TRivia: Self-supervised Fine-tuning of Vision-Language Models for Table Recognition

Supplementary Material

A. More Training Details

		Stage-1	Stage-2	Stage-3
Vision	**Max Resolution**	$1280 \times 28 \times 28$	$1280 \times 28 \times 28$	$1280 \times 28 \times 28$
	#Tokens per Image	$256 \sim 1280$	$256 \sim 1280$	$256 \sim 1280$
Data	**Dataset**	Synthetic Data	Real-world Data	TRivia Curated Data
	#Samples	700K	50K	50K
Model	**Trainable**	LLM	All	All
	Sequence Length	8192	8192	8192
Training	**Batch Size**	32	32	128
	LR: ψ_{ViT}	2×10^{-6}	2×10^{-6}	2×10^{-7}
	LR: $\{\theta_{\text{MLP}}, \phi_{\text{LM}}\}$	1×10^{-5}	1×10^{-5}	1×10^{-6}
	Epoch	1	2	1

Table 4. Training setup and hyperparameters in three training stages.

The training configurations for the three stages are summarized in Table 4. We use Qwen2.5-VL-3B as the backbone model. Across all stages, the number of image tokens ranges from 256 to 1280, corresponding to image resolutions from $256 * 28 * 28$ to $1280 * 28 * 28$. The prompt templates used for table recognition are provided in Table 11. In the third stage, which incorporates GRPO training, the sampling temperature is set to 1.2. We generate $G = 16$ samples per step and use a constant learning rate scheduler. All experiments are conducted with 8 X A100 80GB GPUs. Stage 1 training requires approximately one day, Stage 2 about two hours, and Stage 3 (GRPO) around two days.

B. Dataset Construction

For stage 1 and 2, we removes all samples with incomplete HTML tags during dataset construction. In stage 2, when only table structure tags were available [21, 49], we employ Qwen2.5-VL-72B to perform OCR over each cell's bounding box to recover textual content.

In stage 3, we begin with 100K unlabeled images. We employ response-consistency sampling using the stage-2 model with a temperature of 1.0, producing eight outputs per image. The consistency score is computed via pairwise TEDS among these outputs. Then, we calculate the consistency score using pairwise TEDS among the 8 outputs. Images are uniformly sampled across consistency score intervals from 0.4 to 1.0 with a step size of 0.1. To promote diversity, we ensure that all table images originate from distinct PDFs. The filtering results in 50K selected images.

We then generate QA pairs using the attention-guided QA generation for these 50K images. We use Qwen2.5-VL-72B as M_{QA}, generating 16 QAs per image with a temperature of 1.0 and prompt as shown in Table 8. Invalid JSON outputs are discarded. For cross-checking filtering, we adopt InternVL3-78B as M_{Val} to answer each question with and without table images as input and not having table images as input. We retain QA pairs whose F1 score exceeds 0.9 with the image but falls below 0.3 without it. Furthermore, we use M_{QA} to obtain the visual source of each QA pair and greedily search the final sets of QA pairs such that the IOU between any two QA pairs is less than 0.3. We remove images with fewer than 3 valid QAs to maintain reliable QA reward estimation. The final dataset comprises 48,470 images, each associated with an average of 28.3 QAs.

C. More Details about Experiments

All experiments are conducted using eight A100 80GB GPUs. For inference, we employ vLLM by default for all compatible models.

General-purpose VLMs for TR. When using general-purpose VLMs for table recognition, we perform prompt optimization to achieve the best results. We compare various templates from benchmarks such as OmnidocBench, CC-OCR, and OCRBench v2, and find that a unified prompt design (shown in Table 11) achieves the best overall performance. During generation, we set the sampling temperature to 0.2, which reduces repetitions relative to temperature 0 and mitigates hallucinations compared to temperature 1.0, yielding around a 3% performance gain. For Gemini 2.5 Pro, we enable "thinking mode", which improves performance by approximately 3%. For the Qwen2.5-VL and Qwen3-VL, we set the number of image tokens to $256 \sim 1280$, which is the same as TRivia-3B.

Document parsing VLMs for TR. For document parsing VLMs, we adopt their specialized table recognition prompts. In the case of PaddleOCR-VL, we disable unwrapping and document orientation classification modules, as disabling them empirically improves performance.

D. More experimental results

D.1. Extension to Multi-Task OCR Learning

	(a)	(b)	(c)	(d)
Base	80.73	31.52	70.01	56.73
Stage 2	88.84	40.19	80.11	60.55
TRivia	**90.20**	**61.95**	**86.10**	**72.96**

Table 5. Complex Table Types

	TR	KIE
Base	80.24	76.06
TRivia: TR	82.07	76.08
TRivia: TR+KIE	81.95	91.01

Table 6. Multiple OCR tasks

As TRivia is defined through task-specific QA rewards, it

can in principle be generalized to other OCR and document understanding tasks. In this section, we provide a preliminary study on extending TRivia beyond table recognition. As a first step, we study a joint table recognition (TR) and key information extraction (KIE) setting using a small-scale mixed training setup. For table recognition, we sample 3K data from our stage 3 data and use the overall TEDS on three benchmarks. For KIE, we employ CORD [27] for training and evaluation. The results are shown in Table 6.

Applying TRivia only to the TR component already improves table recognition performance while maintaining KIE accuracy. When QA rewards are further introduced for both TR and KIE, the model achieves a substantial gain on KIE while preserving TR performance. Although a comprehensive study on additional tasks such as text recognition, formula recognition, and layout analysis is beyond the scope of this work, these results provide initial evidence that TRivia is compatible with multi-task OCR learning and can serve as a general self-supervised adaptation strategy.

D.2. Performance on Complex Table Types

To better understand where TRivia brings the largest gains, we evaluate it on several representative complex table scenarios covered by OmniDocBench, CC-OCR, and OCRBench, including tables containing formulas, rotated tables, large tables, and borderless tables with merged cells. The results are summarized in Table 5.

TRivia improves performance consistently across all four settings. The most significant gain is observed on rotated tables, where the improvement over the base model reaches +21.76 TEDS. Large gains are also obtained on large tables and borderless merged-cell tables. These results indicate that the proposed self-supervised training strategy is especially effective for visually and structurally challenging cases that are underrepresented in conventional synthetic or benchmark-style training data.

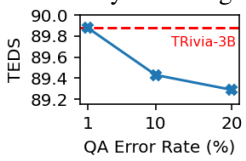

Figure 7. QA Error Impact

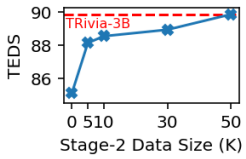

Figure 8. Stage-2 Data Size

D.3. Robustness to QA Noise

An important practical question is whether the effectiveness of TRivia depends critically on the correctness of the automatically generated QA pairs. To evaluate this, we conduct an additional robustness study by artificially injecting incorrect QA pairs into the training set. The results are shown in Figure 7. We observe that TRivia is relatively insensitive to moderate QA corruption: even when 20% of the QA pairs are intentionally replaced with incorrect ones, the degradation in table recognition performance remains limited.

This behavior can be explained by the reward design of TRivia and the relative optimization mechanism of GRPO. For a given table image, each sampled recognition response is assigned a scalar reward defined as the average QA score over the associated QA set. GRPO does not optimize these rewards in absolute terms; instead, it updates the policy according to the relative advantage of each response within the sampled group. Consequently, what primarily matters is whether a QA pair changes the reward ordering among candidate responses, rather than its standalone score.

When a QA pair is incorrect or ill-posed, it is unlikely to be answered correctly by any of the sampled responses, regardless of the recognition quality. In this case, the corresponding QA contributes nearly the same low score to all responses in the group. Such a shared contribution acts approximately as a common offset in the aggregated reward and is therefore largely removed during the within-group normalization in GRPO. As a result, these noisy QA pairs have limited influence on the relative preference structure among candidate responses, and thus do not substantially alter the optimization direction.

From this perspective, the main effect of noisy QA pairs is not to introduce systematic bias, but to reduce the number of informative reward components available for estimating response quality. Our proposed Attention-guided QA generation aims to mitigate this by providing more reliable reward estimation with diverse QA samples.

D.4. Effect of Stage-2 Data Size

We also investigate the relationship between Stage-2 supervised fine-tuning data size and the final performance of TRivia. As shown in Figure 8, strong performance can already be achieved with a relatively small amount of Stage-2 data.

This result suggests that the main role of Stage 2 is to provide a stable and reliable policy initialization for subsequent GRPO optimization, rather than to serve as the primary source of performance gains. Once a reasonable initialization is obtained, the Stage-3 TRivia training contributes most of the improvement.

D.5. Results on Standard Large-Scale Benchmarks

TR	Complex*		PubTabNet		FinTabNet	
Method	TEDS	S-TEDS	TEDS	S-TEDS	TEDS	S-TEDS
SLANNet-plus	68.19	79.21	86.57	96.43	73.77	84.84
UniTable	70.86	80.81	86.44	95.66	75.15	82.97
MuTabNet	59.24	77.41	94.42	97.51	97.24	98.17
TRivia (Stage-1)	77.85	83.23	95.52	96.43	96.74	97.68
TRivia (Stage-2)	88.57	92.48	90.81	93.04	89.60	91.82
TRivia-3B	89.88	93.60	91.79	93.81	91.66	93.71
TRivia (SFT)	89.58	93.24	95.00	96.05	96.52	97.63

* 1,512 complex tables from OmniDocBench, CC-OCR, and OCRBench.

Table 7. TEDS on Complex and Standard Large Benchmarks.

To complement the main experiments on challenging real-world tables, we further evaluate TRivia on the standard large-scale benchmarks PubTabNet and FinTabNet, and compare it with representative table recognition systems, including MuTabNet [14]. The results are reported in Table 7.

The comparison reveals two consistent observations. First, TRivia-3B remains substantially stronger on complex real-world tables collected from OmniDocBench, CC-OCR, and OCRBench, where existing specialized models exhibit a clear performance gap. Second, because PubTabNet and FinTabNet are only used in Stage 1, the subsequent stages of real-world adaptation may reduce in-domain performance on these benchmarks. When these benchmark-style samples are incorporated again in the final supervised stage, the resulting model restores strong performance on PubTabNet and FinTabNet while preserving the advantage on complex tables. These results suggest that TRivia is particularly beneficial for improving robustness to diverse real-world table layouts, while remaining compatible with conventional benchmark-oriented training when stronger in-domain accuracy is desired.

E. Prompt Template

This section provides all prompt designs used in this paper.

Table 8 shows the prompts for QA generation. We include several heuristic constraints in it. For example, the questions should avoid direct references to rows or columns, such as "the third row", which is heavily limited to the structural parsing ability of the M_{QA} model. The questions should be simple, avoiding complex reasoning, to reduce the impact of M_{LLM}'s capability on reward estimation.

Since our task involves bilingual table recognition in Chinese and English, prompts for both question answering with the LLM and the VLM are prepared in both languages. This prevents the model from outputting answers in the wrong language, as shown in Table 9.

Table 10 shows the prompts that M_{Val} use for cross-check filtering.

Finally, the table recognition prompt for the general-purpose VLM is shown in Table 11, which is developed through multiple iterations of optimization for best performance.

<image>
Given an image, your task is to generate 10 reasonable and natural QA pairs based on the following rules:

1. The questions should be contextually appropriate and natural.
2. The answer to each question must be a short word or two or a numerical value.
3. For tables in Chinese, generate QA pairs in Chinese. Ensure the question and answer are both in the same language.
4. Each question should have one and only one answer.
5. Distribute the QA pairs across different parts of the table to cover multiple data points, avoiding any concentration on a single row or column.
6. Avoid questions that involve reasoning, such as numerical comparisons, maximum/minimum values, or calculations.
7. Do not directly mention the table structure in the question; instead, incorporate natural references. For example, instead of asking, "What is the journal account in the first row?" ask, "What is the journal account for serial number 1?"
8. Exclude questions that could be answered by only one data point, such as "Does the table include future goals?" or "Does the table list the budget?"
9. Ensure the questions can be answered using an HTML-formatted table, and avoid referencing visual orientation or relative positioning (e.g., "What is to the left of a T-account?").

If you cannot generate QA pairs that meet the above criteria, output "None". Otherwise, output the generated QAs in JSON format.

Example Output Format:

```json
  [
    "question": "What is the market cap in Rmb mn?", "answer": "13,650.6",
    "question": "What is the 12 month price target?", "answer": "24.80",
  ]
```

Table 8. Q&A Generation Prompt

For question in English:
Given an HTML-formatted table and a corresponding question, your task is to respond appropriately based on table. If the table do not contain the answer of question, output "Not answerable".
Your answer should be a short phrase of only few words. Output the answer within <answer> </answer>.

HTML Table: {html_table}

Question: {question}

For question in Chinese:
给定一个HTML格式的表格以及一个相应的问题，你的任务是根据表格回答该问题。如果该表格不包含该问题的答案，请输出"无法回答"。你的答案必须简短、仅有一两个词语。输出答案时用<answer></answer>包裹。

HTML表格: {html_table}

问题: {html_table}

Table 9. LLM QA Prompt

For question in English with image input:
<image>
Given a table image and a corresponding question, your task is to respond appropriately based on table image. If the table do not contain the answer of question, output "Not answerable".
Your answer should be a short phrase of only few words. Output the answer within <answer> </answer>.

Question: {question}

For question in Chinese with image input:
<image>
给定一个表格图像以及一个相应的问题，你的任务是根据表格图片回答该问题。如果该表格不包含该问题的答案，请输出"无法回答"。你的答案必须简短、仅有一两个词语。输出答案时用<answer></answer>包裹。

问题: {question}
For question in English without image input:
Answer the following question. Your answer should be a short phrase of only few words. If you cannot answer this question, output "Not answerable".
Your answer should be a short phrase of only few words. Output the answer within <answer> </answer>.

Question: {question}
For question in Chinese without image input:
回答下面的问题。你的答案必须简短、仅有一两个词语。如果你无法回答该问题，请输出"无法回答"。你的答案必须简短、仅有一两个词语。输出答案时用<answer></answer>包裹。

问题: {question}

Table 10. VLM QA cross-checking Prompt

For general purpose VLMs:

<image>

You are an AI specialized in recognizing and extracting table from images. Your mission is to analyze the table image and generate the result in HTML format using specified tags. Output only the results without any other words and explanation.

For TRivia-3B:

<image>

You are an AI specialized in recognizing and extracting table from images. Your mission is to analyze the table image and generate the result in OTSL format using specified tags. Output only the results without any other words and explanation.

Table 11. Table Recognition Prompt