SpatialThinker: Reinforcing 3D Reasoning in Multimodal LLMs via Spatial Rewards

Hunar Batra[1]* **Haoqin Tu**[2] **Hardy Chen**[2] **Yuanze Lin**[1] **Cihang Xie**[2] **Ronald Clark**[1]

[1]University of Oxford [2]University of California, Santa Cruz

Abstract

Multimodal large language models (MLLMs) have achieved remarkable progress in vision–language tasks, but they continue to struggle with spatial understanding. Existing spatial MLLMs often rely on explicit 3D inputs or architecture-specific modifications, and remain constrained by large-scale datasets or sparse supervision. To address these limitations, we introduce SpatialThinker, a 3D-aware MLLM trained with RL to integrate structured spatial grounding with multi-step reasoning. The model simulates human-like spatial perception by constructing a scene graph of task-relevant objects and spatial relations, and reasoning towards an answer via dense spatial rewards. SpatialThinker consists of two key contributions: (1) a data synthesis pipeline that generates STVQA-7K, a high-quality spatial VQA dataset, and (2) online RL with a multi-objective dense spatial reward enforcing spatial grounding. SpatialThinker-7B outperforms supervised fine-tuning and the sparse RL baseline on spatial understanding and real-world VQA benchmarks, nearly doubling the base-model gain compared to sparse RL, and surpassing GPT-4o. These results showcase the effectiveness of combining spatial supervision with reward-aligned reasoning in enabling robust 3D spatial understanding with limited data and advancing MLLMs towards human-level visual reasoning.

1 Introduction

Spatial reasoning is central to human intelligence, enabling us to perceive, localize, and manipulate objects in complex environments. This ability is critical for embodied AI tasks such as robotic manipulation [40, 27, 61], navigation [36], and augmented reality [43], where spatial awareness underpins real-world decision-making [23, 73]. While multimodal large language models (MLLMs) excel at general vision–language tasks [39, 50, 21, 7, 24, 52, 29], they continue to struggle with 3D spatial understanding [10, 75, 41, 89, 74, 56], which requires capturing geometry, structure, and relations beyond 2D projections.

Existing approaches remain data-hungry or architecturally specialized. They rely on massive synthetic datasets derived from 3D scene graphs (e.g., SpatialVLM was trained on 2B Spatial VQA samples, SpatialRGPT on 700k) [10, 19, 17], architectural changes [35], explicit 3D inputs such as point clouds [34, 17, 8], or reinforcement learning (RL) with sparse rewards [58, 78, 86, 87, 68, 99].

We present SpatialThinker, a 3D-aware MLLM that integrates scene graph grounding with multi-step reasoning through online policy RL. The model builds question-focused scene subgraphs consisting of objects, their relations, and localized coordinates, and reasons over them under a lexicographically-ordered multi-objective reward: format rewards enforce structured reasoning, count penalties regulate regional focus, accuracy rewards prioritize correctness, and CIoU-based spatial rewards encourage precise localization. This design promotes human-like reasoning: observe, localize, think, answer.

*Correspondence to {hunar.batra, ronald.clark}@cs.ox.ac.uk

2nd Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2025).

By training on only 7K samples with our synthesized STVQA-7K dataset, SpatialThinker-7B outperforms supervised fine-tuning (+6%) and conventional RL baselines (+3.2%) across twelve spatial understanding, real-world and generic VQA benchmarks, surpassing GPT-4o (+3.4% avg.) and Claude 3.5 Sonnet (+10.1% avg.) [39, 5], particularly a +12.1% gain over GPT-4o on 3DSRBench [56]. While sparse RL improves the base model by +4% avg., our dense spatial reward design yields +7.2%, nearly doubling the benefit. These results show that models can learn effective spatial reasoning by focusing on relevant regions, constructing internal scene representations, and accurately localizing objects through dense rewards, without relying on large-scale data alone [10, 55].

Our contributions are:

- SpatialThinker, a Spatial MLLM that integrates scene graph-based grounding with online RL for spatial reasoning, achieving strong results with only 7K samples.
- STVQA-7K, a high-quality spatial VQA dataset grounded in scene graphs.
- A dense, lexicographically gated multi-objective reward that guides regionally focused spatial reasoning, achieving superior in- and out-of-distribution generalization across spatial, generic VQA, and real-world benchmarks.

2 SpatialThinker: Spatially-Aware Reasoning MLLMs

2.1 Multi-Objective Reward Design

SpatialThinker is trained with a fine-grained, multi-objective reward that guides visually grounded reasoning. Unlike prior RLVR methods relying on sparse correctness signals [63, 99, 69], we combine four complementary components, including: format, accuracy, count, and spatial rewards, aligned with the reasoning stages: observe, localize, think, answer. We present reward ablation in Appendix E

Format Reward. Responses must follow a structured template with <*observe*>, <*scene*>, <*think*>, and <*answer*> tags. The scene JSON must be parseable, with valid object fields (ID, bbox) and triplet relations. The format reward $R_f \in \{0, 1\}$ (weight $w_{format} = 0.1$) enforces this structure.

Accuracy Reward. To prioritize task performance, we assign $R_a = 1$ if the predicted answer exactly matches the ground truth, else 0. This component receives the highest weight ($w_{\text{accuracy}} = 0.5$) to prioritize task performance while the other rewards guide how the model arrives at correct answers.

Count Reward. The count reward encourages the model to predict the appropriate number of objects and relations relevant to the spatial query. It penalizes both under- and over-generation, using a weighted error term based on the deviation between predicted and ground-truth counts: $R_c = w_{count} \cdot (0.7 \cdot \text{obj-score} + 0.3 \cdot \text{rel-score})$, where $w_{count} = 0.2$. This guides the model to stay focused on question-relevant regions. Without it, models tend to game the spatial reward by generating excessive objects and relations to boost match likelihood.

Spatial Reward. To supervise object localization, we compute the spatial reward only when the final answer is correct. Predicted and ground-truth objects are matched using the Hungarian algorithm with a cost function that combines Complete IoU (CIoU) and semantic similarity: $C(o_i^{\text{pred}}, o_j^{\text{gt}}) = \lambda_{\text{spatial}}(1 - \text{IoU}(b_i, b_j)) + \lambda_{\text{semantic}}(1 - \text{sim}(l_i, l_j))$, where b and l denote bounding boxes and labels, respectively. The reward is then computed as the average CIoU across matched pairs: $R_{\text{spatial}} = \frac{1}{|\mathcal{M}|} \sum_{(i,j)\in\mathcal{M}} \text{CIoU}(b_i^{\text{pred}}, b_j^{\text{gt}}); w_{spatial} = 0.2$. CIoU offers dense supervision over IoU, even for non-overlapping boxes by incorporating distance and aspect ratio terms [97].

Lexicographic Gating. To avoid reward gaming across objectives, we apply lexicographic ordering with conditional gating [70], prioritizing format $\succ$ {count, accuracy} $\succ$ spatial. The model must first satisfy formatting, then jointly optimize count and accuracy, and receives spatial reward only when the answer is correct. This ensures spatial grounding reinforces valid reasoning. Without accuracy gating, we observe that models overfit to spatial localization while sacrificing task correctness. The final reward is computed as the following with $\mathbb{I}[\cdot]$ as the indicator function:

$$R_{\text{total}} = \mathbb{I}[R_{\text{format}} > 0] \cdot (w_f R_f + w_c R_c + w_a R_a + \mathbb{I}[R_{\text{accuracy}} > 0] \cdot w_s R_s)$$

2.2 Online RL Policy Optimization

To train SpatialThinker with dense, lexicographically gated rewards, we adopt Group-Relative Policy Optimization (GRPO) [20, 67], an online RL method that avoids critic networks by estimating advantages through intra-group comparisons. Given an input $\mathbf{x}$, we sample N trajectories $\{y^{(1)}, \dots, y^{(N)}\}$ from the current policy $\pi_{\theta_{\text{old}}}$. Each response is scored via our dense spatial reward function (Section 2.1), and advantages are computed using group-normalized scores: $A^{(i)} = \frac{r^{(i)} - \mu}{\sigma + \varepsilon}$, where μ and σ are the group mean and standard deviation, and $\varepsilon = 10^{-6}$. We then update the policy using a PPO-style clipped loss with KL regularization:

$$\mathcal{L}_{\text{RL}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y^{(i)}|}\sum_{t=1}^{|y^{(i)}|}\Big[\min\Big(r^{i,t}A^{(i)},\, \text{clip}(r^{i,t}, 1-\epsilon_l, 1+\epsilon_h)A^{(i)}\Big) - \beta\, D_{\text{KL}}^{i,t}\Big],$$

where $r^{i,t} = \frac{\pi_\theta(y_t^{(i)} | \mathbf{x}, y_{<t}^{(i)})}{\pi_{\theta_{\text{old}}}(y_t^{(i)} | \mathbf{x}, y_{<t}^{(i)})}$ is the importance ratio between new and old policies, and $D_{\text{KL}}^{i,t}$ is the token-level KL divergence against a reference model. We set $\epsilon_l = 0.2$, $\epsilon_h = 0.3$, and $\beta = 10^{-2}$. This objective balances learning from dense spatial rewards while constraining policy divergence to ensure stability and generalization.

2.3 STVQA-7K: Dataset Construction

To facilitate reward-aligned spatial reasoning, we construct STVQA-7K, a synthetic visual question answering (VQA) dataset built from human-annotated scene graphs in Visual Genome [44]. STVQA-7K comprises 7,587 spatially grounded multiple-choice VQA pairs spanning both 2D and 3D spatial understanding. We augment the original VG150 predicate set with 34 additional spatial relations—covering distance (e.g., near, far), size (e.g., bigger, taller), orientation (e.g., facing away), and containment (e.g., inside, beneath)—to enrich the relational vocabulary beyond the standard 50 predicates. Each QA pair is generated from a scene graph using Claude Sonnet 4 [6], then verified for semantic correctness using GPT-4o [39] through a consistency-based filtering pipeline. From an initial pool of 56,224 questions, we retain the top 7.5K high-quality samples after automated rating, difficulty estimation, and validation. Finally, we align each question with a subgraph of relevant objects and relations, enabling localized scene graph supervision during training. This results in a richly annotated, task-aligned dataset for developing and evaluating grounded spatial reasoning models. Complete data construction details are provided in Appendix C.

3 Experiments

Implementation Details. We build SpatialThinker upon two strong open-source multimodal base models: Qwen2.5-VL-3B and Qwen2.5-VL-7B [7]. No supervised fine-tuning is performed prior to RL training on our STVQA-7K dataset (Section C). We employ GRPO [67] as the advantage estimator as described in Section 2.2, using a rollout size of 8 samples per query and a sampling temperature of 1.0. The models are trained with a maximum context length of 16,384 tokens. The rollout batch size is set to 512, and the global batch size is 128. We train for 75 training steps i.e., 5 training episodes) on 4 $\times$ NVIDIA H100 80GB GPUs. Training time totals around 13 hours for the 3B model and 15 hours for the 7B model. The models are trained on high-resolution image inputs ranging from 512×512 to 2048×2048 pixels, to preserve fine-grained spatial information. All model parameters, including the vision encoder, are updated during training. We use the AdamW optimizer with `bf16` precision, a learning rate of 1×10^{-6}, and a weight decay of 1×10^{-2}. The KL penalty coefficient is set to 10^{-2} (Appendix F). STVQA-7K is partitioned with a 90/10 train–validation split.

Experimental Setup. We evaluate SpatialThinker on six spatial reasoning benchmarks spanning 2D and 3D understanding: CV-Bench [74], BLINK [25], 3DSRBench [56], MMVP [75], SpatialBench [8], and RealWorldQA [85]. Comparisons include both proprietary (GPT-4o [39]) and open-source models—Qwen2.5-VL [7], Cambrian-1 [74], LLaVA-Next [46], VLAA-Thinker [12]—as well as spatially-specialized models such as SpatialRGPT [17], SpatialBot [8], SpaceLLaVA [10], SpaceThinker [3], and RoboPoint [93]. We also evaluate training variants including supervised fine-tuning (SFT) and vanilla GRPO (using only format and accuracy rewards) to isolate the contribution of dense spatial rewards. Detailed experimental setup, evaluation settings, and prompts are shared in Appendix D.

Model	3DSRBench [56]	CV-Bench [74]		Avg.	BLINK$_{val}$ [25]		Avg.
		2D	3D		Spatial Relation	Relative Depth	
Proprietary Models							
GPT-4o [39]	44.3	75.8	**83.0**	**79.4**	82.5	**78.2**	**80.4**
Claude 3.5 Sonnet [5]	48.2	60.2	71.5	65.9	58.7	67.7	63.2
Open-Source General MLLMs							
Qwen2.5-VL-3B [7]	44.0	59.9	60.2	60.0	66.4	54.0	60.2
Qwen2.5-VL-7B [7]	48.4	69.1	68.0	68.6	84.0	52.4	68.2
VLAA-Thinker-Qwen2.5-VL-7B [12]	52.2	60.8	60.3	60.6	81.2	71.0	76.1
LLaVA-NeXT-8B [46]	48.4	62.2	65.3	63.8	-	-	-
Cambrian-1-8B [74]	42.2	72.3	72.0	72.2	69.9	73.4	71.7
Open-Source Spatial MLLMs							
RoboPoint-13B [93]	-	-	61.2	-	60.8	61.3	61.1
SpatialBot-3B [8]	41.1	-	69.1	-	67.8	67.7	67.8
SpaceLLaVA-13B [1]	42.0	-	68.5	-	72.7	62.9	67.8
SATORI-R1 [68]	47.5	50.9	62.8	56.9	60.1	52.4	56.3
Spatial-RGPT-7B w/ depth [17]	48.4	-	60.7	-	65.7	82.3	74.0
SpaceThinker [3]	51.1	65.1	65.9	65.5	73.4	59.9	66.7
SpaceOm [2]	52.2	72.1	69.3	70.7	81.1	65.3	73.2
Method Comparison (Trained on STVQA-7K)							
Qwen2.5-VL-3B + SFT	50.8	53.9	68.4	61.1	65.0	66.9	66.0
Qwen2.5-VL-3B + Vanilla GRPO	50.1	70.6	66.6	68.6	73.4	55.6	64.5
SpatialThinker-3B (Ours)	52.9	71.0	76.3	73.6	81.8	66.9	74.4
Qwen2.5-VL-7B + SFT	53.6	56.1	71.3	63.7	75.5	64.5	70.0
Qwen2.5-VL-7B + Vanilla GRPO	54.7	68.9	76.5	72.7	80.4	75.0	77.7
SpatialThinker-7B (Ours)	**56.4**	**77.7**	78.7	78.2	**86.0**	72.6	79.3

Table 1: Performance over 2D & 3D Spatial Understanding Benchmarks across different model types.

3.1 Results

Performance across spatial benchmarks. As shown in Tables 1 and 2, SpatialThinker-7B achieves strong performance across all benchmarks: 78.2% on CV-Bench (vs. GPT-4o's 79.4%), 79.3% on BLINK tasks (vs. GPT-4o's 80.4%), 78.0% on MMVP (vs. GPT-4o's 70.7%), 56.4% on 3DSRBench, outperforming GPT-4o by 12.1%, and 66.4% on SpatialBench (vs. GPT-4o's 67.0%). Our 7B model outperforms all baselines on MMVP, and all open-source baselines on SpatialReasonerEval. Despite using only RGB inputs and 7K training samples, SpatialThinker-7B matches or surpasses larger proprietary and spatially-specialized open-source models, and further enhances visual understanding on real-world VQA benchmarks (see Appendix G.2 & H).

RL Training with Dense Rewards Enables Stronger Generalization. Compared to SFT and vanilla GRPO, SpatialThinker-7B achieves +6% and +3.2% higher average accuracy, respectively over 6 spatial and 6 VQA tasks. Similarly, the 3B variant shows +5.5% and +4.1% avg. gains respectively. Notably, while vanilla GRPO gives modest gains over base model (+4% for 7B, +4.9% for 3B), training with our dense spatial reward nearly doubles $\times 1.8$ this gain (+7.2% for 7B, +9% for 3B), underscoring the complementary learning signal provided by count and spatial objectives. The same trend holds under out-of-distribution evaluation, with dense rewards enabling significantly better real-world transfer (see Appendix G.4).

4 Conclusion

We introduced SpatialThinker, a 3D-aware MLLM that achieves strong spatial reasoning by combining scene graph grounding with dense spatial rewards. Trained on just 7K samples, it surpasses GPT-4o on spatial benchmarks while outperforming models trained on larger datasets and specialised spatial MLLMs. Dense spatial rewards nearly double the gains of standard RL, underscoring the value of rich supervision signals. Future work could explore implicit spatial reasoning with latent tokens, and design unified multi-objective policies covering diverse visual tasks.

Model	MMVP [75]	SpatialReasonerEval [58]	SpatialBench [8]
Proprietary Models			
GPT-4o [39]	70.7	**85.8**	**67.0**
Claude 3.5 Sonnet [5]	71.3	84.1	63.2
Open-Source General & Spatial MLLMs			
Qwen2.5-VL-3B [7]	67.0	68.0	49.9
Qwen2.5-VL-7B [7]	72.3	70.6	62.5
VLAA-Thinker-7B [12]	75.3	61.2	66.2
SpaceThinker [3]	63.0	69.6	57.9
SATORI-R1 [68]	63.7	64.0	60.3
SpaceOm [2]	66.3	68.9	58.6
SpatialReasoner [58]	64.0	76.4	59.2
Visionary-R1 [86]	70.3	72.9	59.8
Method Comparison (Trained on STVQA-7K)			
Qwen2.5-VL-3B + SFT	62.7	67.5	56.3
Qwen2.5-VL-3B + Vanilla GRPO	68.3	69.3	56.9
SpatialThinker-3B (Ours)	69.0	76.5	61.5
Qwen2.5-VL-7B + SFT	68.3	70.8	63.5
Qwen2.5-VL-7B + Vanilla GRPO	74.3	79.6	64.2
SpatialThinker-7B (Ours)	**78.0**	82.7	66.4

Table 2: Performance on additional spatial benchmarks.

References

[1] Remyx AI and Salma Mayorquin. “SpaceLLaVA Models”. In: *Hugging Face* (Mar. 2025). URL: `https://huggingface.co/remyxai/SpaceLLaVA`.

[2] Remyx AI and Salma Mayorquin. “SpaceOm Models”. In: *Hugging Face* (2025). URL: `https://huggingface.co/remyxai/SpaceOm`.

[3] Remyx AI and Salma Mayorquin. “SpaceThinker Models”. In: *Hugging Face* (Apr. 2025). URL: `https://huggingface.co/remyxai/SpaceThinker-Qwen2.5VL-3B`.

[4] Anthropic. “Claude 3.7 Sonnet System Card”. In: *Anthropic* (Feb. 2025).

[5] Anthropic. “Model Card Addendum: Claude 3.5 Haiku and Upgraded Claude 3.5 Sonnet”. In: *Anthropic* (Oct. 2024).

[6] Anthropic. “System Card: Claude Opus 4 & Claude Sonnet 4”. In: *Anthropic System Cards* (May 2025).

[7] Shuai Bai et al. “Qwen2.5-VL Technical Report”. In: *ArXiv* abs/2502.13923 (2025).

[8] Wenxiao Cai et al. “Spatialbot: Precise spatial understanding with vision language models”. In: *arXiv preprint arXiv:2406.13642* (2024).

[9] Nicolas Carion et al. “End-to-end object detection with transformers”. In: *European conference on computer vision*. Springer. 2020, pp. 213–229.

[10] Boyuan Chen et al. “Spatialvlm: Endowing vision-language models with spatial reasoning capabilities”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 14455–14465.

[11] Guikun Chen, Jin Li, and Wenguan Wang. “Scene Graph Generation with Role-Playing Large Language Models”. In: *ArXiv* abs/2410.15364 (2024).

[12] Hardy Chen et al. “SFT or RL? An Early Investigation into Training R1-Like Reasoning Large Vision-Language Models”. In: *ArXiv* abs/2504.11468 (2025).

[13] Kaiyuan Chen et al. *Robo2VLM: Visual Question Answering from Large-Scale In-the-Wild Robot Manipulation Datasets*. 2025. arXiv: `2505.15517 [cs.RO]`.

[14] Lin Chen et al. “Are We on the Right Way for Evaluating Large Vision-Language Models?” In: *ArXiv* abs/2403.20330 (2024).

[15] Zuyao Chen et al. “Compile scene graphs with reinforcement learning”. In: *arXiv preprint arXiv:2504.13617* (2025).

[16] Zuyao Chen et al. “Gpt4sgg: Synthesizing scene graphs from holistic and region-specific narratives”. In: *arXiv preprint arXiv:2312.04314* (2023).

[17] An-Chieh Cheng et al. “Spatialrgpt: Grounded spatial reasoning in vision-language models”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 135062–135093.

[18] Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. “Reltr: Relation transformer for scene graph generation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.9 (2023), pp. 11169–11183.

[19] Erik Daxberger et al. “Mm-spatial: Exploring 3d spatial understanding in multimodal llms”. In: *arXiv preprint arXiv:2503.13111* (2025).

[20] DeepSeek-AI et al. “DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning”. In: *ArXiv* abs/2501.12948 (2025).

[21] Matt Deitke et al. “Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models”. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 91–104.

[22] Yihe Deng et al. “OpenVLThinker: Complex Vision-Language Reasoning via Iterative SFT-RL Cycles”. In: 2025.

[23] Danny Driess et al. “PaLM-E: An Embodied Multimodal Language Model”. In: *International Conference on Machine Learning*. 2023.

[24] Kimi Team Angang Du et al. “Kimi-VL Technical Report”. In: *ArXiv* abs/2504.07491 (2025).

[25] Xingyu Fu et al. “BLINK: Multimodal Large Language Models Can See but Not Perceive”. In: *ArXiv* abs/2404.12390 (2024).

[26] Kanishk Gandhi et al. “Cognitive Behaviors that Enable Self-Improving Reasoners, or, Four Habits of Highly Effective STaRs”. In: *ArXiv* abs/2503.01307 (2025).

[27] Jensen Gao et al. “Physically Grounded Vision-Language Models for Robotic Manipulation”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)* (2023), pp. 12462–12469.

[28] Gemini Team and Google. “Gemini 1.5: Unlocking Multimodal Understanding across Millions of Tokens of Context”. In: *Google DeepMind* (2024).

[29] Google. “Gemini 2.0 Flash: Model Card”. In: *Technical Report* (Apr. 2025). Published April 15, 2025.

[30] Qiao Gu et al. “ConceptGraphs: Open-Vocabulary 3D Scene Graphs for Perception and Planning”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)* (2023), pp. 5021–5028.

[31] Tianrui Guan et al. “Hallusionbench: An Advanced Diagnostic Suite for Entangled Language Hallucination and Visual Illusion in Large Vision-Language Models”. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023), pp. 14375–14385.

[32] Marcel Hildebrandt et al. “Scene Graph Reasoning for Visual Question Answering”. In: *ArXiv* abs/2007.01072 (2020).

[33] Yining Hong et al. “3D concept learning and reasoning from multi-view images”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 9202–9212.

[34] Yining Hong et al. “3D-LLM: Injecting the 3D world into large language models”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 20482–20494.

[35] Yining Hong et al. “3d-llm: Injecting the 3d world into large language models”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 20482–20494.

[36] Chen Huang et al. “Visual Language Maps for Robot Navigation”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)* (2022), pp. 10608–10615.

[37] Wenxuan Huang et al. “Vision-R1: Incentivizing Reasoning Capability in Multimodal Large Language Models”. In: *ArXiv* abs/2503.06749 (2025).

[38] Drew A Hudson and Christopher D Manning. “Gqa: A new dataset for real-world visual reasoning and compositional question answering”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 6700–6709.

[39] Aaron Hurst et al. “Gpt-4o system card”. In: *arXiv preprint arXiv:2410.21276* (2024).

[40] Physical Intelligence et al. “π0.5: a Vision-Language-Action Model with Open-World Generalization”. In: *ArXiv* abs/2504.16054 (2025).

[41] Amita Kamath, Jack Hessel, and Kai-Wei Chang. “What’s" up" with vision-language models? investigating their struggle with spatial reasoning”. In: *arXiv preprint arXiv:2310.19785* (2023).

[42] Kibum Kim et al. “Llm4sgg: Large language models for weakly supervised scene graph generation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 28306–28316.

[43] Mikhail Konenkov et al. “VR-GPT: Visual Language Model for Intelligent Virtual Reality Applications”. In: *ArXiv* abs/2405.11537 (2024).

[44] Ranjay Krishna et al. “Visual genome: Connecting language and vision using crowdsourced dense image annotations”. In: *International journal of computer vision* 123.1 (2017), pp. 32–73.

[45] Chengzu Li et al. “Topviewrs: Vision-language models as top-view spatial reasoners”. In: *arXiv preprint arXiv:2406.02537* (2024).

[46] Feng Li et al. “LLaVA-NeXT-Interleave: Tackling Multi-image, Video, and 3D in Large Multimodal Models”. In: *ArXiv* abs/2407.07895 (2024).

[47] Lin Li et al. “Relation-R1: Progressively Cognitive Chain-of-Thought Guided Reinforcement Learning for Unified Relation Comprehension”. In: *arXiv preprint arXiv:2504.14642* (2025).

[48] Lin Li et al. “Zero-shot Visual Relation Detection via Composite Visual Cues from Large Language Models”. In: *ArXiv* abs/2305.12476 (2023).

[49] Yanwei Li et al. “Mini-Gemini: Mining the Potential of Multi-modality Vision Language Models”. In: *ArXiv* abs/2403.18814 (2024).

[50] Ji Lin et al. “Vila: On pre-training for visual language models”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 26689–26699.

[51] Haotian Liu et al. “Improved baselines with visual instruction tuning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 26296–26306.

[52] Haotian Liu et al. “Visual Instruction Tuning”. In: *ArXiv* abs/2304.08485 (2023).

[53] Yuqi Liu et al. “Seg-Zero: Reasoning-Chain Guided Segmentation via Cognitive Reinforcement”. In: *ArXiv* abs/2503.06520 (2025).

[54] Ziyu Liu et al. “Visual-RFT: Visual Reinforcement Fine-Tuning”. In: *ArXiv* abs/2503.01785 (2025).

[55] Chenyang Ma et al. “Spatialpin: Enhancing spatial reasoning capabilities of vision-language models through prompting and interacting 3d priors”. In: *Advances in neural information processing systems* 37 (2024), pp. 68803–68832.

[56] Wufei Ma et al. “3DSRBench: A Comprehensive 3D Spatial Reasoning Benchmark”. In: *ArXiv* abs/2412.07825 (2024).

[57] Wufei Ma et al. “Spatialllm: A compound 3d-informed design towards spatially-intelligent large multimodal models”. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 17249–17260.

[58] Wufei Ma et al. “SpatialReasoner: Towards Explicit and Generalizable 3D Spatial Reasoning”. In: *ArXiv* abs/2504.20024 (2025).

[59] Fanqing Meng et al. “MM-Eureka: Exploring the Frontiers of Multimodal Reasoning with Rule-based Reinforcement Learning”. In: 2025.

[60] Roshanak Mirzaee et al. “Spartqa:: A textual question answering benchmark for spatial reasoning”. In: *arXiv preprint arXiv:2104.05832* (2021).

[61] Soroush Nasiriany et al. “PIVOT: Iterative Visual Prompting Elicits Actionable Knowledge for VLMs”. In: *ArXiv* abs/2402.07872 (2024).

[62] Michael Ogezi and Freda Shi. “SpaRE: Enhancing Spatial Reasoning in Vision-Language Models with Synthetic Data”. In: *arXiv preprint arXiv:2504.20648* (2025).

[63] Yi Peng et al. “LMM-R1: Empowering 3B LMMs with Strong Reasoning Abilities Through Two-Stage Rule-Based RL”. In: *ArXiv* abs/2503.07536 (2025).

[64] Zhiliang Peng et al. “Kosmos-2: Grounding multimodal large language models to the world”. In: *arXiv preprint arXiv:2306.14824* (2023).

[65] André Susano Pinto et al. “Tuning computer vision models with task rewards”. In: *ArXiv* abs/2302.08242 (2023).

[66] Hanoona Rasheed et al. “Glamm: Pixel grounding large multimodal model”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 13009–13018.

[67] Zhihong Shao et al. “DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models”. In: *ArXiv* abs/2402.03300 (2024).

[68] Chuming Shen et al. “SATORI-R1: Incentivizing Multimodal Reasoning with Spatial Grounding and Verifiable Rewards”. In: *ArXiv* abs/2505.19094 (2025).

[69] Haozhan Shen et al. “VLM-R1: A Stable and Generalizable R1-style Large Vision-Language Model”. In: *ArXiv* abs/2504.07615 (2025).

[70] Joar Skalse et al. “Lexicographic Multi-Objective Reinforcement Learning”. In: *ArXiv* abs/2212.13769 (2022).

[71] Chan Hee Song et al. “Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics”. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 15768–15780.

[72] Kexian Tang et al. “LEGO-Puzzles: How Good Are MLLMs at Multi-Step Spatial Reasoning?” In: *ArXiv* abs/2503.19990 (2025).

[73] Gemini Robotics Team et al. “Gemini Robotics: Bringing AI into the Physical World”. In: *ArXiv* abs/2503.20020 (2025).

[74] Shengbang Tong et al. “Cambrian-1: A Fully Open, Vision-Centric Exploration of Multimodal LLMs”. In: *ArXiv* abs/2406.16860 (2024).

[75] Shengbang Tong et al. “Eyes wide shut? exploring the visual shortcomings of multimodal llms”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 9568–9578.

[76] Jean Vassoyan, Nathanael Beau, and Roman Plaud. “Ignore the KL Penalty! Boosting Exploration on Critical Tokens to Enhance RL Fine-Tuning”. In: *ArXiv* abs/2502.06533 (2025).

[77] Johanna Wald et al. “Learning 3D Semantic Scene Graphs From 3D Indoor Reconstructions”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 3960–3969.

[78] Peiyao Wang and Haibin Ling. “SVQA-R1: Reinforcing Spatial Reasoning in MLLMs via View-Consistent Reward Optimization”. In: *ArXiv* abs/2506.01371 (2025).

[79] Peng Wang et al. “Qwen2-VL: Enhancing Vision-Language Model’s Perception of the World at Any Resolution”. In: *ArXiv* abs/2409.12191 (2024).

[80] Xingrui Wang et al. “3d-aware visual question answering about parts, poses and occlusions”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 58717–58735.

[81] Xingrui Wang et al. “Compositional 4d dynamic scenes understanding with physics priors for video question answering”. In: *arXiv preprint arXiv:2406.00622* (2024).

[82] Zehan Wang et al. “SpatialCLIP: Learning 3D-aware Image Representations from Spatially Discriminative Language”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2025), pp. 29656–29666. DOI: 10.1109/CVPR52734.2025.02761.

[83] Jason Wei et al. “Chain of Thought Prompting Elicits Reasoning in Large Language Models”. In: *ArXiv* abs/2201.11903 (2022).

[84] Penghao Wu and Saining Xie. “V*: Guided Visual Search as a Core Mechanism in Multimodal LLMs”. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023), pp. 13084–13094.

[85] xAI. “Grok-1.5 Vision Preview”. In: *xAI Blog* (Apr. 2024).

[86] Jiaer Xia et al. “Visionary-R1: Mitigating Shortcuts in Visual Reasoning with Reinforcement Learning”. In: *ArXiv* abs/2505.14677 (2025).

[87] Tong Xiao et al. “Advancing Multimodal Reasoning Capabilities of Multimodal Large Language Models via Visual Perception Reward”. In: *ArXiv* abs/2506.07218 (2025).

[88] Yutaro Yamada et al. “Evaluating spatial understanding of large language models”. In: *arXiv preprint arXiv:2310.14540* (2023).

[89] Jihan Yang et al. “Thinking in space: How multimodal large language models see, remember, and recall spaces”. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 10632–10643.

[90] Yi Yang et al. “R1-Onevision: Advancing Generalized Multimodal Reasoning through Cross-Modal Formalization”. In: *ArXiv* abs/2503.10615 (2025).

[91] Haoxuan You et al. “Ferret: Refer and ground anything anywhere at any granularity”. In: *arXiv preprint arXiv:2310.07704* (2023).

[92] Qiying Yu et al. “DAPO: An Open-Source LLM Reinforcement Learning System at Scale”. In: *ArXiv* abs/2503.14476 (2025).

[93] Wentao Yuan et al. “RoboPoint: A Vision-Language Model for Spatial Affordance Prediction for Robotics”. In: *ArXiv* abs/2406.10721 (2024).

[94] Yi-Fan Zhang et al. “MME-RealWorld: Could Your Multimodal LLM Challenge High-Resolution Real-World Scenarios that are Difficult for Humans?” In: *ArXiv* abs/2408.13257 (2024).

[95] Yaowei Zheng et al. “EasyR1: An Efficient, Scalable, Multi-Modality RL Training Framework”. In: *arXiv preprint arXiv:2501.12345* (2025).

[96] Yaowei Zheng et al. “LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models”. In: *ArXiv* abs/2403.13372 (2024).

[97] Zhaohui Zheng et al. “Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation”. In: *IEEE Transactions on Cybernetics* 52 (2020), pp. 8574–8586.

[98] Hengguang Zhou et al. “R1-Zero’s "Aha Moment" in Visual Reasoning on a 2B Non-SFT Model”. In: *ArXiv* abs/2503.05132 (2025).

[99] Fangrui Zhu et al. “Struct2D: A Perception-Guided Framework for Spatial Reasoning in Large Multimodal Models”. In: *ArXiv* abs/2506.04220 (2025).

Appendix

A Related Work

3D Spatial Reasoning in MLLMs. While multimodal large language models have achieved notable success in fundamental visual tasks [39, 50, 21, 51], their ability to perform complex spatial reasoning remains limited. Multiple evaluations have highlighted persistent shortcomings in this domain [60, 75, 41, 88, 45, 89, 56], which can be partially attributed to the predominance of datasets centered around visual perception rather than explicit spatial or relational grounding [38]. In response, considerable research has focused on incorporating 3D spatial information into MLLMs. Early approaches embed explicit representations such as point clouds or multi-view reconstructions [34, 33], while others generate structured spatial states or world models guided by physical priors [80, 81]. More recent systems have trained large-scale models with 3D-enhanced VQA datasets, such as SpatialVLM with 2B samples [10], and extensions like SpatialPIN [55] or SpatialBot [8], which inject 3D priors or auxiliary depth signals. SpatialRGPT [17] builds 3D scene graphs from RGB-depth data to produce a large 700k-sample spatial QA dataset for training, improving performance but requiring extensive pre-processing and data. Similarly, MM-Spatial [19], SpatialLLM [57], and SpaRE [62] address spatial reasoning with hundreds of thousands to millions of synthetic or reconstructed samples. Despite this progress, existing methods are either data-heavy, reliant on specialized 3D inputs, or restricted in modeling structured relational understanding. In contrast, SPATIALTHINKER achieves robust 3D spatial reasoning including object localization, and relational and regional understanding, using only 7K high-quality structured QA samples combined with reinforcement learning over dense spatial rewards.

Structured Visual Grounding in MLLMs. Scene graphs provide a structured representation of objects and their relations and have been widely explored for visual reasoning [32, 77, 30]. Classical scene graph generation builds on detection-relation pipelines [9, 18], but often struggles with multi-role or open-vocabulary reasoning. With the advent of LLMs, text-augmented approaches such as LLM4SGG and GPT4SGG convert captions into structured graphs [42, 16], while more advanced open-vocabulary SGG methods leverage VLMs or MLLMs to generalize beyond fixed ontologies [11, 48]. Recent RL-driven frameworks, such as R1-SGG and Relation-R1, train models to construct scene graphs directly with dense structural or cognitive rewards [15, 47], highlighting the utility of structured supervision. In parallel, region-aware MLLMs like KOSMOS-2 [64], Ferret [91], and GLaMM [66] improve spatial grounding by integrating region information through bounding boxes and textual region descriptions, enabling more precise localization within images.SPATIALTHINKER builds on these advances by explicitly grounding reasoning on scene subgraphs focused on the question-specific region of interest, combining structured scene understanding with interpretable, reward-guided spatial reasoning.

Multimodal Reinforcement Learning. Reinforcement learning (RL) has been widely adopted to enhance reasoning in MLLMs, extending chain-of-thought prompting [83] and fine-grained verifiable rewards to multimodal reasoning tasks. Recent works have applied RL for math reasoning [90, 59], classification and grounding [54], semantic segmentation [53], structured reasoning pipelines [68] or referring expressions comprehension and open vocabulary detection [69, 65, 54]. Spatial RL strategies have emerged as well: SVQA-R1 incorporates view-consistency rewards [78], while SpatialReasoner adds coordinate-aware supervision in reasoning [69, 58]. Despite these efforts, most existing methods rely on relatively simple or sparse reward signals, such as final answer accuracy or coarse coordinate supervision, which provide limited guidance for detailed spatial relational reasoning. SPATIALTHINKER advances this space with a fine-grained multi-objective reward design covering regional subgraph construction, comprising object localisation and relational grounding, and final correctness. The model predicts these structured representations first, then reasons over them for detailed and interpretable spatial inference.

B Preliminaries

Scene Graph Generation. A scene graph provides a structured representation of an image I as a directed graph $G = (V, E)$. Each node $v_i \in V$ denotes an object with a category label c_i and a 2D bounding box $b_i = (x_i, y_i, w_i, h_i)$; each edge $e_{ij} \in E$ is a relationship triplet $\langle v_i, r_{ij}, v_j \rangle$ capturing spatial or interactive relations (e.g., *left of*, *on*, *under*) [32, 77]. Classical SGG decomposes prediction into object detection and relation recognition [9, 18], while open-vocabulary methods leverage language/vision priors to generalize beyond fixed ontologies [11, 48]. We refer to *question-focused scene subgraphs* as $G_q = (V_q, E_q) \subseteq G$ that retain only objects and relations relevant to a given query q.

Reasoning in Multimodal Large Language Models. Multimodal large language models (MLLMs) define autoregressive policies π_θ over sequences of interleaved visual and textual tokens. Given an image $\mathbf{x}_{\text{img}}$ and a spatial question $\mathbf{x}_{\text{text}}$, the model generates a reasoning trace $\mathbf{y} = (a_1, \ldots, a_T)$, where each a_t represents a token from intermediate reasoning steps or the final answer. This policy is factorized as:

$$\pi_\theta(\mathbf{y} \mid \mathbf{x}_{\text{img}}, \mathbf{x}_{\text{text}}) = \prod_{t=1}^{T} \pi_\theta(a_t \mid \mathbf{x}_{\text{img}}, \mathbf{x}_{\text{text}}, a_{<t}) \tag{1}$$

While supervised fine-tuning enables models to imitate reasoning traces observed during training, reinforcement learning offers a principled way to optimize generation using explicit reward signals, often resulting in better generalization to out-of-distribution inputs and improved adherence to task-specific structure [26, 20, 37]. The reinforcement learning objective seeks to maximize expected reward over trajectories:

$$\max_\theta \; \mathbb{E}_{Q \sim \mathcal{D},\, \mathbf{y} \sim \pi_\theta(\cdot|Q)} \left[R(Q, \mathbf{y}) \right] \tag{2}$$

where $Q = \{\mathbf{x}_{\text{img}}, \mathbf{x}_{\text{text}}\}$ is the input query, $\mathcal{D}$ is the dataset distribution, and R is a verifiable reward function evaluating task correctness, formatting, and spatial grounding.

Task Formulation We cast spatial reasoning in MLLMs as the task of producing a visually grounded response $\mathbf{y}$ to a query $Q = \mathbf{x}_{\text{img}}, \mathbf{x}_{\text{text}}$. Unlike generic reasoning, our formulation explicitly requires constructing question-focused scene subgraphs G_q and reasoning over objects, bounding boxes, and relations. The policy π_θ is trained on spatially grounded VQA samples from STVQA-7K C using our multi-objective spatial reward R (Section 2.1), which enforces structural validity, count fidelity, answer accuracy, and precise spatial grounding.

C STVQA-7K: Dataset Construction

High-quality spatial VQA datasets remain scarce, as most existing benchmarks either lack grounded scene-graph annotations (i.e., explicit spatial coordinates for objects and relations) or fail to comprehensively cover both 2D and 3D spatial reasoning categories. Visual Genome [44] provides dense, human-annotated scene graphs that support strict grounding of both question generation and answer verification within a unified representational framework. Using Visual Genome, we synthetically constructed a spatial visual question answering dataset called SpatialThinker Visual Question Answering dataset i.e., STVQA-7K comprising 7,587 samples, fully grounded in human-annotated scene graphs [44], which we employed for post-training the SpatialThinker models. Importantly, our pipeline is scalable and can be extended to generate up to 108K samples, the maximum supported by Visual Genome, enabling future large-scale post-training or RL fine-tuning.

The original VG150 predicate set is limited to 50 relations, missing several important categories such as positional relations (e.g., left, right, beside), distance-based relations (e.g., near, far, next to), comparative size (e.g., smaller, taller, bigger), orientation (e.g., facing towards/away), and containment (e.g., inside, beneath). To address this gap, we extended the scene graph relation space with an additional 34 predicates, ensuring richer spatial coverage in both 2D and 3D reasoning. Bounding box coordinates are retained in absolute pixel space, rather than normalized values, to preserve real-world scale and spatial alignment, to enable both improved spatial reasoning and effective use of CIoU-based supervision during reward optimization. The dataset construction pipeline proceeds in three stages: (1) synthetic question generation from ground-truth scene graphs, (2) automated quality filtering with external verification, and (3) scene graph adaptation for regional alignment with individual questions.

Synthetic Question Generation. Visual Genome scene graphs serve as our foundational ground truth, providing object categories, bounding boxes, and relational triplets for over 150,000 images. We synthetically generate question-answer pairs for a given scene graph data using Claude Sonnet

4 [6], synthesizing multiple-choice questions based on the salient objects and meaningful spatial relations explicitly present in each graph. Each question-answer pair is accompanied with a rating generated out of 10 and the difficulty level. Our question generation encompasses nine distinct spatial reasoning categories: spatial relations (above, behind, near, etc.), physical reach and interaction (holding, touching), comparative size, orientation from specific viewpoints, instance location within image frames, depth ordering relative to the camera, distance comparisons to reference objects, object counting, and existence verification. This comprehensive taxonomy spans both 2D and 3D spatial understanding, providing a broad coverage of visual-spatial reasoning capabilities. To promote robust perception, we also include questions involving objects that are partially visible or occluded in the scene, encouraging the model to reason about spatial arrangements and fine-grained details. For each question, we generate a rating out of 10.

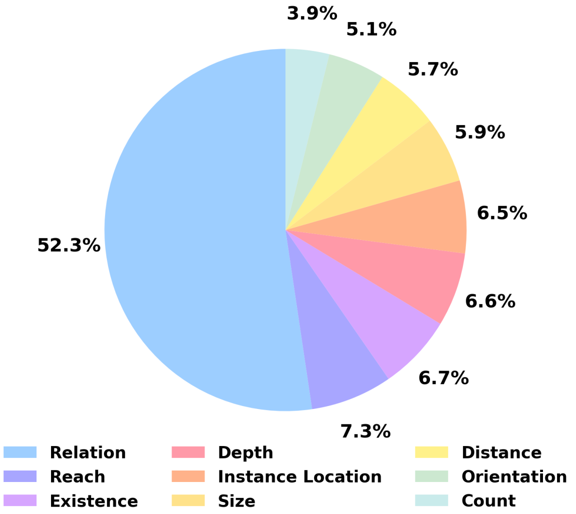
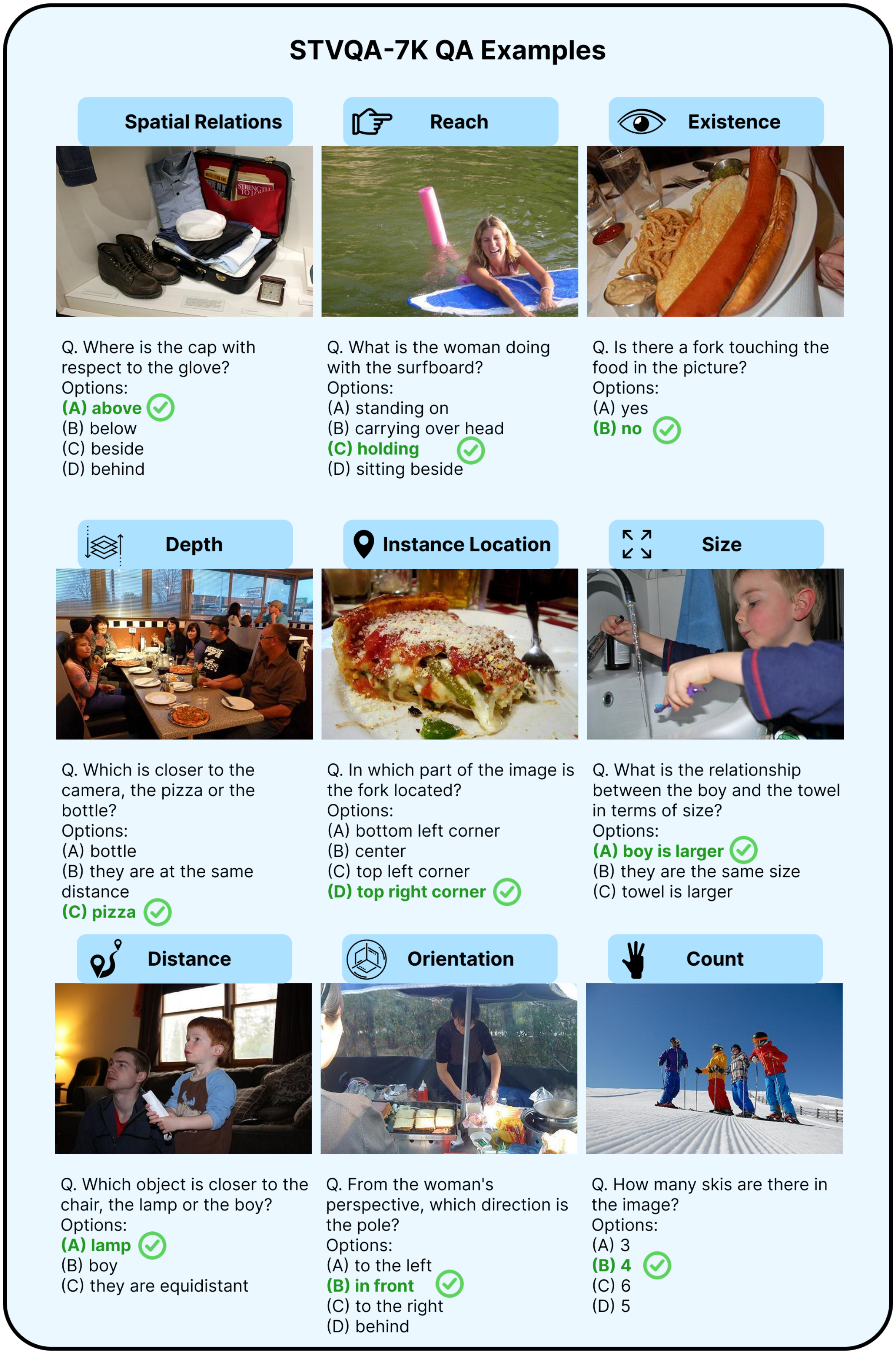


Figure 1: Distribution of QA types in STVQA-7K. The dataset spans a diverse range of spatial reasoning skills, with an emphasis on spatial relations while also balancing other categories such as localization, depth, distance, size, and orientation.

Quality Filtering and Validation. To ensure semantic correctness at scale, we implement a consistency-based verification procedure using GPT-4o [39] as an external validation model. For each generated question-answer pair, we assess agreement between the external model and our synthetic ground truth label using a pass@2 criterion. Questions that fail this initial consistency check undergo additional evaluation with two supplementary model responses. Items for which all four collected responses disagree with the generated label are discarded as potentially incorrect or ambiguous. This filtering process begins with 56,224 initially generated questions by Claude Sonnet 4 [6]. We select the 10,000 highest-rated samples based on the questions complexity and rating towards its contribution to enhance spatial intelligence as judged by Claude Sonnet 4. Following consistency filtering, we retain 6,895 training samples and 692 validation samples (75%), indicating high label reliability. The final set consists of 50% samples from the relation category, and the remaining 50% distributed across the eight other categories. To prevent positional bias, answers are uniformly distributed across options A, B, C, and D. Figure Figure 1 illustrates the distribution of QA types in STVQA-7K, highlighting the emphasis on spatial relations while maintaining balanced coverage across the remaining reasoning categories. Representative examples of generated QA pairs across the nine spatial reasoning categories are shown in Figure 2, illustrating the diversity of question types in STVQA-7K.

Scene Graph Adaptation. Since each question focuses on specific objects and relationships within the broader scene, we derive question-aligned scene subgraphs that capture only the relevant spatial context. For each question, we extract content words through tokenization and lemmatization to obtain both singular and plural word forms. We then filter the original scene graph to retain only object nodes whose labels appear in the extracted question vocabulary. Relational triplets are preserved when both the subject and object entities are retained and the predicate appears in the question context. The resulting focused scene graph representations enable training the model to generate question-aligned region-of-interest subgraphs, encouraging it to localize attention, ground reasoning in relevant entities and relations, and ultimately learn where to focus within complex visual scenes.



STVQA-7K QA Examples

Spatial Relations

Q. Where is the cap with respect to the glove?
Options:
(A) above ✓
(B) below
(C) beside
(D) behind

Reach

Q. What is the woman doing with the surfboard?
Options:
(A) standing on
(B) carrying over head
(C) holding ✓
(D) sitting beside

Existence

Q. Is there a fork touching the food in the picture?
Options:
(A) yes
(B) no ✓

Depth

Q. Which is closer to the camera, the pizza or the bottle?
Options:
(A) bottle
(B) they are at the same distance
(C) pizza ✓

Instance Location

Q. In which part of the image is the fork located?
Options:
(A) bottom left corner
(B) center
(C) top left corner
(D) top right corner ✓

Size

Q. What is the relationship between the boy and the towel in terms of size?
Options:
(A) boy is larger ✓
(B) they are the same size
(C) towel is larger

Distance

Q. Which object is closer to the chair, the lamp or the boy?
Options:
(A) lamp ✓
(B) boy
(C) they are equidistant

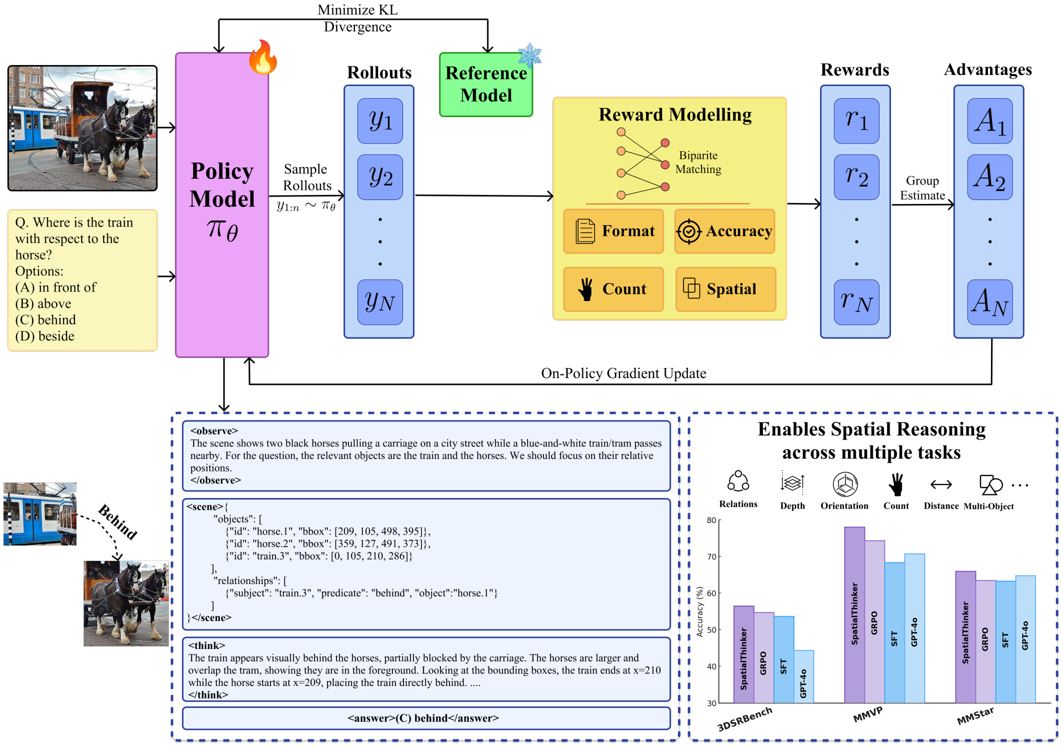
Orientation

Q. From the woman's perspective, which direction is the pole?
Options:
(A) to the left
(B) in front ✓
(C) to the right
(D) behind

Count

Q. How many skis are there in the image?
Options:
(A) 3
(B) 4 ✓
(C) 6
(D) 5

Figure 2: Examples of generated QA pairs across the nine spatial reasoning categories in STVQA-7K. Each category highlights distinct reasoning skills, ranging from relative spatial relations and depth ordering to distance, size, orientation, reach, location, count and existence.



Q. Where is the train with respect to the horse?
Options:
(A) in front of
(B) above
(C) behind
(D) beside

<observe>
The scene shows two black horses pulling a carriage on a city street while a blue-and-white train/tram passes nearby. For the question, the relevant objects are the train and the horses. We should focus on their relative positions.
</observe>

```
<scene>{
        "objects": [
            {"id": "horse.1", "bbox": [209, 105, 498, 395]},
            {"id": "horse.2", "bbox": [359, 127, 491, 373]},
            {"id": "train.3", "bbox": [0, 105, 210, 286]}
        ],
        "relationships": [
            {"subject": "train.3", "predicate": "behind", "object":"horse.1"}
        ]
}</scene>
```

<think>
The train appears visually behind the horses, partially blocked by the carriage. The horses are larger and overlap the tram, showing they are in the foreground. Looking at the bounding boxes, the train ends at x=210 while the horse starts at x=209, placing the train directly behind. ....
</think>

<answer>(C) behind</answer>

Figure 3: Method overview of SpatialThinker. Our framework integrates structured scene-graph grounded reasoning with multi-objective dense RL to enhance 3D spatial understanding in multimodal large language models.

D Experimental Setup Details

This section presents comprehensive evaluations of SpatialThinker across multiple spatial reasoning benchmarks, demonstrating the effectiveness of our multi-objective dense reward design and data-efficient training approach.

D.1 Implementation Details

We build SpatialThinker upon two strong open-source multimodal base models: Qwen2.5-VL-3B and Qwen2.5-VL-7B [7], using them as backbones for policy optimization with reinforcement learning. No supervised fine-tuning is performed prior to RL training on our STVQA-7K dataset (Section C). We employ GRPO [67] as the advantage estimator as described in Section 2.2, using a rollout size of 8 samples per query and a sampling temperature of 1.0. The models are trained with a maximum context length of 16,384 tokens. The rollout batch size is set to 512, and the global batch size is 128. We train for 75 training steps i.e., 5 training episodes) on 4 × NVIDIA H100 80GB GPUs. Training time totals around 13 hours for the 3B model and 15 hours for the 7B model.

The models are trained on high-resolution image inputs ranging from 512×512 to 2048×2048 pixels, to preserve fine-grained spatial information. All model parameters, including the vision encoder, are updated during training. We use the AdamW optimizer with `bf16` precision, a learning rate of 1×10^{-6}, and a weight decay of 1×10^{-2}. The KL penalty coefficient is set to 10^{-2}. STVQA-7K is partitioned with a 90/10 train–validation split.

D.2 Experimental Setup

We evaluate SpatialThinker across a diverse suite of 12 spatial understanding and real-world VQA benchmarks, covering both 2D and 3D understanding aspects to assess fine-grained spatial reasoning capabilities and real-world generalization. We compare against both proprietary and open-source baselines, including models specifically trained for spatial reasoning tasks. Our experiments address two key questions: (Q1) Does our spatial VQA data generation pipeline, combined with dense reward RL, improve MLLMs' general spatial reasoning capabilities? (Q2) How effectively can MLLMs learn spatial understanding from just 7K synthetic training samples, and how does this compare to models trained on orders-of-magnitude larger datasets?

Benchmarks. We evaluate models across six core spatial benchmarks, and six general-purpose VQA and real-world understanding datasets. The spatial benchmarks includes CV-Bench [74] that measures 2D spatial relations, object counting, depth ordering, and distance reasoning. BLINK's Spatial Relations and Relative Depth tasks [25] test directional and positional understanding, and fine-grained point-level depth perception—particularly challenging as SPATIALTHINKER receives no explicit point-level supervision during training 3DSRBench [56] assesses egocentric 3D spatial reasoning via relational and multi-object comparisons. MMVP [75] examines visual pattern recognition across attributes such as orientation, positional relations, existence, viewpoint, and size. SpatialBench [8] assesses general spatial comprehension across counting, existence, positional relationships, physical interactions such as reach, and size comparisons. Finally, SpatialReasonerEval [58] emphasizes depth and distance reasoning within 3D spatial tasks.

To assess broader generalization, we further evaluate models on six diverse real-world benchmarks. VStarBench [84] measures accurate localization and recognition of key objects in complex natural scenes. RealWorldQA [85] requires integrating visual inputs with commonsense and multi-step reasoning for real-world understanding. MME-RealWorld [94] spans five challenging domains including optical character recognition in the wild, remote sensing, diagram and table interpretation, autonomous driving, and scene monitoring. RoboSpatial-Home [71] simulates embodied spatial reasoning tasks involving object-object relationships, compatibility, and reference-frame switching (ego-centric, object-centric, and world-centric). We only use Configuration and Compatibility subsets of RoboSpatial-Home. MM-Star [14] provides a holistic benchmark covering math, logical reasoning, instance recognition, and fine/coarse visual perception. HallusionBench [31] evaluates hallucination resistance in multimodal models, requiring accurate visual grounding to counteract entangled linguistic or perceptual illusions. Together, these benchmarks allow us to probe spatial and perceptual reasoning across synthetic, embodied, and naturalistic settings.

Closed-Source MLLM Baselines. Among proprietary models, we evaluate GPT-4o (GPT-4O-0513) [39] and Claude 3.5 Sonnet (CLAUDE-3.5-SONNET-0620) [5], which represent the current state-of-the-art in commercial multimodal reasoning. These serve as upper bounds for spatial generalization under non-public training regimes.

Open-Source Generalist MLLM Baselines. We compare against generalist open-source MLLMs including Qwen2.5-VL 3B and 7B models [7], LLaVA-NeXT [46], Cambrian-1 [74], and VLAA-Thinker (3B and 7B) [12]. These models represent state-of-the-art vision-language architectures, offering strong general visual reasoning but without specific spatial tuning.

Open-Source Spatial MLLM Baselines. We benchmark against specialized open-source models designed for spatial reasoning: SpaceLLaVA-13B [1, 10] – a public re-implementation of SpatialVLM, SpatialRGPT-7B [17] incorporates region-level supervision and explicit depth maps into training, RoboPoint-13B [93], which instruction-tunes an MLLM to predict image key-point affordances for robotics and spatial affordance tasks, SpaceThinker [3], a fine-tuned VLAA-Thinker model for spatial reasoning, and its improved successor SpaceOm [2], which incorporates deeper chain-of-thought traces and Robo2VLM data [13]. Other baselines include SpatialReasoner [58], trained with RL and explicit 3D representations, and SpatialBot [8], which integrates RGB and depth inputs for robust spatial perception.

In addition to the above, we compare against our training variants including supervised fine-tuning (SFT) baselines and vanilla GRPO trained with sparse rewards (accuracy and format only) to isolate the contribution of our dense spatial reward framework.

In addition to external baselines, we evaluate ablations on variants of our model trained with the STVQA-7K dataset: a supervised fine-tuning (SFT) baseline, and a sparse-reward RL baseline that optimizes only format and accuracy rewards, each weighted equally at 0.5. These ablations allow us to isolate the contribution of our proposed multi-objective dense spatial reward function.

Evaluation Setting. We report accuracy as the primary evaluation metric across all benchmarks. All models are evaluated under zero-shot settings, using greedy decoding (temperature = 0.0, max_new_tokens = 2048) to ensure deterministic and reproducible outputs. For models with specific reasoning templates such as VLAA-Thinker, SpaceThinker, and SpaceOm, we utilize their corresponding structured prompts. In line with their original training setup, SpatialRGPT receives depth inputs, while all other models are evaluated using RGB images alone. Our evaluation pipeline builds upon OpenVLThinker's evaluation framework [22], adapted to support our new benchmark and dataset formats.

D.3 SpatialThinker Prompt Format

We use a structured prompt to guide the model through a four-stage reasoning process, explicitly separated using the tags `<observe>`, `<scene>`, `<think>`, and `<answer>`. This format is enforced during training via a binary format reward $R_f \in \{0, 1\}$, with weight $w_{\text{format}} = 0.1$, which verifies the presence, ordering, and validity of all required tags. The `<scene>` section must contain a JSON-encoded subgraph with object IDs, bounding boxes, and relational triplets, while the final answer must be clearly placed within the `<answer>` tags.

Each prompt also includes the input image dimensions in the form `Image size: {Width}` × `{Height}`, which are dynamically replaced with actual values. Including this information helps the model constrain predicted bounding box coordinates within image bounds, enabling better spatial localization. These coordinates are directly evaluated using IoU-based spatial rewards such as Complete IoU (CIoU), making dimension-aware prediction essential for optimizing structured spatial grounding.

SpatialThinker Prompt

You FIRST observe the image in <observe> </observe> tags, then visualise the relevant scene graph in <scene> </scene> tags, followed by thinking about the reasoning process as an internal monologue within <think> </think> tags and then provide the final answer. The final answer MUST BE put within <answer> </answer> tags, and only return the final choice including the correct option and answer within the answer tags, e.g., <answer> (C) The red cube is left of the green sphere </answer>.
Image size: {Width} × {Height}

D.4 Details on SFT Training

To establish a comprehensive baseline for comparison with our reinforcement learning approach, we conduct supervised fine-tuning (SFT) experiments using the same base models (Qwen2.5-VL-3B and Qwen2.5-VL-7B) and training dataset (STVQA-7K). The SFT implementation utilizes LLaMA-Factory framework [96] with Low-Rank Adaptation (LoRA) for parameter-efficient fine-tuning.

The training configuration employs LoRA with rank 8 applied to all available modules within the model architecture, enabling comprehensive adaptation while maintaining computational efficiency. Models are trained for 3 epochs totaling 645 training steps, using a context window length of 2048 tokens. We adopt BF16 mixed precision training with a learning rate of 1×10^{-4}, following a cosine learning rate schedule with a warmup ratio of 0.1.

For the SFT experiments, we train models directly on question-answer pairs without intermediate reasoning traces or chain-of-thought prompting. This design choice reflects the practical constraint that generating ground-truth reasoning traces would require additional dataset processing, annotation, and API credits budget. In contrast, reinforcement learning approaches with verifiable rewards (RLVR) naturally enables training with answer supervision alone, as the model learns to generate its own reasoning strategies through environmental feedback rather than imitating pre-specified reasoning patterns.

The SFT baseline serves a critical role in our experimental evaluation, providing direct evidence of the generalization advantages offered by reinforcement learning with dense spatial rewards compared to traditional supervised learning on the same dataset.

D.5 Details on RL Training

We implement reinforcement learning training using the EasyR1 framework [95], building upon Qwen2.5-VL-3B and Qwen2.5-VL-7B as base models without any prior supervised fine-tuning. This direct application of RL to the base models enables us to isolate the effects of reward-driven learning from potential confounding factors introduced by intermediate training stages. Additionally, performing an SFT stage prior to RL would require generating ground-truth reasoning traces, which is limited by API budget. Moreover, explicit reasoning supervision is not strictly necessary—our multi-objective dense spatial rewards encourage the model to acquire structured reasoning and self-reflection abilities directly during RL training.

The training employs Group Relative Policy Optimization (GRPO) [67] as the advantage estimation method, configured with a rollout size of 8 samples per query at a sampling temperature of 1.0. This configuration balances exploration diversity with computational efficiency, allowing the model to discover multiple reasoning strategies while maintaining stable convergence. The training process utilizes a rollout batch size of 512 and a global batch size of 128, processing data through 75 training steps (approximately 5 training episodes) to achieve convergence. The entire training pipeline runs on $4 \times$ NVIDIA H100 80GB GPUs, requiring approximately ~ 13 hours for the 3B model and ~ 15 hours for the 7B variant.

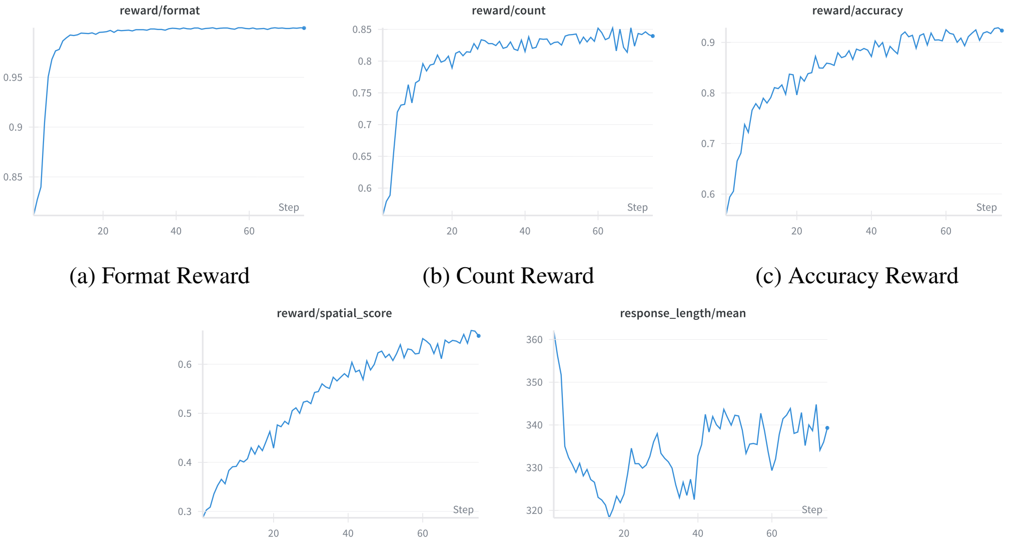
To preserve fine-grained spatial information critical for accurate object localization and spatial reasoning, models process high-resolution image inputs ranging from 512×512 to 2048×2048 pixels. The training configuration updates all model parameters including the vision encoder, enabling comprehensive adaptation to spatial reasoning tasks. Optimization employs AdamW with BF16 mixed precision, a conservative learning rate of 1×10^{-6}, and weight decay of 1×10^{-2}. The KL penalty coefficient is set to 10^{-2} to prevent excessive divergence from the base model distribution while allowing sufficient exploration for spatial reasoning strategies. The training utilizes a 90/10 train-validation split of the STVQA-7K dataset, with a maximum context length of 16,384 tokens to accommodate detailed scene descriptions and reasoning traces.

For baseline comparisons, we train vanilla GRPO models (Qwen2.5-VL-3B + Vanilla GRPO and Qwen2.5-VL-7B + Vanilla GRPO) using a simplified reward structure consisting solely of accuracy ($w_{acc} = 0.5$) and format rewards ($w_{format} = 0.5$), without the spatial grounding and count penalty components. This configuration represents standard RLVR approaches that rely on sparse final-answer supervision [20, 69, 12]. The full multi-objective reward design employed for SpatialThinker training, incorporating format, count, accuracy, and spatial rewards with lexicographic gating, is detailed in Section 2.1. The substantial performance improvements of SpatialThinker over vanilla GRPO baselines demonstrate the critical importance of dense spatial supervision in teaching models to perform visually-grounded reasoning.

D.5.1 SpatialThinker RL Training Curves

Throughout reinforcement learning, all four reward components: format, accuracy, count, and spatial; demonstrate consistent and interpretable improvement, reflecting stable learning under our lexicographically gated, multi-objective reward structure. The format reward quickly converges early in training, indicating the model learns to produce structurally valid outputs that adhere to the required scene-grounded reasoning format. Accuracy steadily improves across steps, highlighting the model's increasing ability to provide correct answers. Count reward rises consistently, showing that the model learns to focus on predicting only question-relevant objects and relations, rather than describing the entire scene. The spatial reward also improves gradually, indicating better object localization and grounding, as the model increasingly aligns predicted bounding boxes with ground truth annotations. Together, these trends reflect how each reward component scaffolds a different stage of the reasoning process, enforcing structure, correctness, focus, and grounding in tandem.

Response length initially declines, then rises again as it begins producing more deliberate, structured reasoning, signaling an "aha moment" where the model starts to produce more deliberate reasoning traces [20, 98]. This emergent behavior suggests the development of internal problem-solving strategies, as the model learns to spend more "thinking time" before answering, consistent with the emergence of self-reflection and structured planning in its spatial reasoning process.



(a) Format Reward

(b) Count Reward

(c) Accuracy Reward

(d) Spatial Reward

(e) Response Length

Figure 4: RL training dynamics of SPATIALTHINKER. All reward components (a–d) improve consistently, reflecting stable optimization. Response length (e) shows a non-monotonic trend, indicating emergent reasoning strategies.

E Reward Design Process

This section details our approach to designing a robust reward system that guides models toward genuine spatial reasoning while preventing degenerate solutions. Our reward design emerged from iterative refinement to address systematic reward hacking behaviors observed during training. Early experiments revealed that models readily exploit loopholes in reward functions—particularly when spatial localization rewards were provided without proper constraints. To empirically motivate our design choices, we first present an ablation over successive reward components on the STVQA-$_{\text{val}}$ split, followed by details of observed behaviours and analysis.

E.1 Reward Design Ablation

To empirically validate our design choices, we conduct a controlled ablation study on the STVQA-7K$_{\text{val}}$ set, progressively introducing each reward component and constraint. The ablation results support our design rationale by highlighting how each component mitigates specific failure modes. Adding spatial rewards naively without generation constraints, causes performance to collapse by over 50% (from 74.9% to 23.7%), as models exploit the reward by generating cluttered bounding boxes to game the CIoU metric. Introducing the count reward addresses this issue, improving accuracy by 38% relative (to 61.7%), as it constrains overgeneration and forces models to focus on question-relevant elements. However, residual overfitting persists because rewarding spatial alignment across all scene objects biases the model toward exhaustive global descriptions. To address this, we shift from global to local spatial supervision—rewarding only Regions of Interest (RoIs) derived from question-relevant objects and relations—thereby training the model to attend selectively to meaningful spatial cues rather than densely describing the entire scene. Lexicographic gating further ensures that spatial rewards are only applied when the final answer is correct, effectively serving as a bonus signal rather than a competing objective. This ordering stabilizes learning—without it, models overfit by optimizing auxiliary rewards more aggressively than accuracy itself. Together, these interventions restore and slightly surpass the original performance (76.3%), demonstrating the importance of grounding rewards in both correctness and relevance. Finally, dataset filtering using pass@2 correctness verification with gpt-4o [39] amplifies these effects, yielding a substantial gain and culminating in the best validation accuracy of 87.9%. This step ensures that only high-quality, verifiable supervision signals contribute to training, reinforcing the alignment between spatial grounding and task success.

Reward Components	**STVQA-7K$_{\text{val}}$**
Format + Accuracy	74.9
+ Spatial	23.7
+ Count	61.7
+ Lexicographic Gating and RoI Filtering	76.3
+ Filtered Dataset (pass@2)	**87.9**$_{(+13.0)}$

Table 3: Reward ablation on STVQA-7K$_{\text{val}}$. Progressive addition of constraints and filtering restores stable optimization and improves grounding quality. The final configuration (**yellow**) represents our full reward design.

E.2 Reward Design Rationale

Mitigating Spatial Reward Hacking. Our initial reward formulation, which directly rewarded spatial localization quality, led to unexpected model behavior. Without constraints on generation quantity, models discovered they could maximize spatial rewards by generating numerous bounding boxes with varying coordinates. Through Hungarian matching that selects the best-matching boxes, even random predictions would occasionally yield high Complete IoU (CIoU) scores. This reward hacking manifested as models producing excessive, hallucinated objects while achieving poor task accuracy—the spatial reward was inflated despite the clutter of irrelevant predictions degrading actual performance. To address this exploitation, we introduced the Count Reward that penalizes deviations from expected object and relation counts. This reward serves dual purposes: (1) preventing reward hacking by constraining the generation space, and (2) encouraging models to focus on question-relevant scene elements rather than exhaustively describing the entire image. The count reward formulation provides a linear penalty proportional to relative deviations from ground truth counts, normalized to prevent domination by scenes with many objects.

Scene Graph Filtering. Another form of overfitting emerged when training with complete Visual Genome scene graphs. Models would memorize exhaustive scene descriptions, including irrelevant background objects, leading to poor generalization. We addressed this by filtering ground truth scene graphs to retain only objects and relations relevant to the given question, focusing supervision on task-critical information.

CIoU over IoU for Spatial Reward. For spatial localization, we adopt Complete IoU (CIoU) instead of standard IoU to compute the spatial reward. Unlike IoU, which returns zero when predicted and ground-truth boxes do not overlap, CIoU provides meaningful gradients by incorporating center distance, aspect ratio, and overlap [97]. This makes CIoU a denser and more robust supervisory signal during training.

Balancing Supervision with Exploration. Our experiments reveal a crucial insight: models learn simple reward functions significantly faster than complex ones. Tasks with straightforward rewards (e.g., format compliance) show rapid improvements, while multi-component rewards require careful balancing. However, counterintuitively, highly detailed reward functions that attempt to supervise every aspect often degrade performance. Models overfit to maximize minute reward components, converging to template-style answers that score well on individual metrics while losing flexibility. We observed accuracy drops mid-training when rewards became too prescriptive, as models focused on reward optimization rather than genuine task understanding. Effective reinforcement learning requires providing guidance while preserving exploration space. Our final design addresses this by providing soft signals through format checks, count constraints, and accuracy rewards, with spatial localization rewards activated only for correct answers. This maintains the delicate balance between guidance and exploration necessary for robust learning.

Sequential Optimization via Lexicographic Gating. To prevent models from gaming individual reward components at the expense of task accuracy, we implement lexicographic gating [70]. Rewards are applied in a strict hierarchy: format $\succ$ {count, accuracy} $\succ$ spatial. This forces models to first master output formatting, then simultaneously learn to control generation scope and achieve correctness, before optimizing spatial grounding:

$$\mathrm{R}_{\text{total}} = \mathbb{I}[R_{\text{format}} = 1] \cdot (w_{\text{format}} \cdot R_f + w_{\text{count}} \cdot R_c + w_{\text{accuracy}} \cdot R_a + \mathbb{I}[R_{\text{accuracy}} = 1] \cdot w_{\text{spatial}} \cdot R_s)$$

where $\mathbb{I}[\cdot]$ is the indicator function, with weights $w_{\text{format}} = 0.1$, $w_{\text{count}} = 0.2$, $w_{\text{accuracy}} = 0.5$, $w_{\text{spatial}} = 0.2$. This gated design ensures spatial rewards are only applied when the final answer is correct, aligning grounding quality with task success and preventing scenarios where models achieve high spatial scores through precise but irrelevant localizations.

F Ablation on Divergence Constraints

Recent works such as DAPO [92, 76] argue that KL regularization can unnecessarily constrain policy updates and recommend removing the KL penalty entirely to allow freer exploration. In contrast, **?** revisit divergence regularization and propose using a chi-squared penalty to better control overoptimization. Motivated by these findings, we ablate the effect of different divergence constraints in our reinforcement learning setup for spatial reasoning.

Table 4 reports results on CV-Bench 2D and 3D tasks [74] for three variants of SPATIALTHINKER-3B: (i) no KL penalty, (ii) chi-squared divergence penalty with a coefficient of 0.01, and (iii) our default KL divergence penalty with a coefficient of 0.01. Removing the KL penalty leads to a noticeable drop in performance, particularly on 3D tasks. Using a chi-squared divergence penalty underperforms both the no-penalty and KL variants on several subtasks, especially depth and distance reasoning. The KL-regularized model achieves the best overall performance, yielding a CV-Bench average of 73.7% and providing the strongest results on 3D reasoning tasks.

These findings suggest that a modest KL penalty stabilizes policy updates and prevents reward overoptimization in our spatial reasoning setting, leading to more reliable improvements. While recent language-only alignment work has advocated for removing divergence constraints, our results indicate that retaining a small KL term remains beneficial for multimodal reasoning tasks where stability and coherent spatial grounding are crucial.

Model Variant	Count	Relation	Depth	Distance	CV-Bench 2D	CV-Bench 3D	CV-Bench Avg.
SpatialThinker-3B + No KL Penalty	65.5	**76.8**	74.8	70.2	**71.2**	72.5	71.9
SpatialThinker-3B + Chi^2 (0.01)	64.5	73.7	71.2	66.2	69.1	68.7	68.9
SpatialThinker-3B + KL (0.01)	**68.5**	73.5	**79.7**	**72.8**	71.0	**76.3**	**73.7**

Table 4: Ablation on divergence constraints for SPATIALTHINKER-3B on CV-Bench tasks. KL-regularization with $\beta = 0.01$ yields the highest overall average and strongest 3D reasoning performance.

G Detailed Results and Discussion

We evaluate SpatialThinker across six spatial reasoning and six generalist VQA benchmarks to assess its effectiveness in learning spatial understanding and real-world VQA from limited training data through dense reward supervision.

G.1 Performance across Spatial Benchmarks.

We evaluate SPATIALTHINKER across six spatial reasoning benchmarks that collectively span 2D relational understanding, 3D spatial alignment, counting, depth ordering, and distance comparison. As shown in Tables 1 and 2, SPATIALTHINKER-7B achieves strong and consistent performance across all spatial tasks. On CV-Bench, the model attains an average accuracy of 78.2% across 2D and 3D tasks, nearing GPT-4o's 79.4% while outperforming all other open-source models, and Claude 3.5 Sonnet. On the challenging 3DSRBench, which requires orientation and multi-object reasoning, it achieves 56.4%, surpassing GPT-4o by +12%. On BLINK's spatial relation and relative depth tasks, it achieves 86.0% and 72.6%, respectively, yielding a 79.3% average—closely matching GPT-4o (80.4%) and outperforming other spatial MLLMs like Spatial-RGPT-7B (74.0%), which uses depth inputs and 700K training samples. On SpatialBench, our model reaches 66.4%, approaching GPT-4o's 67.0%.

Despite being trained on just 7K synthetic samples and using only RGB inputs, SpatialThinker-7B consistently outperforms open-source baselines, including VLAA-Thinker-7B, Cambrian-1-8B, Spatial-RGPT, SpaceLLaVA, and RoboPoint-13B, all of which are trained on orders of magnitude more data. Notably, it exceeds specialized spatial models as well: on CV-Bench 3D, it outperforms SpaceLLaVA-13B (78.7% vs. 68.5%), and on BLINK tasks, it surpasses Spatial-RGPT-7B by +5.3%, and SpatialBot by +11.5% despite their reliance on depth information. Further, SpatialThinker-7B outperforms all models on MMVP, and all open-source baselines on SpatialReasonerEval that measures 3D spatial understanding tasks like depth and distance. These results highlight the effectiveness of our dense reward design in enabling generalizable spatial reasoning without the need for explicit geometric inputs or large-scale pretraining.

Model	MM-Star [14]	VStarBench [84]	RealWorldQA [85]	MME-RealWorld-Lite [94]	RoboSpatial-Home [71]	HallusionBench [31]
Proprietary and Open-Source MLLMs						
GPT-4o [39]	64.7	66.0	**75.4**	**51.6**	68.4	55.0
Claude 3.5 Sonnet [5]	65.1	51.8	60.1	45.2	57.0	55.5
Qwen2.5-VL-3B [7]	55.9	74.9	58.2	41.9	58.7	46.3
Qwen2.5-VL-7B [7]	63.9	75.9	68.4	44.1	70.6	52.9
VLAA-Thinker-7B [12]	63.8	58.1	66.4	44.6	68.9	**68.9**
SpaceThinker [3]	54.5	56.5	61.6	-	52.6	65.4
SpaceOm [2]	57.7	56.5	53.3	-	68.9	62.9
Method Comparison (Trained on STVQA-7K)						
Qwen2.5-VL-3B + SFT	53.9	73.3	64.8	43.0	69.8	58.9
Qwen2.5-VL-3B + Vanilla GRPO	56.7	74.3	64.4	46.7	64.0	59.0
SpatialThinker-3B (Ours)	57.6	78.0	66.3	46.5	70.6	62.5
Qwen2.5-VL-7B + SFT	63.2	78.0	65.4	47.4	72.4	66.2
Qwen2.5-VL-7B + Vanilla GRPO	63.4	73.9	66.6	46.3	76.2	60.7
SpatialThinker-7B (Ours)	**65.9**	**81.7**	69.2	48.3	**76.3**	66.4

Table 5: Performance on VQA and Real-World benchmarks. Top-1 & Top-2 accuracies are represented using **bold text**, and underlines.

G.2 Performance across Real-World and General VQA Benchmarks

We further assess our model's generalization to real-world visual question answering using six diverse benchmarks: MM-Star, RealWorldQA, VStarBench, MME-RealWorld-Lite, RoboSpatial-Home, and HallusionBench (Table 5). SpatialThinker-7B achieves the highest overall performance across these datasets. It obtains 65.9% on MM-Star, 81.7% on VStarBench, and 76.3% on RoboSpatial-Home, surpassing all open-source and proprietary baselines. It also performs competitively on hallucination-sensitive and real-world benchmarks, scoring 66.4% on HallusionBench, 69.2% on RealWorldQA, and 48.3% on MME-RealWorld-Lite benchmarks.

These results show that training with dense spatial rewards generalizes beyond synthetic benchmarks to real-world settings. Gains on MM-Star, RoboSpatial-Home, and VStarBench highlight the benefit of structured scene grounding, even with a small synthetic training set. Compared to generalist and open-source spatial MLLM baselines, SpatialThinker delivers greater robustness, fewer hallucinations, and higher task fidelity, reinforcing our hypothesis that spatial grounding via reward optimization not only improves spatial reasoning but also enhances visual understanding in the wild.

G.3 RL Training with Dense Rewards Enables Stronger Generalization

To isolate the contributions of our multi-objective spatial reward design, we compare against two ablation variants: supervised fine-tuning (SFT) and reinforcement learning with sparse rewards using only format and answer accuracy. As shown in Table 6, SpatialThinker-7B achieves an average accuracy of 71.2% across all 12 benchmarks—exceeding the SFT baseline by +6.0% and the sparse GRPO variant by +3.2%. These gains are consistent across the 3B variant as well, where SpatialThinker-3B outperforms its SFT and GRPO counterparts by +5.5% and +4.1% average gains, respectively. Notably, even

Model	Avg. Acc. (12)	Δ_{Base}	$\Delta_{\text{GPT-4o}}$	$\Delta_{\text{Claude 3.5 Sonnet}}$
Proprietary and Base MLLMs				
GPT-4o [39]	67.8	-	-	-
Claude 3.5 Sonnet [5]	61.1	-	-	-
Qwen2.5-VL-3B [7]	57.3	-	-	-
Qwen2.5-VL-7B [7]	64.0	-	-	-
Method Comparison (Trained on STVQA-7K)				
Qwen2.5-VL-3B + SFT	60.8	+3.5	-7.0	-0.3
Qwen2.5-VL-3B + Vanilla GRPO	62.2	+4.9	-5.6	+1.1
SpatialThinker-3B (Ours)	66.3	+9.0	-1.5	+5.2
Qwen2.5-VL-7B + SFT	65.2	+1.2	-2.6	+4.1
Qwen2.5-VL-7B + Vanilla GRPO	68.0	+4.0	+0.2	+6.9
SpatialThinker-7B (Ours)	**71.2**	**+7.2**	**+3.4**	**+10.1**

Table 6: Average accuracy across all 12 benchmarks with relative improvements (Δ). *SpatialThinker models consistently outperform SFT and vanilla GRPO, with SpatialThinker-7B surpassing GPT-4o by* ***+3.4*** *points and Claude 3.5 Sonnet by* ***+10.1*** *points.*

vanilla GRPO provides modest improvements over the base model (+4.0 for 7B, +4.9 for 3B), but our dense spatial reward nearly doubles ×1.8 this gain (+7.2% for 7B, +9.0% for 3B), underscoring the complementary learning signal provided by count and spatial objectives.

Beyond aggregate accuracy, lexicographic reward gating stabilizes training by enforcing format and answer correctness before applying spatial rewards. This encourages structured task completion prior to spatial grounding, resulting in steady and interpretable reward curves during training (Section **??**). Overall, these results affirm that structured reinforcement learning with dense spatial supervision significantly enhances the capabilities of multimodal LLMs, even in low-data regimes.

G.4 Out-of-Distribution Generalization: Dense Rewards Enable Stronger Transfer

While both SFT and sparse-reward GRPO improve spatial reasoning over base models, their ability to generalize to out-of-distribution (OOD) real-world tasks is limited, when compared to SpatialThinker models. As shown in Table 7, sparse-reward GRPO provides large spatial gains (+4.3% for 3B, +4.7% for 7B), but offers only marginal improvements on real-world benchmarks (+6.0 and +2.7 respectively)—nearly matching or underperforming SFT (+5.9% for 3B, +2.9% for 7B). In contrast, SpatialThinker, trained with dense spatial and count rewards, achieves significantly stronger OOD generalization: +8.5 for 3B and +5.2 for 7B, outperforming all baselines at both scales. Notably, SpatialThinker-7B provides nearly double the real-world VQA benchmarks gains compared to sparse-reward GRPO (+5.2% vs. +2.7%), highlighting the robustness of our dense reward framework. The combination of structured reasoning formats and lexicographically gated dense rewards encourages models to internalize spatial priors and compositional patterns that transfer effectively to out-of-distribution tasks, even without explicit domain-specific supervision. Appendix H further demonstrates generalization to abstract reasoning tasks.

Model Variant	**Spatial VQA Δ_{Base}**	**Real-World VQA Δ_{Base}**
Qwen2.5-VL-3B + SFT	+2.3	+5.9
Qwen2.5-VL-3B + GRPO	+4.3	+6.0
SpatialThinker-3B	**+9.3**	**+8.5**
Qwen2.5-VL-7B + SFT	+0.3	+2.9
Qwen2.5-VL-7B + GRPO	+4.7	+2.7
SpatialThinker-7B	**+8.3**	**+5.2**

Table 7: Average accuracy gains (Δ) over respective base models on (6) spatial and (6) real-world VQA (OOD) benchmarks.

H Additional Results: Abstract Reasoning

To further evaluate the generalization capacity of SpatialThinker, we examine its performance on two abstract reasoning benchmarks: **Lego Puzzles** [72], which test compositional object reasoning and multi-step spatial reasoning, and **BLINK Multi-View** [25], which requires integrating spatial cues across multiple viewpoints, including visual-spatial understanding and perspective understanding. These tasks are not part of the training distribution and measure the ability of models to extrapolate structured reasoning skills to abstract domains.

Model	**Lego Puzzles [72]**	**BLINK Multi-View [25]**
Proprietary and Open-Source MLLMs		
GPT-4o [39]	**57.7**	**54.1**
Claude 3.5 Sonnet [5]	53.6	51.9
Qwen2.5-VL-3B [7]	29.9	42.9
Qwen2.5-VL-7B [7]	35.8	44.4
VLAA-Thinker-7B [12]	33.4	51.1
SpaceThinker [3]	31.5	50.4
SpaceOm [2]	32.0	48.9
Method Comparison (Trained on SpatialThinkerVQA)		
Qwen2.5-VL-3B + SFT	34.7	42.1
Qwen2.5-VL-3B + Vanilla GRPO	27.0	45.9
SpatialThinker-3B (Ours)	33.9	45.1
Qwen2.5-VL-7B + SFT	36.6	44.4
Qwen2.5-VL-7B + Vanilla GRPO	29.7	51.9
SpatialThinker-7B (Ours)	37.7	52.6

Table 8: Results on abstract reasoning benchmarks. Lego Puzzles measure compositional reasoning over object arrangements, while BLINK Multi-View requires integrating multi-view spatial cues.

Across both tasks, SPATIALTHINKER-7B achieves the highest open-source performance improving over generalist and spatial MLLMs, and scoring 37.7% on Lego Puzzles and 52.6% on BLINK Multi-View, closely approaching GPT-4o and surpassing Claude 3.5 Sonnet on the latter. Interestingly, we observe that vanilla GRPO provides competitive performance on BLINK Multi-View but underperforms on Lego Puzzles, suggesting that dense spatial rewards offer complementary signals that better support compositional reasoning. These results demonstrate that the spatial grounding learned through reinforcement learning transfers to more abstract domains that require compositional and multi-view integration skills.

I Detailed Results: CV-Bench

Model	**CV-Bench Tasks**				**CV-Bench**		**Avg.**
	Count	Relation	Depth	Distance	2D	3D	
Proprietary Models							
GPT-4o [39]	65.9	85.7	87.8	78.2	75.8	83.0	79.4
Gemini-1.5-Pro [28]	70.4	85.2	82.4	72.8	77.8	77.6	77.7
Claude 3.7 Sonnet [4]	-	74.2	85.8	84.2	-	85.0	-
Open-Source General MLLMs							
Qwen2-VL-2B [79]	54.7	22.6	16.7	31.7	38.7	24.2	31.5
Qwen2.5-VL-3B [7]	61.5	58.3	67.3	53.0	59.9	60.2	60.1
Qwen2.5-VL-7B [7]	55.9	82.2	70.0	66.0	69.1	68.0	68.6
VLAA-Thinker-3B [12]	61.6	83.5	53.0	46.8	72.6	49.9	61.3
VLAA-Thinker-7B [12]	47.0	74.6	61.3	59.2	60.8	60.3	60.6
LLaVA-NeXT-34B [46]	-	-	-	-	73.0	74.8	73.9
Mini-Gemini-HD-34B [49]	-	-	-	-	71.5	79.2	75.4
Cambrian-1-34B [74]	-	-	-	-	74.0	79.7	76.9
Open-Source Spatial MLLMs							
Spatial-LLaVA-7B [82]	-	-	57.3	52.2	-	54.8	-
VisualThinker-R1-2B [98]	59.6	66.8	54.2	56.7	63.2	55.45	59.3
Spatial-RGPT-7B w/ depth [17]	-	-	62.3	59.0	-	60.7	-
RoboPoint-13B [93]	-	75.6	77.8	44.5	-	61.15	-
SpaceThinker-3B [3]	61.0	69.2	70.5	61.3	65.1	65.9	65.5
SpaceLLaVA-13B [1]	-	63.7	66.8	70.2	-	68.5	-
SpatialBot-3B [8]	-	69.4	77.3	60.8	-	69.05	-
Method Comparison (Trained on STVQA-7K)							
Qwen2.5-VL-3B + SFT	30.2	77.5	61.2	75.5	53.9	68.4	61.2
Qwen2.5-VL-3B + Vanilla GRPO	67.5	73.7	64.0	69.2	70.6	66.6	68.6
SpatialThinker-3B (Ours)	68.5	73.5	79.7	72.8	71.0	76.3	73.7
Qwen2.5-VL-7B + SFT	33.3	78.9	64.8	77.7	56.1	71.3	63.7
Qwen2.5-VL-7B + Vanilla GRPO	58.9	78.8	79.3	73.7	68.9	76.5	72.7
SpatialThinker-7B (Ours)	68.7	86.7	81.2	76.2	77.7	78.7	78.2

Table 9: Detailed breakdown of CV-Bench [74] results across Count, Relation, Depth, and Distance subtasks.

J Detailed Results: 3DSRBench

Model	**3DSRBench Tasks**				**Avg.**
	Height	Location	Orientation	Multi-Object	
Proprietary Models					
GPT-4o [39]	53.2	59.6	21.6	39.0	44.3
Claude 3.5 Sonnet [5]	53.5	63.1	31.4	41.3	48.2
Gemini 2.0 Flash [29]	49.7	68.9	32.2	41.5	49.9
Gemini 2.0 Flash (thinking) [29]	53.0	67.1	35.8	43.6	51.1
Open-Source MLLMs					
Qwen2.5-VL-3B [7]	45.2	56.8	35.7	35.7	44.0
Qwen2.5-VL-7B [7]	44.1	62.7	40.6	40.5	48.4
Qwen2.5-VL-72B [7]	53.3	71.0	43.1	46.6	54.9
Cambrian-1-8B [74]	23.2	53.9	35.9	41.9	42.2
LLaVA-NeXT-8B [46]	50.6	59.9	36.1	43.4	48.4
VLAA-Thinker-7B [12]	54.0	60.2	42.9	49.1	52.2
Open-Source Spatial MLLMs					
SpatialBot-3B [8]	40.4	54.4	31.9	33.5	41.1
SpaceLLaVA-13B [1]	49.3	54.4	27.6	35.4	42.0
SpatialLLM-8B [57]	45.8	61.6	30.0	36.7	44.9
SpatialRGPT-7B w/ depth [17]	55.9	60.0	34.2	42.3	48.4
SpaceThinker-3B [3]	53.1	57.3	41.9	49.6	51.1
Method Comparison (Trained on STVQA-7K)					
Qwen2.5-VL-3B + SFT	51.1	58.3	42.7	48.1	50.8
Qwen2.5-VL-3B + Vanilla GRPO	48.9	57.9	42.5	47.2	50.1
SpatialThinker-3B (Ours)	52.6	61.8	43.4	49.8	52.9
Qwen2.5-VL-7B + SFT	50.6	66.3	43.8	47.9	53.6
Qwen2.5-VL-7B + Vanilla GRPO	54.3	64.7	45.5	50.4	54.7
SpatialThinker-7B (Ours)	52.0	70.3	45.5	50.9	56.4

Table 10: Detailed Breakdown of 3DSRBench [56] Height, Location, Orientation, and Multi-Object tasks.