Multimodal Table Understanding

Mingyu Zheng[1,2*†], Xinwei Feng[3*], Qingyi Si[1,2*], Qiaoqiao She[3], Zheng Lin[1,2‡], Wenbin Jiang[4], Weiping Wang[1]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]Baidu Inc, Beijing, China
[4]School of Artificial Intelligence, Beijing Normal University, Beijing, China
{zhengmingyu,siqingyi,linzheng,wangweiping}@iie.ac.cn
{fengxinwei,sheqiaoqiao}@baidu.com
{jiangwenbin}@bnu.edu.cn

Abstract

Although great progress has been made by previous table understanding methods including recent approaches based on large language models (LLMs), they rely heavily on the premise that given tables must be converted into a certain text sequence (such as Markdown or HTML) to serve as model input. However, it is difficult to access such high-quality textual table representations in some real-world scenarios, and table images are much more accessible. Therefore, how to directly understand tables using intuitive visual information is a crucial and urgent challenge for developing more practical applications. In this paper, we propose a new problem, multimodal table understanding, where the model needs to generate correct responses to various table-related requests based on the given table image. To facilitate both the model training and evaluation, we construct a large-scale dataset named MMTab, which covers a wide spectrum of table images, instructions and tasks. On this basis, we develop Table-LLaVA, a generalist tabular multimodal large language model (MLLM), which significantly outperforms recent open-source MLLM baselines on 23 benchmarks under held-in and held-out settings. The code and data is available at https://github.com/SpursGoZmy/Table-LLaVA.

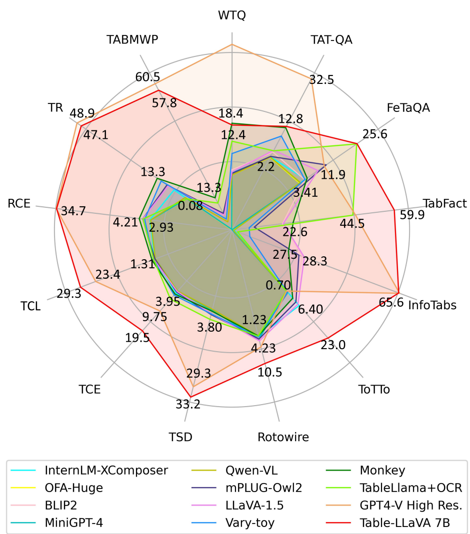

Figure 1: An overall performance comparison of Table-LLaVA 7B and existing MLLMs on various multimodal table understanding benchmarks. Table-LLaVA outperforms recent open-source MLLMs and is even competitive with the powerful GPT-4V on most tasks.

1 Introduction

Tables are widely used to store and present data across various fields, e.g., financial analysis, scientific research and government reports (Lautert et al., 2013; Shigarov, 2023). To make the most of the abundant tabular data, the table understanding (TU) technique has been proposed to automatically understand tables and perform table-based tasks, such as question answering (Pasupat and Liang, 2015) and text generation (Parikh et al., 2020). As a technique that could significantly elevate work efficiency in different industries, it has attracted ever-increasing research interest in recent years.

Though considerable efforts have been dedicated to the table understanding problem (Herzig et al., 2020; Liu et al., 2022), most previous models can only fulfill very limited tasks until the emergence of large language models (LLMs) (Brown et al., 2020; Chowdhery et al., 2022). With the help of powerful LLMs, we are getting closer to the vision that a versatile model can perform a variety of table-based tasks. However, existing table-oriented LLMs (Zhang et al., 2023b; Li et al., 2023c; Zha et al., 2023) rely heavily on the prerequisite that all given tables must be converted into a certain text sequence (like Markdown or HTML) to be input to LLMs. Under some practical scenarios like scanned documents and webpage screenshots, it

* Indicates equal contribution.
†This work was done during an internship at Baidu Inc.
‡ Corresponding author: Zheng Lin.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9102–9124
August 11-16, 2024 ©2024 Association for Computational Linguistics

is difficult to obtain such high-quality textual table representations, and yet table images are more accessible. Moreover, humans can directly understand two-dimensional tables using the intuitive visual information, whereas LLMs can only interpret tables in a one-directional textual perspective, which may increase the difficulty of comprehending diverse table structures and colored table elements. In summary, for the sake of convenience and intuitiveness, it is a crucial and urgent challenge to explore how to directly digest tables using visual information.

To promote the advancement of table understanding and its real-world applications, we propose the **multimodal table understanding** problem, where the model is required to generate correct responses to different table-related requests (e.g., questions) in an end-to-end fashion based on the table image. Despite the fact that recent multimodal large language models (MLLMs) have demonstrated excellent capabilities in many multimodal tasks, they cannot be directly extended to the proposed task. As shown in Figure 1, the performance of popular MLLMs like MiniGPT-4 (Zhu et al., 2023) and BLIP2 (Li et al., 2023b) is close to zero on most tasks, revealing their weakness in understanding tabular data. More importantly, there is a lack of a comprehensive dataset that can support both the development and evaluation of generalist MLLMs towards multimodal table understanding.

To address the above issue, we construct **MMTab**, the first open-source large-scale dataset for multimodal table understanding problem, based on 14 publicly available table datasets of 8 domains. We carefully design scripts to convert original textual tables in these datasets into table images highlighting a broad coverage of table structures and styles, and transform all task-specific samples into multimodal instruction-tuning samples with a unified format of `<table image, input request, output response>`. The resulting dataset contains (1) 150K table recognition samples on 97K table images for pre-training (named **MMTab-pre**). (2) 232K samples of 14 table-based tasks on 82K table images for instruction tuning (named **MMTab-instruct**). (3) 49K test samples on 23K table images composing 17 held-in and 7 held-out benchmarks (named **MMTab-eval**). During the dataset construction, data augmentations at multiple levels (e.g., table-level, task-level) were adopted to further improve the data diversity, and we also introduce multimodal table structure understanding tasks that have been overlooked in previous studies.

Based on the curated dataset, we develop a versatile tabular MLLM named **Table-LLaVA** with an enhanced two-stage training paradigm. In the first stage, we pre-train LLaVA-1.5 (Liu et al., 2023a) with an extra table recognition task on the MMTab-pre, which requires the model to generate textual sequences (like HTML) given table images. This stage aligns the structures and elements within table images to textual modality and thus enhances the comprehension of the basic table structure and content. In the second stage, we continue to instruction-tuning the model with diverse table-based downstream tasks on the MMTab-instruct, which endows the model with the multimodal instruction-following ability for table-related requests.

We compare Table-LLaVA with a series of open-source (M)LLMs and closed-source GPT-4V. Experimental results show that Table-LLaVA beats strong MLLM baselines on 17 held-in and 6 held-out benchmarks, and is even competitive with the powerful GPT-4V on 14 benchmarks with a subset of test samples. Extensive ablation experiments are conducted to reveal the contributions of different training data (e.g., the influence of table recognition pre-training data). We also explore the mutual influence between model's capacity for tabular tasks and non-tabular tasks. We hope this work could establish a strong base for future research on the multimodal table understanding problem and facilitate the progress of more general MLLMs.

We conclude our contributions as follows:

1) We make the first systematic exploration of the multimodal table understanding problem, which is complementary to the traditional text-only problem setting.

2) Accordingly, we construct and release a large-scale dataset MM-Tab which covers diverse tables and data for different tasks, including a series of novel table structure understanding tasks.

3) We develop a versatile tabular MLLM Table-LLaVA, which significantly outperforms a range of strong MLLM baselines under both held-in and held-out settings (Figure 1).

2 Related Work

2.1 Table Understanding

The table understanding (TU) problem concentrates on how to automatically extract, transform and interpret essential information from tabular data, and it has attracted significant attention in the past years (Bonfitto et al., 2021; Shigarov, 2023). Many tasks fall under the umbrella of table understanding problem, e.g., Table Question Answering (TQA) (Nan et al., 2022; Zheng et al., 2023), Table Fact Verification (TFV) (Chen et al., 2020) and Table-to-Text (T2T) generation (Cheng et al., 2022).

Different approaches have been proposed to solve specific TU tasks, ranging from early rule-based systems (Gatterbauer et al., 2007) to later tabular language models (TaLMs) (Liu et al., 2022; Chen et al., 2023a; Dong et al., 2022), which are pre-trained from general language models like BERT (Devlin et al., 2019) with extra large-scale table corpus. Nevertheless, these methods can only support limited TU tasks and handle tables of specific types. Recently, the emerging LLMs have opened up new possibilities for utilizing one single model to fulfill multiple table tasks. Researchers have attempted to enhance the TU ability of existing LLMs with different strategies such as prompt engineering (Chen, 2023; Sui et al., 2023), instruction tuning (Zhang et al., 2023b; Li et al., 2023c; Liu et al., 2023c) and combining external tools (Lu et al., 2023a; Li et al., 2023a). The resulting tabular LLMs like TableLlama (Zhang et al., 2023b) and TableGPT (Li et al., 2023c) can possess better TU ability and respond to wide-ranging table-related requests. However, previous TU approaches including tabular LLMs are unable to directly understand table images, which limits the potential application scenarios of TU technique.

2.2 Multimodal Large Language Models

With LLMs experiencing rapid advancements, recent studies have tried to endow the purely textual LLMs with understanding and perception capabilities of other modalities such as image and video, leading to the emergence of MLLMs (Alayrac et al., 2022; Li et al., 2022). Flamingo (Alayrac et al., 2022) proposes a gated cross-attention mechanism between vision encoder and LLM, which is trained on billions of image-text pairs to align vision and language modalities. BLIP2 (Li et al., 2023b) introduces a Q-Former with learnable query vectors to abstract the visual information from vision encoder into features of a fixed number. LLaVA (Liu et al., 2023b) uses a linear layer as a simpler cross-modal connector and achieve powerful performance with better data efficiency.

Though previous MLLMs demonstrated remarkable performance on multiple multimodal tasks (Liu et al., 2023d; Yu et al., 2023), their ability to digest table images and perform downstream tasks has not been thoroughly investigated. In this work, we build the first large-scale multimodal table understanding dataset and develop Table-LLaVA, a versatile tabular MLLM for diverse table-based tasks. To stimulate future endeavours on this problem, we also provide a comprehensive benchmark and fully evaluate the table understanding ability of existing models. More recently, researchers also tried to develop MLLMs like Vary (Wei et al., 2023) and Monkey (Li et al., 2023d) to understand document pictures with enhanced visual encoders, e.g., scaling up the vision vocabulary and image resolution. These models focus on the unified visual understanding of different document images and can be further improved with the proposed dataset.

3 MMTab Dataset

3.1 Data Collection

As shown in Table 1, with a pursuit of diverse table structures, tasks, and domains, we collect samples from 14 public table datasets of 8 domains (the first 14 rows in Table 1), covering 9 representative academic tasks. The detailed definition of each task can be found in Table 6. The original tables in these datasets are stored in divergent textual formats such as HTML or Markdown. We carefully design Python scripts with external packages like html2image to convert textual tables into high-quality table images. The task-specific input and output texts are transformed into the instruction-following format with pre-defined instruction templates. To minimize errors during answering parsing, we also add extra instructions, requiring models to output the final answer in the JSON format. As shown in the Figure 2, the rendered table images and processed input-output pairs constitute the final multimodal instruction-tuning samples with a unified format of `<table image, input request, output response>`. We adhere to the original dataset partitioning and select 11 datasets for training and held-in evaluation. 3 small-scale datasets with non-overlapping domains are used for held-out evaluation. The overview of sample construction process is depicted in Figure 3.

3.2 Data Augmentations

Previous works have shown that the diversity of instruction-following data is crucial to the capa-

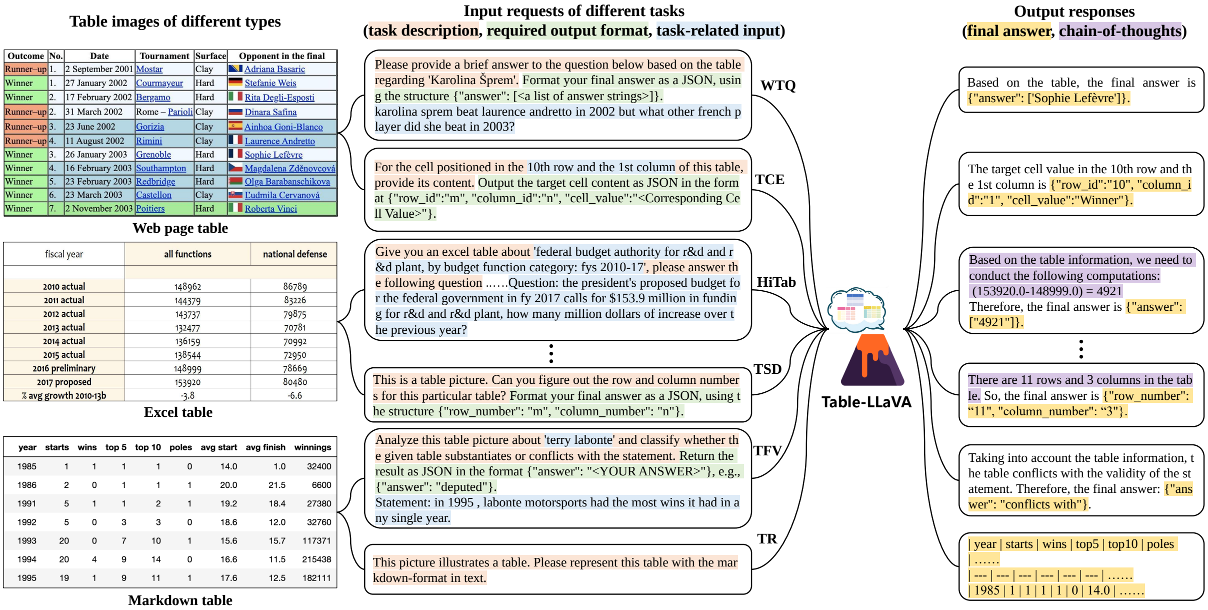

Figure 2: MMTab contains diversified table images and instruction following data, covering a wide range of tabular tasks (see Table 1). More dataset examples are shown in Figure 5-7 in Appendix A.1.

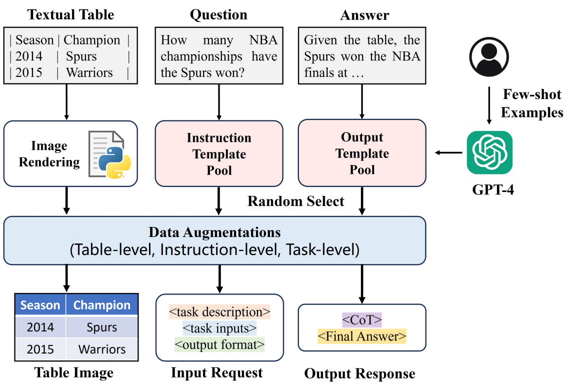

Figure 3: The overview of sample construction process.

bility of the resulting instruction-following models (Zhou et al., 2023; Si et al., 2023; Li et al., 2023c). To create more data diversity and avoid over-fitting in the model training, we perform additional data augmentations at multiple levels.

Table-level augmentations. Real-world tables often have varied structures and styles. An ideal table understanding model should be able to process divergent tables like a human reader. Since our dataset already includes diverse table structures from academic datasets, we separately design scripts to render table images with three different styles: Web-page (70.8%), Excel (19.4%) and Markdown (9.8%). Fine-grained adjustments such as font type and cell colors are also considered.

Instruction-level augmentations. In practical scenarios, user instructions for the same task are likely to vary from user to user. To improve models' robustness towards such variations, we resort to GPT-4 to generate new instruction templates and descriptions about JSON output format in a few-shot fashion based on several manually annotated demonstrations. Generated instruction templates with grammar mistakes or deviation from the original task are filtered out. When we construct input requests of each dataset, we randomly select an instruction template 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 input request. This combination strategy can bring more diversity of input requests. Using the TABMWP as an example, we show instruction templates and Python code for building input requests in Figure 8.

Task-level augmentations. Though the collected 14 public datasets highlight 9 academic tabular tasks (e.g., Flat TQA and Cell Description) which demand table-based reasoning capabilities, it is still a question whether existing MLLMs are truly aware of the basic table structures. Prior study has found that, despite achieving great performance on downstream tasks, tabular LLMs may still exhibit poor capacity for perceiving table structures (Sui et al., 2023). To further strengthen the fundamental table structure understanding ability of MLLMs, 6 table structure understanding tasks (the 6 rows with 'Structure Understanding' task category in Table 1) are devised, e.g., table size detection (TSD) task.

For each task, we use the above-mentioned method to generate input requests and design scripts to automatically extract the final answer from the textual tables in collected datasets. Finally, 8K training samples, 1K or 1.25K evaluation samples were constructed for each structure understanding task. Except above strategies, we also combine single-turn samples of the same table to compose 37K multi-turn conversation samples. At last, we obtain 232K samples on 82K table images for instruction-tuning (named **MMTab-instruct**), and 45K held-in and 4K held-out test samples on 23K table images for evaluation (named **MMTab-eval**).

Inspired by existing MLLMs which align textual descriptions with input images through image-text pre-training, we introduce the table recognition task as an important pre-training task for multimodal table understanding. In this task, MLLMs learn to generate a textual table representation such as an HTML sequence given the table image, which helps aligning structure and text information in the table image with the ground-truth. We additionally collect 20K table images from the ToTTo (Parikh et al., 2020) training split and merge them with table images in the MMTab-instruct to construct sufficient pre-training data. Based on these table images and their original textual tables, we write scripts to construct table representations of three formats (HTML, Markdown and Latex), and then build instruction-following samples in the same way of MMTab-instruct. The resulting pre-training data contains 150K table recognition samples (HTML: 96K, Markdown: 27K, Latex: 27K) on 97K table images, which is denoted as **MMTab-pre**. More details about MMTab are given in Appendix A.

3.3 Dataset Analysis

MMTab offers the following advantages: *(1) Large volume of data.* It contains 150K samples for pre-training, 232K samples for instruction-tuning, 45K samples and 4K samples for held-in and held-out evaluation, respectively. *(2) Including tables of diverse structures, styles and domains.* It includes 105K table images covering a broad range of structures (e.g., simple tables with flat structures as well as complex tables with merged cells and hierarchical headers), divergent styles (i.e., Web page, Excel, and Markdown tables) and multiple domains (e.g., Wikipedia and financial reports). *(3) Encompassing a wide range of tabular tasks.* In addition to 9 academic tasks which mainly evaluate the advanced table-based reasoning ability, MMTab also comprises 6 tasks aimed at assessing models' basic understanding of table structures. The broad coverage of tables and tasks can not only improve the generalization of the resulting model, but also provide a comprehensive testbed for MLLM research.

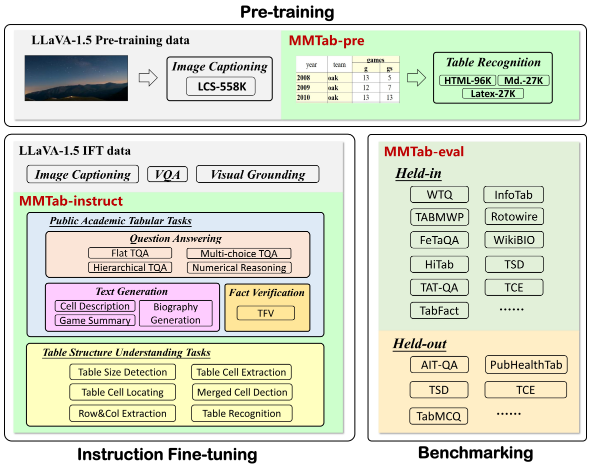

Figure 4: The two-stage training tasks and evaluation of Table-LLaVA. The red font represents our contribution.

4 Table-LLaVA

After constructing the MMTab dataset, we endeavor to fully leverage this data to promote models' multimodal table understanding ability. Inspired by the widely adopted training paradigm of previous MLLMs (Li et al., 2023b; Liu et al., 2023b; Zhu et al., 2023), we devise an enhanced two-stage training procedure and choose LLaVA-1.5 (Liu et al., 2023a) as the backbone to develop a versatile tabular MLLM named Table-LLaVA. The whole training process is illustrated in the Figure 4.

4.1 Model Architecture

Following LLaVA-1.5, the proposed Table-LLaVA consists of three modules: a pre-trained ViT model (Radford et al., 2021) as the visual encoder, a two-layer MLP as the vision-language connector and a Vicuna model (Chiang et al., 2023) as the backbone LLM. The ViT model encodes the input image into visual features, which are then projected into the word embedding space of LLM by the MLP connector. The Vicuna takes as input the concatenation of processed visual features and embedded textual features to generate responses.

4.2 Model Training

Pre-training. As depicted in the top-left region of Fig. 4, the vision-language connector is first

MMTab	Task Category	Task Name	Dataset	Table Style	Domain	Held-in	# Tables		# Samples		Avg. Length
							Train	Test	Train	Test	(input/output)
MMTab-instruct	Table Question Answering (TQA)	Flat TQA	WTQ (2015)	W	Wikipedia	Yes	1.6K	0.4K	17K	4K	45.9/10.4
		Free-form TQA	FeTaQA (2022)	W	Wikipedia	Yes	8K	2K	8K	2K	32.3/18.69
		Hierarchical TQA	HiTab (2022)	E	Wikipedia gorverment reports	Yes	3K	0.5K	8K	1.5K	63.5/12.6
			AIT-QA (2021)	E	Airline	No	-	0.1K	-	0.5K	41.8/10.2
		Multi-choice TQA	TabMCQ (2016)	M	science exams	No	-	0.05K	-	1K	47.9/13.2
		Tabular Numerical Reasoning	TABMWP (2023b)	W	math exams	Yes	30K	7K	30K	7K	54.2/51.9
			TAT-QA (2021)	M	financial reports	Yes	1.7K	0.2K	5.9K	0.7K	40.1/16.5
	Table Fact Verification (TFV)	TFV	TabFact (2020)	E, M	Wikipedia	Yes	9K	1K	31K	6.8K	49.9/18.3
			InfoTabs (2020)	W	Wikipedia	Yes	1.9K	0.6K	18K	5.4K	54.2/18.6
			PubHealthTab (2022)	W	public health	No	-	0.3K	-	1.9K	71.9/18.4
	Table to Text (T2T)	Cell Description	ToTTo (2020)	W	Wikipedia	Yes	15K	7.7K	15K	7.7K	31.1/14.8
			HiTab_T2T (2022)	E	Wikipedia gorverment reports	Yes	3K	1.5K	3K	1.5K	39.1/14.7
		Game Summary	Rotowire (2017)	E	NBA games	Yes	3.4K	0.3K	3.4K	0.3K	27.6/291.7
		Biography Generation	WikiBIO (2016)	E	Wikipedia	Yes	4.9K	1K	4.9K	1K	18.1/84.2
	Table Structure Understanding (TSU)	Table Size Detection	TSD	W, E, M	-	Yes	8K	1.25K	8K	1.25K	30.1/17.9
		Table Cell Extraction	TCE	W, E, M	-	Yes	8K	1.25K	8K	1.25K	51.6/19.9
		Table Cell Locating	TCL	W, E, M	-	Yes	8K	1.25K	8K	1.25K	72.5/45.6
		Merged Cell Detection	MCD	W, E, M	-	Yes	8K	1K	8K	1K	57.49/28.2
		Row&Column Extraction	RCE	W, E, M	-	Yes	8K	1.25K	8K	1.25K	45.6/55.1
		Table Recognition	TR	W, E, M	-	Yes	8K	1K	8K	1K	16.3/389.2
	Total						82K	-	232K	-	66.1/66.9
MMTab-eval	Total						-	23K	-	49K	46.3/32.6
MMTab-pre	Table Recognition		TR	W, E, M	-	-	97K	-	150K	-	16.4/397.5

Table 1: Breakdown statistics of the constructed **MMTab** dataset. W, E and M represents Web page, Excel, and Markdown tables, respectively. Task descriptions are shown in Table 6 in Appendix A.1. For TSD, TCE, TCL, RCE datasets, their test samples contains 1K held-in and 0.25K held-out evaluation samples.

pre-trained with an extra table recognition task on the MMTab-pre dataset, where the model is required to output a textual table representation (e.g., an HTML string) which encompasses both the table structure and table content. This process aims at aligning the visual features of diversified table images with the ground-truth textual table representations, which endows the model with augmented table structure perceiving and OCR ability and thus lays the foundation of more advanced tabular tasks.

Instruction fine-tuning. In the second stage, the pre-trained vision-language connector and the LLM are jointly fine-tuned with instruction following data of multimodal table tasks in MMTab-instruct and traditional multimodal tasks. While a plethora of multimodal instruction following datasets have been previously constructed (Liu et al., 2023b; Lyu et al., 2023; Xu et al., 2023), none of them have adequately solved the multimodal table understanding problem. The proposed MMTab-instruct contributes to addressing this gap and we use it to endow the model with the advanced ability to perform downstream table tasks. We also include the original pre-training and fine-tuning data of LLaVA-1.5 during the training process to improve the generalization of the resulting model and we analyze their influence in the ablation study.

5 Experiments

5.1 Experimental Setup

Baselines. We consider baselines of three genres: (1) Open-source MLLMs including BLIP (Li et al., 2022), OFA-Huge (Wang et al., 2022), BLIP2 (Li et al., 2023b), MiniGPT-4 (Zhu et al., 2023), Qwen-VL (Bai et al., 2023), InternLM-XComposer (Zhang et al., 2023a), mPLUG-Owl (Ye et al., 2023a) and mPLUG-Owl2 (Ye et al., 2023b), LLaVA-1.5 (Liu et al., 2023a), Vary-toy (Wei et al., 2024) and Monkey (Li et al., 2023d). (2) Open-source LLMs including Llama2 (Touvron et al., 2023) and its counterpart TableLlama (Zhang et al., 2023b), which uses LongLoRA (Chen et al., 2023c) to instruction-tune LLama2 on a series of textual tabular tasks. (3) The GPT-4V with low and high image resolution. Considering the high cost of GPT-4V, we randomly select 100 or 200 samples from each benchmark to compare Table-LLaVA with GPT-4V. To enable LLMs to digest table images, we consider an ideal scenario where LLMs are provided with oracle table sequences to explore the performance upper bound, and a more practical scenario where available table sequences are recognized from images by a competitive OCR engine (PaddleOCR, 2024). For all methods, the zero-shot setting was adopted during evaluation.

Method	LLM	Res.	Question Answering					Fact Verification		Text Generation			
			TABMWP	WTQ	HiTab	TAT-QA	FeTaQA	TabFact	InfoTabs	ToTTo	HiTab_T2T	Rotowire	WikiBIO
			Acc.	Acc.	Acc.	Acc.	BLEU	Acc.	Acc.	BLEU	BLEU	BLEU	BLEU
MLLM													
BLIP	385M	384	3.94	1.24	0.12	0.13	0.02	0.17	0.22	0	0.18	0.04	0.02
OFA-Huge	930M	-	0	0.06	0.07	0	0.07	0.26	0.11	0.20	0.15	0	0
BLIP2	Flan-T5 3B	224	3.34	2.01	1.52	2.20	2.34	18.62	27.53	4.3	2.63	1.08	0.72
MiniGPT-4	Vicuna 7B	224	0.22	0.90	0.20	0.13	0.39	0	0.10	0.20	0.11	1.26	0.33
Qwen-VL	Qwen 7B	448	3.30	0.09	0.06	0.13	0.45	1.12	0.65	0.80	0.18	0	0
InternLM-XComposer	InternLM 7B	224	0.06	0.05	0.12	0.26	2.62	1.19	1.11	7.10	3.25	0.43	1.52
mPLUG-Owl	Llama 7B	224	1.76	0.62	0.25	0.13	7.42	7.46	5.53	3.50	1.75	1.96	1.37
mPLUG-Owl2	Llama-2 7B	448	6.83	0.67	0.13	0.39	11.91	8.21	26.19	5.30	2.11	1.23	2.16
LLaVA v1.5	Vicuna-1.5 7B	336	6.05	1.24	2.03	2.97	8.24	18.9	28.31	6.40	2.07	1.92	2.34
Vary-toy	Qwen 1.8B	1024	4.42	7.96	3.42	8.81	2.44	6.33	6.98	0.70	0.27	0.46	0.37
Monkey	Qwen 7B	896	13.26	19.07†	6.41	12.31	3.41	22.56†	22.11	3.50	1.12	0.03	2.77
LLM													
Llama2+Oracle	*Llama-2 7B*	-	*17.88*	*4.26*	*1.21*	*3.62*	*5.54*	*4.21*	*7.55*	*6.20*	*1.84*	*4.67*	*1.33*
Llama2+OCR	Llama-2 7B	-	16.35	3.91	0.77	5.27	5.15	4.32	7.17	-	1.56	3.90	1.28
TableLlama+Oracle	*Llama-2 7B*	-	*12.98*	*31.63*‡	*64.71*‡	*2.84*	*39.05*‡	*82.55*‡	*2.85*	*20.77*‡	*0.19*	*0.13*	*0.39*
TableLlama+OCR	Llama-2 7B	-	11.09	12.49	**13.51**†	2.72	25.44†	44.54†	2.18	-	0.12	0.13	0.31
Ours													
Table-LLaVA 7B	Vicuna-1.5 7B	336	57.78	18.43	10.09	12.82	25.60	59.85	65.26	23.00	9.74	**10.46**	**9.68**
Table-LLaVA 13B	Vicuna-1.5 13B	336	**59.77**	**20.41**	10.85	**15.67**	**28.03**	**65.00**	**66.91**	**24.10**	**10.40**	8.83	9.67

Table 2: Evaluation results on 11 held-in academic tabular benchmarks. '+*Oracle*' and '+OCR' represents that the ground truth or OCR-extracted textual table representations are provided to LLMs, respectively. We only report model performance in the ideal '+*Oracle*' setting and compare with models in the more practical '+OCR' setting. † indicates the model has been trained on the corresponding dataset, ‡ denotes results from original papers.

Implementation details can be found in App. B.

Evaluation metrics. For TQA, TFV, and T2T benchmarks, we use accuracy or BLEU (Papineni et al., 2002). For TSD, we compute accuracy for predicted row and column numbers separately. For TCE and TCL, we compute accuracy at cell-level. For MCD, we use cell-level F1. For RCE, we compute cell-level F1 for extracted rows and columns, respectively. For table recognition (TR) task, we follow Zhong et al. (2020) and use the Tree-Edit-Distance-based Similarity (TEDS) score, which is based on the tree structure of HTML table sequence and can measure both the structure similarity and the cell content similarity between the prediction and the ground truth. The score is normalized between 0 and 1, where 1 means perfect matching. For TR testing samples whose target sequences are in the Markdown or Latex format, we convert the predicted sequences into the HTML format to compute their TEDS scores.

5.2 Results and Analysis

Public academic tabular benchmark results. *Performance of open-source MLLMs.* As we can see from the MLLM rows in Table 2, early MLLMs (e.g., MiniGPT-4, BLIP) exhibited minimal proficiency in multimodal table understanding due to the lack of tabular training data, but recent MLLMs (e.g., LLaVA-1.5 and Monkey) have yielded better capacity for comprehending table images, which can be attributed to their improvements on the OCR and text-rich scenarios. Especially, among existing MLLMs, Monkey performs the best in most question answering tasks and fact verification tasks because of the training on relevant table datasets (i.e., WTQ and TabFact).

Performance of LLMs. As shown in Table 2, TableLlama+OCR performs better than Llama2+OCR on several tasks (e.g., HiTab, FeTaQA, TabFact) through fine-tuning on the corresponding training data, but this also damages its generalization ability on unseen tasks (e.g., InfoTabs and TABMWP). Compared to Llama2+OCR, Llama2+Oracle does not achieve notable improvements, indicating that its bottleneck is the ability to understand tables and follow related instructions, rather than the table recognition ability. On the contrary, TableLlama+Oracle consistently outperforms TableLlama+OCR in all tasks, because it has been instruction-tuned on large-scale tabular data, which leads to better table understanding and instruction-following ability. Thus, the provided oracle table sequences break the bottleneck of the OCR engine's table recognition capability, resulting in significant improvements.

Comparison between Table-LLaVA and existing models. Compared to previous open-source MLLMs and LLMs+OCR, Table-LLaVA 7B and

Method	LLM	Res.	TSD		TCE	TCL	MCD	RCE		TR		
			Row Acc.	Col. Acc.	Acc.	Acc.	F1	Row F1	Col. F1	HTML TEDS	Markdown TEDS	Latex TEDS
MLLM												
BLIP	385M	384	0	0.10	0.76	0	0	0	0	0	0.18	0
OFA-Huge	930M	-	0	0.10	0.26	0	0	0	0	0	0.16	0
BLIP2	Flan-T5 3B	224	0.20	0.30	0.15	0	0	0.06	0	0	0.25	0
MiniGPT-4	Vicuna 7B	224	0.40	0.40	0	0	0	0	0	0	0.34	0
Qwen-VL	Qwen 7B	448	0	0	0.03	0.03	0.38	0	0	0	2.51	0
InternLM-XComposer	InternLM 7B	224	0.90	3.00	0.89	0.28	0.14	0.28	0.25	13.33	2.61	1.34
mPLUG-Owl	Llama 7B	224	1.20	3.90	0.13	0.16	0.34	2.04	1.38	15.31	7.36	3.13
mPLUG-Owl2	Llama-2 7B	448	0.50	3.50	0.51	0.17	0.45	3.49	2.38	15.71	6.67	4.43
LLaVA v1.5	Vicuna-1.5 7B	336	0.80	2.50	0.22	0.62	1.26	1.66	4.13	12.88	10.74	1.55
Vary-toy	Qwen 1.8B	1024	1.30	2.20	1.96	0.73	0.52	2.01	2.38	10.13	12.72	11.67
Monkey	Qwen 7B	896	0.80	0.60	1.46	1.31	0.67	3.89	4.53	21.96	13.29	4.54
LLM												
Llama2+Oracle	*Llama-2 7B*	-	*1.70*	*3.60*	*0.62*	*0.17*	-	*9.36*	*18.03*	-	-	-
Llama2+OCR	Llama-2 7B	-	1.30	3.40	0.35	0.15	-	8.15	10.45	-	-	-
TableLlama+Oracle	*Llama-2 7B*	-	*5.30*	*4.40*	*9.35*	*0.82*	-	*4.34*	*5.26*	-	-	-
TableLlama+OCR	Llama-2 7B	-	3.90	3.70	3.95	0.65	-	2.82	2.39	-	-	-
Ours												
Table-LLaVA 7B	Vicuna-1.5 7B	336	33.10	**33.20**	19.45	29.31	**17.14**	**31.43**	37.93	50.24	44.82	46.11
Table-LLaVA 13B	Vicuna-1.5 13B	336	**34.40**	27.60	**19.53**	**29.68**	16.52	31.07	**41.49**	**51.44**	**46.00**	**46.50**

Table 3: Evaluation results on 6 held-in table structure understanding benchmarks. For all evaluation metrics, higher values indicate better performance. HTML, Markdown and Latex represents the format of target textual table representations in the table recognition (TR) task, and TEDS is its evaluation metric. See Section 5.1 for the details.

13B both surpass them with large margins, demonstrating the effectiveness of our methods and the value of MMTab dataset. One exception is the accuracy of TableLlama+OCR on HiTab, which maybe because table images in this benchmark are relatively large, leading to information loss when resizing them into desired resolutions of Table-LLaVA (i.e., 336×336). We believe there is great potential for using more powerful MLLMs to perform diverse multimodal table understanding tasks.

Table structure understanding benchmark results. Table structure understanding is a fundamental ability for fulfilling more advanced tabular tasks. As can been found in Table 3, both previous MLLMs and LLMs failed to generalize well on these relatively simple tabular benchmarks that are almost trivial for humans. What's more, their performance is even worse than that on more challenging academic benchmarks in Table 2. This shows that these powerful (M)LLMs may rely on some superficial correlations (Geirhos et al., 2020) to perform downstream tabular tasks that require complex reasoning, and they actually lack the important ability to perceive basic table structures.

Held-out tabular benchmark results. Table 10 reports model performance on 7 held-out benchmarks whose data do not appear in the model training. We can find that previous open-source models excel at different benchmarks respectively, and no model can consistently outperform others on all these tasks. By contrast, Table-LLaVA achieves best performance on most benchmarks, except for the accuracy of Vary-toy on AIT-QA, which is because AIT-QA contains large tables extracted from annual reports of airline companies and Vary-toy might have seen similar tables in its training data of company document images. Besides, Vary-toy supports higher input image resolution (1024), which is more friendly for large tables.

Comparison with GPT-4V. The average performance of Table-LLaVA and GPT-4V on five types of benchmarks is shown in the upper part of Table 4. GPT-4V achieves remarkable results under both low (512×512) and high (768×2000) image resolution. The average performance of Table-LLaVA 7B (336×336) is better than GPT-4V with low resolution (512×512) on four types of benchmarks, while GPT-4V surpasses Table-LLaVA in the held-out scenario, indicating its strong generalization ability. As can be seen from detailed benchmark performance in Table 8, Table 9 and Table 10, Table-LLaVA achieves better or competitive results with GPT-4V on 14 out of 24 benchmarks. Besides,

Method	TQA	TFV	T2T	TSU	Held-out
GPT-4V (On a subset of test samples)					
Low Resolution	24.15	52.00	2.42	28.11	30.40
High Resolution	**35.91**	55.55	3.05	31.16	**44.49**
Ours (On a subset of test samples)					
Table-LLaVA 7B	24.55	**65.25**	**9.49**	34.24	23.16
Table-LLaVA 13B	26.63	64.50	9.12	**34.36**	24.71
Table-LLaVA 13B	26.95	65.96	13.25	34.42	25.62
Table-LLaVA 7B	24.94	**62.56**	**13.22**	**34.27**	**24.46**
w/o LLaVA-pre	24.06	61.45	12.40	31.18	21.50
△	-0.88	-1.11	-0.82	-3.09	-2.96
w/o MMTab-pre	23.45	60.32	12.26	29.55	21.73
△	-1.49	-2.24	-0.97	-4.73	-2.72
w/o LLaVA-instruct	**24.98**	61.85	12.87	33.98	23.90
△	+0.04	-0.71	-0.36	-0.29	-0.56
w/o MMTab-instruct	2.82	20.57	4.08	5.68	3.02
△	-22.12	-41.99	-9.14	-28.60	-21.43
w/o TSU-instruct	24.34	62.28	12.39	5.99	13.24
△	-0.60	-0.28	-0.83	-28.28	-11.22
w successively IFT	24.76	61.99	13.06	33.89	23.85
△	-0.18	-0.57	-0.16	-0.38	-0.61

Table 4: Upper: Comparison with GPT-4V. Lower: Ablation experiment results. We report average performance over benchmarks under five types, respectively. △ stands for the performance gap between Table-LLaVA 7B and its variants. 'TSU-instruct' stands for 6 table structure understanding datasets in MMTab-instruct. 'successively IFT' represents that 'LLaVA-instruct' and 'MMTab-instruct' are used to fine-tune the model in a sequential order rather than mixed together.

GPT-4V can obtain significant improvements from high image resolution, which helps the model comprehend fine-grained table elements and structures in large tables. We also analyze the influence of input image resolution on the performance of Table-LLaVA in Appendix C.1.

Ablation study. We conduct sufficient ablation experiments to validate the effectiveness of our proposed dataset and training strategy. We divide the ablation study into three parts: (1) *Ablation of pre-training*. As shown in Table 4, both 'w/o LLaVA-pre' and 'w/o MMTab-pre' cause negative effects, and the latter results in a larger decline. This is because both LLaVA-pre and MMTab-pre help align visual and textual modalities, while MMTab-pre is more suitable for multimodal alignment of table understanding. (2) *Ablation of instruction fine-tuning.* 'w/o LLaVA-instruct' causes a slight performance decrease, indicating that though LLaVA-instruct has different image domains and task settings from MMTab, it has benefits for the multimodal table understanding due to the enhancement of instruction-following ability. 'w/o MMTab-instruct' leads to a significant performance drop on all types of tasks, resulting in extremely poor performance (e.g., 3.02 on held-out benchmarks). This further confirms that our constructed data can supplement the missing table understanding capability of the current MLLMs. If the table structure understanding data in MMTab-instruct is removed (i.e., 'w/o TSU-instruct'), we can find that, although it does not cause obvious performance damage to traditional academic tasks like TQA and TFV, it has a huge negative impact on TSU and Held-out tasks. This indicates that the proposed TSU datasets also help with model generalization. (3) *Ablation of training strategies.* We compare models instruction-tuned with LLaVA-pre and MMTab-pre in sequence ('w successfully IFT') or mixed together. We find that 'w successfully IFT' has slightly weaker performance, which suggests that mixed data is more conducive to model performance.

The influence of MMTab on non-tabular tasks. Table 7 lists performance of Table-LLaVA and its backbone LLaVA-1.5 on two non-tabular benchmarks: TextVQA (Singh et al., 2019) and LLaVA-Bench (In-the-Wild) (Liu et al., 2023b). Table-LLaVA beats LLaVA-1.5 in most cases under both model sizes, which demonstrates that MMTab actually has positive impact on the performance of non-tabular tasks. Combing this with ablation of non-tabular training data, we can find that there are mutual benefits between model's capacity for tabular tasks and non-tabular tasks, which shows that table understanding is one fundamental and necessary ability of MLLM and it deserves more investigations. **More results and analysis such as case study are shown in Appendix C.**

6 Conclusion

This paper proposes a novel multimodal table understanding problem, together with a large-scale open-source dataset MMTab, which covers a broad range of table structures and tabular tasks. This dataset provides a comprehensive testbed for MLLM research with held-in and held-out multimodal tabular benchmarks. On the basis of MMTab, we empower LLaVA 1.5 to be a generalist tabular MLLM Table-LLaVA. Experimental results show that Table-LLaVA significantly outperforms existing MLLMs on multiple benchmarks, and is even on par with the powerful GPT-4V. In conclusion, the contributions of this paper lie at promoting the research on multimodal table understanding from the task, dataset and model perspectives.

7 Limitations

Though this work makes the first comprehensive exploration towards the multimodal table understanding problem, there are certain limitations that can be left to the follow-ups. First, the proposed dataset mainly focus on the single table in English. The multi-table scenario together with broader language coverage have not yet been considered. Second, MMTab is based on real-world tables from carefully selected table datasets and it contains diverse high-quality table images rendered by automatic scripts. Nevertheless, table images in the wild can be low-quality. For instance, blurred, handwritten or incomplete table images. To further bridge the gap between the academic research and the real application scenarios, more diversified table images from the wild could be collected in the future, and their corresponding instruction following data needs to be constructed. We believe this could significantly promote the applications of MLLM-based table understanding systems. In the end, though the proposed Table-LLaVA demonstrates great performance on a wide range of table-based tasks, the resolution of input images is relatively low and may limit the upper bound of its capacity. Luckily, with the emergence of MLLMs which possess higher input image resolution (e.g., Monkey (Li et al., 2023d), LLaVA-Next (Liu et al., 2024)), we can use MMTab to develop more powerful tabular MLLM in the future research.

8 Ethical Considerations

The proposed MMTab dataset is constructed based on the academic datasets like WTQ and TabFact, which are free and open datasets for research use with licenses like MIT License[1] or CC-BY-SA-4.0 License [2]. We write Python scripts to render textual table sequences (like HTML) in these datasets to obtain table images, and build multimodal instruction-following data based on original samples. The resulting dataset MMTab is also a free and open resource for the community to study the multimodal table understanding problem. Thus, the authors foresee no ethical concerns with the research in this paper.

[1] https://opensource.org/license/mit/

[2] https://creativecommons.org/licenses/by-sa/4.0/deed.en

References

Mubashara Akhtar, Oana Cocarascu, and Elena Simperl. 2022. PubHealthTab: A public health table-based dataset for evidence-based fact checking. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1–16, Seattle, United States. Association for Computational Linguistics.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond.

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.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Pei Chen, Soumajyoti Sarkar, Leonard Lausen, Balasubramaniam Srinivasan, Sheng Zha, Ruihong Huang, and George Karypis. 2023a. Hytrel: Hypergraph-enhanced tabular data representation learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Wenhu Chen. 2023. Large language models are few(1)-shot table reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1120–1130, Dubrovnik, Croatia. Association for Computational Linguistics.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023b. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact: A large-scale dataset for table-based fact verification.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023c. Longlora: Efficient fine-tuning of long-context large language models.

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 (Volume 1: Long Papers)*, pages 1094–1110, Dublin, Ireland. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Haoyu Dong, Zhoujun Cheng, Xinyi He, Mengyu Zhou, Anda Zhou, Fan Zhou, Ao Liu, Shi Han, and Dongmei Zhang. 2022. Table pre-training: A survey on model architectures, pre-training objectives, and downstream tasks.

Wolfgang Gatterbauer, Paul Bohunsky, Marcus Herzog, Bernhard Krüpl, and Bernhard Pollak. 2007. Towards domain-independent information extraction from web tables. In *Proceedings of the 16th international conference on World Wide Web*, pages 71–80.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. INFOTABS: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, Online. Association for Computational Linguistics.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Sujay Kumar Jauhar, Peter Turney, and Eduard Hovy. 2016. Tabmcq: A dataset of general knowledge tables and multiple-choice questions.

Yannis Katsis, Saneem Chemmengath, Vishwajeet Kumar, Samarth Bharadwaj, Mustafa Canim, Michael Glass, Alfio Gliozzo, Feifei Pan, Jaydeep Sen, Karthik Sankaranarayanan, and Soumen Chakrabarti. 2021. Ait-qa: Question answering dataset over complex tables in the airline industry.

Larissa R. Lautert, Marcelo M. Scheidt, and Carina F. Dorneles. 2013. Web table taxonomy and formalization. *SIGMOD Rec.*, 42(3):28–33.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Hongxin Li, Jingran Su, Yuntao Chen, Qing Li, and Zhaoxiang Zhang. 2023a. Sheetcopilot: Bringing software productivity to the next level through large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*.

Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2023c. Table-gpt: Table-tuned gpt for diverse table tasks.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. 2023d. Monkey: Image resolution and text label are important things for large multi-modal models. *arXiv preprint arXiv:2311.06607*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. Llava-next: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. In *NeurIPS*.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022. TAPEX: Table pre-training via learning a neural SQL executor. In *International Conference on Learning Representations*.

Qian Liu, Fan Zhou, Zhengbao Jiang, Longxu Dou, and Min Lin. 2023c. From zero to hero: Examining the power of symbolic tasks in instruction tuning.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2023d. Mmbench: Is your multi-modal model an all-around player?

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023a. Chameleon: Plug-and-play compositional reasoning with large language models. In *The 37th Conference on Neural Information Processing Systems (NeurIPS)*.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2023b. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *International Conference on Learning Representations (ICLR)*.

Chenyang Lyu, Minghao Wu, Longyue Wang, Xinting Huang, Bingshuai Liu, Zefeng Du, Shuming Shi, and Zhaopeng Tu. 2023. Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration. *arXiv preprint arXiv:2306.09093*.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, and Dragomir Radev. 2022. Fetaqa: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

PaddleOCR. 2024. table recognition model from pp-structure. `https://github.com/PaddlePaddle/PaddleOCR/tree/release/2.7/ppstructure/table`. Accessed on: 2024-02-14.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

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.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Alexey Shigarov. 2023. Table understanding: Problem overview. *WIREs 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.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2023. Gpt4table: Can large language models understand structured table data? a benchmark and empirical study.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas

Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *CoRR*, abs/2202.03052.

Haoran Wei, Lingyu Kong, Jinyue Chen, Liang Zhao, Zheng Ge, Jinrong Yang, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. 2023. Vary: Scaling up the vision vocabulary for large vision-language models.

Haoran Wei, Lingyu Kong, Jinyue Chen, Liang Zhao, Zheng Ge, En Yu, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. 2024. Small language model meets with reinforced vision vocabulary. *arXiv preprint arXiv:2401.12503*.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Zhiyang Xu, Ying Shen, and Lifu Huang. 2023. Multi-instruct: Improving multi-modal zero-shot learning via instruction tuning.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023a. mplug-owl: Modularization empowers large language models with multimodality.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2023b. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration.

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

Liangyu Zha, Junlin Zhou, Liyao Li, Rui Wang, Qingyi Huang, Saisai Yang, Jing Yuan, Changbao Su, Xiang Li, Aofeng Su, Tao Zhang, Chen Zhou, Kaizhe Shou, Miao Wang, Wufang Zhu, Guoshan Lu, Chao Ye, Yali Ye, Wentao Ye, Yiming Zhang, Xinglong Deng, Jie Xu, Haobo Wang, Gang Chen, and Junbo Zhao. 2023. Tablegpt: Towards unifying tables, nature language and commands into one gpt.

Pan Zhang, Xiaoyi Dong, Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Haodong Duan, Songyang Zhang, Shuangrui Ding, Wenwei Zhang, Hang Yan, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Jiaqi Wang. 2023a. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2023b. Tablellama: Towards open large generalist models for tables.

Mingyu Zheng, Yang Hao, Wenbin Jiang, Zheng Lin, Yajuan Lyu, QiaoQiao She, and Weiping Wang. 2023. IM-TQA: A Chinese table question answering dataset with implicit and multi-type table structures. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5074–5094, Toronto, Canada. Association for Computational Linguistics.

Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. 2020. Image-based table recognition: data, model, and evaluation.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini 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.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.

A More Details about MMTab

A.1 Task Descriptions and More Dataset Examples

Detailed description and evaluation metric of each task are given in Table 6, and more dataset examples are illustrated in Figure 5, 6, 7. When we collect tables from TabMCQ dataset, we filter extremely long tables more than 50 rows. For the hybrid-QA dataset TAT-QA, we only preserve samples whose questions can be answered with the table information. For the ToTTo dataset, its training set contains 35K tables and we randomly select 15K tables for MMTab-instruct in order to reduce the cost of transforming HTML tables into images.

Except augmentation strategies mentioned in Section 3.2, we also perform additional data augmentations including: (1) "response-level augmentations", where we synthesize chain-of-thoughts using annotated intermediate computational procedures in the original datasets and use them to augment the final answer. (2) "conversation-level augmentations", where we randomly choose samples of the same table to compose multi-turn conversation samples.

Hyperparameter	Pre-train	Fine-tune
training data	MMTab-pre (150K), LLaVA-pre (558K)	MMTab-instruct (232K), LLaVA-instruct (665K)
batch size	256	128
max length	2560	
learning rate (lr)	1e-3	2e-5
lr schedule	cosine decay	
warmup ratio	0.03	
weight decay	0	
optimizer	AdamW	
epoch	1	
Deepspeed Stage	2	3
machine	one machine with 8 80GB A800	
training time (w/o flash-attention)	32 hours	26 hours

Table 5: Hyperparameter setting and training details of Table-LLaVA.

A.2 Instruction Templates

The diversity of the instruction-following data has a significant impact on the performance of the resulting model. As discussed in the Section 3.2, we utilize in-context learning to ask GPT-4 to generate new instruction templates and create more diversity of input request. When we build input requests of each dataset, we randomly choose an instruction template and an output format description from the candidate pool, and then combine them with the task-specific input such as the question to produce the final input request. Figure 8 shows the Python code for this combination process, together with all instruction templates and JSON output format descriptions for the TABMWP dataset. Previous textual instruction-following datasets for tabular tasks such as TableInstruct (Zhang et al., 2023b) usually adopt one fixed instruction template for each dataset. By contrast, we construct at least 20 instruction templates for each dataset while considering their respective characteristics.

B Implementation Details

Following LLaVA-1.5 (Liu et al., 2023a), we use the well-trained CLIP-ViT-L-336px (Radford et al., 2021) as the visual encoder and input images are resized to 336×336. We develop two Table-LLaVA models with Vicuna-1.5 7B and 13B as the backbone LLM, and we denote the resulting models as Table-LLaVA 7B and Table-LLaVA 13B, respectively. We follow the original hyper-parameter setting of LLaVA-1.5 except that We increased the max sequence length from 2048 to 2560 to accommodate longer text sequences. The training hyper-parameters for both the pre-training and the visual instruction tuning are listed in Table 5. In this paper, all experiments including baseline experiments were conducted on a single machine with 8 80GB A800. Without using flash-attention (Dao et al., 2022), the pre-training process and the instruction-tuning takes about 32 hours and 26 hours for one epoch, respectively. Unless otherwise specified, we evaluate performance of baseline models on our benchmarks with the official implementations. As mentioned in the Section 3.1, we add extra instructions to the input request which require models to output the final answer in the JSON format, and we write Python scripts with regular expressions to extract the final answer for a fair comparion. Some baselines like Monkey cannot follow instructions to output the answer in the desired JSON format, which may only output a short answer due to the overfitting of their training data. Thus, we relaxed requirements and specifically designed answer extraction scripts to calculate their accuracy. For ToTTo benchmark, since the ground-truth of testing samples have not been open-sourced, we submit the output results of different models to the official website to get evaluation results.

MMTab	Task Category	Task Name	Dataset	Task Description	Metric
MMTab-instruct	Question Answering	Flat TQA (F TQA)	WTQ	TQA based on tables which usually possesses a flat structure with the first row as the sole column header.	Accuracy(↑)
		Free-form TQA	FeTaQA	TQA with a free-form text answer rather than a short text span copied from the table.	BLEU(↑)
		Hierarchical TQA (H TQA)	HiTab	TQA based on tables which usually possesses hierachical headers and merged cells.	Accuracy(↑)
			AIT-QA		Accuracy(↑)
		Multi-choice TQA	TabMCQ	TQA with multi-choice questions.	Accuracy(↑)
		Tabular Numerical Reasoning	TABMWP	TQA requiring mathematical reasoning operations such as finding the largest number or do math computations.	Accuracy(↑)
			TAT-QA		Accuracy(↑)
	Fact Verification	Table Fact Verification	TabFact	Given a table as evidence and a statement, the task is to distinguish whether the given statement is entailed or refuted by the table.	Accuracy(↑)
			InfoTabs		Accuracy(↑)
			PubHealthTab		Accuracy(↑)
	Text Generation	Cell Description	ToTTo	Generate a one-sentence description for the highlighted table cells.	BLEU(↑)
			HiTab_T2T	Generate a one-sentence description for the highlighted table cells using the provided operators such as SUM, DIVISION.	BLEU(↑)
		Game Summary	Rotowire	Given a table recording box- and line-scores of an NBA game, the task is to generate a detail game summary which is sourced from rotowire.com.	BLEU(↑)
		Biography Generation	WikiBIO	Given a table containing information of a person, the task is to generate a biography to introduce this person.	BLEU(↑)
	Structure Understanding	Table Size Detection	TSD	Determine the row number and column number of the given table.	Accuracy at row or column level(↑)
		Table Cell Extraction	TCE	Given a group of (row_id, column_id), the task is to extract the corresponding table cells.	Accuracy(↑)
		Table Cell Locating	TCL	Given a group of cells, the task is to find positions of these cells in the table and return their position in theformat of (row_id, column_id).	Accuracy(↑)
		Merged Cell Detection	MCD	Determine whether the table contains merged cells and return postions of top-left and bottom-right cells in the merged regions.	F1(↑)
		Row&Column Extraction	RCE	Given a group of row_id or column_id, the task is to extract the corresponding table cells in the target rows or target columns.	F1 at row or column level(↑)
		Table Recognition	TR	Given a table image, the task is to return a textual representation of the table in the format of HTML, Markdown or Latex Same	TEDS(↑)
MMTab-pre	Table Recognition		TR for pre-training		

Table 6: Detailed description of each task and their evaluation metrics.

Models	TextVQA	LLaVA-Bench (in-the-wild)			
		Conversation	Detail description	Complex reasoning	Overall
LLaVA v1.5 7B	58.2*	54.3	49.6	72.4	61.4
Table-LLaVA 7B	**59.2**	**58.3**	**50.9**	**73.2**	**63.2**
LLaVA v1.5 13B	61.3*	72.0	**53.8**	72.0	67.5
Table-LLaVA 13B	**61.9**	**72.0**	53.7	**77.1**	**69.6**

Table 7: Comparison of Table-LLaVA and its backbone on non-tabular tasks. $*$ indicates results are from the original LLaVA-1.5 paper.

C More Experimental Results and Analysis

C.1 Influence of Input Image Resolution

To shed more light on the influence of image resolution on multimodal table understanding, we divide test samples into 5 groups by their image resolution and evaluate model performance on different groups. The results, illustrated in Figure 9, demonstrate that image resolution has a great impact on model performance. The model performance gradually degenerates with the increasing image resolution, which reveals that it is necessary to enlarge the input image solution of MLLMs in order to process extremely large table images.

C.2 Influence of MMTab on Non-tabular Tasks

We compare Table-LLaVA with its backbone LLaVA-1.5 on two non-tabular benchmarks: TextVQA (Singh et al., 2019), a VQA benchmark requiring the understanding of image texts, and LLaVA-Bench (In-the-Wild) (Liu et al., 2023b), a recent general benchmark for MLLMs including 3 task categories (conversation, detail description and complex reasoning). The results are listed in the Table 7. Table-LLaVA beats LLaVA-1.5 in most cases under both model sizes, which demonstrates that tabular training data has positive impact on the performance on non-tabular tasks.

C.3 Influence of OCR Success Rate on LLM Performance

We compute the cell-level OCR success rates on 4 benchmarks and show the performance of textual

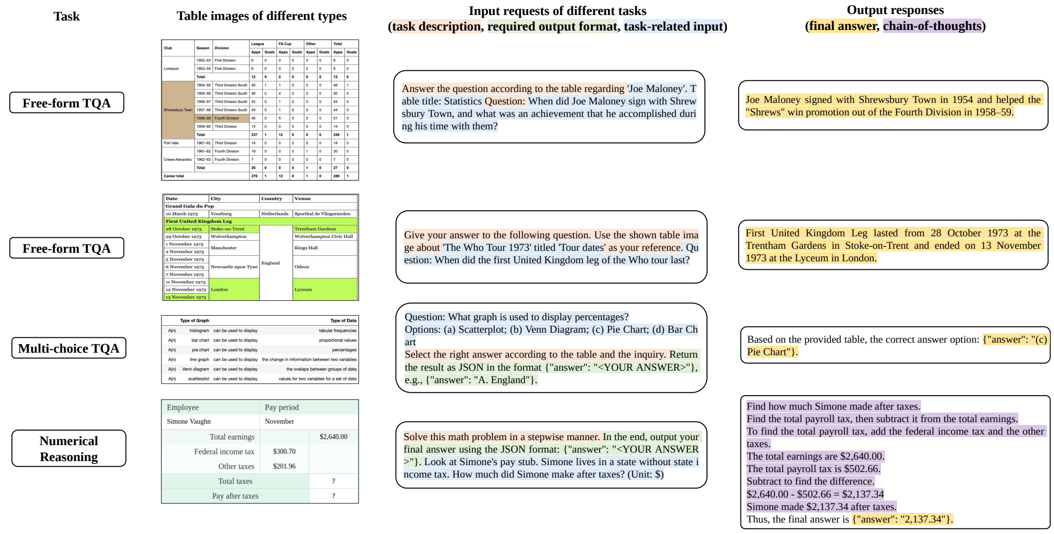

Figure 5: More dataset examples.

Method	LLM	Res.	Question Answering					Fact Verification		Text Generation			
			TABMWP	WTQ	HiTab	TAT-QA	FeTaQA	TabFact	InfoTabs	ToTTo	HiTab_T2T	Rotowire	WikiBIO
			Acc.	Acc.	Acc.	Acc.	BLEU	Acc.	Acc.	BLEU	BLEU	BLEU	BLEU
Ours (on all test samples)													
Table-LLaVA 7B	Vicuna-1.5 7B	336	57.78	18.43	10.09	12.82	25.60	59.85	65.26	23.00	9.74	**10.46**	**9.68**
Table-LLaVA 13B	Vicuna-1.5 13B	336	**59.77**	**20.41**	10.85	**15.67**	**28.03**	**65.00**	**66.91**	**24.10**	**10.40**	8.83	9.67
GPT-4V (on a subset of test samples)													
Low Resolution	GPT-4	512	60.00	22.50	9.50	19.50	9.26	45.50	58.50	-	1.85	3.89	1.55
High Resolution	GPT-4	768*2000	**60.50**	**48.00**	**27.50**	**32.50**	11.04	45.50	65.60	-	2.98	4.23	1.94
Ours (on a subset of test samples)													
Table-LLaVA 7B	Vicuna-1.5 7B	336	57.00	18.00	7.50	11.00	29.23	**63.50**	67.00	-	9.34	**10.08**	**9.04**
Table-LLaVA 13B	Vicuna-1.5 13B	336	60.00	21.50	8.00	14.00	**29.63**	59.50	**69.50**	-	**9.53**	9.00	8.84

Table 8: Comparison between GPT-4V and Table-LLaVA on 11 held-in public academic tabular benchmarks. Note that we randomly select a subset of testing samples for each tasks due to the high cost of GPT-4V and we also evaluate Table-LLaVA on the same subset.

LLMs in Table 11. As shown in the table, OCR success rates vary a lot among 4 benchmarks, ranging from 11.05% to 75.35%. Intuitively, table images with large sizes (i.e. large Ave. Cell Numer) pose greater challenge to OCR engines and thus often lead to low OCR success rates. With OCR success rate decreasing, the performance gap of TableLlama between '+Oracle' and '+OCR' settings significantly increases, which reveals the importance of correct table recognition results. Moreover, compared with TableLlama, the performance gap of Llama2 between two settings is much more lower and less significant, which shows its bottleneck is the ability to understand and follow table-related instructions, rather than OCR results.

By manually inspecting the OCR results, we find that typical error types include (1) character-level mistakes, e.g., missing the first or last letter, (2) cell-level mistakes, e.g., missing whole cells, mistakenly splitting text in one cell into two cells, very wrong cell text especially for cells with long and intensive text, (3) row or column level mistakes, e.g., missing rows or inserting non-existing rows. (4) structure-level mistakes, e.g., falsely recognizing a merged cell as a non-merged cell or vice versa.

C.4 Case Study

We conduct a side-by-side qualitative analysis to compare Table-LLaVA with other (M)LLMs on different benchmarks, as illustrated in Figure 10-16. The results demonstrate that Table-LLaVA can handle a series of table tasks and possesses better multimodal table understanding ability than existing open-source MLLMs. For instance, as can be seen in Figure 10, Table-LLaVA provides both the intermediate reasoning steps and the correct final answer for the math word problem based on table image, whereas other MLLMs including GPT-4V fail to give the correct answer. The proposed MMTab dataset can be directly utilized in the training process of future MLLMs to boost their multimodal table understanding ability.

Method	LLM	Res.	TSD		TCE	TCL	MCD	RCE		TR		
			Row Acc.	Col. Acc.	Acc.	Acc.	F1	Row F1	Col. F1	HTML TEDS	Markdown TEDS	Latex TEDS
Ours (on all test samples)												
Table-LLaVA 7B	Vicuna-1.5 7B	336	33.10	**33.20**	19.45	29.31	**17.14**	**31.43**	37.93	50.24	44.82	46.11
Table-LLaVA 13B	Vicuna-1.5 13B	336	**34.40**	27.60	**19.53**	**29.68**	16.52	31.07	**41.49**	**51.44**	**46.00**	**46.50**
GPT-4V (on a subset of test samples)												
Low Resolution	GPT-4	512	6.00	24.00	3.57	14.41	2.12	**30.32**	**56.86**	41.55	45.74	34.46
High Resolution	GPT-4	768*2000	12.50	**46.00**	9.75	23.38	3.50	26.44	43.17	48.58	**60.58**	37.66
Ours (on a subset of test samples)												
Table-LLaVA 7B	Vicuna-1.5 7B	336	32.00	30.50	17.72	30.45	**18.44**	29.55	40.40	51.66	40.74	50.94
Table-LLaVA 13B	Vicuna-1.5 13B	336	**34.50**	26.00	**18.41**	**30.54**	15.88	29.87	42.88	**52.03**	41.65	**51.85**

Table 9: Comparison between GPT-4V and Table-LLaVA on 6 held-in table structure understanding benchmarks.

Method	AIT-QA	PubHealthTab	TabMCQ	TSD		TCE	TCL	RCE	
	Acc	Acc	Acc	Row Acc.	Col Acc.	Acc.	Acc.	Row F1.	Col. F1.
Previous Best	Vary-toy	Monkey	Monkey	LLaVA-1.5	mPLUG-Owl2	Monkey	LLaVA-1.5	Monkey	LLama2+OCR
	9.39	18.89	17.89	2.40	3.60	0.76	0.93	4.29	4.54
Ours									
Table-LLaVA 7B	5.48	**51.03**	44.51	25.20	**16.40**	11.28	26.10	**21.97**	18.14
Table-LLaVA 13B	6.06	48.46	**51.51**	**31.60**	14.80	**11.38**	**26.17**	21.94	**18.67**
GPT-4V (on a subset of test samples)									
Low Resolution	19.00	59.50	66.00	8.00	15.00	10.29	17.73	27.69	50.36
High Resolution	**62.50**	**67.00**	**66.00**	19.00	**38.00**	**14.36**	**27.91**	**48.52**	**57.14**
Ours (on a subset of test samples)									
Table-LLaVA 7B	5.00	52.50	43.50	22.00	16.00	12.73	26.27	16.57	13.91
Table-LLaVA 13B	6.50	53.50	45.50	**30.00**	15.00	11.92	25.45	20.77	13.78

Table 10: Evaluation results on 7 held-out tabular benchmarks.

OCR Accuracy	**TABMWP**	**TabFact**	**WTQ**	**HiTab**
Cell-level OCR Accuracy (%)	75.35	51.48	27.09	11.05
Table Size				
Ave. Row Number	6.45	14.40	26.4	23.38
Ave. Col Number	2.19	6.23	6.2	8.17
Ave. Cell Number (Row*Col)	14.13	83.71	163.68	191.01
Image Resolution (px)				
Ave. Width*Height	269*190	2354*875	1996*1137	3194*870
Methods				
Llama2+Oracle	17.88	4.32	4.26	1.21
Llama2+OCR	16.35	4.21	3.91	0.77
Gap between Oracle and OCR	1.53	0.11	0.35	0.44
TableLlama+Oracle	12.98	82.55	31.63	64.71
TableLlama+OCR	11.09	44.54	12.49	13.51
Gap between Oracle and OCR	1.89	38.01	19.14	51.20

Table 11: LLM performance on benchmarks with different OCR success rates.

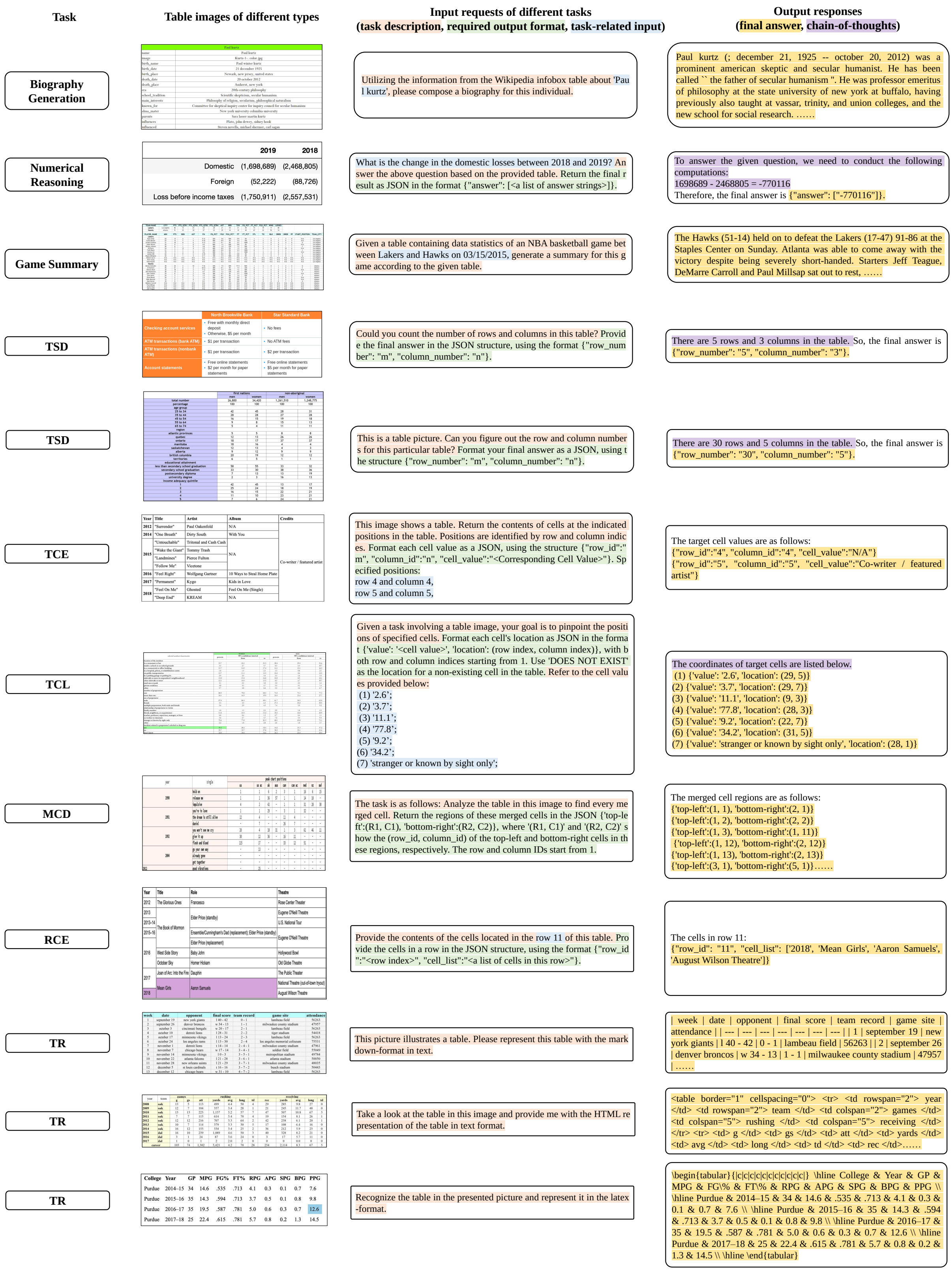

Figure 6: More dataset examples.

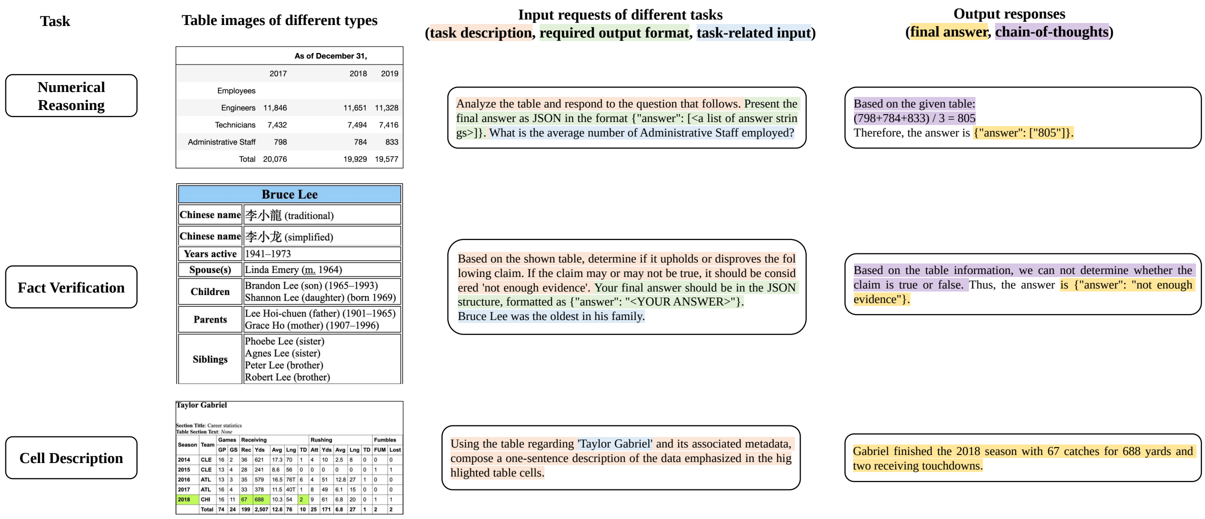

Figure 7: More dataset examples.

```python
# JSON_output_format_description_pool
JSON_output_instruction_list = [
    'Output the final answer as JSON in the format {"answer": "<YOUR ANSWER>"}.',
    'Conclude your response with a final answer in the JSON format {"answer": "<YOUR ANSWER>"}.',
    'Provide a concluding answer in a JSON structure, using the format {"answer": "<YOUR ANSWER>"}.',
    'The final result should be presented in the JSON format of {"answer": "<YOUR ANSWER>"}.',
    'The concluding answer should be in the JSON structure, formatted as {"answer": "<YOUR ANSWER>"}.',
    'Format the ultimate answer as a JSON, using the structure {"answer": "<YOUR ANSWER>"}.',
    'In the end, output your final answer using the JSON format: {"answer": "<YOUR ANSWER>"}.',
    'Present the final answer in a JSON format, outlined as {"answer": "<YOUR ANSWER>"}.',
    'Conclude your response with the final answer in the JSON format, structured as {"answer": "<YOUR ANSWER>"}.',
    'Finally, your final answer should be in the JSON format of {"answer": "<YOUR ANSWER>"}.',
    'In the last of your solution, output the final answer as JSON in the format {"answer": "<YOUR ANSWER>"}.',
    'At the end of your output, present the final answer as JSON in the format {"answer": "<YOUR ANSWER>"}.',
]
def build_TABMWP_input_request(table_title, question):
    # select one instruction describing the JSON output format
    JSON_INSTRUCTION = random.sample(JSON_output_instruction_list, 1)[0]
    # instruction_template_pool
    instruction_template_list = [
        f"Given the table about '{table_title}', solve the following math problem step by step. {JSON_INSTRUCTION}\n{question}",
        f"Refer to the provided table and work through the question step by step. {JSON_INSTRUCTION}\nTable title: {table_title}\nProblem: {question}",
        f"Using the displayed table concerning the '{table_title}', solve the subsequent math problem in a stepwise manner. {JSON_INSTRUCTION}\n\n{question}",
        f"Look at the table titled '{table_title}' and methodically tackle the math problem that follows. {JSON_INSTRUCTION}\n{question}",
        f"With the shown table image as your reference, carefully work out a detailed solution to the following question. {JSON_INSTRUCTION}\nTable title: {table_title}\nQuestion: {question}",
        f"Consider the table regarding to '{table_title}' to sequentially solve the problem presented below. {JSON_INSTRUCTION}\n\n{question}",
        f"Based on the table picture with the title '{table_title}', unfold the steps to solve the problem given next. {JSON_INSTRUCTION}\nProblem: {question}",
        f"With the table titled '{table_title}' in mind, please break down and resolve the question below step by step. {JSON_INSTRUCTION}\n\n{question}",
        f"Examine the table of '{table_title}' and proceed to solve the following math word problem in a stepwise manner. {JSON_INSTRUCTION}\n{question}",
        f"Using the table of '{table_title}', unfold the math word problem presented below, detailing every step of your calculation. {JSON_INSTRUCTION}\n{question}",
        f"Based on this table about '{table_title}', solve the following problem. {JSON_INSTRUCTION}\n{question}",
        f"Take a look at this table about '{table_title}', and tackle the math word problem below in a sequential manner. {JSON_INSTRUCTION}\n{question}",
        f"Considering the table of '{table_title}', answer the question below by showing each progressive step toward the answer. {JSON_INSTRUCTION}\n{question}",
        f"Check the table regarding to '{table_title}', and sequentially solve the math word problem, writing out each step of your reasoning process. {JSON_INSTRUCTION}\n{question}",
        f"Based on this table about '{table_title}', answer the following question in a stepwise manner. {JSON_INSTRUCTION}\n{question}",
        f"According to the table titled '{table_title}', solve this problem and give detail solutions. {JSON_INSTRUCTION}\n\n{question}",
        f"Solve the problem according to the provided table image. Please provide detailed solution. {JSON_INSTRUCTION}\nTable title: {table_title}\nProblem: {question}",
        f"This image shows a table of '{table_title}'. Solve the following math word problem based on the table.\n\nProblem: {question}\nLet's think step by step. {JSON_INSTRUCTION}",
        f"Table title: {table_title}\nMath word problem: {question}\nSolve the above problem based on the table information. Let's think step by step. {JSON_INSTRUCTION}",
        f"Table title: {table_title}\nQuestion: {question}\nGive a detailed response to the above question. {JSON_INSTRUCTION}",
        f"Table title: {table_title}\nQuestion: {question}\nSolve the above question. {JSON_INSTRUCTION}\nYour detailed solution: ",
        f"Based on this table of '{table_title}', answer the following question. Give detailed solution consisting of each step. {JSON_INSTRUCTION}\n{question}",
        f"Give you a table image, solve this math word problem based on the table. Let's think step by step. {JSON_INSTRUCTION}\n\nTable title: {table_title}\n\nProblem: {question}",
        f"Solve this math word problem according to the provided table of '{table_title}'. {JSON_INSTRUCTION}\n{question}",
        f"Show the detailed solution to solve the following problem. {JSON_INSTRUCTION} The problem is related to the given table titled '{table_title}'.\n\nProblem: {question}",
        f"Please solve the problem based on the given table about '{table_title}'. {JSON_INSTRUCTION} \n\nProblem: {question}\n\nYour Solution:",
        f"Problem: \n{question}\nSolve the above problem based on the table titled '{table_title}'. {JSON_INSTRUCTION}",
      ]
    # combine the randomly selected task description, output format description with task-related input (i.e., question) to obtain the final input request
    final_input_request = random.sample(instruction_template_list, 1)[0]
    return final_input_request
```

Figure 8: Exemplary instruction templates, JSON output format descriptions, and the Python Code for constructing the input requests. Taking the TABMWP dataset as an example.

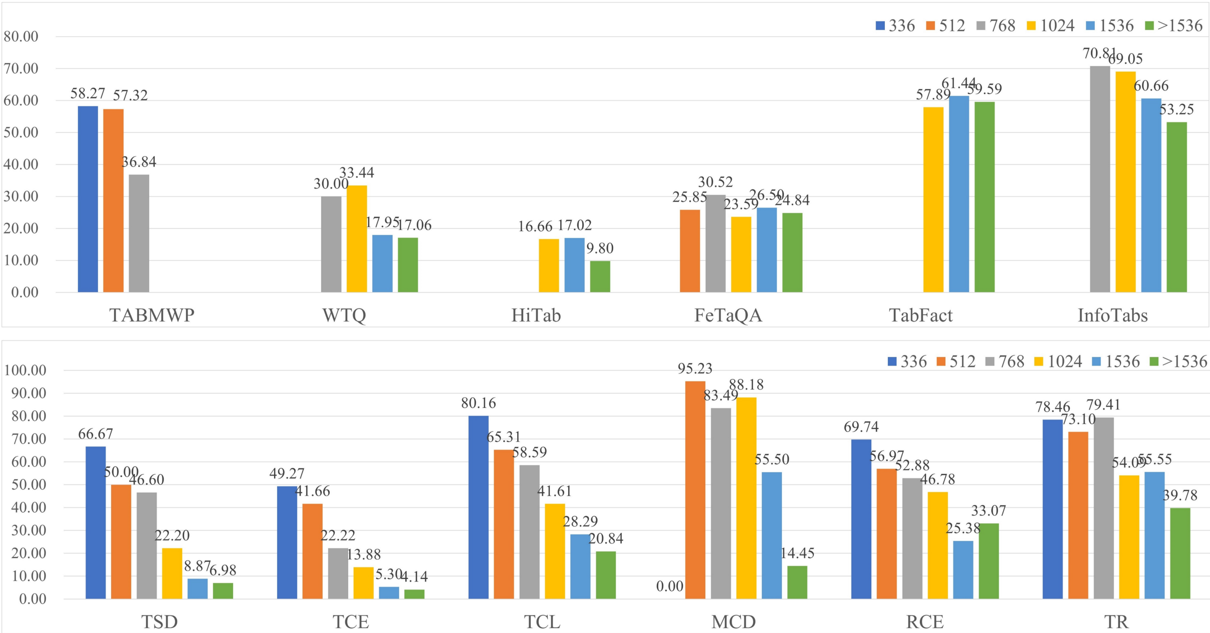

Figure 9: Break-down experimental results of Table-LLaVA 7B by different input image resolution. We divide test samples into 5 groups according to their image resolution, e.g., '512' represents the input image resolution is smaller than 512×512 but larger than 336×336. For TSD, MCD, RCE and TR, we report averaged results.

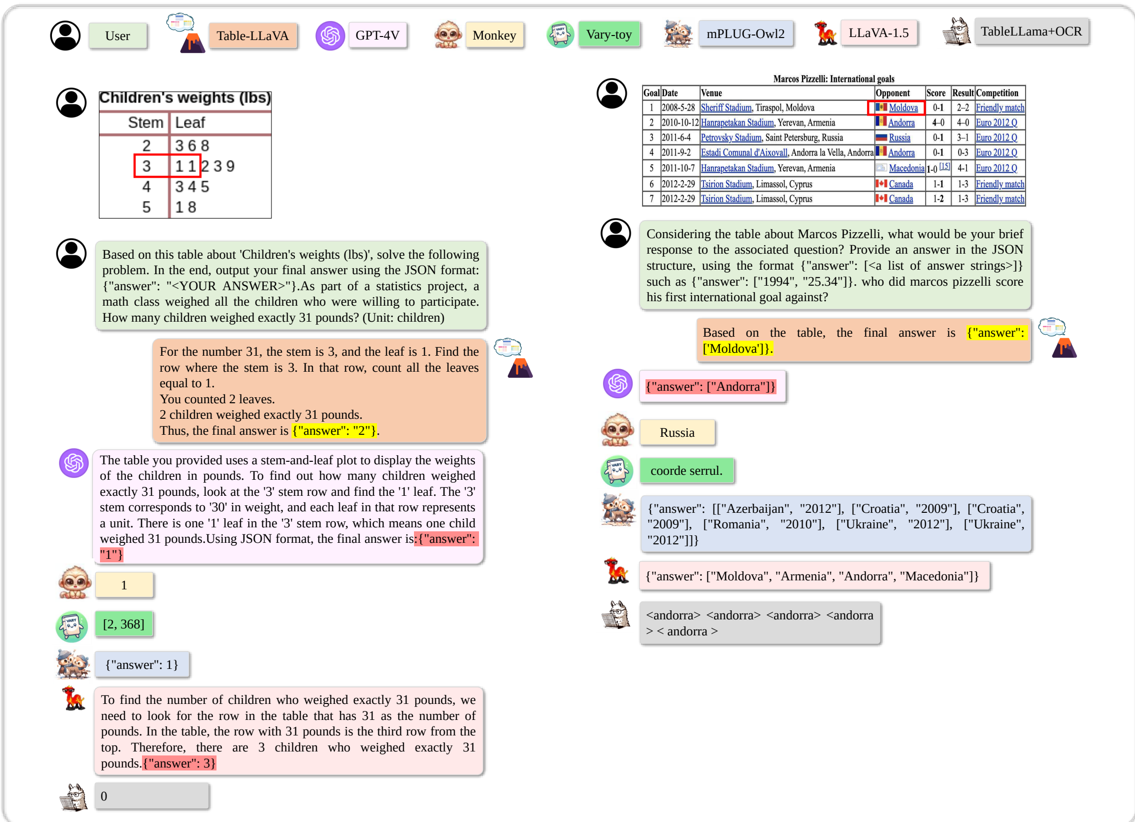

Figure 10: Case study on the TABMWP and WTQ benchmark. For the TABMWP benchmark, the model needs to conduct multi-step reasoning to obtain the final answer.

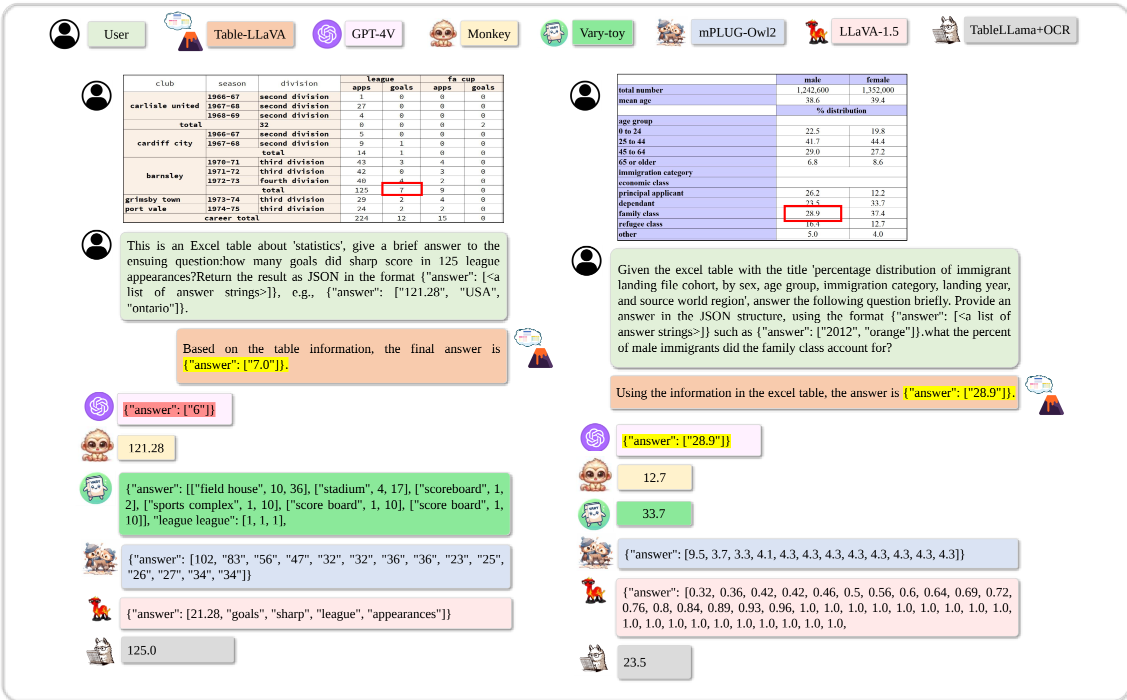

Figure 11: Case study on the HiTab benchmark, where the model is required to comprehend hierarchical tables with merged cells.

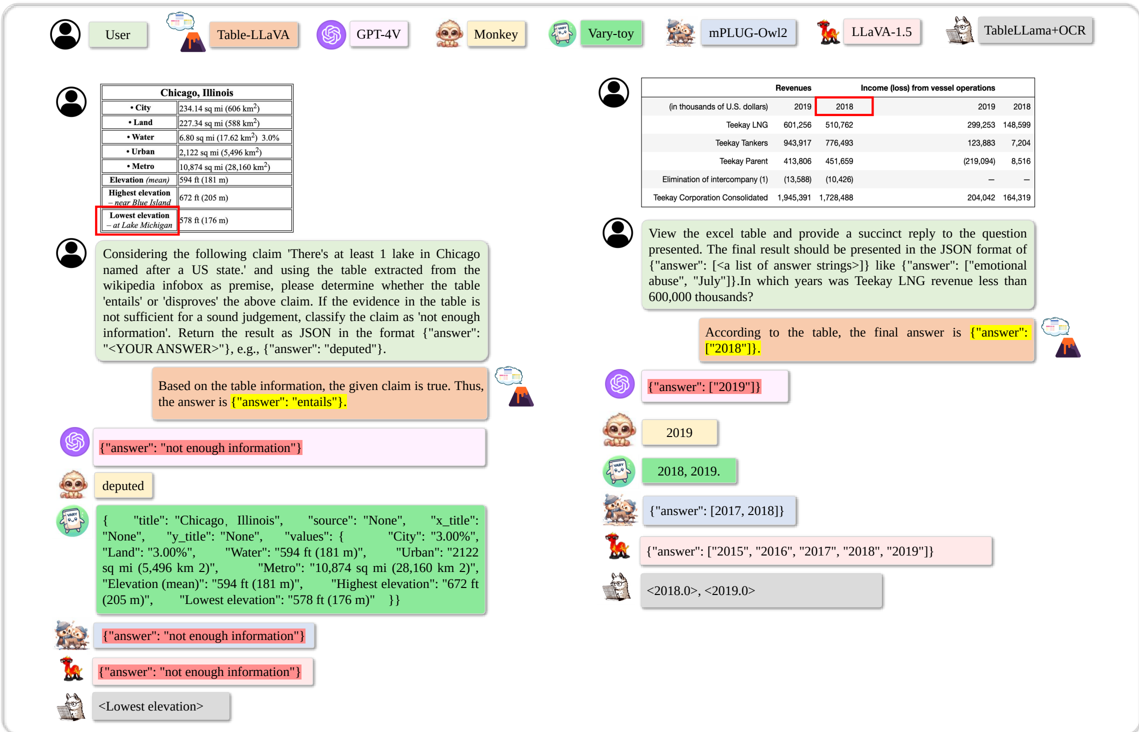

Figure 12: Case study on the InfoTab and TAT-QA benchmarks. Common table structures usually have the first row as column header, but table structures in IntoTab are quite different, where the first column contains row headers. Diverse table structures in MMTab pose unique challenge to existing MLLMs.

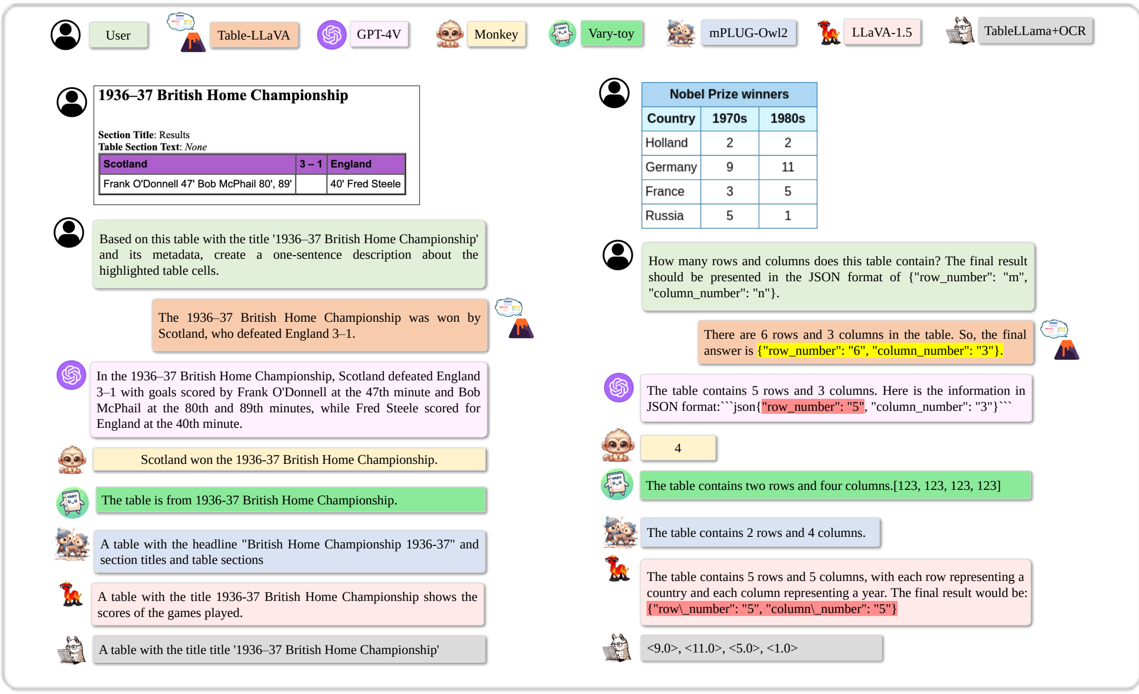

Figure 13: Case study on the ToTTo and TSD benchmarks. Though facing a relatively small and simple table, existing powerful MLLMs may fail to determine the row number and column number of the table. The basic ability to understand diverse table structures has been overlooked by previous MLLM study and the proposed MMTab alleviates this problem.

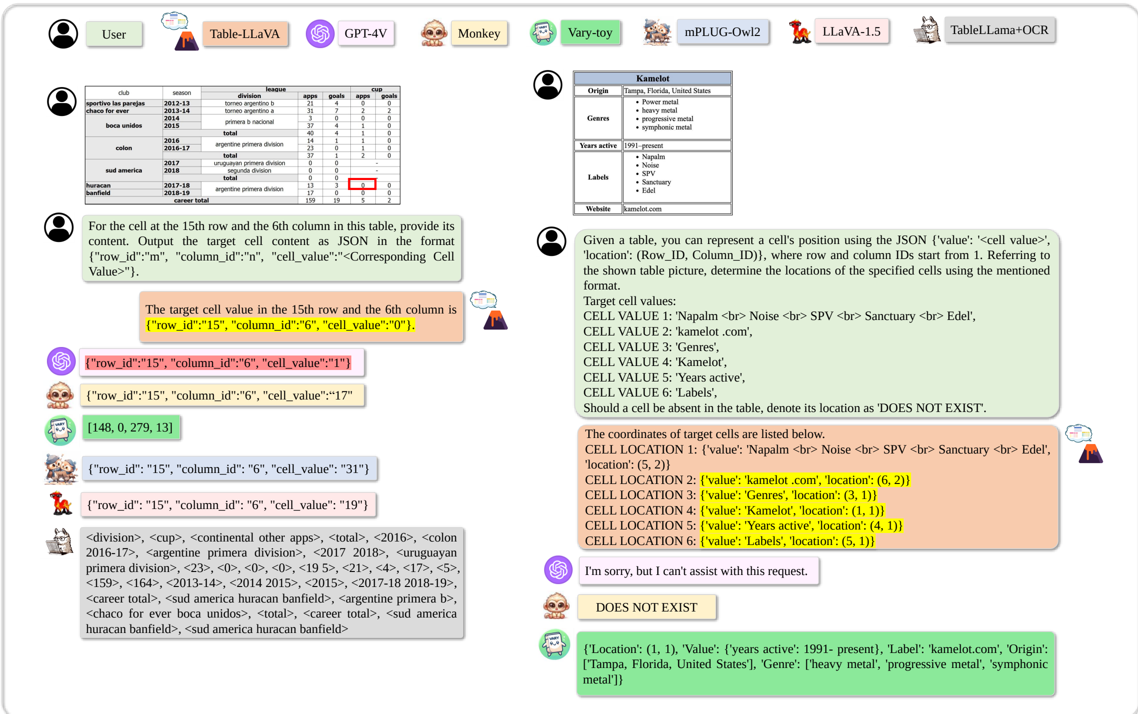

Figure 14: Case study on the TCE and TCL benchmarks, where the model is required to extract the target cell content or find the target cell location based on the table image. This task is trivial for human readers yet is challenging for existing MLLMs, which reveals the gap between current MLLMs and the human-level table understanding ability.

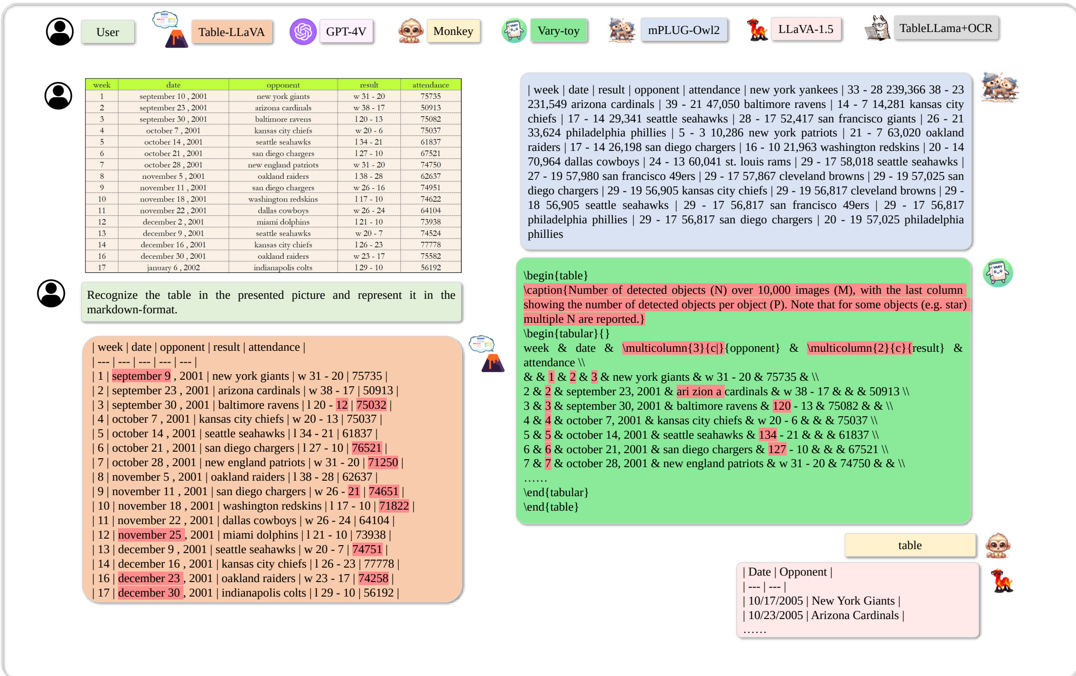

Figure 15: Case study on the TR (table recognition) benchmark. Though Table-LLaVA generate the markdown sequence with correct table structure, its OCR ability is restricted by the input resolution and needs to be enlarged. In the future work, the proposed MMTab dataset can be combined with MLLMs with stronger OCR ability and larger input image resolution like Monkey and LLaVA-Next to build more powerful tabular MLLMs.

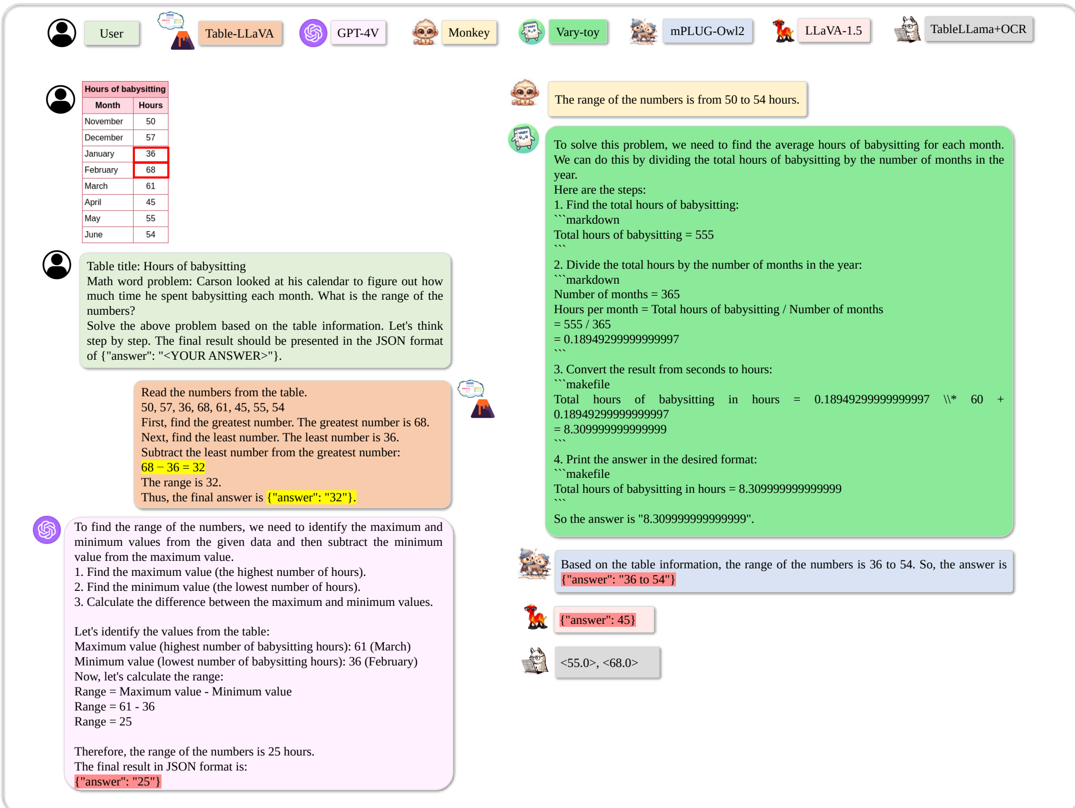

Figure 16: Case study on the TABMWP benchmark. In this case, the model needs to conduct table-based mathematical reasoning such as finding the largest number in the table or do math computations. Moreover, more external tools like Python Interpreter (Chen et al., 2023b) could be integrated with Table-LLaVA to build MLLM-based table agents.