LTD-Bench: Evaluating Large Language Models by Letting Them Draw

Liuhao Lin[*†]
Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University
Xiamen, China

Ke Li[†‡]
Tencent Youtu Lab
Shanghai, China

Zihan Xu
Tencent Youtu Lab
Shanghai, China

Yuchen Shi
Tencent Youtu Lab
Shanghai, China

Yulei Qin
Tencent Youtu Lab
Shanghai, China

Yan Zhang[§]
Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University
Xiamen, China

Xing Sun
Tencent Youtu Lab
Shanghai, China

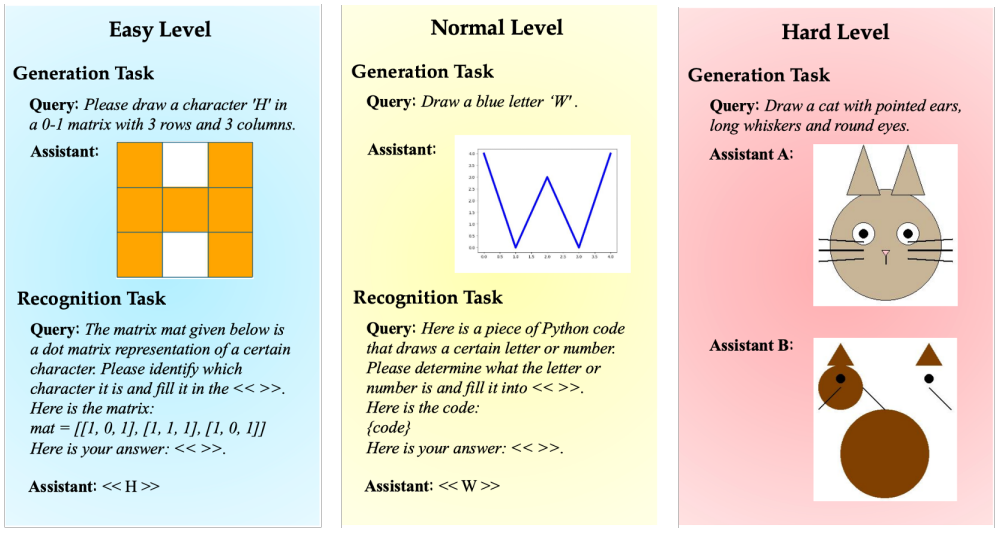
Rongrong Ji
Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University
Xiamen, China

Abstract

Current evaluation paradigms for large language models (LLMs) represent a critical blind spot in AI research—relying on opaque numerical metrics that conceal fundamental limitations in spatial reasoning while providing no intuitive understanding of model capabilities. This deficiency creates a dangerous disconnect between reported performance and practical abilities, particularly for applications requiring physical world understanding. We introduce LTD-Bench, a breakthrough benchmark that transforms LLM evaluation from abstract scores to directly observable visual outputs by requiring models to generate drawings through dot matrices or executable code. This approach makes spatial reasoning limitations immediately apparent even to non-experts, bridging the fundamental gap between statistical performance and intuitive assessment. LTD-Bench implements a comprehensive methodology with complementary generation tasks (testing spatial imagination) and recognition tasks (assessing spatial perception) across three progressively challenging difficulty levels, methodically evaluating both directions of the critical language-spatial mapping. Our extensive experiments with state-of-the-art models expose an alarming capability gap: even LLMs achieving impressive results on traditional benchmarks demonstrate profound deficiencies in establishing bidirectional mappings between language and spatial concepts—a fundamental limitation that undermines their potential as genuine world models. Furthermore, LTD-Bench's visual outputs enable powerful diagnostic analysis, offering a poten-

[*]This work was done during an internship at Tencent Youtu Lab.
[†]These authors have contributed equally.
[‡]Project lead.
[§]Corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks.



(a) Easy Level (b) Normal Level (c) Hard Level

Figure 1: The data examples of three levels in LTD-Bench. The model outputs in the generation tasks have all been rendered into images.

tial approach to investigate model similarity. Our dataset and codes are available at `https://github.com/walktaster/LTD-Bench`

1 Introduction

Large language models (LLMs) have demonstrated remarkable progress in recent years [26][25][3], achieving impressive results on numerous benchmarks spanning language understanding [14][17], mathematical reasoning [8][15][7][6], code generation [4][2] and instruction following [33][16][20]. However, these apparent successes mask a critical blind spot: current evaluation paradigms, which rely heavily on aggregate scores and opaque metrics, offer limited insight into models' true capabilities—particularly their understanding of the physical world. This disconnect is especially concerning as LLMs are increasingly deployed in domains such as robotics, autonomous systems, and design tools [30][24][28][21], where spatial reasoning is essential.

What makes this problem particularly pernicious is the abstract nature of traditional evaluation. When a model scores 85% on a benchmark, what specific capabilities and limitations does this number reveal? How can researchers, developers, and end-users gain an intuitive understanding of what the model can and cannot do in spatial domains? These questions remain largely unanswered by current methodologies.

To address this gap, we introduce **LTD-Bench** (Let Them Draw Benchmark), a novel evaluation framework that shifts LLM assessment from abstract numerical scores to directly observable visual outputs. Figure 1 shows the data examples of LTD-Bench. Unlike conventional benchmarks, LTD-Bench requires models to generate visual artifacts—either as dot matrices or executable code—based on textual instructions, making their spatial reasoning abilities immediately apparent even to non-experts. This approach bridges the disconnect between statistical metrics and intuitive understanding of model capabilities.

LTD-Bench comprises two complementary evaluation paths: generation tasks, which assess spatial imagination by requiring models to translate textual descriptions into visual representations, and recognition tasks, which evaluate spatial perception by asking models to interpret visual patterns. These tasks span three progressively challenging levels, from basic character representation to complex real-world object visualization, enabling fine-grained analysis of spatial reasoning abilities.

Our experiments with state-of-the-art LLMs reveal a significant capability gap overlooked by existing benchmarks. Even models that perform well on traditional reasoning tasks exhibit profound deficien-

cies in mapping between language and spatial concepts, undermining their potential as genuine world models. Furthermore, LTD-Bench's visual outputs facilitate diagnostic analyses not possible with traditional benchmarks, such as comparing stylistic characteristics of generated images to investigate model similarities.

By making model limitations visible rather than obscured behind abstract metrics, LTD-Bench represents a paradigm shift in the evaluation and understanding of large language models. Our framework lays the foundation for developing AI systems with more robust spatial reasoning—a critical requirement for applications that must interact with and reason about the physical world.

Our contributions are summarized as follows:

- We introduce LTD-Bench, the first benchmark that transforms LLM evaluation from opaque metrics to visually interpretable outputs. By requiring models to generate visual artifacts through drawing, we enable direct human assessment of spatial reasoning capabilities, addressing a fundamental gap between statistical performance and intuitive understanding of model limitations.
- We design a structured evaluation methodology with complementary generation and recognition tasks across three difficulty levels, providing a comprehensive assessment of both how LLMs translate language into spatial arrangements (imagination) and interpret spatial patterns into language (perception).
- Our experimental results quantify a significant capability gap in current LLMs, showing that even models with strong reasoning abilities struggle to establish the bidirectional mapping between language and spatial concepts - a critical finding that identifies a priority direction for improvement in the next generation of AI systems.
- We demonstrate how visual output comparison provides a powerful diagnostic tool for model development, revealing stylistic similarities among various models and offering insights into the model similarity that is not well captured by traditional evaluation metrics.

2 Related Work and Discussion

Existing Benchmarks. Current LLM evaluation frameworks primarily emphasize symbolic and procedural competencies. Comprehensive benchmarks such as MMLU [14] and TruthfulQA [17] assess cross-domain knowledge retention and factual accuracy, while mathematical reasoning datasets like GSM8K [8] and MATH [15] focus on multi-step problem-solving in abstract domains. Code generation benchmarks (e.g., HumanEval [4], MBPP [2]) further evaluate the translation of natural language into executable algorithms. While these benchmarks effectively quantify core symbolic manipulation skills, they are confined to a text-to-symbol paradigm and lack intuitive, visual, and directly interpretable assessments of model capabilities. Consequently, they do not reveal whether LLMs can establish robust, bidirectional mappings between linguistic symbols and spatial entities.

Spatial Perception and Imagination. The lack of spatial evaluation in LLMs is partly rooted in the assumption that visual perception is essential for spatial reasoning. However, neurocognitive studies of congenitally blind individuals demonstrate that robust spatial cognition can arise through nonvisual modalities, such as linguistic descriptions and haptic feedback. For instance, Striem-Amit et al. [22] provide evidence for a neural dissociation between abstract semantic knowledge and sensory attributes, while Cooney et al. [9] show that spatial reasoning relies on innate neural mechanisms rather than visual experience. These findings challenge the necessity of visual input for spatial reasoning and establish a biological precedent for text-based spatial cognition. This suggests that text-based LLMs, even without visual input, should be capable of developing spatial understanding. Moreover, Transformer architectures, which underpin modern LLMs, excel at modeling relationships between abstract tokens, theoretically enabling the inference of geometric and topological patterns from textual descriptions. Recent work supports this potential: LLMs have demonstrated the ability to generate code for rendering simple shapes [1][13], indicating latent spatial reasoning capabilities. Nevertheless, no existing benchmark systematically evaluates these abilities, despite their importance for meaningful interaction with the physical world.

Intuitive Visual Evaluation. LTD-Bench addresses this gap by providing an intuitive and visual assessment of LLMs' spatial perception and imagination. In generation tasks, models are required

to produce either renderable dot matrices or Python code for image drawing, both of which can be visualized as images. Prior research has shown that prompting LLMs for evaluation is effective not only in NLP tasks [32][5][19][10] but also in multimodal domains [31][29]. Accordingly, LTD-Bench leverages GPT-4.1 to assess the quality of images generated by LLMs in certain open-ended generative tasks, enabling a more direct and interpretable evaluation of spatial reasoning abilities.

3 LTD-Bench

The gap between reported LLM performance and actual spatial reasoning capabilities represents a critical blind spot in AI evaluation. LTD-Bench addresses this gap through a novel approach that transforms abstract metrics into directly observable visual evidence, enabling intuitive assessment of how well models can establish bidirectional mappings between language and spatial concepts.

3.1 Design Principles and Problem Addressing

LTD-Bench addresses three fundamental problems in current LLM evaluation approaches through carefully designed principles:

Problem 1: Invisibility of Spatial Reasoning Limitations. Current benchmarks provide numerical scores that obscure whether models can actually establish bidirectional mappings between language and spatial concepts.

Solution: Visual Interpretability. LTD-Bench's core innovation is its transformation of abstract model capabilities into concrete visual artifacts. All outputs from generation tasks are rendered into images, enabling direct inspection by both humans and automated systems. This approach makes model limitations immediately apparent to anyone - regardless of technical background - revealing capabilities that remain hidden in traditional benchmarks.

Problem 2: Incomplete Assessment of Spatial Cognition. Existing evaluations rarely assess both directions of the critical language-spatial mapping.

Solution: Dual-Path Evaluation. LTD-Bench systematically evaluates both aspects of spatial cognition through complementary pathways:

- **Generation Tasks (Spatial Imagination):** Models translate textual descriptions into visual representations (dot matrices or drawing code), testing their ability to convert linguistic concepts into spatial arrangements.
- **Recognition Tasks (Spatial Perception):** Models interpret visual patterns from given representations, testing their ability to understand spatial configurations through language.

Problem 3: Inability to Pinpoint Capability Thresholds. Traditional benchmarks often fail to identify precisely where models begin to struggle with increasingly complex spatial reasoning.

Solution: Progressive Complexity. LTD-Bench implements a hierarchical structure with three difficulty levels:

1. **Easy Level:** Basic character representation using discrete dot matrices in finite grid space, establishing baseline spatial capabilities.
2. **Normal Level:** Character drawing using continuous curves in infinite coordinate space, requiring more sophisticated spatial reasoning.
3. **Hard Level:** Complex real-world object representation, requiring advanced spatial conceptualization and compositional understanding.

Through these design principles, LTD-Bench not only evaluates model performance but fundamentally transforms how we understand and interpret model capabilities, making limitations visible that were previously hidden behind abstract metrics.

Table 1: The structure of LTD-Bench

Level	Task		Total
	Generation	Recognition	
Easy	50	36	86
Normal	36	36	72
Hard	25	-	25
Total	111	72	183

3.2 Benchmark Structure and Task Design

Based on the design principles outlined in Section 3.1, LTD-Bench comprises a comprehensive evaluation framework with 183 distinct data distributed across three difficulty levels, as summarized in Table 1. Each level presents unique challenges designed to assess specific aspects of spatial reasoning.

3.2.1 Easy Level: Discrete Grid-Based Spatial Understanding

The Easy level is designed to assess the fundamental spatial abilities of LLMs within a two-dimensional finite grid space represented in the form of a dot matrix. In accordance with the dual-path evaluation principle, this level consists of both generation and recognition tasks:

Generation Task: Given a textual instruction (e.g. "Please draw a character 'H' in a 0-1 matrix with 3 rows and 3 columns."), the model is required to output a dot matrix, where '1' denotes a filled cell and '0' denotes a blank cell, representing the specified character. The output dot matrix can be rendered into a grid-like image for intuitive and visual display.

This task evaluates whether models can:

- Conceptualize the spatial arrangement of simple characters
- Translate this conceptualization into a precise grid-based representation
- Maintain correct proportions and spatial relationships within constraints

Recognition Task: In the complementary recognition pathway, models are presented with a dot matrix and must identify which character it represents. This evaluates the reverse mapping—from spatial arrangement to symbolic representation—testing models' ability to interpret visual patterns expressed as matrices.

As shown in Figure 1(a), the outputs of generation tasks at this level can be directly verified through visual inspection, while recognition tasks have unambiguous ground-truth answers. This level establishes whether models possess even the most basic capacity for bidirectional mapping between language and discrete spatial representations.

3.2.2 Normal Level: Curve Composition in Infinite 2D Coordinate Space

The Normal level increases complexity by transitioning from discrete grid spaces to continuous coordinate systems, requiring models to reason about characters as compositions of mathematical curves in an unbounded space.

Generation Task: The model is tasked with generating Python code that draws specified characters (such as letters or digits) using only combinations of curves. Prompts are designed to enforce the constraint that only curve-based drawing methods are allowed, explicitly prohibiting the use of direct text rendering functions (e.g., TextPrint). The generated Python code can be executed directly to produce images, enabling an intuitive and visual evaluation of the correctness of the model's drawings.

This level evaluates whether models can:

- Translate character concepts into continuous rather than discrete representations
- Generate executable code that correctly implements spatial understanding

- Create visual outputs that maintain character recognizability despite implementation constraints

Recognition Task: For recognition, models are presented with Python code that draws a character through curve combinations and must identify which character the code would render. This requires parsing code, understanding how mathematical functions translate to visual shapes, and recognizing the resulting pattern.

As illustrated in Figure 1(b), this level significantly increases the complexity of the bidirectional mapping between language and spatial concepts, requiring models to operate in continuous rather than discrete space and to translate between linguistic, mathematical and visual representations.

3.2.3 Hard Level: Real-world Object Drawing in Infinite 2D Space

The Hard level represents the most advanced spatial reasoning challenge, requiring models to conceptualize and render complex real-world objects as compositions of multiple curves.

Generation Task: Models receive open-ended instructions to draw real-world objects with specific attributes, such as "Draw a cat with pointed ears, long whiskers and round eyes." This requires not only understanding what these objects look like but also decomposing them into geometric primitives and implementing them through code.

This level evaluates whether models can:

- Conceptualize complex multi-part objects from linguistic descriptions
- Translate abstract object features into concrete spatial relationships
- Generate code that produces a coherent and recognizable visual representation

Evaluation Approach. Due to the open-ended and subjective nature of these tasks, evaluation at this level employs GPT-4.1 as an automated assessor following established practices in LLM-based evaluation [29][32][5][19][10]. The evaluation protocol assigns scores between 0.0 and 1.0 based on predefined criteria that consider both adherence to specified features and overall visual coherence.

Additionally, as shown in Figure 1(c), the stylistic diversity of outputs at this level enables comparative analysis between different model architectures. By examining stylistic similarities in generated images, researchers can identify shared tendencies across model families, providing insights into the similarity among various LLMs that is not captured by traditional evaluation metrics.

Together, these three levels form a comprehensive framework for evaluating spatial perception and imagination capabilities in LLMs, making visible the specific strengths and limitations that remain hidden in traditional text-based benchmarks.

4 Experiments

4.1 Experiment Settings

Models. Since our approach places certain demands on both the reasoning and coding abilities of LLMs, and its tasks are relatively challenging for smaller LLMs, we primarily selected some of the most advanced models for evaluation, including DeepSeek-R1 [12], DeepSeek-V3 [18], GPT-4o, GPT-4.1-mini, QwQ32B [23], Qwen2.5-72B-Instruct [27] and Llama3.3-70B-Instruct [11].

Evaluation Methods. The evaluation of our benchmark is tailored to the nature of each task type and difficulty level.

For generation tasks, which lack fixed ground-truth answers, we employ both human evaluation and GPT-4.1-based automated evaluation at the Easy and Normal levels. An analytical comparison between human and GPT-4.1 evaluations is provided in Appendix A.3. For Hard-level generation tasks, the open-ended nature of the outputs introduces considerable subjectivity into human assessment, as personal aesthetic preferences and other factors may influence scoring. Therefore, we rely exclusively on GPT-4.1 for evaluation at this level, utilizing a detailed system prompt with explicit scoring criteria to guide GPT-4.1 in assigning a score between 0.0 and 1.0 to each generated image.

Table 2: The model output examples on the generation tasks across different difficulty levels in LTD-Bench. All outputs are rendered as images to facilitate an intuitive visual assessment of the model's capabilities.

	Easy Generation *Please draw a character 'K' in a 0-1 matrix with 5 rows and 4 columns*	Normal Generation *Draw a green letter Q*	Hard Generation *Draw a cat with pointed ears, long whiskers and round eyes*
Deepseek-r1			
Deepseek-v3			
GPT-4.1-mini			
GPT-4o			
QwQ-32B			
Qwen2.5-72B-Instruct			
Llama3.3-70B-Instruct			

If a model's output code fails to execute and does not produce a valid image, a score of 0 is assigned for that instance.

For recognition tasks, which have well-defined ground-truth answers, we directly compare model outputs to the correct answers to compute accuracy.

More implementation details are provided in Appendix A.1, and the prompt templates used in LTD-Bench are included in Appendix B.

4.2 Result Analyses

Since the intuitive visual evaluation of LLM capabilities is a central contribution of our work, we first present representative model outputs for generation tasks across different difficulty levels in Table 2. These examples clearly illustrate performance disparities among models. For instance, advanced models such as Deepseek-r1 and GPT-4.1-mini significantly outperform smaller models like Qwen2.5-72B-Instruct and Llama3.3-70B-Instruct. Overall performance metrics are summarized in Table 3, from which we draw the following conclusions:

LLMs generally exhibit poor spatial perception and imagination. As shown in Table 3, only Deepseek-r1 achieves an average accuracy above 70%, with GPT-4.1-mini exceeding 60%. In contrast, Qwen2.5-72B-Instruct and Llama3.3-70B-Instruct achieve only around 30%. Notably, human experts can solve Easy and Normal tasks with near-perfect accuracy, even in text-only settings, whereas LLMs fall far short. These results indicate that current LLMs lack robust spatial reasoning and fail to

Table 3: LTD-Bench evaluation results. **Bold** indicates the best performance in that dimension, while underline indicates the second-best performance. For generation tasks at Easy and Normal level, the data in blue is the results evaluated by human and data in orange is the results evaluated by GPT-4.1. Numbers are presented in % with a full score of 100%.

Model	Easy		Normal		Hard	Average
	Generation	Recognition	Generation	Recognition	Generation	
Deepseek-r1	82.00 (80.00 / 84.00)	**69.44**	65.28 (55.56 / 75.00)	**77.78**	63.20	**71.54**
Deepseek-v3	72.00 (66.00 / 78.00)	36.11	54.17 (47.22 / 61.11)	63.89	66.40	58.51
GPT-4.1-mini	**85.00 (82.00 / 88.00)**	38.89	**70.83 (66.67 / 75.00)**	55.56	**71.60**	64.38
GPT-4o	81.00 (76.00 / 86.00)	41.67	45.83 (36.11 / 55.56)	44.44	48.00	52.19
QwQ-32B	65.00 (58.00 / 72.00)	36.11	38.89 (33.33 / 44.44)	58.33	42.00	48.07
Qwen2.5-72B-Instruct	56.00 (42.00 / 70.00)	13.89	18.06 (13.89 / 22.22)	25.00	40.80	30.75
Llama3.3-70B-Instruct	46.00 (32.00 / 60.00)	11.11	23.61 (16.67 / 30.56)	19.44	35.20	27.07

Table 4: Model performance on generation and recognition tasks. **Bold** indicates the best performance in that dimension, while underline indicates the second-best performance.

Model	Generation	Recognition	Average
Deepseek-r1	72.88	**73.61**	**73.24**
Deepseek-v3	64.71	50.00	57.36
GPT-4.1-mini	**77.46**	47.22	62.34
GPT-4o	62.07	43.06	52.56
QwQ-32B	51.33	47.22	49.28
Qwen2.5-72B-Instruct	39.17	19.44	29.31
Llama3.3-70B-Instruct	36.62	15.28	25.95

establish reliable bidirectional mappings between linguistic symbols and spatial entities—an essential capability for genuine world comprehension. Further analysis is provided in Appendix C.

Deep reasoning improves recognition but not generation tasks. Table 4 shows that Deepseek-r1, equipped with deep reasoning, outperforms GPT-4.1-mini by over 25% in recognition accuracy, but lags behind in generation tasks. Similarly, within the Deepseek family, Deepseek-r1 consistently surpasses Deepseek-v3, yet the relative improvement is much less pronounced for generation than for recognition. QwQ-32B, another model with deep reasoning, matches GPT-4.1-mini on recognition but underperforms on generation. We hypothesize that recognition tasks benefit from enhanced spatial perception via reasoning, while generation tasks, which rely more on spatial imagination, are less amenable to such improvements.

Table 5 further supports this: Llama3.3-70B distilled with Deepseek-r1 data shows an 18.05% accuracy gain on recognition tasks, but even a 2.91% decline on generation tasks. This suggests that deep reasoning may lead to overthinking in tasks where LLMs' inherent abilities are insufficient, potentially degrading performance.

Multimodal LLMs do not show clear advantages on text-based spatial tasks. Contrary to human intuition—where visual experience enhances spatial abilities—multimodal models like GPT-4.1-mini and GPT-4o do not consistently outperform text-only models such as Deepseek-r1 and Deepseek-v3 on LTD-Bench (Tables 3 and 4). While GPT-4.1-mini excels in generation tasks, its overall performance is still lower than Deepseek-r1, and GPT-4o underperforms compared to Deepseek-v3. Although these results may be influenced by differences in language modeling capabilities, they challenge expectations based on human cognition and highlight the need for further research on aligning visual and textual features in multimodal learning.

Table 5: Comparison of the performance of Llama3.3-70B-Instruct and Deepseek -r1-distill-Llama3.3-70B on generation and recognition tasks.

Model	Generation	Recognition	Average
Llama3.3-70B-Instruct	36.62	15.28	25.95
Deepseek-r1-distill-Llama3.3-70B	33.71	33.33	33.52
Δ	2.91↓	18.05↑	7.57↑

Table 6: Style similarity comparison on the generation task at Hard level. (1) is Qwen2.5-72B-Instruct, (2) is Qwen2.5-32B-Instruct and (3) is GPT-4.1-mini. The total sample size is 22, excluding the images that failed to be generated.

	(1) and (2) more similar	(1) and (3) more similar	(2) and (3) more similar	All three are different
Rate	12/22	1/22	2/22	7/22

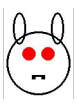
4.3 Model Similarity

For Hard-level generation tasks, different models tend to produce images with distinct stylistic characteristics. Analyzing the stylistic similarity among these images provides a promising avenue for investigating model similarity. As shown in Table 6, we conduct a style similarity comparison among three models: Qwen2.5-72B-Instruct, Qwen2.5-32B-Instruct, and GPT-4.1-mini. The results indicate that the highest proportion of stylistically similar images occurs between the two models from the Qwen2.5 series, accounting for over 50% of the total valid samples. In contrast, only three samples were found to be more similar to GPT-4.1-mini. Table 2 further presents representative examples, offering a more intuitive illustration of the stylistic similarities among images generated by the three models. It is evident from these examples that the Qwen2.5 series models exhibit greater stylistic resemblance to each other. This suggests a positive correlation between model similarity and the stylistic similarity of the open-ended images they generate. Through this exploratory study, we identify a potential new approach for assessing model similarity, which may serve as a valuable reference for future research in this area.

5 Limitations and Future Work

While LTD-Bench offers an intuitive and visual framework for evaluating LLM capabilities, several limitations remain. First, the current benchmark features a relatively small dataset and assesses a narrow range of abilities, focusing solely on spatial perception and imagination. This limits both the comprehensiveness and generalizability of our findings. Future work will expand the dataset and incorporate a wider array of tasks to enable more thorough evaluation of LLMs.

Table 7: The specific image examples generated by Qwen2.5-72B-Instruct, Qwen2.5-32B-Instruct and GPT-4.1-mini.

	Qwen2.5-72B-Instruct	Qwen2.5-32B-Instruct	GPT-4.1-mini
Flower			
House			
Rabbit			

Second, our analysis of model similarity is preliminary, relying only on stylistic comparisons of generated images as a proxy. More systematic and quantitative approaches are needed to rigorously assess similarities between models. In future research, we aim to develop more sophisticated metrics and analytical methods to achieve a deeper understanding of model similarity.

6 Conclusion

In this paper, we present LTD-Bench, a novel benchmark that enables intuitive and visual evaluation of LLMs by letting them draw. LTD-Bench specifically targets two fundamental aspects: spatial perception and spatial imagination. Our experimental results demonstrate that current LLMs face substantial challenges in both areas, often failing to establish robust bidirectional mappings between linguistic symbols and spatial concepts. Additionally, LTD-Bench provides a promising new avenue for exploring model similarity, offering valuable insights to guide future research in this domain.

Acknowledgments and Disclosure of Funding

This work was supported by the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No. U21B2037, No. U22B2051, No. U23A20383, No. 62176222, No. 62176223, No. 62176226, No. 62072386, No. 62072387, No. 62072389, No. 62002305 and No. 62272401), and the Natural Science Foundation of Fujian Province of China (No. 2021J06003, No.2022J06001).

References

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

[2] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[3] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690, 2024.

[4] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[5] Cheng-Han Chiang and Hung-yi Lee. Can large language models be an alternative to human evaluations? *arXiv preprint arXiv:2305.01937*, 2023.

[6] Francois Chollet, Mike Knoop, Gregory Kamradt, and Bryan Landers. Arc prize 2024: Technical report. *arXiv preprint arXiv:2412.04604*, 2024.

[7] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[9] Sarah Cooney, Corinne A Holmes, Giulia Cappagli, Elena Cocchi, Monica Gori, and Fiona Newell. Express: Susceptibility to spatial illusions does not depend on visual experience: evidence from sighted and blind children. *Quarterly Journal of Experimental Psychology*, page 17470218251336082, 2024.

[10] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*, 2023.

[11] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[12] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[13] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962, 2023.

[14] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[15] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[16] Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. Followbench: A multi-level fine-grained constraints following benchmark for large language models. *arXiv preprint arXiv:2310.20410*, 2023.

[17] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

[18] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[19] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023.

[20] Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. Infobench: Evaluating instruction following ability in large language models. *arXiv preprint arXiv:2401.03601*, 2024.

[21] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

[22] Ella Striem-Amit, Xiaoying Wang, Yanchao Bi, and Alfonso Caramazza. Neural representation of visual concepts in people born blind. *Nature communications*, 9(1):5250, 2018.

[23] Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025.

[24] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

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

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

[27] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[28] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[29] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

[30] Fanlong Zeng, Wensheng Gan, Yongheng Wang, Ning Liu, and Philip S Yu. Large language models for robotics: A survey. *arXiv preprint arXiv:2311.07226*, 2023.

[31] Chanjin Zheng, Zengyi Yu, Yilin Jiang, Mingzi Zhang, Xunuo Lu, Jing Jin, and Liteng Gao. Artmentor: Ai-assisted evaluation of artworks to explore multimodal large language models capabilities. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2025.

[32] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

[33] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We describe in detail the contribution of this paper in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the paper and future work in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not include theorems.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide our detailed implementations in Appendix A.1 and our dataset and codes are available at `https://anonymous.4open.science/r/LTD-Bench-D324`.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: our dataset and codes are available at `https://anonymous.4open.science/r/LTD-Bench-D324`.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the detailed implementations in Appendix A.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report our experimental results in Section 4 and Appendix A.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Since our work is a benchmark, we provide the specific implementation used for model inference in Appendix A.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research is in accordance with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our broader impact is mentioned in our contributions included in Section 1 and Section 6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators and original owners of all assets (e.g., code, data, models) used in this paper are properly credited. The relevant citations and attributions are provided for all external resources, and the licenses and terms of use for each asset are explicitly mentioned and fully respected. For models and datasets, we have adhered to the usage guidelines and licensing terms specified by their respective creators, ensuring that all necessary permissions were obtained where applicable.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets`

has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We have attached our prompts in Appendix B, and our dataset and codes are available at `https://anonymous.4open.science/r/LTD-Bench-D324`.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Experiment details

A.1 Implementation details

For open-source LLMs used in this paper, we utilized the API on the Bailian* platform, which hosts models identical to those on HuggingFace. For GPT-4.1, GPT-4.1-mini and GPT-4o, we use their official API†. In our experiments, we set the temperature to 0 for all models. During GPT-4.1-based automated evaluation, it is found that the outputs of GPT-4.1 still exist variance, although the temperature is set as 0. Therefore, we utlize GPT-4.1 to evaluate the outputs of all models by 5 times.

A.2 More details of human evaluation

For human evaluation, we assigned independent annotators to tasks of different difficulty levels, with 10 annotators dedicated to each level. The annotators cover a diverse range of technical backgrounds, including newly enrolled undergraduates (computer beginners), master's students, and laboratory engineers—ensuring evaluations are not biased toward a single expertise group. For each difficulty level, the final experimental result is determined by averaging the scores from the 10 annotators, which helps mitigate individual subjectivity.

A.3 Additional evaluation methods experiment

As mentioned in Section 4.1, we employ both human evaluation and GPT-4.1-based automated evaluation for generation tasks at the Easy and Normal levels. Human evaluation provides more accurate and reliable results, as it avoids the hallucinations and occasional misjudgment issues that GPT-4.1 may suffer from. However, human evaluation is labor-intensive and costly, requiring manual inspection of each sample. In contrast, GPT-4.1-based evaluation offers a convenient and fully automated approach, enabling rapid, end-to-end computation of accuracy metrics. For further analysis, we conduct an additional experiment with both evaluation methods. As shown in Figure 2, although GPT-4.1 tends to yield slightly higher accuracy scores due to occasional hallucinations, our experiment confirms that the relative ranking of model performance remains consistent between human and GPT-4.1 evaluations. Therefore, GPT-4.1 can be reliably used for large-scale, automatic evaluation, while human evaluation serves as a more precise but resource-intensive reference.

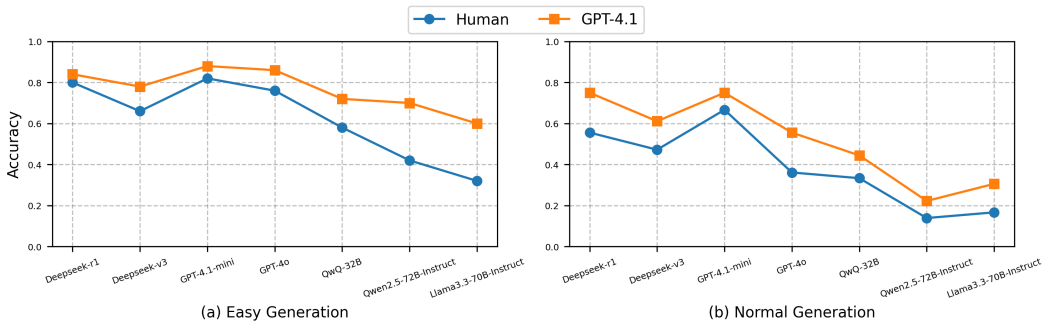


Figure 2: Comparison of human evaluation results and GPT-4.1-based automated evaluation results.

*https://bailian.console.aliyun.com/
†https://platform.openai.com/

B Prompt templates

We list all prompt templates used in LTD-Bench here.

You are a dot matrix drawing robot. I will ask you to draw a specified character on a 0-1 matrix with a specified number of rows and columns.
Requirements:
1. You need to draw the specified character on a 0-1 matrix with a specified number of rows and columns by setting the elements to 1;
2. Please strictly follow the following format to output the 0-1 matrix you drew in <Mat></Mat>:
<Mat>
mat = []
</Mat>

Here is the question:
question

Figure 3: The prompt for the the Easy-level generation task.

The matrix mat given below is a dot matrix representation of a certain character. Please identify which character it is and fill the answer in the « ».
Here is the matrix: {matrix}

Here is your answer: « »

Figure 4: The prompt for the Easy-level recognition task.

You are a code generation robot. You need to generate runnable Python code based on the drawing requirements provided by the user to create the image the user needs.

Requirements:
1. Draw the pattern required by the user in a two-dimensional coordinate system, ensuring that the axes are hidden at the end, and do not use the 'Text' or 'TextPath' functions directly for drawing
2. The generated image should be saved as "test.jpg"
3. Please output in the following format, filling in the generated Python code within the <Code></Code> tags, without adding comments at the beginning or end
<Code>
</Code>

Here is the question:
{question}

Figure 5: The prompt for the Normal-level generation task.

Here is a piece of Python code that draws a certain letter or number. Please determine what the letter or number is and fill the answer in the « ».
Here is the code: {code}

Here is your answer: « »

Figure 6: The prompt for the Normal-level recognition task.

You are a code generation robot. The user will provide drawing requirements for a certain object. You need to generate directly executable Python code according to the drawing requirements to draw the image required by the user.

Requirements:
1. First, analyze the basic features of the drawing object. On this basis, according to the additional drawing requirements proposed by the user, sort out all the feature details that need to be drawn, and conceive how to draw it with Python code
2. Then, generate Python code according to your ideas to draw the image required by the user. Pay attention to the correctness of the library function call
3. Save the drawn image as "test.jpg"
4. Please output in the following format. Fill in all the features and details you need to draw and your ideas in <Thought></Thought>, and fill in the Python code you generated in <Code></Code>. Do not add comments at the beginning and end
<Thought>
</Thought>
<Code>
</Code>

Here is the question:
{question}

Figure 7: The prompt for the Hard-level generation task.

The following is a dot matrix, where 1 represents fill and 0 represents blank. You need to determine whether the character drawn by the dot matrix is ground truth. If so, output [[Yes]], otherwise output [[No]]. Here is the dot matrix: output matrix

Figure 8: The prompt for evaluating the model outputs on the Easy-level generation task.

Please judge whether the character drawn in the given image is ground truth. If so, output [[Yes]], otherwise output [[No]].

Figure 9: The prompt for evaluating the model outputs on the Normal-level generation task.

{System prompt}
You are an evaluation assistant. Please analyze and score the input image according to the given object and drawing requirements.
Requirements:
1. First determine whether the image can be identified as the given object, then determine whether the image meets the drawing requirements, and finally score based on the analysis
2. The score range (scoring standard) is:

0.0: The image cannot identify the object at all
0.1: The image can hardly identify the object
0.2: The image is difficult to identify the object
0.3: The image can barely identify the object, but the main features are blurred and do not meet the drawing requirements
0.4: The image can basically identify the object, but does not meet the drawing requirements
0.5: The image can identify the object, but only meets a few drawing requirements
0.6: The image can identify the object, but a few drawing requirements are not met
0.7: The image can identify the object and basically meets all drawing requirements
0.8: The image can clearly identify the object and fully meets all drawing requirements, but the painting details and overall aesthetics are poor
0.9: The image can clearly identify the object, fully meets all drawing requirements, and the drawing details and overall aesthetics are also excellent
1.0: The image can perfectly identify the object, fully meets all drawing requirements, the details are extremely rich, and the overall effect is excellent

3. Strictly follow the format below to output your analysis and final score

<Analysis>***</Analysis>
<Score>***</Score>

{User prompt}
Object: {object}
Drawing requirements: {question}

Figure 10: The system prompt and user prompt for evaluating model outputs on the Hard-level generation task.

You are an impartial judge. Please evaluate which two of the three provided images are most similar in style only. Begin your evaluation by comparing the three images and provide a short explanation. Avoid any position biases and ensure that the order in which the images were presented does not influence your decision. After providing your explanation, output your final verdict by strictly following this format: '[[A]]' if the first and second images are more similar, '[[B]]' if if the first and third images are more similar, '[[C]]' if the second and third images are more similar, and '[[D]]' if all three images have different styles.

Figure 11: The prompt for comparing the style similarity of images generated by three different models on the Hard-level generation task.

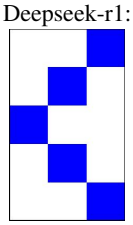
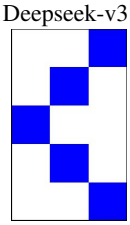
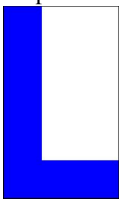
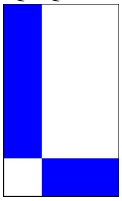
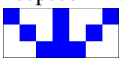
C Case study

In this section, we list several failed cases of different models on generation tasks of three difficulty levels, to further analyze the current limitations in model capabilities.

Easy level. Table 8 shows the failed cases on the generation task at Easy level. We can observe that models often mistakenly generate the characters '>' and 'J' as their mirrored counterparts '<' and 'L' within the dot matrix, revealing their insufficient understanding of basic spatial orientations such as left-right and up-down. Additionally, the way models render the character 'W' in the dot matrix further highlights their limitations in spatial imagination.

Normal level. And for the generation task at Normal level, the failed cases shown in Table 9 more clearly reveal the limitations of the models' spatial capabilities. When questioned with "Draw a blue letter W", QwQ-32B produced an image with the same issue observed in the Easy level earlier: the letter was upside down. Other than that, images generated by other models for other questions listed in the table are even more problematic, featuring completely incorrect outputs with numerous chaotic lines. This suggests that the models may not have a proper understanding of how their actions correspond to spatial states, resulting in outputs that deviate significantly from the intended results. Such shortcomings are critical obstacles for LLMs in achieving a true understanding of the world.

Table 8: The failed cases on the Easy-level generation task.

Question	**Model Outputs**	
Please draw a character '>' in a 0-1 matrix with 5 rows and 3 columns.	Deepseek-r1: 	Deepseek-v3: 
Please draw a character 'J' in a 0-1 matrix with 5 rows and 3 columns.	Deepseek-r1: 	QwQ-32B: 
Please draw a character 'W' in a 0-1 matrix with 3 rows and 7 columns.	Deepseek-r1: 	GPT-4.1-mini: 

Hard level. Table 10 further shows the failed cases in Hard level. Firstly, the "Draw a clock with..." case demonstrates that when spatial requirements are introduced, the models tend to perform poorly in easy math and coding tasks. Even advanced models like Deepseek-r1, GPT-4o, and QwQ-32B made mistakes. Specifically, these models may easily understand what "the pointer pointing to 9:30" means, but they often make various errors when asked to translate this understanding into a spatial representation. Further more, in the latter two cases, 'Airplane' and 'Leaf', it is even more apparent that the models' limited spatial imagination, combined with their inadequate ability to map linguistic symbols to spatial entities, leads to these unsatisfactory results.

Through the failed cases and analyses above, we provide a clear and intuitive visualization of the current models' significant shortcomings in spatial capabilities. These findings offer valuable insights for future research on how large language models understand the world.

Table 9: The failed cases on the Normal-level generation task.

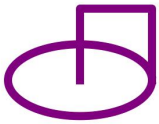
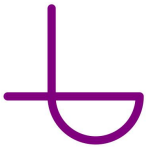
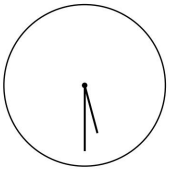
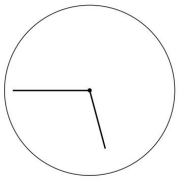
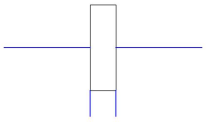
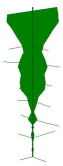
Question	Model Outputs	
Draw a purple number 9.	GPT-4.1-mini: 	GPT-4o: 
Draw a blue letter W.	Deepseek-v3: 	QwQ-32B: 
Draw a purple letter J.	Deepseek-v3: 	GPT-4.1-mini: 

Table 10: The failed cases on the Hard-level generation task.

Question	Model Outputs		
Draw a clock with a round face and the pointer pointing to 9:30	Deepseek-r1: 	GPT-4o: 	QwQ-32B: 
Draw a plane, from a bird's-eye view, with two wings and two tails, and the wings should be much longer than the tail	Deepseek-r1: 	GPT-4.1-mini: 	QwQ-32B: 
Draw a leaf with veins and irregular jagged edges	GPT-4.1-mini: 	QwQ-32B: 	Qwen2.5-72B-Instruct: