Exploring Multimodal Relation Extraction of Hierarchical Tabular Data with Multi-task Learning

Xinyu Zhang[1], Aibo Song[1*], Jingyi Qiu[1], Jiahui Jin[1], Tianbo Zhang[1], Xiaolin Fang[1],

[1]School of Computer Science and Engineering, Southeast University, Nanjing, China

{zxyseu,absong,jingyi_qiu,jjin,tianbozhang,xiaolin}@seu.edu.cn

Abstract

Relation Extraction (RE) is a key task in table understanding, aiming to extract semantic relations between columns. However, complex tables with hierarchical headers are hard to obtain high-quality textual formats (e.g., Markdown) for input under practical scenarios like webpage screenshots and scanned documents, while table images are more accessible and intuitive. Besides, existing works overlook the need of mining relations among multiple columns rather than just the semantic relation between two specific columns in real-world practice. In this work, we explore utilizing Multimodal Large Language Models (MLLMs) to address RE in tables with complex structures. We creatively extend the concept of RE to include calculational relations, enabling multi-task learning of both **semantic** and **calculational** RE for mutual reinforcement. Specifically, we reconstruct table images into graph structure based on neighboring nodes to extract graph-level visual features. Such feature enhancement alleviates the insensitivity of MLLMs to the positional information within table images. We then propose a Chain-of-Thought distillation framework with self-correction mechanism to enhance MLLMs' reasoning capabilities without increasing parameter scale. Our method significantly outperforms most baselines on wide datasets. Additionally, we release a benchmark dataset for calculational RE in complex tables.

1 Introduction

Tables are widely used to present and store data in various fields, and table understanding involves extracting structured information from tables for further analysis (Shigarov, 2023; Zhang et al., 2024a). Relation extraction (RE) is a crucial task in table understanding (Zhang et al., 2020; Deng et al., 2022; Suhara et al., 2022), focusing on extracting semantic relations between two table columns. For example, the "*birthplace*" relation type can connect the "*person*" and "*city*" columns, providing valuable information for downstream tasks like table question answering (Cheng et al., 2023).

Quarter	Total			Domestic			Overseas		
	division	quantity	sales	price	quantity	sales	price	quantity	sales
Q1	first	5	61	13	3	39	11	2	22
Q2		2	22	9	1	9	13	1	13
Q3	second	1	8	6	0	0	8	1	8
Q4		7	78	12	4	48	10	3	30
total		15	169	-	8	96	-	7	73

Figure 1: An illustration of a complex table with hierarchical headers. For example, the sub-columns "*price*" and "*sales*" share the same header "*domestic*", which represents a hierarchical structure we mainly focus on.

With advancements in generative Large Language Models (LLMs) like GPT-4 (Achiam et al., 2023), LLaMa (Touvron et al., 2023), and Qwen (Bai et al., 2023a), LLMs overcome the disadvantages of traditional models that typically require massive training data on specific tasks to avoid overfitting. Fine-tuning LLMs bring new possibilities for RE (Zhang et al., 2024a; Korini and Bizer, 2024). However, these studies focus on the simplest table structure composed of a matrix of rows and columns, and merely deal with the semantic relations between two specified columns. These overly strong constraints do not meet reality requirement. Wild tables often feature complex layouts like hierarchical headers and merged cells, containing various types of relations, as illustrated in Fig.1. Such tables are hard to be clearly expressed in textual format (Zheng et al., 2024). LLMs interpret tables in a one-directional textual perspective and their Optical Character Recognition (OCR) (Shi et al., 2023) capability of complex tables is poor, thereby we employ Multimodal Large Language Models (MLLMs) to directly digest two-dimensional tables via intuitive visual information from images.

Furthermore, given the demand for mining relations between multiple columns in real-world sce-

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 26770–26781

narios, we take multicolumn relation into account besides primitive semantic relation between just two columns. Given that wild tables tend to contain rich numerical data which is valuable for downstream applications like automated table analysis (Chen et al., 2023; Zhang et al., 2024b), we propose the concept of **calculational relation** which means specific columns can be calculated from related columns, promoting mutual task enhancement as both require understanding of headers and column contents. As shown in Fig.1, columns "*Overseas/price*" and "*Overseas/quantity*" may be considered related in semantic RE. While in calculational RE, they can be multiplied to get the "*Overseas/sales*". Such calculational relations enhance feature settings, enabling models to better grasp the interplay among sub-columns, thus improving semantic accuracy and serving as a check against errors. By sharing feature representations, the model leverages additional relevant data and task insights to mitigate overfitting. Through pilot experiments, we found that vanilla MLLMs with small parameters still struggle with numerical reasoning, a common issue in LLMs.

To cope with these challenges, we improve a visual algorithm, dividing table images into deformable patches and rebuilding as graph structure based on neighbor nodes to approximate the row and column positions of tabular data. Such method integrates graph-level visual features, helping MLLMs better capture hierarchical structures. The framework contains two stages: (1) In pre-alignment stage, we apply table recognition task to align the visual and textual embeddings; (2) Then we conduct multi-task instruction tuning for both semantic and calculational RE between columns, boosting performance through mutual reinforcement. We also propose a Chain-of-Thought (CoT) (Wei et al., 2022) distillation framework with self-correction mechanism, which generates high-quality reasoning steps from a powerful teacher model to aid student model in numerical task. In summary, our main contributions are as follows:

- Our work creatively extends RE task and propose the concept of calculational relation between multiple columns. We pioneer exploration of RE in hierarchical tables through multimodal mode, improving MLLMs' performance on semantic RE through mutual task enhancement.
- We enhance visual embeddings with graph-level features to improve perception of complex table structures. And we further propose a CoT distillation framework with a self-correction mechanism to assist smaller MLLMs in numerical reasoning.
- We release a benchmark dataset for the proposed task of calculational RE in complex tables, including manually annotated relation labels expressed as calculation expressions.

2 Related Work

2.1 Relation Extraction in Tables

Table understanding (Bonfitto et al., 2021) focuses on automatic extracting, transforming, and interpreting valuable information from tabular data, including Relation Extraction (RE) as a key subtask. Early methods fine-tune Pretrained Language Models (PLMs) like BERT (Devlin et al., 2019) with textual table serialization to learn tabular embeddings (Deng et al., 2022; Suhara et al., 2022). The advent of LLMs in recent years have opened up new possibilities. TableLlama (Zhang et al., 2024a) develops the first open-source generalist model for tables and evaluates on RE task.

2.2 Multimodal Large Language Models

Multimodal Large Language Models (MLLMs) (Zhu et al., 2024; Liu et al., 2023; Bai et al., 2023b) are mainly built upon a LLM, a visual module, and a cross-modality projector. The visual module encodes images into dense representations, and the projector, e.g., MLP, converts these representations into LLM-readable visual tokens. Finally, visual and textual tokens are fed into the LLM to perform next-token prediction. MLLMs aim to endow textual LLMs with comprehension for other modalities like images and videos. Table-LLaVA (Zheng et al., 2024) develops a versatile tabular MLLM for diverse table-based tasks, and fully evaluates the table understanding ability of existing models.

3 Task Definition

We define the multimodal RE task as follows:

Definition 1 (RE-semantic)**.** *Given an image of table T and a pair of columns (c_i, c_j) in T, a model $\mathcal{M}$ with given candidate relation set $\mathcal{R}$ predicts a relation $r_{ij} \in \mathcal{R}$ that best describes the semantic relationship between c_i and c_j.*

Definition 2 (RE-calculational)**.** *Given an image of table T containing a set of columns $(c_1, c_2, ..., c_n)$, a model $\mathcal{M}$ forms a calculation expression $E_{p,q,...,m}$ with math **operators** and takes column headers as **operands**, which describes the*

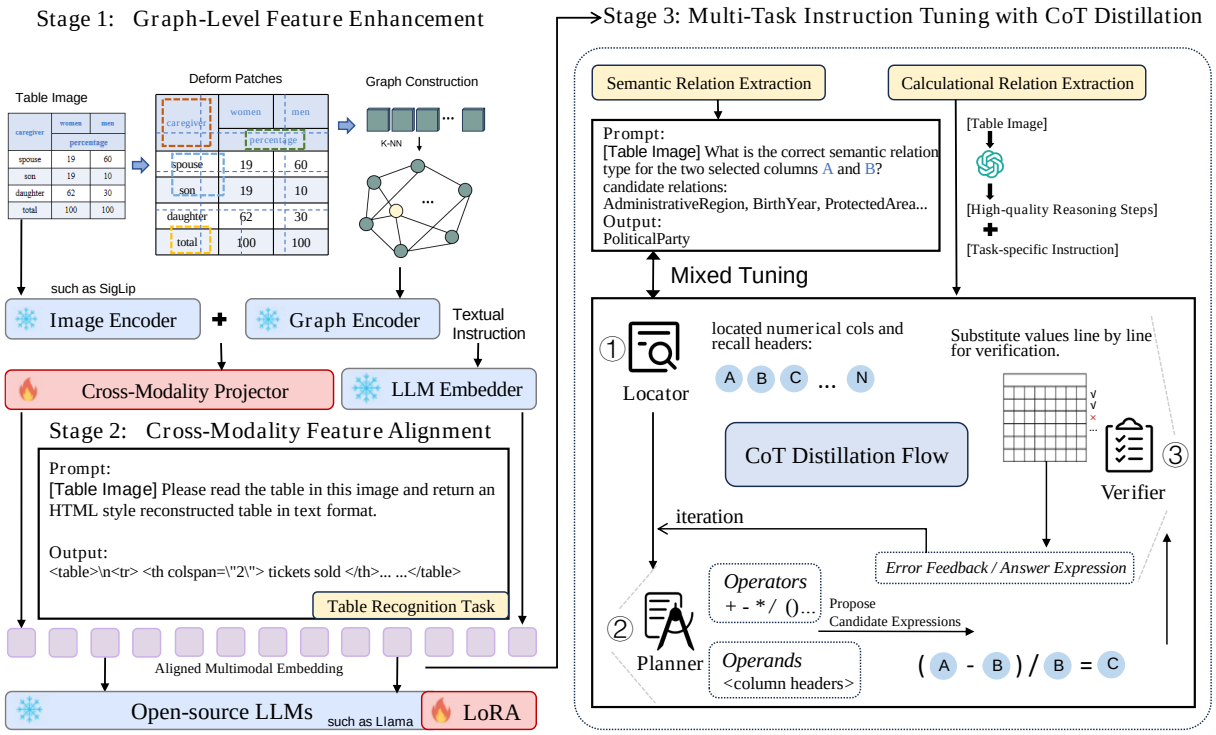

Figure 2: Overview of model structure and workflow.

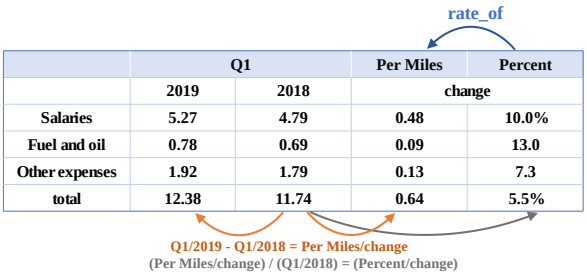

Figure 3: Demonstrations of semantic relation and calculational relation between table columns.

calculational relation between numerical columns $c_p, c_q, ..., c_m$.

Note that semantic relation exists between a pair of columns with a given set of candidate semantic relations, while the calculational RE is an open extraction task involves multiple columns. Here is an example for illustration:

Example 1. *As shown in Fig.3, the semantic relation between the "Per ASM/change" column and the "Percent/change" column can be predicted as* ***Rate_of****, while the calculational relation implied in the table can be extracted as* ***Q1/2019 - Q1/2018 = Per Miles/change****. For columns with hierarchical headers, we adopt a splicing method that connects multi-level header text with a "/" symbol.*

4 Methodology

The overall workflow is illustrated in Fig.2, with two-stage training: During pre-alignment, we fuse graph-level visual features from table's image and apply table recognition task to align the embeddings from different modals. Then we conduct multi-task instruction tuning for both semantic and calculational RE tasks to boost performance through mutual reinforcement.

4.1 Graph-Level Feature Enhancement

Recent works (Dosovitskiy et al., 2021; Tolstikhin et al., 2021; Han et al., 2022) have proved that viewing an image as a graph is more flexible and effective for visual perception. We naturally think of dividing a table image composed of grid cells into patches and reconstructing it into an undirected graph based on the proximity relationships between nodes. This process enables the model to effectively capture structural information in hierarchical layouts. Unlike previous methods that use fixed-size patch embeddings, we introduce an adaptive patch-partitioning approach (Chen et al., 2021) to minimize positional information loss of the cell objects contained in the image.

Embedding of Deformable Patches. As shown in Stage 1 of Fig.2, the initial input table image denoted as $A \in \mathcal{R}^{H \times W \times 3}$, is firstly divided into N fixed-size patches with size $a \times a \times 3 (a = \lfloor H\sqrt{N} \rfloor$, assume $H = W$ for simplicity) denoted as $\{p^{(i)}\}$, $1 \leq i \leq N$. Each patch is mapped to a high dimensional vector of length D via a linear

layer $\mathcal{L}$. Then we introduce an added branch (Chen et al., 2021) to predict a pair of offset for center coordinate and width-height dimension parameters for each patch with Eq. (1) and (2):

$$o_x, o_y = \sigma_1(W_\Delta \cdot \mathcal{L}(p)) \quad (1)$$

$$d_w, d_h = \sigma_2\left(\sigma_1\left(W_s \cdot \mathcal{L}(p) + b_s\right)\right) \quad (2)$$

Here, W_Δ and W_s are weight matrices that perform transformation on features, σ_1 and σ_2 are activation functions used to constrain offset amplitude and ensure non negative scale, while o_x, o_y, d_w, d_h represent predicted offsets and size, respectively. We can determine a new patch region based on predicted values and overlaps are allowed between patches. Finally we obtain a new feature map $\mathcal{P}$ = $[p_1, p_2, ..., p_N]$ of original table image, where p_i is the embedding of new i-th patch.

Graph Construction via k-NN. Feature of patches in $\mathcal{P}$ are then regarded as a set of unordered nodes $\mathcal{V}$ =$[v_1, v_2, .., v_N]$. We identify k nearest neighbors $\mathcal{N}(v_i)$ for each node v_i and establish an edge e_{ij} between v_i and v_j for $v_j \in \mathcal{N}(v_i)$. Thus we construct a graph representation $\mathcal{G}$ based on the visual features, where nodes are linked without the constraint of local position. We employ Pyramid ViG-B (Han et al., 2022) in experiments for graph convolution and follow its original settings to perform further training, k is set to 9 by default. Training details and model can be found in Appendix A. We take it as integrated vision encoder to obtain graph-level visual feature of tables.

4.2 Cross-Modality Feature Alignment

In order to develop MLLMs' ability to digest visual tabular data, we employ table recognition task (Zheng et al., 2024) to perform futher feature alignment. We conduct dimension adjustment on the graph-level visual embeddings obtained in Section 4.1 and concat it with conventional visual embeddings from ViT. A cross-modality projector is then trained to align visual features from various table images with ground-truth textual representations.

As shown in Stage 2 of Fig.2, the model learns to generate textual table representation (e.g., HTML string) based on instructions and corresponding table images with next-token prediction. This process improves the base LLM's perception of hierarchical table structure and character content, facilitating effective multi-task instruction tuning in later stages.

Prompt

[Table Image]
[Task Descriptions]

Please read the table image, identify the columns composed of numerical values, and return corresponding headers. If the column header is hierarchical, please concatenate it with / in sequence... ...

Response

Headers of columns composed of numerical values:
Q1/2019
Q1/2018
Per ASM/change
Percent/change

Figure 4: **Locator**. Read table image, locate numerical columns and recall corresponding hierarchical headers.

4.3 Multi-Task Instruction Tuning

Afterwards, we perform jointly multi-task instruction tuning that combines semantic and calculational relation extraction, mutually reinforcing performance in both tasks.

When dealing with diverse tables with varying cells distributions, MLLMs with limited parameters struggle to generate accurate and coherent responses (Tang et al., 2024), particularly when involving numerical values. We attempt to address this challenge by utilizing CoT prompting (Wei et al., 2022) to guide the model in generating intermediate reasoning steps. For the challenging calculational RE task, we devise a progressive CoT distillation framework with self-correction mechanism which extracts robust steps from large-scale closed-source LLMs like GPT-4V, and use them as supervisory signals to tune our smaller model (Tang et al., 2024).

The framework comprises three distinct modules: **Locator**, **Planner**, and **Verifier**. These modules guide the model to realize step-by-step numerical reasoning by generating intermediate steps, simultaneously ensure transparent and understandable reasoning and decision-making process.

Locator. As shown in Fig. 4, given the input table image and task instruction, the Locator identifies all columns that composed of numerical values and recalls their headers as the basis for constructing calculation expressions later.

Planner. Existing solutions for mathematical problems usually require models to generate expressions directly, but some calculations are too complex for accurate single-step generation. To address this issue, we decompose the expression into ***operators*** and ***operands*** in separate perspectives to simplify the generation of complete expressions. In most common cases, the operands are typically extracted from table cells, while operators are determined by the inter-column semantics (e.g., the

Prompt

[Table Image]
[Task Descriptions]
[Reasoning History]

Please infer the possible calculational relation to exist between these columns, use the corresponding headers as the operands and choose appropriate operators to form a complete arithmetic expression.... ...

Response

Based on the analysis, the most probable calculational relation is :
(Q1/2019- Q1/2018) / (Q1/2018) = (Percent/change)

[probable further analysis]

Figure 5: **Planner**. Interpret headers and infer the most appropriate calculation expressions.

Prompt

[Table Image]
[Task Descriptions]
[Reasoning History]

Please verify the correctness of the expression you proposed in the last step by sequentially substituting the values for each row corresponding to these columns and provide your calculation process. If the number of failed validation rows is reached <threshold>, please summarize these errors and end the validation… …

Response

The verification process involves the following steps:
[line-by-line value substitution for validation]

Situation 1: Validation succeeded
Return calculational relation expression

Situation 2: Validation failed
[miscalculation summary] to Planner for iteration.

Figure 6: **Verifier**. Validate line by line, collect error information for iterative self-correction when necessary.

"Percent" in the header suggests the division operator).

As shown in Fig. 5, the Planner first interprets the extracted header list sequentially, which interactively promotes the ability of semantic relation annotation. Then it infers the most probable calculational relation to exist, recalling relevant headers as operands and selecting appropriate operators to construct a preliminary arithmetic expression.

Verifier. For the arithmetic expression proposed by Planner, the Verifier performs sequential verification on the value sets within each row of the relevant columns, as shown in Fig. 6. Its goal is to accurately identify calculational errors and provide feedback. Considering the presence of noise in the numerical values of wild tables, a certain degree of tolerance is necessary, thereby we introduce an adjustable threshold η. If the number of rows that fail validation exceeds η, Verifier extracts the error details and returns them to the Planner for introspection and re-inference. The Planner then proposes the next possible expression or returns None if no satisfactory answer is found. During the process, we ensure robustness to floating-point arithmetic errors through a tolerance-based setting.

Through this workflow, we extract high-quality inference paths from the teacher and let our model learn to generate target answers and reasoning paths. This trains the logical reasoning ability of our smaller MLLM and enhances its calculational skills by learning the process of understanding, decomposing, and planning numerical problems. To improve consistency in task settings, we employ basic CoT paradigm by prompting the teacher model to directly generate reasoning evidence for the semantic RE task.

5 Experiments

5.1 Instruction Tuning Dataset Construction

Semantic RE datasets. We design a simple data processing pipeline which includes converting column type labels into relation labels and performing modal transformation on semantic RE datasets. We carefully design Python scripts to convert textual tables into high-quality images. Task-specific input and output are transformed into a pre-defined instruction-following format.

Calculational RE datasets. We collect over 1,200 tables containing rich numerical relations from datasets HiTab, WTQ and AIT-QA, most of which have hierarchical headers. We then conduct careful manual annotation of calculational relation in these tables. We obtain their images from MMTab (Zheng et al., 2024) and finally propose a benchmark dataset **CCR** for the **C**olumn **C**alculational **R**elation extraction task with about 2.2K samples [1], the demonstrations and detailed statistical information of CCR is presented in Tab. 1.

More details on dataset construction and sample demo can be found in Appendix B.

5.2 Experimental Settings

Datasets. We select a wide range of datasets to validate the effectiveness of our model on multi-relation extraction task, including TURL (Deng et al., 2022), Wiki (Bhagavatula et al., 2015), SemTab (Qiu et al., 2023), AIT-QA (Katsis et al., 2022), HiTab (Cheng et al., 2022) and WTQ (Pasupat and Liang, 2015), which cover basic and hierarchical tables for a comprehensive evaluation.

Baselines. We consider baselines of two main genres: **Open-source MLLMs** including BLIP (Li et al., 2022), Monkey (Li et al., 2024), BLIP2 (Li et al., 2023), MiniGPT-4 (Zhu et al., 2024), Qwen2-VL (Wang et al., 2024), LLaVA-1.5 (Liu et al., 2024), Table-LLaVA (Zheng et al., 2024)

[1] https://huggingface.co/datasets/1728-zxy/CCR

Table 1: Demonstrations of calculational relation between multiple table columns with hierarchical headers.

Operators	Demonstration
+	playoffs/w + playoffs/l = playoffs/gc
-	aboriginal population/percent - nonaboriginal population/percent = difference/percentage points
*	Fuel Expense (in millions) = Gallons Consumed (in millions) * Average Price Per Gallon
/	(regular season/w) / (regular season/gc) = regular season/win%
-/	(2012/number of jobs - 2011/number of jobs) / (2011/number of jobs) = 2012/percentage change
+/	Year = (Jan + Feb + Mar + Apr + May + Jun + Jul + Aug + Sep + Qct + Nov + Dec) / 12

and Bunny (He et al., 2024). **Open-source LLMs** including Llama-3[2], Baichuan2 (Yang et al., 2023), Qwen2.5 (Team, 2024) and TableLlama (Zhang et al., 2024a). To enable LLMs to digest table images, we follow (Zheng et al., 2024) to provide HTML sequences recognized from images via a competitive OCR engine PaddleOCR 2024[3].

Implementation Details. We take Bunny-v1.1 (He et al., 2024) as the base model, which contains three modules: a well pre-trained ViT model, SigLIP (Zhai et al., 2023) with S^2-Wrapper (Shi et al., 2024) enabled as the vision encoder, a two-layer MLP as the cross-modality projector and Llama-3 8B[2] as the base LLM. We conduct two-stage training of feature alignment and instruction tuning. We first optimize the cross-modality projector for one epoch to align visual and textual features. Then the MLLM is fine-tuned with instructions to adapt to downstream RE tasks, training the projector and LLM backbone with LoRA (Hu et al., 2022) applied. We adopt 6:4 ratio for instruction tuning and testing across all datasets and baselines. Considering the limited size of most datasets, improvement of data distribution diversify to avoid overfitting is necessary. We perform multi-level data augmentation on the training set by rewriting the prompt in chunks, and conduct random set partitions for three times and repeat the experiment. We report the average metrics for the final results and apply same settings for all baselines to ensure fairness. Detailed constructive process and examples are shown in Appendix C.

Metrics. For semantic RE task with a candidate relation set, we adopt Micro-F1 which is suitable for multi-classification tasks. For the calculational RE which requires exanct matchig of expressions, we use Accuracy for evaluation. And for precise evaluation of the calculation expression, we combine rule-based Python scripts and manual methods to address the issue that exact match cannot identify equivalent expressions with different formulations.

[2]https://github.com/meta-llama/llama3

[3]https://github.com/PaddlePaddle/PaddleOCR

5.3 Main Results

The main results are presented in Tab. 2 and prove that our framework achieves the best performance on most datasets. We additionally conduct single-task experiments for a comprehensive evaluation. We skip the experiment on the TURL dataset using TableLlama, as this model is tabular-targeted and has already been trained on this dataset.

Performance on single-task. As shown in Tab. 2, our method achieves superior performance in both semantic RE and the more challenging calculational RE tasks, respectively surpassing baseline models by a notable difference of 6.27% and 4.96% at most. One exception is the RE-semantic performance on the Wiki dataset, which is 3.64% lower than the best result, because the specification of columns in this dataset is achieved through column index instead of textual column names, and there are too many candidate relation types. Such factors depends heavily on the model's OCR capabilities, making it slightly inferior to the latest Qwen2-VL. The other exception is on HiTab-C, which is attributed to the high resolution of images it contains, leading to quality loss during resizing. Early MLLMs like BLIP exhibit minimal proficiency in multimodal table understanding due to the lack of tabular training data, but recent ones like LLaVA-1.5 make progress on comprehending table images, owing to their improvements on OCR scenarios. Moreover, TableLlama+OCR performs better that most other baselines on SemTab, because it has been instruction-tuned on large-scale tabular data, which leads to better table understanding and instruction-following ability. We skip the separate experiment on AIT-C for its limited sample size.

Performance on multi-task. Most methods significantly outperform single-task performance in multi-relation extraction by over 5%, owing to the inherent relationship between two tasks, which leverages the multi-task learning capabilities of LLMs. Our method achieves superior scores of 58.76% and 47.19%, exceeding the highest baselines by 4.69% and 3.09% separately. And most baselines strug-

Table 2: Overall evaluation results with best **bolded** and runner-up underlined. "RE-multi" refers to mixed instruction tuning of multi-task learning paradigm, which means tuning with all mixed 6 datasets and test separately on sum of 3 semantic/calculational datasets denoted as "Sem"/"Cal". While "RE-semantic" and "RE-calculational" refer to experiments focusing solely on two separate single tasks. The symbol † indicates that the model is designed for table tasks. "-C" represents the calculational relation dataset we annotated.

Method	LLM	RE-multi		RE-semantic			RE-calculational	
		Sem	Cal	TURL	Wiki	SemTab	HiTab-C	WTQ-C
MLLMs								
BLIP	385M	3.55	0	1.06	2.38	0.97	0	0.02
Monkey	Qwen 7B	15.17	7.54	13.30	5.27	11.96	7.44	10.95
BLIP2	Flan-T5 3B	5.45	7.01	2.17	4.40	2.01	0.54	1.21
MiniGPT-4	Vicuna 7B	29.11	19.07	13.74	6.73	23.98	5.19	11.43
LLaVA-1.5	Vicuna-1.5 7B	38.77	19.20	35.03	12.14	36.41	18.10	17.59
Table-LLaVA†	Vicuna-1.5 7B	33.52	16.38	30.77	17.21	39.67	13.06	22.38
Qwen2-VL	Qwen 7B	51.79	40.02	54.57	**31.15**	61.98	38.82	40.46
Bunny-v1.1	Llama-3 8B	54.07	31.16	53.31	25.51	58.19	26.60	35.10
LLMs								
Llama3+OCR	Llama-3 8B	53.10	37.80	49.86	26.85	54.70	37.94	33.08
Baichuan2+OCR	Baichuan2 7B Chat	42.83	29.32	37.15	15.28	26.17	26.23	19.03
Qwen2.5+OCR	Qwen2.5 8B	52.09	44.10	46.29	13.96	38.00	**43.22**	41.10
TableLlama+OCR†	Llama-2 7B	-	24.47	-	23.39	61.21	20.72	27.40
Ours								
OUR	Llama-3 8B	**58.76**	**47.19**	**60.84**	27.51	**63.52**	39.81	**45.12**

gle with the challenging calculational RE task, as this requires both strong numerical analysis skills and a deep understanding of complex table structures. Both the language models and multimodal models in the Qwen series achieve outstanding performance among open-source models. Overall, the experimental result trend can reflect that mixed instruction data of multi-task is more conducive to model performance.

5.3.1 Ablation Study

We conduct sufficient ablation experiments on both types of datasets to validate the effectiveness of each framework component, as shown in Tab. 3.

We divide the ablation study into three parts: (1) *Ablation of graph-level visual features.* Our first remove of graph-level features results in respectively 6.28% and 7.87% performance drop in multi-task scenario, shining the effectiveness of graph visual embeddings for enhancing understanding of table structures. (2) *Ablation of alignment task.* We remove the TR task to assess its individual impacts. It causes significant drop with performance

Table 3: Ablation results on three scenarios same as the setup of the main experiments.

Modules	RE-multi		RE-sem	RE-cal
	sem	cal	SemTab	HiTab-C
FULL	**58.76**	**47.19**	**63.52**	**39.81**
w/o gf	52.48	39.32	57.76	31.63
w/o TR	54.28	41.22	60.14	35.31
w/o CoT	54.71	36.80	60.03	29.70

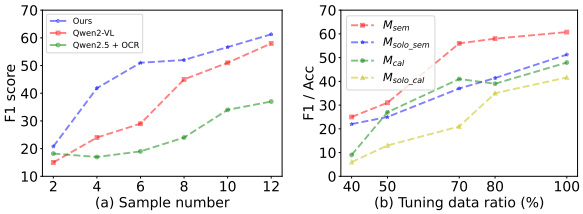

Figure 7: (a) Sample number m_s effect on semantic RE, ranging from 2 to 12. The whole table will be input when it has less rows than m_s. (b) Performance improvements over increasing the tuning data size.

decreasing by 4.48% and 5.97% respectively. This indicates that alignment is crucial for robust understanding of the mapping from table images to text especially for calculational RE task. (3) *Ablation of CoT.* We compare original instruction tuning with our approach of integrating distilled reasoning steps for CoT prompting. Our performance shows an impressive 10.39% gain in the calculation scenario, suggesting that the model better mimics the reasoning strategies of teacher model and enhances its intrinsic reasoning capability.

5.4 Data Efficiency Analysis

Input Data Efficiency. A common flaw in LLMs' semantic RE capacity is that they cannot handle a full wide table due to input length restriction. Our approach, however, is *input data efficient* and can make wise predictions with only a few samples from each column. This makes our model more attractive in practice as it can handle large long tables.

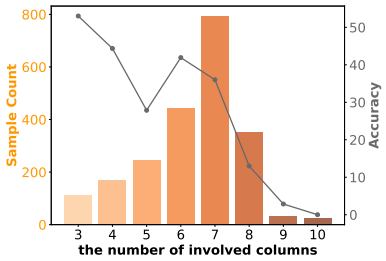

Figure 8: Effects of varying involved column numbers on calculational RE.

We train different variants of our model with fewer table rows to discuss their input efficiency, by conducting random row sampling when converting tables into images. For better comparison, we conduct same training on the models with best RE performance in MLLMs and LLMs. As shown in Fig.7, we find that our F1 score exceeded 0.42 with just 4 sampled rows and then gradually level off as the number of sampled rows increases, while the other two methods have relatively slow growth. This confirms that our model has a nice input data efficiency property and is practical for long tables on semantic RE task.

Tuning Efficiency. The setting of multi-task learning should further stabilize the performance with fewer tuning data for each task. To verify the tuning efficiency of our model $\mathcal{M}$, we compare variants with different high-quality instruction-tuning training data sizes (20% to 100%) and evaluate the performance on two single tasks where the model is denoted as $\mathcal{M}_{solo}$. As Fig.7 shows, $\mathcal{M}$ consistently outperforms $\mathcal{M}_{solo}$ and respectively achieves over 0.55 F1 scores and almost 0.4 Accuracy on two tasks with only 70% of the tuning data. This indicates that our model can effectively utilize limited data resources, which is crucial in low-resource RE scenarios.

5.5 Effect of Calculational RE Complexity

Intuitively, the complexity of a calculational RE task should be related to the number of items involved (refers to columns specifically in our task). Therefore, we delve into the performance of our method on complex calculational ability in tables which may include mixed operations. Specifically, we group and test the effectiveness of table samples in CCR with calculational expressions involving different numbers of columns, and the results are shown in the Fig.8. Consistent with our expectations, the extraction of calculational relation between 3 table columns works best with an Accuracy

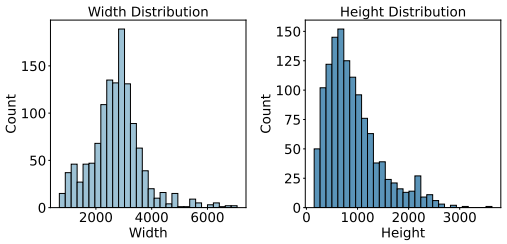

Figure 9: Resolution distribution of table images in CCR dataset.

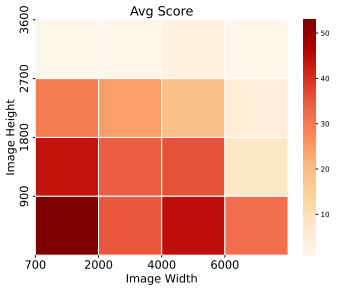

Figure 10: Calculational RE performance on images with wide and high resolutions in different ranges.

of 54.02% because it only involves fundamental arithmetic operations and very few columns. And then the performance sharply drops off as the number of involved columns increase, which implies the increase in calculational RE complexity.

Surprisingly, the model effects slightly rebound in the cases involving 6 and 7 columns. We further examine table samples in these two subsets and find that they contain a certain number of homogeneous tables and labels from HiTab-C. It can be speculated that the model may have learned enough inference information during tuning process, so as to perform better on similar tables in test stage.

5.6 Effect of Resolution Size

Image resolution is currently a key limited factor for the performance of MLLMs. To evaluate its impact on the performance of our model, we further conduct tests on table images in CCR. Fig.9 shows their wide and high-resolution distribution.

We divide table images with different width and height ranges into groups and separately test their performance, as shown in Fig. 10. We can see that table images with lower resolution have better effects on RE, within the acceptable resolution range of the vanilla SigLIP with S^2-Wrapper of 1152×1152. And performance drops off as the resolution increases. It is worth noting that the model is not so sensitive to increase in width, comparing to that in height. When the image width increases to over 2000, the model's performance can still reach about 35% -45%. While as the height

increases to the same level, the model's effectiveness rapidly decreases. Such results imply that our model can well perform RE tasks on wide tables with rich columns with a high upper bound of capacity.

6 Conclusion

This work pioneers the exploration of relation extraction task of hierarchical tables in multimodal mode and introduces the innovative concept of calculational relations among multicolumn, along with a manually annotated corresponding benchmark dataset. We propose a framework tailored for multi-RE in hierarchical tables, which demonstrates superior performance in experiments.

Limitations

While this work represents a pioneering effort in addressing multi-relation extraction from multimodal tables, several limitations remain for future exploration. The dataset available for research is still relatively scarce, and in the future, it is necessary to construct a large scale hierarchical table dataset that covers both semantic and calculational relations. Besides, the current calculational RE experiment still has some limitations, as the data volume is relatively small and could not represent generalization performance. Addressing these issues presents valuable foundation for future research.

Acknowledgments

We would like to acknowledge the support from the National Natural Science Foundation of China under the grant numbers [62061146001, 62232004] and the Big Data Computing Center of Southeast University.

References

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

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023a. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023b. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. 2015. Tabel: Entity linking in web tables. In *Proceedings of the 14th International Semantic Web Conference*.

Sara Bonfitto, Elena Casiraghi, and Marco Mesiti. 2021. Table understanding approaches for extracting knowledge from heterogeneous tables. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 11(4):e1407.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.

Zhiyang Chen, Yousong Zhu, Chaoyang Zhao, Guosheng Hu, Wei Zeng, Jinqiao Wang, and Ming Tang. 2021. Dpt: Deformable patch-based transformer for visual recognition. In *Proceedings of the 29th ACM International Conference on Multimedia*.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2022. HiTab: A hierarchical table dataset for question answering and natural language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, et al. 2023. Binding language models in symbolic languages. In *Proceedings of the 11th International Conference on Learning Representations*.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. Turl: Table understanding through representation learning. *ACM SIGMOD Record*, 51(1):33–40.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the 9th International Conference on Learning Representations*.

Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. 2022. Vision gnn: An image is worth graph of nodes. In *Advances in Neural Information Processing Systems*.

Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. 2024. Efficient multimodal learning from data-centric perspective. *arXiv preprint arXiv:2402.11530*.

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 *Proceedings of the 10th International Conference on Learning Representations*.

Yannis Katsis, Saneem Chemmengath, Vishwajeet Kumar, Samarth Bharadwaj, Mustafa Canim, Michael Glass, Alfio Gliozzo, Feifei Pan, Jaydeep Sen, Karthik Sankaranarayanan, and Soumen Chakrabarti. 2022. AIT-QA: Question answering dataset over complex tables in the airline industry. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*.

Keti Korini and Christian Bizer. 2024. Column property annotation using large language models. In *Proceedings of the 21st Extended Semantic Web Conference*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning*.

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*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems*.

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*.

Jingyi Qiu, Aibo Song, Jiahui Jin, Tianbo Zhang, Jingyi Ding, Xiaolin Fang, and Jianguo Qian. 2023. Dependency-aware core column discovery for table understanding. In *The Semantic Web – ISWC 2023*, pages 159–178, Cham. Springer Nature Switzerland.

Baifeng Shi, Ziyang Wu, Maolin Mao, Xin Wang, and Trevor Darrell. 2024. When do we not need larger vision models? In *Computer Vision – ECCV 2024*, pages 444–462, Cham. Springer Nature Switzerland.

Yongxin Shi, Dezhi Peng, Wenhui Liao, Zening Lin, Xinhong Chen, Chongyu Liu, Yuyi Zhang, and Lianwen Jin. 2023. Exploring ocr capabilities of gpt-4v (ision): A quantitative and in-depth evaluation. *arXiv preprint arXiv:2310.16809*.

Alexey Shigarov. 2023. Table understanding: Problem overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(1):e1482.

Qingyi Si, Tong Wang, Zheng Lin, Xu Zhang, Yanan Cao, and Weiping Wang. 2023. An empirical study of instruction-tuning large language models in Chinese. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4086–4107, Singapore. Association for Computational Linguistics.

Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çağatay Demiralp, Chen Chen, and Wang-Chiew Tan. 2022. Annotating columns with pre-trained language models. In *Proceedings of the 2022 International Conference on Management of Data*.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2024. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

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

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. 2021. Mlp-mixer: An all-mlp architecture for vision. In *Advances in Neural Information Processing Systems*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision*.

Dan Zhang, Madelon Hulsebos, Yoshihiko Suhara, Çağatay Demiralp, Jinfeng Li, and Wang-Chiew Tan. 2020. Sato: contextual semantic type detection in tables. In *Proceedings of the VLDB Endowment*.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024a. TableLlama: Towards open large generalist models for tables. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Xiaokang Zhang, Jing Zhang, Zeyao Ma, Yang Li, Bohan Zhang, Guanlin Li, Zijun Yao, Kangli Xu, Jinchang Zhou, Daniel Zhang-Li, et al. 2024b. Tablellm: Enabling tabular data manipulation by llms in real office usage scenarios. *arXiv preprint arXiv:2403.19318*.

Mingyu Zheng, Xinwei Feng, Qingyi Si, Qiaoqiao She, Zheng Lin, Wenbin Jiang, and Weiping Wang. 2024. Multimodal table understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 55006–55021. Curran Associates, Inc.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *Proceedings of the 12th International Conference on Learning Representations*.

A Training Details of Graph Encoder

We take the model of pvigb_224_gelu (Han et al., 2022) and its code library as basis, and continue training on mini-ImageNet after adding deformable patch prediction block. The training parameters are summarized in the Tab. 4, achieving 83.9% top-1 accuracy. The trained model is publicly available[4].

[4]https://huggingface.co/1728-zxy/pvig-b-deformable

Table 4: Training hyper-parameters for the graph encoder.

Pyramid ViG-B with deformable patches	Parameter
Epochs	30
Batch size	128
Optimizer	AdamW
Learning rate schedule	Cosine
lr	2e-6
weight-decay	0.05
k	9

B Dataset Construction Details

Semantic RE. For the RE samples in TURL dataset, we utilize its test (2072 column pairs from 1467 tables) and validation (2,175 column pairs from 1,560 tables) set in experiments due to the large number of complete sets. Finally, we obtain a total number of 4247 column pairs from 3027 tables for practice.

Due to the scarcity of suitable table RE datasets, we design a processing pipeline to construct training data. Semantic RE aims to identify relations between primary and non-primary columns. In other words, we aim to identify the attributes in non-primary columns as relations connecting them to the primary column. At the experimental level, the type labels of non-primary columns, e.g., "age" or "birth date" can naturally be interpreted as relation labels between them and primary column.

Based on this insight, we transform two classic column type annotation datasets (SemTab, Wiki) into relation label through a simple pipeline. For each table, we pair the primary column with each non-primary columns, using the type label of the non-primary column as corresponding relation label. We filter extremely long tables with too many rows or make appropriate truncation, and preprocess noise such as vacancy and garbled text.

Then we carefully design Python scripts to convert textual tables into high-quality images. Task-specific input and output are transformed into a predefined instruction-following format. We add extra requirement for the model to answer in JSON format to minimize errors when generating answers.

Calculational RE. We collect over 1,200 tables containing rich numerical relations from datasets HiTab, AIT-QA and WTQ, most of which have hierarchical headers. We then conduct a careful manual annotation process to identify calculational relations within these tables. We ensure robustness to floating-point arithmetic errors through a tolerance-based setting and implement difficulty-appropriate annotation filtering criteria. Finally we collect about 2.2K samples and obtain their images from MMTab (Zheng et al., 2024).

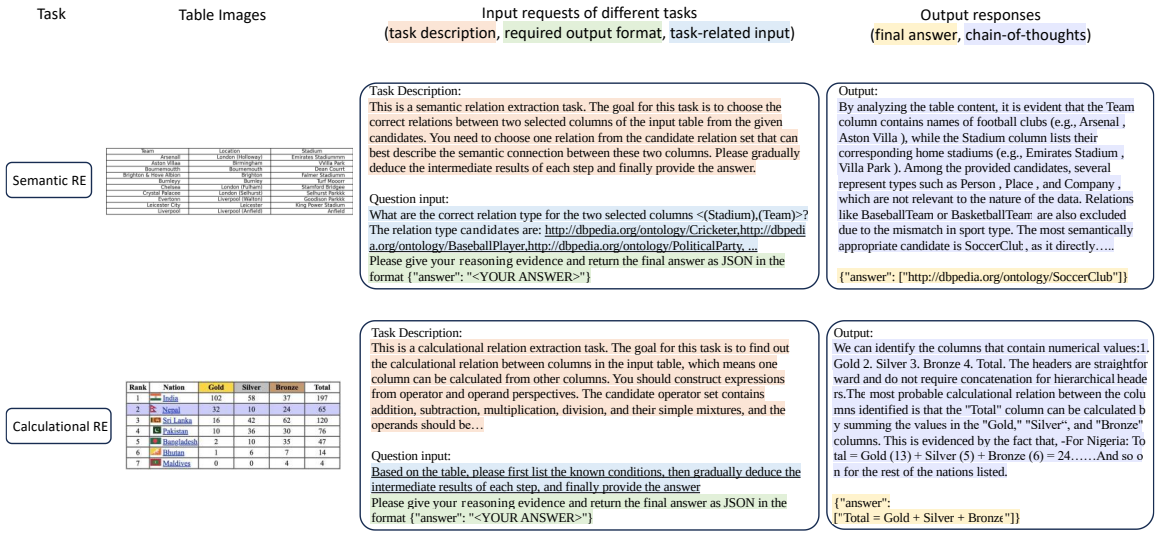

Figure 11: Concise task examples.

Demonstrations input and output of two tasks are shown in Fig.11.

C Multi-Level Data Augmentation

Previous works have shown that the diversity of instruction-following data is crucial to the capability of the resulting models (Zhou et al., 2023; Si et al., 2023). To diversify the tuning data distribution and avoid over-fitting, we perform data augmentations at multiple levels by rewriting the prompt in chunks.

We resort to GPT-4V to rewrite multiple versions of instruction templates, task descriptions and JSON output format based on several manually annotated demonstrations. We randomly select an instruction, a task description and an output format description from the candidate pool, and then combine them with the task-specific input such as table-related questions to produce the final sample. This combination strategy can bring more diversity of input requests. Moreover, our tables images are rendered with varied structures and styles such as Web-page, Excel and Markdown with fine-grained adjustments such font type and cell colors. Through random concatenation to produce the final input request and manually review, we ultimately generated multiple times training samples for datasets excluding TURL, as shown in Tab. 5.

D Implementation Details

For the pre-training stage of modal alignment, we employ the MMTab-pre dataset (Zheng et al., 2024) for the Table Recognition (TR) task, which comprises 150K table recognition samples in three text formats (HTML: 96K, Markdown: 27K, Latex: 27K) on 97K table images. To enhance generalization and robustness, we first perform training on general-purpose image-text data (100K samples from Bunny-pretrain-LAION-2M (He et al., 2024)) to warm-up, followed by domain-specific TR data. The projector is trained for one epoch with a learning rate of 1e-3. While in multi-task instruction tuning stage, we take learning rate of 2e-5 and the training lasts three epochs. We use the same cross-entropy loss for next-token prediction throughout.

Table 5: Dataset Statistics. "samples" and "tables" denotes the number of instruction pairs and involved tables. † means that only a portion of the dataset is used for experiments due to its large size. DA represents the multiple by which data augmentation was performed on training set. The symbol $\overline{n}$ refers to average number of columns involved per expression.

Datasets	Annotation Type		Table Info			
	sem	cal	samples	tables	DA	$\overline{n}$
TURL†	✓		4247	3027	-	2
SemTab	✓		370	166	5	2
Wiki	✓		247	221	5	2
HiTab-C		✓	1898	1012	2	6.723
WTQ-C		✓	232	208	5	4.594
AIT-C		✓	58	44	10	3.839
CCR		✓	2188	1264	-	-