe whether the model has internalized a generalized spatial understanding beyond simple format conversion. Compared with the base model, both $\text{FT}_{\text{D}\rightarrow\text{A}}$ and $\text{FT}_{\text{D}\leftrightarrow\text{A}}$ achieve approximately a 9% improvement in accuracy on the description-only reasoning task ($\mathcal{R}_{\text{Desc}}$) with a similarly narrow margin. This indicates that **learning ASCII representations generalizes well and improves spatial reasoning even when ASCII is not explicitly provided.** In contrast, fine-tuning solely on ASCII understanding exacerbates the asymmetry, resulting in a slight drop in answer accuracy compared with the base model.

For $\mathcal{R}_{\text{Desc:(Ans, ASCII)}}$ and $\mathcal{R}_{\text{Desc:(ASCII, Ans)}}$, models fine-tuned on ASCII generation demonstrate significant improvements in ASCII accuracy. Consequently, we observe a reversal in the relationship between generation and reasoning: whereas low-quality layouts in other models hinder answer accuracy, the high-quality intermediate representations here provide a valid scaffold that meaningfully contributes to the final prediction. This suggests that a reliable intermediate state is a prerequisite for improving downstream reasoning.

In tasks $\mathcal{R}_{\text{ASCII}}$ and $\mathcal{R}_{\text{Desc+ASCII}}$ where ASCII is provided as input, fine-tuning on ASCII understanding generally improves answer accuracy and achieves the best performance among all other training settings. This further suggests that **ASCII representations provide a stronger reasoning substrate than natural language descriptions alone**. This mirrors human reasoning. While humans are capable of inferring spatial relationships from textual descriptions, they make substantially fewer errors when a complete visual layout is externally provided, rather than reconstructed mentally. We observe the same pattern in our human validation study (Appendix I), where participants achieve higher accuracy and greater consistency when solving spatial tasks with access to ASCII layouts compared to text-only descriptions.

Table 3 shows that fine-tuning on ASCII generation improves consistency much more than fine-tuning on ASCII understanding. We hypothesize that training solely on interpretation (Understanding) while retaining flawed output capabilities (Generation) exacerbates internal conflict. Specifically, the model becomes hyper-sensitive to discrepancies between its own noisy "sketch" and the textual description. In contrast, fine-tuning on generation resolves this conflict at the source. By ensuring the intermediate representation is accurate and aligns with the text, the model creates more reliable reference. Consequently, it results in higher consistency, and consistent instances are more likely to be correct.

Building on the strong performance observed on our TEXT2SPACE dataset, we evaluate models on the STEPGAME benchmark (Shi et al., 2022) to test generalization. We observe that training models to generate ASCII representations yields higher accuracy on this external task (Table 4). This suggests that the core bottleneck lies in representation construction, and explicitly supervising structured intermediate representations helps models internalize abstract spatial constraints, enabling transfer to broader tasks.

## 6 Conclusion

In this paper, we investigate spatial reasoning in large language models through the lens of explicit representation, ASCII layouts. We introduce TEXT2SPACE, a benchmark that uses ASCII layouts to disentangle representation failures from reasoning failures. Our results reveal a clear read–write asymmetry: while models can reliably interpret spatial information from structured ASCII inputs, they struggle to reconstruct such representations from text, and these errors propagate to downstream reasoning.

Additionally, our consistency analysis demonstrates that the alignment between a model's generated sketch and its final answer serves as a robust unsupervised indicator of correctness. We find that accuracy is significantly higher on instances where the intermediate representation remains consistent with the textual output, even in the absence of ground truth labels.

We further show that reasoning accuracy is primarily constrained by the quality of intermediate spatial representations. Crucially, fine-tuning on Text↔ASCII alignment improves spatial reasoning from natural language even without explicit sketching at inference time, suggesting that the training process instills spatial understanding that generalizes beyond the ASCII format itself. Together, these findings demonstrate that learning to construct spatial representations offers a promising path toward more robust spatial cognition in language models.

## Limitations

This work focuses on spatial reasoning tasks that can be naturally expressed using grid-based ASCII representations. Alternative structured formats, such as coordinate lists or scene graphs, are not explored in this study. Spatial problems involving continuous geometry, pixel-level visual perception, or real-world 3D environments are not explored in this work. While ASCII serves as a text-native visual proxy that leverages the inherent generation capabilities of standard language models without requiring external visual encoders, it remains a discrete abstraction and does not capture the richness of continuous spatial perception, and our synthetic TEXT2SPACE benchmark may not fully represent the noise and ambiguity in real-world scenarios.

Our evaluation relies on answer accuracy and the consistency between generated diagrams and final predictions. Although these metrics provide observable and verifiable signals of spatial reasoning behavior, they do not directly reveal the internal representations used by the model. While the explicit structure of ASCII offers a foundation for iterative verification and self-correction, this work does not explore model self-reflection or corrective mechanisms, which we leave to future research.

## References


AI@Meta. 2024. Llama 3 model card.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ruth M.J Byrne and P.N Johnson-Laird. 1989. Spatial reasoning. *Journal of Memory and Language*, 28(5):564–575.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. 2024. Spatialrgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093.

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

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Anderson Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew J. Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. 2018. Textworld: A learning environment for text-based games. In *Workshop on Computer Games*, pages 41–75. Springer.

Antonia Creswell and Murray Shanahan. 2022. Faithful reasoning using large language models. *arXiv preprint arXiv:2208.14271*.

Junlin Han, Shengbang Tong, David Fan, Yufan Ren, Koustuv Sinha, Philip Torr, and Filippos Kokkinos. 2025. Learning to see before seeing: Demystifying llm visual priors from language pre-training. *arXiv preprint arXiv:2509.26625*.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Zhe Hu, Jing Li, Zhongzhu Pu, Hou Pong Chan, and Yu Yin. 2025. Praxis-vlm: Vision-grounded decision making via text-driven reinforcement learning. *arXiv preprint arXiv:2503.16965*.

Shiyuan Huang, Siddarth Mamidanna, Shreedhar Jangam, Yilun Zhou, and Leilani H Gilpin. 2023. Can large language models explain themselves? a study of llm-generated self-explanations. *arXiv preprint arXiv:2310.11207*.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. 2023. What's" up" with vision-language models? investigating their struggle with spatial reasoning. *arXiv preprint arXiv:2310.19785*.

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

Sawan Kumar and Partha Talukdar. 2020. NILE : Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742, Online. Association for Computational Linguistics.

Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. 2025. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*.

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

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Fangyu Liu, Guy Emerson, and Nigel Collier. 2023. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651.

Li Liu, Diji Yang, Sijia Zhong, Kalyana Suma Sree Tholeti, Lei Ding, Yi Zhang, and Leilani Gilpin. 2024. Right this way: Can vlms guide us to see more to answer questions? *Advances in Neural Information Processing Systems*, 37:132946–132976.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *The 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL 2023)*.

Andreas Madsen, Sarath Chandar, and Siva Reddy. 2024. Are self-explanations from large language models faithful? *arXiv preprint arXiv:2401.07927*.

Margherita Malanchini, Kaili Rimfeld, Nicholas G Shakeshaft, Andrew McMillan, Kerry L Schofield, Maja Rodic, Valerio Rossi, Yulia Kovas, Philip S Dale, Elliot M Tucker-Drob, and Robert Plomin. 2020. Evidence for a unitary structure of spatial cognition beyond general intelligence. *npj Science of Learning*, 5(1):9.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjmashidi. 2021. Spartqa:: A textual question answering benchmark for spatial reasoning. *arXiv preprint arXiv:2104.05832*.

Nora Newcombe and Janellen Huttenlocher. 2000. *Making space: The development of spatial representation and reasoning*. MIT press.

Michael Ogezi and Freda Shi. 2025. Spare: Enhancing spatial reasoning in vision-language models with synthetic data. *arXiv preprint arXiv:2504.20648*.

OpenAI. 2025. Gpt-4.1 model documentation. https://platform.openai.com/docs/models/gpt-4.1. Accessed: 2025-12-20.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Letitia Parcalabescu and Anette Frank. 2024. On measuring faithfulness or self-consistency of natural language explanations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6048–6089.

Steven T Piantadosi, Harry Tily, and Edward Gibson. 2012. The communicative function of ambiguity in language. *Cognition*, 122(3):280–291.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junmei Song, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv*, abs/2402.03300.

Zhengxiang Shi, Qiang Zhang, and Aldo Lipani. 2022. 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.

Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. 2024. Rethinking interpretability in the era of large language models. *arXiv preprint arXiv:2402.01761*.

Yuda Song, Hanlin Zhang, Carson Eisenach, Sham Kakade, Dean Foster, and Udaya Ghai. 2024. Mind the gap: Examining the self-improvement capabilities of large language models. *arXiv preprint arXiv:2412.02674*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Sharon Li, and Neel Joshi. 2024. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. *Advances in Neural Information Processing Systems*, 37:75392–75421.

Sophie L Wang, Phillip Isola, and Brian Cheung. 2025. Words that make language models perceive. *arXiv preprint arXiv:2510.02425*.

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

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

Yana Wei, Liang Zhao, Jian-Yuan Sun, Kangheng Lin, Jisheng Yin, Jingcheng Hu, Yinmin Zhang, En Yu, Haoran Lv, Zejia Weng, Jia Wang, Chunrui Han, Yuang Peng, Qi Han, Zheng Ge, Xiangyu Tony Zhang, Daxin Jiang, and Vishal M. Patel. 2025. Open vision reasoner: Transferring linguistic cognitive behavior for visual reasoning. *arXiv preprint arXiv:2507.05255*.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. 2024. Mind's eye of llms: visualization-of-thought elicits spatial reasoning in large language models. *Advances in Neural Information Processing Systems*, 37:90277–90317.

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

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Hongjie Zhang, Hourui Deng, Jie Ou, and Chaosheng Feng. 2025a. Mitigating spatial hallucination in large language models for path planning via prompt engineering. *Scientific Reports*, 15(1):8881.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025b. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*.

# Appendix

## A Dataset Details

Table 5 summarizes the possible values for each field in TEXT2SPACE. The dataset is in English, covering various terminologies including cardinal directions (e.g., 'North'), clock-face references (e.g., '3 o'clock'), and standard spatial prepositions (e.g., 'left of'). Figure 3 provides an overview of the full dataset statistics, and Figure 4 shows the corresponding statistics for the test set.

| Field | Possible Values |
|---|---|
| query_type | full, vertical, horizontal |
| label (full) | above, below, left, right, upper-left, upper-right, lower-left, lower-right |
| label (vertical) | above, below, same level |
| label (horizontal) | left, right, same column |
| terminology_used | spatial, cardinal, clock, spatial+cardinal, spatial+clock, cardinal+clock, spatial+cardinal+clock |
| ascii | simple, grid, panel |
| num_components | 2–8 |
| num_relations | 1–12 |
| is_directly_stated | true, false |
| has_unique_layout | true, false |
| ambiguous_stages | 0–11 |

Table 5: TEXT2SPACE Dataset Schema, including spatial query types, terminology variations, and structural constraints (components and relations).

## B ASCII Format Choice

Using three different ASCII representations as input, we conduct two experiments to determine the most suitable format. The first experiment evaluates the interpretation of natural language descriptions from ASCII (*ASCII → Desc*), while the second assesses spatial reasoning directly from ASCII ($\mathcal{R}_{\text{ASCII}}$). As shown in Table 6, performance on *ASCII → Desc* is comparable across all three formats. In contrast, for $\mathcal{R}_{\text{ASCII}}$, the grid-based representation yields substantially higher performance. Accordingly, we adopt the grid format in all subsequent experiments.

| ASCII Format | QWEN3-30B-A3B |
|---|---|
| *ASCII → Desc* | |
| simple | **57.1** |
| grid | 55.8 |
| panel | 54.4 |
| $\mathcal{R}_{\text{ASCII}}$ | |
| simple | 76.8 |
| grid | **83.2** |
| panel | 79.2 |

Table 6: ASCII Format Result

## C Training configurations

| Configuration | Value |
|---|---|
| Base Model | QWEN3-30B-A3B |
| Frameworks | HuggingFace Transformers, TRL, PEFT |
| Fine-tuning Method | LoRA (Rank $r = 16$, $\alpha = 32$) |
| Target Modules | $q, k, v, o, gate, up, down$ projections |
| Precision | bfloat16 |
| Optimizer | AdamW |
| Learning Rate | $2 \times 10^{-4}$ (Cosine Schedule) |
| Batch Size | 4 (per device) |
| Gradient Accumulation | 4 steps |
| Epochs | 1 |
| Dataset Split | 4,000 (Train) / 500 (Validation) |
| Hardware | 4× NVIDIA A40 |

Table 7: Training configuration details.

## D Confusion matrices for spatial reasoning predictions

Figure 5 presents the confusion matrices for spatial reasoning predictions before and after fine-tuning on the ASCII–description conversion tasks. Fine-tuning results in stronger concentration along the diagonal and reduces systematic confusions between spatially related classes, indicating improved internal organization of spatial representations rather than uniform scaling of prediction confidence.

## E F1 Scores

Table 8 reports the F1 scores for spatial reasoning tasks. In line with Table 2, reasoning from ASCII alone generally outperforms reasoning from descriptions alone. While generating ASCII before answering hurts performance without fine-tuning, fine-tuning with explicit ASCII generation, either unidirectional (Desc→ASCII) or bidirectional (Desc↔ASCII), consistently improves performance when only descriptions are available at inference time.

Figure 3: **Statistical overview of the Text2Space dataset.** The figure illustrates the distribution of components and relations per instance (top left), followed by the balanced distribution of query types and spatial terminology (top right). The bottom row details the ratio of direct to indirect relations, the proportion of unique layouts, and the distribution of ambiguous stages. The summary box (bottom right) highlights key metrics, including a total of 20,000 instances with an average of 5.7 components and 6.4 relations.

Figure 4: **Statistical overview of the Test Set.** The figure illustrates the distributions for the 1,000 held-out test instances. The dataset maintains a balanced distribution of query types and spatial terminology (top right).

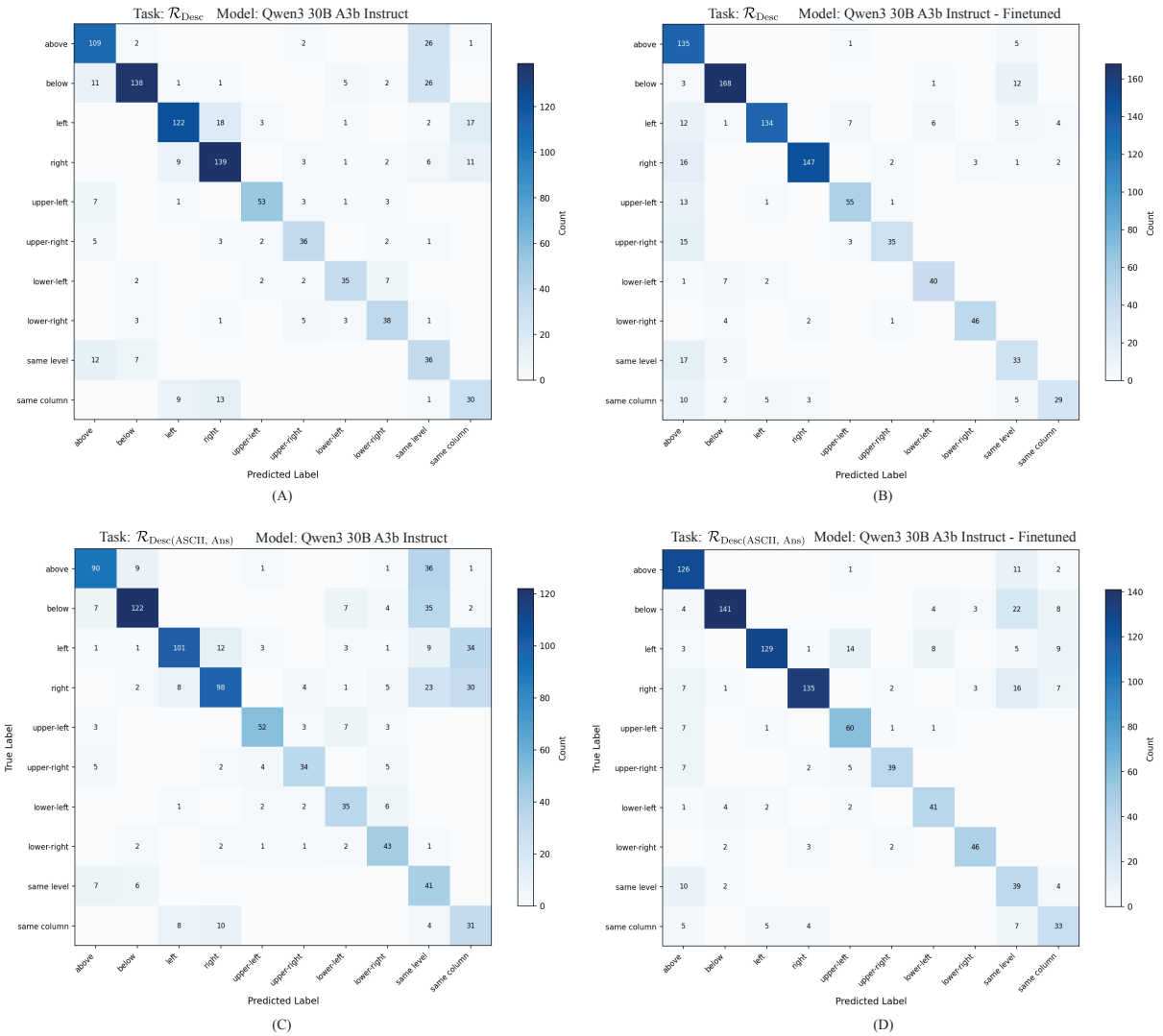


Figure 5: Confusion matrices for spatial reasoning predictions by Qwen3-30B-A3B before and after fine-tuning (Qwen3-30B-A3B $\text{FT}_{\text{D}\to\text{A}}$). The top row ((A)-(B)) shows results on the text-only spatial reasoning task ($\mathcal{R}_{\text{Desc}}$), while the bottom row ((C)-(D)) shows results when ASCII is generated as an intermediate step ($\mathcal{R}_{\text{Desc:(Ans, ASCII)}}$). Left panels ((A),(C)) correspond to the base model, and right panels ((B),(D)) to the fine-tuned model. Rows indicate ground-truth spatial relations and columns indicate predicted relations. Darker diagonal cells indicate correct predictions.

| **Model** | $\mathcal{R}_{\text{Desc}}$ | $\mathcal{R}_{\text{ASCII}}$ | $\mathcal{R}_{\text{Desc(Ans, ASCII)}}$ | $\mathcal{R}_{\text{Desc(ASCII, Ans)}}$ | $\mathcal{R}_{\text{Desc+ASCII}}$ |
|---|---|---|---|---|---|
| *One-turn generation* | | | | | |
| QWEN2.5-7B | 37.7 | 33.4 | 38.6 | 33.9 | **48.3** |
| LLAMA-3-70B | 63.3 | 62.2 | 62.3 | 52.2 | **68.8** |
| GPT-4.1 | 81.2 | 85.8 | 82.2 | 77.5 | **89.6** |
| GPT-OSS-20B | 91.9 | 92.0 | 93.2 | 90.5 | **96.8** |
| QWEN3-235B-A22B | 81.9 | 86.4 | 81.5 | 79.6 | **88.0** |
| QWEN3-30B-A3B | 72.3 | 80.3 | 69.6 | 66.2 | **81.3** |
| QWEN3-30B-A3B $\text{FT}_{\text{DESC}\to\text{ASCII}}$ | 79.4 | 72.7 | 79.5 | 76.9 | **81.9** |
| QWEN3-30B-A3B $\text{FT}_{\text{DESC}\leftarrow\text{ASCII}}$ | 71.4 | 84.1 | 67.4 | 69.4 | **84.7** |
| QWEN3-30B-A3B $\text{FT}_{\text{DESC}\leftrightarrow\text{ASCII}}$ | 78.2 | 73.3 | 80.0 | 79.3 | **82.6** |
| *Two-turn generation* | | | | | |
| QWEN2.5-7B | | | 38.6 | 38.6 | **46.9** |
| LLAMA-3-70B | | | 61.0 | 56.2 | **67.2** |
| GPT-4.1 | | | 81.0 | 76.1 | **90.0** |
| GPT-OSS-20B | | | 92.3 | 86.8 | **94.0** |
| QWEN3-235B-A22B | | | 83.0 | 73.1 | **85.0** |
| QWEN3-30B-A3B | | | 71.8 | 64.1 | **78.8** |
| QWEN3-30B-A3B $\text{FT}_{\text{DESC}\to\text{ASCII}}$ | | | 76.2 | 74.9 | **78.3** |
| QWEN3-30B-A3B $\text{FT}_{\text{DESC}\leftarrow\text{ASCII}}$ | | | 68.2 | 57.8 | **82.6** |
| QWEN3-30B-A3B $\text{FT}_{\text{DESC}\leftrightarrow\text{ASCII}}$ | | | 77.3 | 74.0 | **80.1** |

Table 8: F1-score results. The best result is shown in **bold**, and the second-best is underlined.

## F Data Generation Pipeline

Pseudo-code for the spatial dataset generation pipeline with multi-dimensional balancing.

**Algorithm 1** Text2Space Dataset Generation

**Require:**
$N_{target}$: Target count
$R_{comp}, R_{rel}$: Complexity ranges (components, relations)
**Ensure:**
$\mathcal{D}$: Final spatial dataset
$\mathcal{D} \leftarrow \emptyset$
Initialize $Tracker$ to monitor distribution
**while** $|\mathcal{D}| < N_{target}$ **do**
**1. Select Parameters**
Identify imbalanced categories using $Tracker$
$P \leftarrow$ select parameters (e.g., node count) to balance
**2. Build Connected Structure**
$V \leftarrow \{A, B, C, \dots\}$
$G \leftarrow (V, E = \emptyset)$
$C \leftarrow \{\text{Random}(V)\}$
**while** $V \setminus C \neq \emptyset$ **do**
$u \leftarrow \text{Random}(V \setminus C)$
$v \leftarrow \text{Random}(C)$
$dir \leftarrow$ Random(8 Compass Directions)
**if** edge $(u, v, dir)$ is valid **then**
$E \leftarrow E \cup \{(u, v, dir)\}$
$C \leftarrow C \cup \{u\}$
**end if**
**end while**
Add extra relations to $E$ to reach target complexity
**3. Generate Description**
$S \leftarrow \text{Random}(\{\text{Spatial, Cardinal, Clock}\})$
$Desc \leftarrow \textsc{GraphToNaturalLanguage}(G, S)$
**4. Create Query**
$\mathcal{P}_{infer} \leftarrow$ Identify uniquely inferable pairs in $G$
$(Q, A) \leftarrow$ Sample pair and Type from $\mathcal{P}_{infer}$
Determine if relationship is Direct vs. Inferred
**5. Render Visualizations**
$Vis \leftarrow \textsc{RenderASCII}(G, \{\text{Simple, Grid, Panel}\})$
**6. Apply Rejection Sampling**
$I \leftarrow \{G, Desc, Q, A, Vis, S\}$
$\Delta \leftarrow \textsc{CalcBalanceDeviation}(I, Tracker)$
**if** $I$ worsens balance **then**
**Reject** with probability $P \propto \Delta^2$
**else**
$\mathcal{D} \leftarrow \mathcal{D} \cup \{I\}$
Update $Tracker$ with instance $I$
**end if**
**end while**
**return** $\mathcal{D}$

## G Evaluation Algorithm for ASCII and Descriptions

Pseudo-code for evaluating generated descriptions and ASCII layouts.

**Algorithm 2** Evaluate Description vs. ASCII Layouts

**Require:**
$G$: ASCII Grid (Model Output)
$D$: Natural Language Description
$Mode$: "verify_ascii" or "verify_desc"
**Ensure:**
$R$: Result tuple (Pass, Accuracy, Breakdown)
**1. Parse Description**
$Segs \leftarrow \text{Split}(D, \{., , \})$
$Claims \leftarrow \emptyset, \quad Bad \leftarrow \emptyset$
**for** $s \in Segs$ **do**
Match $s$ with patterns (e.g., "A is left of B")
**if** match found **then**
$(u, v, dir) \leftarrow \text{Extract}(s)$
$Claims.add((u, v, dir))$
**else**
$Bad.add(s)$
**end if**
**end for**
$Objs_T \leftarrow \text{UniqueObjects}(Claims)$
**2. Build Grid Graph**
$Pos \leftarrow \text{ScanCoordinates}(G)$
Normalize $Pos$ (top-left at 0,0)
$Truth \leftarrow \emptyset$
**for** $(u, v) \in Pos$ **do**
$\vec{d} \leftarrow Pos[u] - Pos[v]$
$Truth[u, v] \leftarrow \text{VectorToDir}(\vec{d})$
**end for**
**3. Validate Relations**
$Corr \leftarrow \emptyset, \quad Err \leftarrow \emptyset$
**for** $(u, v, dir) \in Claims$ **do**
**if** $u \notin Pos$ **or** $v \notin Pos$ **then**
$Err.add((u, v, \text{"Missing"}))$
**else**
**if** $dir == Truth[u, v]$ **then**
$Corr.add((u, v, dir))$
**else**
$Err.add((u, v, Truth[u, v]))$
**end if**
**end if**
**end for**
**4. Final Scoring**
**if** $Mode$ == "verify_desc" **then** ▷ ASCII is Truth
$Pass \leftarrow (Err = \emptyset)$
**else** ▷ Text is Truth
$Extra \leftarrow Pos.keys \setminus Objs_T$
$Pass \leftarrow (Err = \emptyset \wedge Bad = \emptyset \wedge Extra = \emptyset)$
**end if**
$Acc \leftarrow |Corr|/|Claims|$
**return** $\{Pass, Acc, Corr, Err, Bad\}$

## H Evaluation Algorithm for Consistency

Pseudo-code for evaluating the consistency between generated ASCII layouts and query relations.

**Algorithm 3** Evaluate Consistency Between ASCII Grid and Reference relations

**Require:**
$G$: ASCII representation (Model Grid)
$Q$: Natural language query
$L$: Reference label (relation)
**Ensure:**
$R$: Evaluation result (correctness, objects, relations)
**1. Parse the Query**
Match $Q$ vs. patterns: "Vertical", "Horizontal", or "Full"
Extract target objects $X$ and $Y$ from $Q$
**if** no valid pattern found **then**
    **return** ERROR("Query parsing failed")
**end if**
**2. Parse the ASCII Grid**
Scan $G$ to find coordinates for all characters
$Pos \leftarrow \{(obj, x, y) \mid obj \text{ found in } G\}$
**if** $X \notin Pos$ **or** $Y \notin Pos$ **then**
    **return** ERROR("Queried object missing in grid")
**end if**
Normalize $Pos$ so top-left object is at $(0, 0)$
**3. Infer Relation from Grid**
$(x_1, y_1) \leftarrow Pos[X], \quad (x_2, y_2) \leftarrow Pos[Y]$
$Rel_{inf} \leftarrow \emptyset$
**if** Type is "Vertical" **then**
    **if** $y_1 < y_2$ **then** $Rel_{inf} \leftarrow$ "above"
    **else if** $y_1 > y_2$ **then** $Rel_{inf} \leftarrow$ "below"
    **else** $Rel_{inf} \leftarrow$ "same level"
    **end if**
**else if** Type is "Horizontal" **then**
    **if** $x_1 < x_2$ **then** $Rel_{inf} \leftarrow$ "left"
    **else if** $x_1 > x_2$ **then** $Rel_{inf} \leftarrow$ "right"
    **else** $Rel_{inf} \leftarrow$ "same column"
    **end if**
**else** ▷ Full Spatial Query
    Compute vector $\vec{v} = (x_1 - x_2, y_1 - y_2)$
    $Rel_{inf} \leftarrow$ Map $\vec{v}$ to {upper-left, below, right, ...}
**end if**
**4. Compare with Reference Label**
$Match \leftarrow (\text{LowerCase}(Rel_{inf}) == \text{LowerCase}(L))$
$R \leftarrow \{\text{Pass} : Match, \text{Exp} : L, \text{Act} : Rel_{inf}\}$
**return** $R$

## I Human Validation

**Human Validation Setup.** To quantify the impact of different input formats on spatial reasoning accuracy, we conducted a controlled human validation study with 11 evaluators divided into two groups: Team A ($n = 7$) and Team B ($n = 4$). We curated three distinct problem sets (Set X, Set Y, and Set Z), each containing 10 spatially complex examples. The design details is shown in Table 9

The study utilized a counterbalanced within-subjects design to disentangle the effect of input format from question difficulty. All participants completed three sessions:

1. **Session 2 (Visual Baseline):** Both teams evaluated Set Z using the **ASCII-only** format. Performance was highly consistent across groups (Team A: 0.986, Team B: 0.950), confirming that both teams possessed comparable visual reasoning baselines.

2. **Sessions 1 & 3 (Crossover):** We swapped the assignment of problem sets and formats between teams. Team A solved Set X using **Description-only** (Session 1) and Set Y using **Description+ASCII** (Session 3). Conversely, Team B solved Set Y using **Description-only** (Session 1) and Set X using **Description+ASCII** (Session 3).

This cross-over design ensures that Sets X and Y were evaluated in both text-only and combined formats, allowing for a direct comparison of performance on semantically identical questions.

| **Task Sessions** | **Team A** ($n = 7$) | **Team B** ($n = 4$) |
|---|---|---|
| **Session 1** *(Single Format)* | **Description Only** (Set X) | **Description Only** (Set Y) |
| **Session 2** *(Visual Baseline)* | **ASCII Only** (Set Z) | **ASCII Only** (Set Z) |
| **Session 3** *(Combined)* | **Desc + ASCII** (Set Y) | **Desc + ASCII** (Set X) |

Table 9: Human Validation Study Design. The experiment followed a three-session protocol with counterbalanced question sets.

**Validation Results** Table 10 shows the substantial improvement achieved when reasoning from an ASCII-augmented description compared to reasoning from the description alone.

| **Question Set** | **Desc Only** (NL Format) | **Desc + ASCII** (Combined) | **Improvement** |
|---|---|---|---|
| Set X | 0.857 (Team A) | 0.975 (Team B) | +11.8% |
| Set Y | 0.875 (Team B) | 0.986 (Team A) | +11.1% |
| **Average** | **0.866** | **0.981** | **+11.5%** |

Table 10: Human performance comparison across different input formats. By counterbalancing Sets X and Y, we isolate the improvement attributed to the format.

**Participants and Ethics** All participants were paid at an hourly rate consistent with local labor standards. The research involves benign behavioral tasks with no collection of personally identifiable information (PII) and currently under review for an IRB exemption.

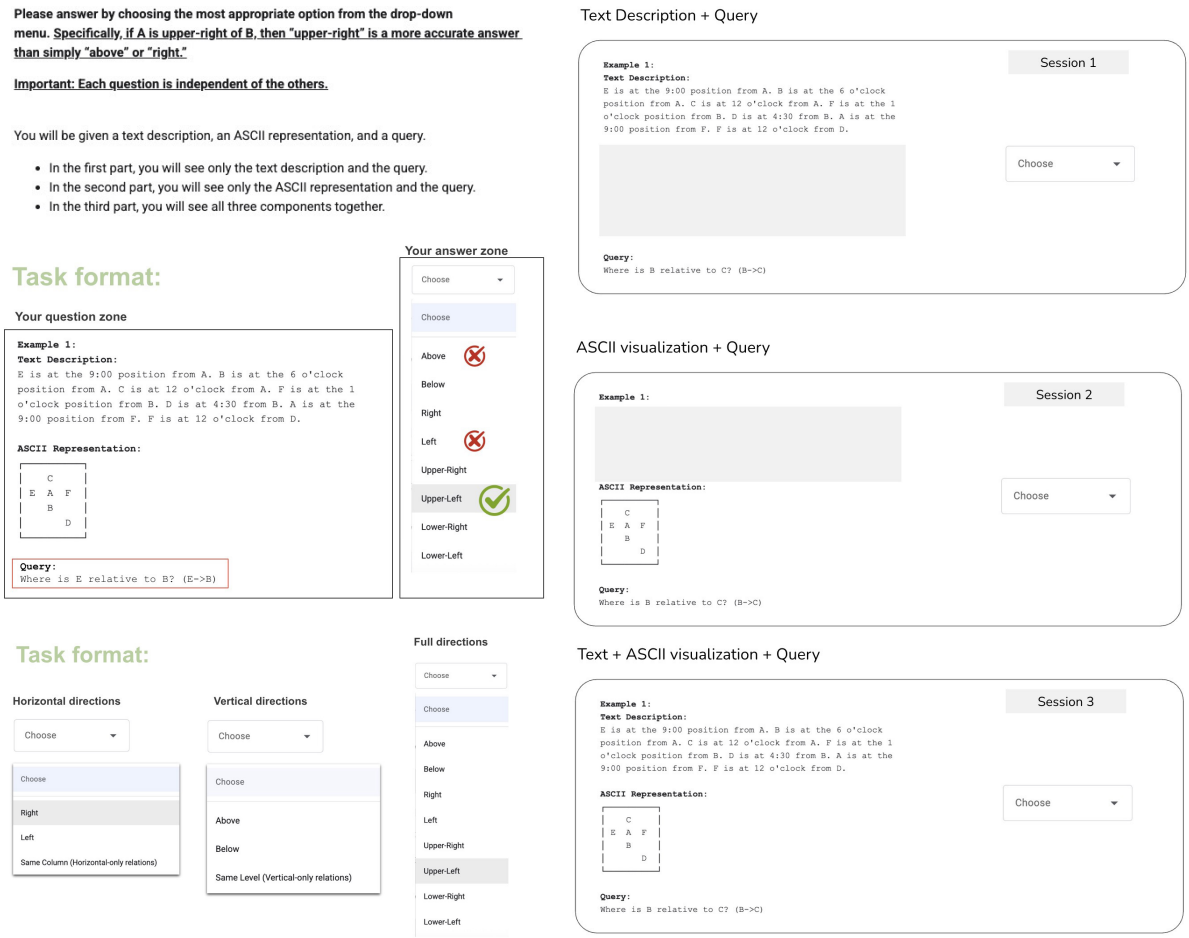


Figure 6: Overview of the human validation interface and instruction protocol. Participants solved spatial reasoning tasks across three counterbalanced sessions: (1) Text Description Only, (2) ASCII Visualization Only, and (3) Combined Text + ASCII. We use Google Form to collect the response from the participants. The interface requires participants to select the most precise directional term from a standardized drop-down menu.

# J Prompts and Outputs

## J.1 Task Description → ASCII

```
---------
SYSTEM
---------
Assume you have expertise in spatial
reasoning. Your task is to answer the
following question to the best of your
knowledge.
You will receive a scene description.
Your job is to generate the requested
ASCII visualization and return the
result as a compact JSON object.

Description -> ASCII Diagram.
Read the textual scene description and
render the requested ASCII layout.

- Treat the canvas like a map: the first
 printed row is the top, and the last
row is the bottom.
- The first column is the left edge; the
 last column is the right edge.
- Above = higher row; below = lower row
(do not invert the vertical axis).
- Left = smaller column index; right =
larger column index (do not mirror
horizontally).

Formats:
- grid: <grid using + - | for borders>

Rendering & Positioning Rules

Use given uppercase letters (A, B, C,
...); don't invent new ones.
Grid size is flexible - expand rows/cols
 as needed to fit all points.
Place all letters; if two overlap, stack
 them (e.g., "AF").

Relations are first -> second (e.g., "D
to F" means D relative to F).

Positioning:
above/below -> same column
left/right -> same level
diagonals (upper/lower-left/right) ->
one cell diagonally.
Keep perfect alignment for cardinal
directions.
Adjacent items occupy neighboring cells;
 no diagonal drift.
Maintain uniform cell sizes and
consistent borders.

Keep row alignment when depicting left/
right relations; do not shift labels
vertically.
Keep column alignment when depicting
above/below relations; do not shift
labels horizontally.
For diagonal relations, ensure the two
entities occupy different rows and
different columns.
Avoid inconsistent cell widths and do
not introduce extra or missing letters.

Additional ASCII Guidance:
- Label every entity exactly as named in
 the description.

Input Notes:
- Use only the provided description to
infer entity positions.
- Do not invent additional entities or
relations when information is missing.

Output Notes:
- Produce ASCII views that faithfully
match the described scene.

OUTPUT SCHEMA (JSON):
{
  "ascii": {
    "grid": "<grid using + - | for
    borders>"
  }
}

Return ONLY a compact JSON object.
Do not include markdown fences or any
extra text.

---------
USER
---------
Description:
B is to the lower-right of A. D is to
the right of B. C is at the bottom-right
 of A. C is located below B.

---------
ASSISTANT
---------
{"ascii":{"grid":"+--+--+--+\n|A |  |
|\n+--+--+--+\n|  |B |D |\n+--+--+--+\n|
  |C |  |\n+--+--+--+"}}
```

## J.2 Task ASCII → Description

```
---------
SYSTEM
---------
Assume you have expertise in spatial
reasoning. Your task is to answer the
following question to the best of your
knowledge.
You will receive ASCII diagram(s).
Your job is to write a concise spatial
summary and return the result as a
compact JSON object.

ASCII Diagram -> Description.
Interpret the ASCII evidence and rewrite
 it as a concise natural-language
description.

- Treat the canvas like a map: the first
 printed row is the top, and the last
row is the bottom.
- The first column is the left edge; the
 last column is the right edge.
- Above = higher row; below = lower row
(do not invert the vertical axis).
- Left = smaller column index; right =
larger column index (do not mirror
horizontally).

Formats:
- grid: <grid using + - | for borders>

Input Notes:
- Treat the ASCII diagram as
authoritative evidence.

Output Notes:
- Summarize the spatial arrangement in
clear declarative sentences.

Description Guidelines:
- Mention every entity shown in the
ASCII input and describe their relative
```

```
placement.
- Keep the description short and factual
; avoid speculative language.
- The spatial summary will consist of
sentences in the format "A is [relation]
 B." Each sentence should describe the
relationship between exactly two objects
.

OUTPUT SCHEMA (JSON):
{
  "description": "<natural language
  spatial summary>"
}

Return ONLY a compact JSON object.
Do not include markdown fences or any
extra text.

---------
USER
---------
ASCII Reference:
grid:
+--+--+--+
|A |  |  |
+--+--+--+
|  |B |D |
+--+--+--+
|  |C |  |
+--+--+--+

---------
ASSISTANT
---------
{"description":"B is to the lower-right
of A. D is to the right of B. C is at
the bottom-right of A. C is located
below B."}
```

### J.3 Task $\mathcal{R}_{\text{Desc}}$

```
---------
SYSTEM
---------
Assume you have expertise in spatial
reasoning. Your task is to answer the
following question to the best of your
knowledge.
You will receive a directional question
and a scene description.
Your job is to produce the correct
directional answer and return the result
 as a compact JSON object.

Note that the possible directions are
defined as {left, right, above, below,
upper-left, upper-right, lower-left,
lower-right, same column, same level},
and there will be only one correct
answer.

Description + Query -> Answer.
Answer the spatial relationship question
 using the textual description.

10 Distinct (Mutually Exclusive)
Directions
Pure Directions - perfectly aligned:
- left: directly left, no vertical
offset
- right: directly right, no vertical
offset
- above: directly above, no horizontal
offset
- below: directly below, no vertical
offset
Diagonal Directions - both vertical and
horizontal offset: upper-left, upper-
right, lower-left, lower-right
Horizontal Alignment ONLY: same level;
Vertical Alignment ONLY: same column

Input Notes:
- Use the description to reason about
the positions of the queried entities.

Output Notes:
- Return the spatial relation as a
single direction string.

Direction Notes:
- Ensure the answer directly references
the entities named in the question.

OUTPUT SCHEMA (JSON):
{
  "answer": "<direction>"
}

Return ONLY a compact JSON object.
Do not include markdown fences or any
extra text.

---------
USER
---------
Description:
B is to the lower-right of A. D is to
the right of B. C is at the bottom-right
 of A. C is located below B.

Question: Where is C horizontally
relative to D?

---------
ASSISTANT
---------
{"answer":"left"}
```

### J.4 Task $\mathcal{R}_{\text{ASCII}}$

```
---------
SYSTEM
---------
Assume you have expertise in spatial
reasoning. Your task is to answer the
following question to the best of your
knowledge.
You will receive a directional question
and ASCII diagram(s).
Your job is to produce the correct
directional answer and return the result
 as a compact JSON object.

Note that the possible directions are
defined as {left, right, above, below,
upper-left, upper-right, lower-left,
lower-right, same column, same level},
and there will be only one correct
answer.

ASCII + Query -> Answer.
Answer the spatial relationship question
 using the ASCII diagram as the primary
evidence.

10 Distinct (Mutually Exclusive)
Directions
Pure Directions - perfectly aligned:
- left: directly left, no vertical
offset
- right: directly right, no vertical
offset
- above: directly above, no horizontal
offset
```

```
- below: directly below, no vertical
offset
Diagonal Directions - both vertical and
horizontal offset: upper-left, upper-
right, lower-left, lower-right
Horizontal Alignment ONLY: same level;
Vertical Alignment ONLY: same column

- Treat the canvas like a map: the first
 printed row is the top, and the last
row is the bottom.
- The first column is the left edge; the
 last column is the right edge.
- Above = higher row; below = lower row
(do not invert the vertical axis).
- Left = smaller column index; right =
larger column index (do not mirror
horizontally).

Formats:
- grid: <grid using + - | for borders>

Input Notes:
- Use the ASCII diagram as ground truth;
 the query specifies the entities to
compare.

Output Notes:
- Provide a single directional answer
from the defined set.

Direction Notes:
- Base the answer on the ASCII layout;
ignore assumptions that contradict the
diagram.

OUTPUT SCHEMA (JSON):
{
  "answer": "<direction>"
}

Return ONLY a compact JSON object.
Do not include markdown fences or any
extra text.

---------
USER
---------
ASCII Reference:
grid:
+--+--+--+
|A |  |  |
+--+--+--+
|  |B |D |
+--+--+--+
|  |C |  |
+--+--+--+

Question: Where is C horizontally
relative to D?

---------
ASSISTANT
---------
{"answer":"left"}
```

### J.5 Task $\mathcal{R}_{\text{Desc(Ans, ASCII)}}$

*One-turn Generation*

```
---------
SYSTEM
---------
Assume you have expertise in spatial
reasoning. Your task is to answer the
following question to the best of your
knowledge.
You will receive a directional question
and a scene description.
Your job is to produce the correct
directional answer and generate the
requested ASCII visualization and return
 the result as a compact JSON object.

Note that the possible directions are
defined as {left, right, above, below,
upper-left, upper-right, lower-left,
lower-right, same column, same level},
and there will be only one correct
answer.

Description + Query -> ASCII + Answer.
Answer the spatial question and draw
ASCII that matches the same layout.

10 Distinct (Mutually Exclusive)
Directions
Pure Directions - perfectly aligned:
- left: directly left, no vertical
offset
- right: directly right, no vertical
offset
- above: directly above, no horizontal
offset
- below: directly below, no vertical
offset
Diagonal Directions - both vertical and
horizontal offset: upper-left, upper-
right, lower-left, lower-right
Horizontal Alignment ONLY: same level;
Vertical Alignment ONLY: same column

- Treat the canvas like a map: the first
 printed row is the top, and the last
row is the bottom.
- The first column is the left edge; the
 last column is the right edge.
- Above = higher row; below = lower row
(do not invert the vertical axis).
- Left = smaller column index; right =
larger column index (do not mirror
horizontally).

Formats:
- grid: <grid using + - | for borders>

Rendering & Positioning Rules

Use given uppercase letters (A, B, C,
...); don't invent new ones.
Grid size is flexible - expand rows/cols
 as needed to fit all points.
Place all letters; if two overlap, stack
 them (e.g., "AF").

Relations are first -> second (e.g., "D
to F" means D relative to F).

Positioning:
above/below -> same column
left/right -> same level
diagonals (upper/lower-left/right) ->
one cell diagonally.
Keep perfect alignment for cardinal
directions.
Adjacent items occupy neighboring cells;
 no diagonal drift.
Maintain uniform cell sizes and
consistent borders.

Keep row alignment when depicting left/
right relations; do not shift labels
vertically.
Keep column alignment when depicting
above/below relations; do not shift
```

```
labels horizontally.
For diagonal relations, ensure the two
entities occupy different rows and
different columns.
Avoid inconsistent cell widths and do
not introduce extra or missing letters.

Additional ASCII Guidance:
- Ensure the ASCII visualization aligns
with the answer you provide.

Determine the correct answer first, then
 draw ASCII that matches it.

Input Notes:
- Use the description to understand the
scene before answering.
- Ensure the ASCII rendering is
consistent with the textual evidence.

Output Notes:
- Provide both the answer and the
requested ASCII views in the JSON output
.

Direction Notes:
- Check the ASCII you generate so it
supports the chosen answer.

OUTPUT SCHEMA (JSON):
{
  "answer": "<direction>",
  "ascii": {
    "grid": "<grid using + - | for
    borders>"
  }
}

Return ONLY a compact JSON object.
Do not include markdown fences or any
extra text.

---------
USER
---------
Description:
B is to the lower-right of A. D is to
the right of B. C is at the bottom-right
 of A. C is located below B.

Question: Where is C horizontally
relative to D?

---------
ASSISTANT
---------
{"answer":"left","ascii":{"grid
":"+--+--+--+\n|A |  |  |\n+--+--+--+\n|
  |B |D |\n+--+--+--+\n|  |C |  |\n
+--+--+--+"}}
```

*Two-turn Generation*

```
---------
SYSTEM
---------
Assume you have expertise in spatial
reasoning. Your task is to answer the
following question to the best of your
knowledge.
You will receive a directional question
and a scene description.
Your task is to provide both the correct
 directional answer and the
corresponding ASCII visualization,
including both in a single compact JSON
object.

This is a two-turn task: Answer First,
Then Generate ASCII.

- Turn 1: Answer the spatial question
based on reasoning.
- Turn 2: Generate an ASCII
visualization that matches your answer.

## Turn 1: Answer the Question

The possible directions are {left, right
, above, below, upper-left, upper-right,
 lower-left, lower-right, same column,
same level}. There is only one correct
answer.

10 Distinct (Mutually Exclusive)
Directions
Pure Directions - perfectly aligned:
- left: directly left, no vertical
offset
- right: directly right, no vertical
offset
- above: directly above, no horizontal
offset
- below: directly below, no vertical
offset
Diagonal Directions - both vertical and
horizontal offset: upper-left, upper-
right, lower-left, lower-right
Horizontal Alignment ONLY: same level;
Vertical Alignment ONLY: same column

## Turn 2: Generate ASCII Visualization

Formats:
- grid: <grid using + - | for borders>

Rendering & Positioning Rules

Use given uppercase letters (A, B, C,
...); don't invent new ones.
Grid size is flexible - expand rows/cols
 as needed to fit all points.
Place all letters; if two overlap, stack
 them (e.g., "AF").

Relations are first -> second (e.g., "D
to F" means D relative to F).

Positioning:
above/below -> same column
left/right -> same level
diagonals (upper/lower-left/right) ->
one cell diagonally.
Keep perfect alignment for cardinal
directions.
Adjacent items occupy neighboring cells;
 no diagonal drift.
Maintain uniform cell sizes and
consistent borders.

Keep row alignment when depicting left/
right relations; do not shift labels
vertically.
Keep column alignment when depicting
above/below relations; do not shift
labels horizontally.
For diagonal relations, ensure the two
entities occupy different rows and
different columns.
Avoid inconsistent cell widths and do
not introduce extra or missing letters.

- Treat the canvas like a map: the first
 printed row is the top, and the last
row is the bottom.
- The first column is the left edge; the
 last column is the right edge.
```

```
- Above = higher row; below = lower row
(do not invert the vertical axis).
- Left = smaller column index; right =
larger column index (do not mirror
horizontally).

Output Notes:
- Determine the answer based on spatial
reasoning from the description.
- Then create ASCII that accurately
reflects your answer.
- Turn 1: Provide the directional answer
 as a JSON object with "answer" key.
- Turn 2: Provide the ASCII views as a
JSON object with "ascii" key.

OUTPUT SCHEMA (JSON):
{
  "answer": "<direction>",
  "ascii": {
    "grid": "<grid using + - | for
    borders>"
  }
}

Return ONLY a compact JSON object.
Do not include markdown fences or any
extra text.

---------
USER
---------
Description:
B is to the lower-right of A. D is to
the right of B. C is at the bottom-right
 of A. C is located below B.

Question: Where is C horizontally
relative to D?

Please answer this question.

---------
ASSISTANT
---------
{"answer":"left"}

---------
USER
---------
Now, generate an ASCII visualization
that matches your answer and shows this
spatial arrangement.

---------
ASSISTANT
---------
{"ascii":{"grid":"+--+--+--+\n|A |  |
|\n+--+--+--+\n|  |B |D |\n+--+--+--+\n|
  |C |  |\n+--+--+--+"}}
```

### J.6 Task $\mathcal{R}_{\text{Desc(ASCII, Ans)}}$

*One-turn Generation*

```
---------
SYSTEM
---------
Assume you have expertise in spatial
reasoning. Your task is to answer the
following question to the best of your
knowledge.
You will receive a directional question
and a scene description.
Your job is to produce the correct
directional answer and generate the
requested ASCII visualization and return
 the result as a compact JSON object.

Note that the possible directions are
defined as {left, right, above, below,
upper-left, upper-right, lower-left,
lower-right, same column, same level},
and there will be only one correct
answer.

Description + Query -> ASCII + Answer.
Answer the spatial question and draw
ASCII that matches the same layout.

10 Distinct (Mutually Exclusive)
Directions
Pure Directions - perfectly aligned:
- left: directly left, no vertical
offset
- right: directly right, no vertical
offset
- above: directly above, no horizontal
offset
- below: directly below, no vertical
offset
Diagonal Directions - both vertical and
horizontal offset: upper-left, upper-
right, lower-left, lower-right
Horizontal Alignment ONLY: same level;
Vertical Alignment ONLY: same column

- Treat the canvas like a map: the first
 printed row is the top, and the last
row is the bottom.
- The first column is the left edge; the
 last column is the right edge.
- Above = higher row; below = lower row
(do not invert the vertical axis).
- Left = smaller column index; right =
larger column index (do not mirror
horizontally).

Formats:
- grid: <grid using + - | for borders>

Rendering & Positioning Rules

Use given uppercase letters (A, B, C,
...); don't invent new ones.
Grid size is flexible - expand rows/cols
 as needed to fit all points.
Place all letters; if two overlap, stack
 them (e.g., "AF").

Relations are first -> second (e.g., "D
to F" means D relative to F).

Positioning:
above/below -> same column
left/right -> same level
diagonals (upper/lower-left/right) ->
one cell diagonally.
Keep perfect alignment for cardinal
directions.
Adjacent items occupy neighboring cells;
 no diagonal drift.
Maintain uniform cell sizes and
consistent borders.

Keep row alignment when depicting left/
right relations; do not shift labels
vertically.
Keep column alignment when depicting
above/below relations; do not shift
labels horizontally.
For diagonal relations, ensure the two
entities occupy different rows and
different columns.
Avoid inconsistent cell widths and do
not introduce extra or missing letters.
```

```
Additional ASCII Guidance:
- Ensure the ASCII visualization aligns
with the answer you provide.

Generate the ASCII visualization before
finalizing the answer.

Input Notes:
- Use the description to understand the
scene before answering.
- Ensure the ASCII rendering is
consistent with the textual evidence.

Output Notes:
- Provide both the answer and the
requested ASCII views in the JSON output
.

Direction Notes:
- Check the ASCII you generate so it
supports the chosen answer.

OUTPUT SCHEMA (JSON):
{
  "ascii": {
    "grid": "<grid using + - | for
    borders>"
  },
  "answer": "<direction>"
}

Return ONLY a compact JSON object.
Do not include markdown fences or any
extra text.

---------
USER
---------
Description:
B is to the lower-right of A. D is to
the right of B. C is at the bottom-right
 of A. C is located below B.

Question: Where is C horizontally
relative to D?

---------
ASSISTANT
---------
{"ascii":{"grid":"+--+--+--+\n|A |  |
|\n+--+--+--+\n|  |B |D |\n+--+--+--+\n|
  |C |  |\n+--+--+--+"},"answer":"left"}
```

*Two-turn Generation*

```
---------
SYSTEM
---------
Assume you have expertise in spatial
reasoning. Your task is to answer the
following question to the best of your
knowledge.
You will receive a directional question
and a scene description.
Your task is to provide both the correct
 directional answer and the
corresponding ASCII visualization,
including both in a single compact JSON
object.

This is a two-turn task: Generate ASCII
First, Then Answer.

- Turn 1: Generate an ASCII
visualization from the scene description
.
- Turn 2: Answer the spatial question
using the ASCII you generated as a
reference.

## Turn 1: Generate ASCII Visualization

Formats:
- grid: <grid using + - | for borders>

Rendering & Positioning Rules

Use given uppercase letters (A, B, C,
...); don't invent new ones.
Grid size is flexible - expand rows/cols
 as needed to fit all points.
Place all letters; if two overlap, stack
 them (e.g., "AF").

Relations are first -> second (e.g., "D
to F" means D relative to F).

Positioning:
above/below -> same column
left/right -> same level
diagonals (upper/lower-left/right) ->
one cell diagonally.
Keep perfect alignment for cardinal
directions.
Adjacent items occupy neighboring cells;
 no diagonal drift.
Maintain uniform cell sizes and
consistent borders.

Keep row alignment when depicting left/
right relations; do not shift labels
vertically.
Keep column alignment when depicting
above/below relations; do not shift
labels horizontally.
For diagonal relations, ensure the two
entities occupy different rows and
different columns.
Avoid inconsistent cell widths and do
not introduce extra or missing letters.

- Treat the canvas like a map: the first
 printed row is the top, and the last
row is the bottom.
- The first column is the left edge; the
 last column is the right edge.
- Above = higher row; below = lower row
(do not invert the vertical axis).
- Left = smaller column index; right =
larger column index (do not mirror
horizontally).

## Turn 2: Answer the Question

The possible directions are {left, right
, above, below, upper-left, upper-right,
 lower-left, lower-right, same column,
same level}. There is only one correct
answer.

10 Distinct (Mutually Exclusive)
Directions
Pure Directions - perfectly aligned:
- left: directly left, no vertical
offset
- right: directly right, no vertical
offset
- above: directly above, no horizontal
offset
- below: directly below, no vertical
offset
Diagonal Directions - both vertical and
horizontal offset: upper-left, upper-
right, lower-left, lower-right
Horizontal Alignment ONLY: same level;
Vertical Alignment ONLY: same column
```

```
Output Notes:
- Generate the ASCII visualization
before attempting to answer.
- Use the ASCII you generated to
determine the correct answer.
- Turn 1: Provide the ASCII views as a
JSON object with "ascii" key.
- Turn 2: Provide the directional answer
 as a JSON object with "answer" key.

OUTPUT SCHEMA (JSON):
{
  "answer": "<direction>",
  "ascii": {
    "grid": "<grid using + - | for
    borders>"
  }
}

Return ONLY a compact JSON object.
Do not include markdown fences or any
extra text.

---------
USER
---------
Description:
B is to the lower-right of A. D is to
the right of B. C is at the bottom-right
 of A. C is located below B.

Question: Where is C horizontally
relative to D?

Please generate an ASCII visualization
of this scene.

---------
ASSISTANT
---------
{"ascii":{"grid":"+--+--+--+\n|A |  |
|\n+--+--+--+\n|  |B |D |\n+--+--+--+\n|
  |C |  |\n+--+--+--+"}}

---------
USER
---------
Now, based on the ASCII visualization
you generated, answer the question:
Where is C horizontally relative to D?

---------
ASSISTANT
---------
{"answer":"left"}
```

### J.7 Task $\mathcal{R}_{\text{Desc:(ASCII, Ans)}}$

*One-turn Generation*

```
---------
SYSTEM
---------
Assume you have expertise in spatial
reasoning. Your task is to answer the
following question to the best of your
knowledge.
You will receive a directional question,
 a scene description, and ASCII diagram(
s).
Your job is to produce the correct
directional answer and return the result
 as a compact JSON object.

Note that the possible directions are
defined as {left, right, above, below,
upper-left, upper-right, lower-left,
lower-right, same column, same level},
and there will be only one correct
answer.

Description + ASCII + Query -> Answer.
Combine textual and ASCII evidence to
answer the spatial question.

10 Distinct (Mutually Exclusive)
Directions
Pure Directions - perfectly aligned:
- left: directly left, no vertical
offset
- right: directly right, no vertical
offset
- above: directly above, no horizontal
offset
- below: directly below, no vertical
offset
Diagonal Directions - both vertical and
horizontal offset: upper-left, upper-
right, lower-left, lower-right
Horizontal Alignment ONLY: same level;
Vertical Alignment ONLY: same column

- Treat the canvas like a map: the first
 printed row is the top, and the last
row is the bottom.
- The first column is the left edge; the
 last column is the right edge.
- Above = higher row; below = lower row
(do not invert the vertical axis).
- Left = smaller column index; right =
larger column index (do not mirror
horizontally).

Formats:
- grid: <grid using + - | for borders>

Input Notes:
- Cross-check the description and ASCII
diagram; treat ASCII as authoritative
when conflicts arise.

Output Notes:
- Output a single canonical direction
that matches the combined evidence.

Direction Notes:
- If the two evidence sources disagree,
follow the ASCII diagram.

OUTPUT SCHEMA (JSON):
{
  "answer": "<direction>"
}

Return ONLY a compact JSON object.
Do not include markdown fences or any
extra text.

---------
USER
---------
Description:
B is to the lower-right of A. D is to
the right of B. C is at the bottom-right
 of A. C is located below B.

ASCII Reference:
grid:
+--+--+--+
|A |  |  |
+--+--+--+
|  |B |D |
+--+--+--+
|  |C |  |
+--+--+--+
```

```
Question: Where is C horizontally
relative to D?

---------
ASSISTANT
---------
{"answer":"left"}
```

*Two-turn Generation* with ground truth ASCII

```
---------
SYSTEM
---------
Assume you have expertise in spatial
reasoning. Your task is to answer the
following question to the best of your
knowledge.
You will receive a directional question
and a scene description.
Your task is to provide both the correct
 directional answer and the
corresponding ASCII visualization,
including both in a single compact JSON
object.

This is a two-turn task: Generate ASCII
First, Then Answer.

- Turn 1: Generate an ASCII
visualization from the scene description
.
- Turn 2: Answer the spatial question
using the ASCII you generated as a
reference.

## Turn 1: Generate ASCII Visualization

Formats:
- grid: <grid using + - | for borders>

Rendering & Positioning Rules

Use given uppercase letters (A, B, C,
...); don't invent new ones.
Grid size is flexible - expand rows/cols
 as needed to fit all points.
Place all letters; if two overlap, stack
 them (e.g., "AF").

Relations are first -> second (e.g., "D
to F" means D relative to F).

Positioning:
above/below -> same column
left/right -> same level
diagonals (upper/lower-left/right) ->
one cell diagonally.
Keep perfect alignment for cardinal
directions.
Adjacent items occupy neighboring cells;
 no diagonal drift.
Maintain uniform cell sizes and
consistent borders.

Keep row alignment when depicting left/
right relations; do not shift labels
vertically.
Keep column alignment when depicting
above/below relations; do not shift
labels horizontally.
For diagonal relations, ensure the two
entities occupy different rows and
different columns.
Avoid inconsistent cell widths and do
not introduce extra or missing letters.

- Treat the canvas like a map: the first
 printed row is the top, and the last
row is the bottom.
- The first column is the left edge; the
 last column is the right edge.
- Above = higher row; below = lower row
(do not invert the vertical axis).
- Left = smaller column index; right =
larger column index (do not mirror
horizontally).

## Turn 2: Answer the Question

The possible directions are {left, right
, above, below, upper-left, upper-right,
 lower-left, lower-right, same column,
same level}. There is only one correct
answer.

10 Distinct (Mutually Exclusive)
Directions
Pure Directions - perfectly aligned:
- left: directly left, no vertical
offset
- right: directly right, no vertical
offset
- above: directly above, no horizontal
offset
- below: directly below, no vertical
offset
Diagonal Directions - both vertical and
horizontal offset: upper-left, upper-
right, lower-left, lower-right
Horizontal Alignment ONLY: same level;
Vertical Alignment ONLY: same column

Output Notes:
- Generate the ASCII visualization
before attempting to answer.
- Use the ASCII you generated to
determine the correct answer.
- Turn 1: Provide the ASCII views as a
JSON object with "ascii" key.
- Turn 2: Provide the directional answer
 as a JSON object with "answer" key.

OUTPUT SCHEMA (JSON):
{
  "answer": "<direction>",
  "ascii": {
    "grid": "<grid using + - | for
    borders>"
  }
}

Return ONLY a compact JSON object.
Do not include markdown fences or any
extra text.

---------
USER
---------
Description:
B is to the lower-right of A. D is to
the right of B. C is at the bottom-right
 of A. C is located below B.

Question: Where is C horizontally
relative to D?

Please generate an ASCII visualization
of this scene.

---------
ASSISTANT
---------
{"ascii":{"grid":"+--+--+--+\n|A |  |
|\n+--+--+--+\n|  |B |D |\n+--+--+--+\n|
  |C |  |\n+--+--+--+"}}
```

```
---------
USER
---------
Now, based on the ASCII visualization
you generated, answer the question:
Where is C horizontally relative to D?

---------
ASSISTANT
---------
{"answer":"left"}
```

## K Artifacts, Licensing, and Intended Use

In this work, we introduce a new dataset and utilize several existing open-source artifacts. Below, we detail the licensing, terms of use, and intended usage for each component to ensure transparency and compliance with responsible AI research practices.

### K.1 Our Contributions: The TEXT2SPACE Dataset

**License and Terms** The TEXT2SPACE dataset will be released under the **CC-BY 4.0 International License**, allowing for free use and redistribution with appropriate attribution. The accompanying code base, including the data generation pipeline and evaluation scripts, is released under the **MIT License**. We grant permission for researchers to modify and use these resources for both academic and commercial purposes, provided that the original paper is cited.

**Intended Use** The dataset is designed primarily to evaluate and enhance the spatial reasoning capabilities of Large Language Models (LLMs). It serves as a diagnostic benchmark for assessing specific spatial skills such as orientation, relative positioning, and transitive inference.

**Limitations and Misuse Potential** As the dataset consists entirely of synthetically generated spatial descriptions and ASCII diagrams using geometric primitives, it contains no personally identifiable information (PII) or offensive content. We foresee no substantial risk of misuse. However, users should note that the discrete grid representation is an abstraction and may not fully transfer to continuous real-world spatial tasks, such as autonomous driving or physical navigation validation.

### K.2 Use of Existing Artifacts

We utilize publicly available models, datasets, and software libraries in accordance with their respective licenses and terms of use.

**Models** We employ the following models for training and evaluation:

- **Open-sourced Models:** We utilize LLAMA-3-70B-INSTRUCT (AI@Meta, 2024),GPT-OSS-20B (OpenAI, 2025), and Qwen Models (QWEN2.5-7B-INSTRUCT (Yang et al., 2024; Team, 2024), QWEN3-30B-A3B-INSTRUCT-2507, QWEN3-235B-A22B-INSTRUCT-2507 (Team, 2025)). These are open-weights models, and we adhere to their community license agreements, utilizing them strictly for academic research purposes and complying with its open-source license for research evaluation.
- **Proprietary Models:** We utilize GPT-4.1 (OpenAI, 2025) via its official API as a baseline and evaluator. We comply with the provider's Terms of Service regarding data usage and content generation.

**Computation Costs**

- **Model Fine-tuning:** We conducted our fine-tuning experiments on a node equipped with 4 NVIDIA A40 GPUs. The training process for a single model fine-tuning required approximately 4.5 hours, resulting in a total computational cost of 18 GPU hours per model fine-tuning.
- **Inference:** Each inference task takes approximately 100 minutes for a total of 1000 samples.

**Datasets** To evaluate generalization capabilities, we utilize the **STEPGAME** benchmark (Shi et al., 2022). We use this dataset consistent with its original release terms, which are intended for academic research on spatial reasoning.

**Software Libraries** Our training and evaluation pipeline leverages the Hugging Face ecosystem, including `transformers`, `trl`, and `peft` (von Werra et al., 2020; Hu et al., 2021; Mangrulkar et al., 2022). These libraries are distributed under the Apache 2.0 license, and we use them in accordance with their documentation.