Exploring the Interplay Between Explicit Grounding and Reasoning in Visual TableQA

Anonymous ACL submission

Abstract

This paper investigates how explicit visual grounding within Chain-of-Thought (CoT) sequences impacts the reasoning proficiency of multimodal large language models (MLLMs). Using visual TableQA as a testbed, we examine this interplay through supervised fine-tuning (SFT) and reinforcement learning (RL) with a hierarchical grounding reward. Our analysis reveals an observable performance trade-off, where the requirement for rigid spatial syntax appears to interfere with the model's internal reasoning heuristics. These insights suggest that aligning precise spatial anchoring with logical inference poses substantial challenges for current training regimes, highlighting the need for more sophisticated data synthesis and optimization strategies in complex multimodal tasks.

1 Introduction and Related Works

Multimodal large language models (MLLMs) (Liu et al., 2023; Bai et al., 2025) have demonstrated remarkable potential in complex visual reasoning tasks, such as document understanding (Li et al., 2024; Ye et al., 2023) and GUI navigation (Hong et al., 2023; Wang et al., 2024; You et al., 2024). These tasks involve reasoning over structured visual environments, where logical consistency is closely coupled with the underlying spatial layout and visual semantics (e.g., cell content or functional icons). However, standard Chain-of-Thought (CoT) prompting often exhibits a misalignment between textual inference and visual evidence. In such cases, the model may generate a logically coherent narrative that diverges from the actual visual input, leading to severe hallucinations (Li et al., 2025; Thawakar et al., 2025; Huang et al., 2025).

To bridge this gap, various grounding strategies have been explored, ranging from semantic-level reinforcement via textual descriptions (Ghosh et al., 2025) to structural anchoring through mechanisms such as dynamic regional cropping (Zhang et al., 2025; Qiu et al., 2025) and interleaved spatial representations. In the latter, visual indicators such as latent feature tokens (Wang et al., 2025b; You et al., 2024) or symbolic coordinates (Shao et al., 2024a) are woven directly into the textual sequence to anchor the reasoning process. One noteworthy direction is explicit visual grounding (Shao et al., 2024b; Wang et al., 2025a), which associates reasoning steps with spatial coordinates. This architecture-agnostic approach is theoretically appealing as it leverages the latent localization capabilities acquired during pre-training (Qiu et al., 2025) without requiring structural modifications.

However, a critical question remains unexplored: *Does enforcing rigid spatial syntax inadvertently interfere with the model's internal reasoning heuristics?* In this work, we use visual TableQA task as a representative testbed to investigate this interplay. Our results demonstrate that SFT for explicit grounding effectively enforces the required output syntax but simultaneously induces a significant *alignment tax* on reasoning performance. Although a hierarchical RL reward facilitates better structural adherence, its capacity to mitigate this degradation remains limited. Even with refinement during the RL stage, larger models struggle to restore their baseline reasoning proficiency, indicating a fundamental tension between rigid formatting constraints and logical inference.

We summarize our contributions as follows:

- Data Synthesis: We develop a pipeline to transform Text2SQL execution traces into a grounded CoT dataset with hierarchical bounding box annotations, filling the gap in fine-grained visual-tabular reasoning data.
- Methodology: We design a hierarchical grounding reward that provides dense supervision, facilitating spatial alignment through a fine-grained RL stage.

- Empirical Insights: We characterize the alignment tax of explicit grounding during SFT, revealing how rigid formatting disrupts reasoning heuristics and identifying the limitations of RL as a corrective mechanism.

2 Methodology

2.1 Grounded TableQA Dataset Construction

To evaluate how visual grounding affect reasoning, we first develop a grounded TableQA dataset for SFT, interleaving CoT reasoning with precise spatial anchors. We choose TableQA as a representative testbed because reasoning over structured tables requires both precise spatial perception and multi-step logical inference. To ensure data diversity, we also incorporate general visual question answering (VQA) data from prior work. The construction follows a two-stage pipeline: grounded data processing and reasoning data generation.

Rule-based Bounding Box Generation. Step-wise reasoning data is inherently complex, especially in multimodal settings where textual steps must synchronize with visual cues. Given the high cost and quality-control challenge of manual annotation, we adopt a programmatic approach by leveraging the Text2SQL datasets. A Text2SQL task (Zhong et al., 2017) involves mapping natural language questions to executable SQL queries. Crucially, SQL query execution mirrors human-like reasoning processes, with its sequential selection and filtering operations providing a logical backbone that maps directly to specific spatial regions. This allows us to transform SQL execution traces into step-wise reasoning data, where bounding boxes represent the visual context of each operation. To formalize this mapping, the transformation process leverages the hierarchical structure of SQL execution. First, each table is rendered as an image while simultaneously capturing the pixel coordinates of all row and column boundaries. Then, the procedure follows a top-down refinement strategy: first, we identify all cells in columns specified by both the `SELECT` and `WHERE` clauses, merging adjacent cells into unified bounding boxes to form the contextual set $\mathcal{B}_{\text{ctx}}$ as Contextual Regions. Second, we filter rows from $\mathcal{B}_{\text{ctx}}$ satisfying the `WHERE` predicates to produce the subset $\mathcal{B}_{\text{filt}}$ as Filtered Regions. Finally, we extract the specific cells contributing to the answer to form $\mathcal{B}_{\text{core}}$ as Core Regions. This hierarchical derivation ensures visual regions are narrowed down progressively, maintaining the property $\mathcal{B}_{\text{core}} \subseteq \mathcal{B}_{\text{filt}} \subseteq \mathcal{B}_{\text{ctx}}$.

Reasoning Data Generation. Based on these generated ground-truth bounding box sets, we use GPT-o3 (OpenAI, 2025) to synthesize corresponding step-wise reasoning texts. For each table-question pair, the model is provided with the ordered grounding sets $\{\mathcal{B}_{\text{ctx}}, \mathcal{B}_{\text{filt}}, \mathcal{B}_{\text{core}}\}$ augmented with localized cell texts and prompted to synthesize natural language rationales describing the transitions between these stages. More details can be found in subsection D.1 with a visualized example. To ensure quality, generated texts are evaluated for consistency with SQL semantics and subjected to human verification. The final dataset, sampled from WikiSQL (Zhong et al., 2017), comprises 2,558 high-quality samples aligning textual inference with precise visual grounding. Additionally, we retain a distinct set of 4,851 instances containing only the grounding sequences as supplementary training data for subsequent RL stage.

2.2 Hierarchical Reward Design

Following the SFT phase, we incorporate a RL stage to further refine the alignment between the model's reasoning trajectories and the table's visual structure. This stage is designed to ensure the model internalizes the logic of visual reasoning beyond simple pattern matching. We define the total reward as

$$R_{\text{final}} = w_{\text{acc}} R_{\text{acc}} + w_{\text{bbox}} R_{\text{bbox}} + w_{\text{fmt}} R_{\text{fmt}},$$

where answer reward R_{acc} measures Exact Match (EM) accuracy and format reward R_{fmt} enforces adherence to the `<think>` and `<answer>` schema.

In early experiments, we observed that using only the direct answer bounding boxes $\mathcal{B}_{\text{core}}$ as a reference for bounding box reward R_{bbox} led to very weak intermediate signals. When predicted bounding boxes were far from the answer bounding boxes, the learning signal was almost nonexistent, often leading to catastrophic entropy collapse. To address this, we propose a hierarchical bounding box reward providing more dense supervision. For the core regions, the reward R_{core} combines coverage and geometric alignment:

$$R_{\text{core}} = \frac{\lambda \cdot \text{Hit}(\mathcal{B}_{\text{p}}, \mathcal{B}_{\text{core}}) + \text{GIoU}(\mathcal{B}_{\text{p}}, \mathcal{B}_{\text{core}})}{\lambda + 1},$$

where $\mathcal{B}_{\text{p}}$ denotes predicted boxes and $\lambda = 1.0$. The total grounding reward R_{bbox} is formulated by

Table 1: The overall evaluation of both SFT and RL stages. RL results are color-coded based on the delta relative to SFT: green for improvement, red for decline. **Best** and second-best results are highlighted independently for each stage within model size groups. All performance metrics are reported using greedy decoding.

Size	SFT Dataset	SFT Scale	Rollout	FinTabNetQA		VTabFact		VWTQ		VWTQ-syn		**All**	
				SFT	SFT+RL	SFT	SFT+RL	SFT	SFT+RL	SFT	SFT+RL	SFT	SFT+RL
3B	/	-	Direct	55.3	55.3	60.0	60.0	20.3	20.3	28.0	28.0	35.2	35.2
		-	CoT	52.9	52.9	**68.4**	**68.4**	**24.5**	24.5	31.9	31.9	**38.9**	38.9
	WikiSQL	500	CoT	57.0	**63.1**	57.2	54.0	24.4	25.0	31.4	29.3	37.6	37.6
		1000		**62.3**	62.5	57.2	58.8	23.9	**26.7**	27.5	**34.9**	37.0	**40.7**
		2000		60.3	54.1	57.6	50.0	22.1	21.1	26.6	27.2	35.9	33.4
	MM-GCoT	500		44.1	55.0	54.8	61.2	22.6	24.8	27.7	30.1	33.2	37.6
		1000		47.1	51.6	56.8	58.8	24.3	22.8	32.7	29.8	36.3	35.9
		2000		45.2	58.6	56.8	63.2	**24.5**	21.4	**34.4**	30.6	36.6	37.6
	Mixed	500+500		61.4	55.2	51.6	16.8	21.2	21.0	29.6	27.2	35.7	28.0
		1000+1000		60.9	55.1	60.8	59.2	21.2	19.8	26.5	32.5	36.2	36.4
		2000+2000		56.1	58.5	58.4	62.0	18.4	20.4	23.9	25.4	33.1	35.3
7B	/	-	Direct	73.2	73.2	74.4	74.4	34.3	34.3	45.1	45.1	51.0	51.0
		-	CoT	**77.7**	**77.7**	**80.0**	**80.0**	**46.0**	**46.0**	**55.2**	**55.2**	**59.9**	**59.9**
	WikiSQL	500	CoT	62.7	75.4	60.0	67.2	41.4	43.4	47.3	51.4	50.0	55.3
		1000		63.6	68.9	63.6	65.6	38.4	40.0	47.0	50.7	49.6	52.5
		2000		64.4	70.2	60.8	66.8	32.2	34.9	40.0	42.2	44.9	48.4
	MM-GCoT	500		64.7	66.2	69.2	72.4	36.7	38.1	48.1	42.5	50.5	49.9
		1000		64.9	69.7	74.0	59.6	38.7	38.5	46.3	48.3	51.4	50.4
		2000		65.2	64.6	69.2	72.4	37.6	34.4	44.8	45.8	49.8	49.5
	Mixed	500+500		61.8	64.9	66.0	66.4	37.2	39.4	42.2	50.6	47.7	51.8
		1000+1000		64.6	69.7	66.0	64.8	37.6	37.8	45.5	47.6	49.4	50.8
		2000+2000		59.5	69.8	64.8	66.8	32.0	28.9	42.2	38.2	45.4	45.1

decomposing hierarchical contributions from the contextual and filtered levels:

$$R_{\text{bbox}} = R_{\text{core}} + \alpha \cdot \text{Hit}(\mathcal{B}_{\text{p}}, \mathcal{B}_{\text{filt}}) + \beta \cdot \text{Hit}(\mathcal{B}_{\text{p}}, \mathcal{B}_{\text{ctx}}),$$

where $\alpha = 0.7$ and $\beta = 0.3$ are decay coefficients. These values are chosen to prioritize localized precision while maintaining a smooth gradient from broader contextual regions. More details can be found in Appendix B.

To balance functional correctness with structural grounding, we set the task weights to $w_{\text{acc}} = 1.0$, $w_{\text{bbox}} = 0.25$, and $w_{\text{fmt}} = 0.1$, ensuring that while EM remains the primary objective, the hierarchical bounding box reward provides the necessary structural dense supervision.

3 Experiments

In this section, we present a comprehensive evaluation of the interplay between explicit spatial grounding and logical reasoning. Our primary results are summarized in Table 1.

3.1 Experimental Setup

3.1.1 Involved LLMs

To evaluate the impact of grounding on reasoning across different parameter scales, we employ two popular MLLMs: **Qwen2.5-VL-3B** and **Qwen2.5-VL-7B** (Bai et al., 2025). This series of models utilizes a vision-language architecture capable of processing dynamic resolution inputs, and both have been specifically trained on visual grounding tasks. Implementation details can be found in subsection D.2.

3.1.2 Datasets

We utilize MM-GCoT (Wu et al., 2025) and processed WikiSQL (Zhong et al., 2017) as training data to establish foundational relational reasoning and grounded multimodal capabilities. In the SFT stage, we evaluated three distinct data compositions including pure MM-GCoT, pure WikiSQL, and a balanced 1:1 mixture to observe their respective effects on performance. During the RL stage, we exclusively utilized the processed WikiSQL dataset to focus on refining logical-spatial alignment for reasoning abilities.

For evaluation, we employ TableVQA-Bench (Kim et al., 2024) to assess cross-domain table understanding. Specifically, FinTabNetQA and VTabFact focus more on conditional cell locating and precise data retrieval. In contrast, VWTQ and its synthesized variants (VWTQ-syn) represent the more challenging tasks, requiring complex counting and multi-step reasoning beyond simple retrieval. More details can be found in Appendix A.

3.2 Evaluation on SFT and RL

Based on the results in Table 1, we articulate two key findings regarding the trade-off between explicit grounding format and reasoning performance, and the role of RL in mitigating this trade-off.

> **Finding 1:** SFT for explicit grounding functions as a structural prerequisite that enables structured output at the cost of a substantial alignment tax.

Our first observation is that SFT for explicit grounding incurs a substantial alignment tax, particularly as model scale increases. Since few-shot prompting fails to reliably elicit structured, grounded CoT in either 3B or 7B models, SFT emerges as a necessary foundation to enforce the required output format. However, this structural adaptation generally degrades reasoning performance relative to original models.

Crucially, we observe a phenomenon of negative scaling with respect to data volume, yet its severity is highly dependent on data composition. For the 7B model, increasing task-specific WikiSQL data from 500 to 2,000 samples triggers a sharp performance decline from 50.0% to 44.9%, suggesting that rigid, domain-constrained formats rapidly override the model's logical integrity. In contrast, general-domain data, namely MM-GCoT, exhibits significantly higher resilience to this degradation; its performance remains relatively stable (around 50%–51%) even as the scale increases. This suggests that the semantic diversity of general VQA data acts as a functional buffer, slowing the onset of overfitting to grounded syntax and better preserving the model's inherent reasoning heuristics. Despite these mitigations, however, neither dataset allows the SFT variants to restore the baseline reasoning performance of the original models, confirming that rigid spatial anchoring imposes an inherent constraint on cognitive flexibility.

> **Finding 2:** RL acts as a partial corrective for the SFT alignment tax, yet its ceiling remains intrinsically tied to the SFT initialization.

Our second finding highlights both the corrective power and the inherent limitations of the RL stage. As illustrated by the predominantly green cells in Table 1, adding RL generally improves performance over SFT-only baselines, acting as a recovery mechanism for the reasoning logic suppressed during the SFT stage. For instance, on the 3B model trained with WikiSQL-1000, RL boosts the average score from 37.0% to 40.7%, notably surpassing the base performance (38.9%).

However, this effectiveness is not an autonomous capability because the RL phase depends on the pre-established structural format provided during SFT. Without an initial grounding mechanism, the RL process cannot forcibly optimize spatial localization from scratch. But once this structural foundation exists, RL significantly enhances grounding quality across different model scales as detailed in Appendix C. Interestingly, RL gains are inversely proportional to SFT data resilience. RL yields the most dramatic improvements on models fine-tuned on task-specific data (e.g., WikiSQL), where SFT-induced degradation was most severe. In contrast, on general-domain data like MM-GCoT, where SFT preserved reasoning integrity, RL's marginal improvements are diminished.

Finally, we observe a persistent ceiling effect in the 7B model. While RL successfully optimizes grounding accuracy, it fails to fully restore the model's baseline reasoning performance. This gap is particularly significant given that larger models are expected to better balance new constraints. However, our results suggest that current scaling laws may not naturally resolve the inherent friction between formatting and logic. This also indicates that the structural distortion from rigid grounding syntax persists through late-stage optimization. Consequently, post-training refinements cannot compensate for the lack of a fundamental shift in grounding representation to mitigate reasoning loss.

4 Conclusion

This study characterizes the performance trade-offs within the explicit grounding paradigm. Our findings identify a persistent alignment tax where rigid coordinate syntax appears to interfere with reasoning heuristics. Although grounding-enhanced RL improves formatting adherence, it only partially recovers the reasoning proficiency lost during alignment. Our work highlights the challenge of integrating spatial precision with logical reasoning, suggesting that future research might further investigate the underlying mechanisms of this trade-off to ultimately enhance the visual reasoning capabilities of MLLMs.

5 Limitations

Despite the insights provided by our study, several limitations remain to be addressed in future work.

First, our empirical investigation is primarily centered on the Qwen architecture (Bai et al., 2025). While Qwen represents a popular MLLM with robust visual-language capabilities, its internal attention mechanisms and pre-training priors may inherently influence how it handles spatial syntax and reasoning heuristics.

Second, due to computational constraints, our experiments did not encompass the most expansive parameter scales (e.g., models exceeding 70B parameters) or the use of full-parameter fine-tuning. We primarily focused on moderate-sized models and parameter-efficient strategies. It remains possible that larger-scale models possess greater residual capacity to absorb the cognitive load imposed by rigid formatting, potentially alleviating the reasoning degradation we observed. Future research is needed to verify whether scaling up model size or adopting more intensive optimization regimes can further bridge the gap between explicit grounding and logical proficiency.

6 Ethical Considerations

Generative AI was utilized solely to refine the manuscript's linguistic clarity and readability, with the authors maintaining full responsibility for all original intellectual content.

Regarding data processing, the curation and filtering of LLM-generated outputs were performed exclusively by the authors under ethical labor practices. Furthermore, all data were sourced from publicly available datasets that contain no personally identifiable information or privacy risks. While the research team is primarily of Asian background, the potential for cultural or demographic bias is significantly minimized as the filtering focused strictly on logical reasoning and structural accuracy within the visual TableQA task, which is relatively objective and culturally neutral.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

DeepSeek-AI. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Utkarsh Tyagi, Oriol Nieto, Zeyu Jin, and Dinesh Manocha. 2025. Visual description grounding reduces hallucinations and boosts reasoning in lvlms. *Preprint*, arXiv:2405.15683.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and Jie Tang. 2023. Cogagent: A visual language model for gui agents. *Preprint*, arXiv:2312.08914.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *Preprint*, arXiv:2503.24290.

Mingxin Huang, Yongxin Shi, Dezhi Peng, Songxuan Lai, Zecheng Xie, and Lianwen Jin. 2025. Ocr-reasoning benchmark: Unveiling the true capabilities of mllms in complex text-rich image reasoning. *Preprint*, arXiv:2505.17163.

Yoonsik Kim, Moonbin Yim, and Ka Yeon Song. 2024. Tablevqa-bench: A visual question answering benchmark on multiple table domains.

Xiaoyuan Li, Moxin Li, Wenjie Wang, Rui Men, Yichang Zhang, Fuli Feng, and Dayiheng Liu. 2025. Mathopeval: A fine-grained evaluation benchmark for visual operations of mllms in mathematical reasoning. *Preprint*, arXiv:2507.18140.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. 2024. Monkey: Image resolution and text label are important things for large multi-modal models. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26763–26773.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Preprint*, arXiv:2304.08485.

OpenAI. 2025. Openai o3 and o4-mini system card.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Han Qiu, Peng Gao, Lewei Lu, Xiaoqin Zhang, Ling Shao, and Shijian Lu. 2025. Spatial preference rewarding for mllms spatial understanding. *Preprint*, arXiv:2510.14374.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. *Preprint*, arXiv:1902.09630.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024a. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Preprint*, arXiv:2403.16999.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024b. Visual cot: advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, Hisham Cholakkal, Ivan Laptev, Mubarak Shah, Fahad Shahbaz Khan, and Salman Khan. 2025. Llamav-o1: Rethinking step-by-step visual reasoning in llms. *Preprint*, arXiv:2501.06186.

Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Jiani Zheng, Sule Bai, Zijian Kang, Jiashi Feng, Zhuochen Wang, and Zhaoxiang Zhang. 2025a. Traceable evidence enhanced visual grounded reasoning: Evaluation and methodology. *Preprint*, arXiv:2507.07999.

Jiacong Wang, Zijian Kang, Haochen Wang, Haiyong Jiang, Jiawen Li, Bohong Wu, Ya Wang, Jiao Ran, Xiao Liang, Chao Feng, and Jun Xiao. 2025b. Vgr: Visual grounded reasoning. *Preprint*, arXiv:2506.11991.

Junyang Wang, Haiyang Xu, Jiabo Ye, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. 2024. Mobile-agent: Autonomous multi-modal mobile device agent with visual perception. *Preprint*, arXiv:2401.16158.

Qiong Wu, Xiangcong Yang, Yiyi Zhou, Chenxin Fang, Baiyang Song, Xiaoshuai Sun, and Rongrong Ji. 2025. Grounded chain-of-thought for multimodal large language models. *Preprint*, arXiv:2503.12799.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023. mplug-docowl: Modularized multimodal large language model for document understanding. *Preprint*, arXiv:2307.02499.

Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. 2024. Ferret-ui: Grounded mobile ui understanding with multimodal llms. *Preprint*, arXiv:2404.05719.

Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. 2025. Mllms know where to look: Training-free perception of small visual details with multimodal llms. *arXiv preprint arXiv:2502.17422*.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. 2024. Swift:a scalable lightweight infrastructure for fine-tuning. *Preprint*, arXiv:2408.05517.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103.

A Dataset Details

In this section, we introduce the details of the utilized datasets in this study.

Training Datasets. To establish a robust foundation for table understanding, we utilize two datasets for supervised fine-tuning:

- **WikiSQL** (Zhong et al., 2017): A large-scale Text-to-SQL dataset containing hand-annotated SQL queries and QA pairs distributed across tables extracted from Wikipedia. We leverage this dataset to train the model in fundamental relational reasoning and table structure parsing. Following the procedures in subsection 2.1, we convert the original relational tables into high-resolution images and transform the SQL-based logic into natural language questions, ensuring the model learns to map linguistic queries to specific structural headers and cell values.

- **MM-GCoT** (Wu et al., 2025): A versatile multimodal CoT dataset that spans diverse, general-purpose VQA scenarios. It interleaves reasoning paths with fine-grained object-level grounding, such as bounding boxes and point-level references. By incorporating this broad-spectrum data, the model establishes a foundational capability for generating intermediate rationales. This general reasoning framework is essential for the model to later adapt and decompose complex visual evidence within the more specialized environments of multi-row and multi-column tables.

Evaluation Benchmark. Our primary evaluation suite, TableVQA-Bench (Kim et al., 2024), comprises three specialized subsets to assess cross-domain generalization:

- **FinTabNetQA**: Sourced from the annual reports of S&P 500 companies, this subset evaluates expert-level financial extraction. It requires precise cell locating and scale-unit handling (e.g., million, percentage) within dense, professional layouts.
- **VTabFact**: A fact-checking task adapted into a True/False binary format. It assesses the model's ability to verify claims by retrieving specific evidence and maintaining logical consistency against table content.
- **VWTQ & VWTQ-syn**: These subsets represent the higher cognitive demand, requiring multi-step reasoning such as counting and aggregation. **VWTQ** is constructed by incorporating an image collection into the WTQ (Pasupat and Liang, 2015), while **VWTQ-syn** employs randomized styles via a custom rendering system to test the model's visual robustness and generalization.

B Algorithm and Metric Details

This section provides formal definitions for the RL algorithm and the spatial metrics used within our reward function.

B.1 Group Relative Policy Optimization

To optimize the model, we employ Group Relative Policy Optimization (GRPO) (DeepSeek-AI, 2025), a variant of PPO that eliminates the need for a separate value function by using group-based relative rewards. For each input q, we sample a group of G outputs $\{o_1, o_2, \ldots, o_G\}$ from the old policy $\pi_{\theta_{old}}$. The objective function is:

$$\mathcal{J}_{GRPO}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\Big[\min\Big(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}\hat{A}_i, \text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1-\epsilon, 1+\epsilon\right)\hat{A}_i\Big) - \beta\mathbb{D}_{KL}(\pi_\theta||\pi_{ref})\Big].$$

The advantage $\hat{A}_i$ for each output o_i is computed by normalizing the total reward R_{final} within the group: $\hat{A}_i = \frac{R_i - \text{mean}(R)}{\text{std}(R)}$.

B.2 Reward Components: Hit and GIoU

While the final model performance is evaluated using Exact Match (EM), the reinforcement learning stage utilizes spatial grounding signals to provide dense supervision.

Hit Ratio. The hit ratio measures the spatial coverage between the predicted box set $\mathcal{B}_{\text{p}}$ and the ground-truth set $\mathcal{B}_{\text{gt}}$. We implement two modes to provide different directional supervision:

- **Strict Hit** (for $\mathcal{B}_{\text{core}}$): Measures the recall of target cells, ensuring all required regions are identified. $\mathcal{S} = \mathcal{B}_{\text{gt}}, \mathcal{T} = \mathcal{B}_{\text{p}}$.
- **Loose Hit** (for $\mathcal{B}_{\text{filt}}$ and $\mathcal{B}_{\text{ctx}}$): Measures the precision of predictions, encouraging anchors to fall within valid regions. $\mathcal{S} = \mathcal{B}_{\text{p}}, \mathcal{T} = \mathcal{B}_{\text{gt}}$.

The formula is defined as:

$$\text{Hit}(\mathcal{B}_{\text{p}}, \mathcal{B}_{\text{gt}}) = \frac{1}{|\mathcal{S}|}\sum_{b\in\mathcal{S}}\mathbb{I}\left(\exists\hat{b}\in\mathcal{T}, \text{Center}(\hat{b})\in b\right),$$

where $\text{Center}(\hat{b})$ is the midpoint of box $\hat{b}$, and $\mathbb{I}(\cdot)$ is the indicator function.

Generalized Intersection over Union. To provide non-zero gradients even when boxes do not overlap, we incorporate GIoU (Rezatofighi et al., 2019). For a predicted box A and a ground-truth box B, let C be the smallest convex hull enclosing both:

$$\text{GIoU}(A, B) = \frac{|A\cap B|}{|A\cup B|} - \frac{|C\setminus(A\cup B)|}{|C|}.$$

Since the model may predict multiple boxes, we aggregate the signal by calculating the average maximum alignment for each target cell:

$$R_{\text{GIoU}}(\mathcal{B}_{\text{p}}, \mathcal{B}_{\text{core}}) = \frac{1}{|\mathcal{B}_{\text{core}}|}\sum_{b\in\mathcal{B}_{\text{core}}}\Phi(b, \mathcal{B}_{\text{p}}),$$

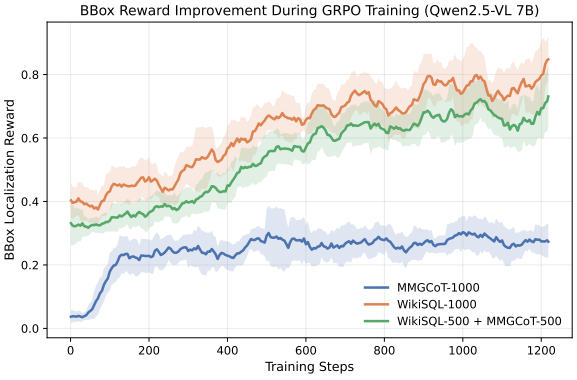
where $\Phi(b, \mathcal{B}_{\text{p}}) = \max_{\hat{b}\in\mathcal{B}_{\text{p}}}\text{GIoU}(\hat{b}, b)$ represents the maximum spatial alignment between the target cell b and the set of predicted boxes. Consistent with our implementation, the raw R_{GIoU} ranges from $[-1, 1]$. To balance its magnitude with the Hit rate, we normalize it to a non-negative interval $[0, 1]$ during the final reward summation:

$$R'_{\text{GIoU}} = \frac{R_{\text{GIoU}}(\mathcal{B}_{\text{p}}, \mathcal{B}_{\text{core}}) + 1}{2}.$$

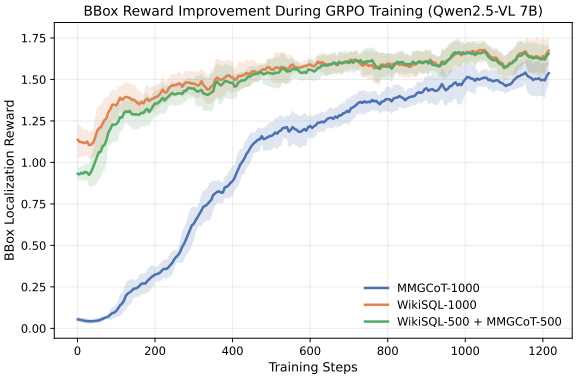
This normalized term provides a continuous penalty that guides the model to shift its focus toward the target even in the absence of initial spatial overlap.

C Evolution of Grounding Capabilities during RL

To further investigate the evolution of spatial understanding during the RL phase, we analyze the bounding box reward across different model scales and SFT initializations. As illustrated in Figure 1, both the Qwen2.5-VL 3B and 7B models demonstrate a consistent upward trend in reward values over the training trajectory. This steady improvement, observed across varying SFT data mixtures, confirms that the RL stage effectively refines grounding performance beyond the initial supervised tuning. Notably, the 7B model reaches a higher reward plateau compared to the 3B model's peak, yet both scales exhibit the same characteristic learning curve. These results validate that the reinforcement learning process successfully optimizes coordinate precision and spatial alignment by leveraging group-based relative rewards, demonstrating its robustness and efficacy in enhancing the model's fundamental grounding capabilities.



(a) Qwen2.5-VL 3B



(b) Qwen2.5-VL 7B

Figure 1: Evolution of the bounding box reward during GRPO training for 3B and 7B models across different SFT initializations. Curves are smoothed using a moving average with window size 10 for visualization, and shaded areas indicate the range of fluctuations in the original (unsmoothed) reward.

D Prompts and Implementation Details

This section details the specific prompts and detailed experimental settings employed in our methodology.

D.1 Prompts and Data Processing

Prompt 1 illustrates the prompt designed for GPT-o3 (OpenAI, 2025) to synthesize the grounded CoT dataset, where we enforce a structured JSON format (see Prompt 2 and Prompt 3) to ensure reasoning steps are strictly aligned with the hierarchical grounding sets.

The synthesized structured data further undergoes a post-processing stage to adapt it into the final CoT training format. During this process, auxiliary hierarchical metadata and redundant object labels are removed to streamline the reasoning trajectory. Crucially, since the initial grounding signals are generated based on logical table indices (i.e., row and column numbers), we perform a coordinate transformation to map these indices into absolute pixel coordinates. To ensure strict spatial consistency, we utilize the official `smart_resize` function provided by the `qwen-vl-utils` library[1], which is the standard utility for the Qwen-VL model series. This function allows us to simultaneously rescale the original images and

[1] https://github.com/QwenLM/Qwen3-VL/tree/main/qwen_vl_utils

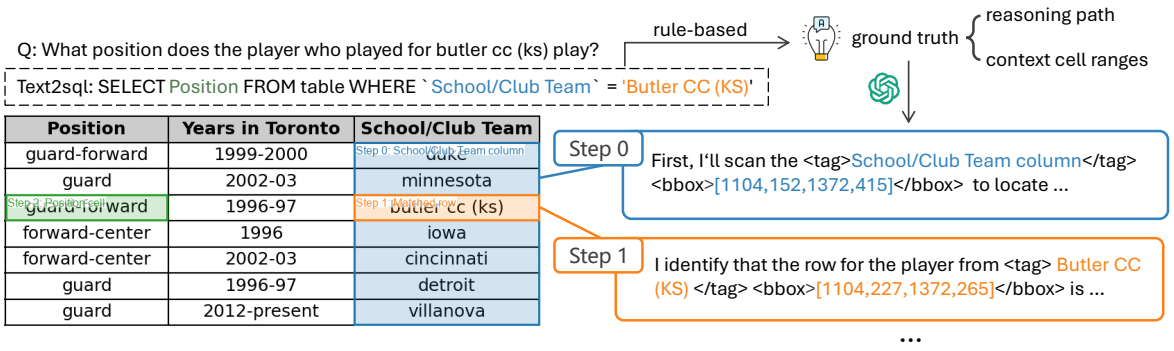


Figure 2: An example of grounded CoT data generation process.

their corresponding bounding box coordinates while maintaining the alignment required by the model's vision-language adapter. This synchronization ensures that the processed visual features and the re-scaled bounding box tokens (e.g., <|box_start|>(y1,x1),(y2,x2)<|box_end|>) remain perfectly aligned, effectively eliminating spatial drift during the RL phase.

During the SFT and RL phases, the model is presented with a multimodal input formatted as <image>{question}. The system prompt used during training and the rollout process is detailed in Prompt 4. This prompt instructs the model to generate its reasoning trajectory within <think> tags—incorporating precise bounding box coordinates as visual anchors—and to provide the final extraction within <answer> tags. A visualized example can be found in Figure 2.

```
You are a helpful assistant specialized in
    table-based reasoning.
Your task is to generate step-by-step
    Chain-of-Thought reasoning steps for the
    user question.

### Requirements:

- Return an object with a key `reasoning_steps`
    which is a list of step objects.
- Each step object must have:
   - `step_number`: integer starting from 1
   - `reasoning_description`: a string
       describing the step, **including
       `<grounding_object> ...
       </grounding_object>` tags to explicitly
       mark the grounding objects** mentioned
       in this step.
   - `groundings`: a list (possibly empty) of
       grounding objects, each having:
     - `grounding_object`: a string, matching
         exactly the text inside the
         `<grounding_object> ...
         </grounding_object>` tags in the
         description.
     - `range`: an object with `rows` and
         `cols` arrays indicating the
         selected table indices.

### Example output:

{get_example()}

### Input:

User question:
{question}

Table:
{table}

Precomputed ranges:
- Select range: {select_range}
- Where range: {where_range}
- Ground truth answer {f"with aggregation
    term '{agg_term}'" if agg_term else
    ""}: {gt_result}
```

Prompt 1: User prompt for grounded CoT data generation.

```
{
"type": "object",
"additionalProperties": False,
"properties": {
   "reasoning_steps": {
   "type": "array",
   "minItems": 1,
   "items": {
      "type": "object",
      "additionalProperties": False,
      "required": ["step_number",
          "reasoning_description",
          "groundings"],
      "properties": {
      "step_number": {
         "type": "integer",
         "minimum": 1
      },
      "reasoning_description": {
         "type": "string",
         "pattern": ".*<grounding_object>.*
             </grounding_object>.*"
      },
      "groundings": {
```

```
          "type": "array",
          "minItems": 0,
          "items": {
          "type": "object",
          "additionalProperties": False,
          "required": ["grounding_object",
              "range"],
          "properties": {
              "grounding_object": { "type":
                  "string" },
              "range": {
              "type": "object",
              "additionalProperties": False,
              "required": ["rows", "cols"],
              "properties": {
                  "rows": {
                  "type": "array",
                  "items": { "type": "integer" }
                  },
                  "cols": {
                  "type": "array",
                  "items": { "type": "integer" }
                  }
                }
              }
            }
          }
        }
      }
    }
  },
"required": ["reasoning_steps"]
}
```

Prompt 2: Json schema for grounded CoT data generation.

```
{
  "reasoning_steps": [
    {
      "step_number": 1,
      "reasoning_description": "Select
          <grounding_object>Player
          column</grounding_object> to identify
          all players.",
      "groundings": [
        {
          "grounding_object": "Player column",
          "range": {
            "rows": [
              0,
              1,
              2,
              3,
              4,
              5,
              6
            ],
            "cols": [
              0
            ]
          }
        }
      ]
    },
    {
      "step_number": 2,
      "reasoning_description": "Filter
          <grounding_object>rows with Position
          = Guard</grounding_object> to narrow
          candidates to players in rows 1, 5,
          6.",
      "groundings": [
        {
          "grounding_object": "rows with
              Position = Guard",
          "range": {
            "rows": [
              1,
              5,
              6
            ],
            "cols": [
              2
            ]
          }
        }
      ]
    },
    {
      "step_number": 3,
      "reasoning_description": "Select
          <grounding_object>No.
          column</grounding_object> for the
          filtered players in rows 1, 5, 6.",
      "groundings": [
        {
          "grounding_object": "No. column",
          "range": {
            "rows": [
              1,
              5,
              6
            ],
            "cols": [
              1
            ]
          }
        }
      ]
    },
    {
      "step_number": 4,
      "reasoning_description": "Identify
          <grounding_object>lowest
          number</grounding_object> among
          selected players: 2 (row 1), 25 (row
          5), 3 (row 6). The lowest number is
          2.",
      "groundings": [
        {
          "grounding_object": "lowest number",
          "range": {
            "rows": [
              1
            ],
            "cols": [
              1
            ]
          }
        }
      ]
    },
    {
      "step_number": 5,
      "reasoning_description": "Final answer is
          <grounding_object>Voshon
```

```
          Lenard</grounding_object> with number
          2 in row 1.",
      "groundings": [
        {
          "grounding_object": "Voshon Lenard",
          "range": {
            "rows": [
              1
            ],
            "cols": [
              0
            ]
          }
        }
      ]
    }
  ]
}
```

Prompt 3: A few-shot example for grounded CoT data generation.

```
You are an expert assistant specializing in
    interpreting tables. For each reasoning
    step, reference the exact visual region
    (bounding box) involved, wrapping your
    thought in <think>example thinking</think>
    tags. Conclude with the result in
    <answer>example answer</answer> tags.
```

Prompt 4: System Prompt for training and rollout process.

D.2 Experimental Configurations

All experiments are implemented using the `ms-swift` framework (Zhao et al., 2024). To maintain computational efficiency, we employ LoRA (Hu et al., 2022) on linear modules with a rank $r = 8$ and $\alpha = 32$. To optimize memory usage and training speed, all parameters are trained using bfloat16 precision.

SFT Stage. In the SFT stage, the model is trained for 1 epoch with a learning rate of 1×10^{-4} and a total effective batch size of 16 (achieved via gradient accumulation). We utilize a cosine learning rate scheduler with a 0.05 warmup ratio. The maximum sequence length is set to $4,096$ tokens to accommodate long execution traces and table contexts.

RL Stage. Starting from the optimal SFT checkpoint, we perform RL using the GRPO algorithm. This stage focuses exclusively on the refined WikiSQL training samples to maximize logical-spatial alignment. We sample $G = 4$ completions per prompt with a temperature of 0.7. The learning rate is reduced to 1×10^{-5} and the KL divergence coefficient β is set to 0.00. Following the default configuration of the Hugging Face `trl` GRPO trainer [2] and recent empirical evidence (Hu et al., 2025) suggesting the KL term is not essential for GRPO-based reasoning tasks, this setting facilitates unconstrained policy exploration within the structured reasoning space. For visual processing, we constrain the image resolution between $3,136$ and $1,003,520$ pixels. The training spans 5 epochs to ensure the stability of the reward convergence.

[2] https://huggingface.co/docs/trl/main/en/grpo_trainer