What Really Matters for Table LLMs?
A Meta-Evaluation of Model and Data Effects

Naihao Deng[1*], **Sheng Zhang**[2], **Henghui Zhu**[3*], **Shuaichen Chang**[2], **Jiani Zhang**[5*], **Alexander Hanbo Li**[4*], **Chung-Wei Hang**[2], **Hideo Kobayashi**[2], **Yiqun Hu**[2], **Patrick Ng**[2*]
[1]University of Michigan [2]AWS AI Labs, New York, USA [3]Figma [4]OKX [5]Google
[1]dnaihao@umich.edu; [2]zshe@amazon.com; [3]hzhu@figma.com

Abstract

Table understanding has evolved over decades of NLP research. In this work, we revisit this trajectory and highlight emerging challenges in the LLM era, particularly the paradox of choice: the difficulty of attributing performance gains amid diverse base models and training sets in the context of table instruction tuning. We replicate four table LLMs by instruction-tuning three foundation models on four existing datasets, yielding 12 models. We then evaluate these models across 16 table benchmarks. Our study is the first to quantitatively disentangle the effects of training data and base model selection, revealing that base model choice plays a more dominant role than the training data itself. Generalization and reasoning remain challenging, inviting future effort on table modeling. We open-source our project at https://github.com/dnaihao/table-sft-eacl-2026. We open-source our replicated table LLMs at https://huggingface.co/collections/dnaihao/table-llms.

1 Introduction

Understanding semi-structured data, such as tables, has been a long-standing challenge in Natural Language Processing (NLP) (Woods, 1972; Warren and Pereira, 1982; Reiter et al., 2005; Pasupat and Liang, 2015; Yu et al., 2018b; Xie et al., 2022; Zhang et al., 2024a). Over the decades, the field has witnessed a series of paradigm shifts, from symbolic rule-based approaches to neural sequence models, to transformer-based architectures, and now to the era of Large Language Models (LLMs). Each shift has come with distinct characteristics and challenges. In this paper, we first offer a retrospective framing of these developments and identify the characteristics and challenges associated with table modeling for each era.

The past few years have witnessed a new era for table modeling, characterized by researchers employing instruction tuning for table-specific tasks, giving rise to a wave of "table LLMs" (Li et al., 2023; Zhang et al., 2024a,b; Zheng et al., 2024; Su et al., 2024; Deng and Mihalcea, 2025). In the meantime, while the long-standing challenges such as generalization (Warren and Pereira, 1982; Yu et al., 2018b; Suhr et al., 2020; Deng and Mihalcea, 2025) and reasoning (Liu et al., 2018; Xie et al., 2022; Wu et al., 2025a) still persist, a new challenge emerges, which we frame as "paradox of choice". Thanks to the numerous foundation LLMs (Touvron et al., 2023; Dubey et al., 2024; Jiang et al., 2023), and the diverse table datasets proposed (Cheng et al., 2022; Nan et al., 2022), these table LLMs vary widely in their base model selection, training data, and evaluation datasets. With so many moving parts, it has become increasingly difficult to attribute improvements to any one factor, raising concerns about reproducibility and comparability.

In this paper, we select four table LLMs and replicate them by training three distinct foundation LLMs on their proposed dataset, respectively. While such replication may appear incremental, it directly addresses a blind spot in current table LLM research. Previous studies (Zhang et al., 2024a,b; Zha et al., 2023) typically fix a single base model and attribute progress solely to new instruction data. However, our controlled experiments reveal that this assumption can be misleading: when the same datasets are applied across three different 7B-scale foundations (Mistral, OLMo, Phi), **base-model choice alone accounts for 81.6% of performance variance**, whereas **instruction data explains 13.8%**. We highlight that this finding turns an intuition long held by practitioners into **quantified, reproducible evidence**, offering the first cross-model, cross-dataset replication for table LLMs. As a side product, during the replication

*Work done during Naihao's internship at AWS

Findings of the Association for Computational Linguistics: EACL 2026, pages 3755–3782

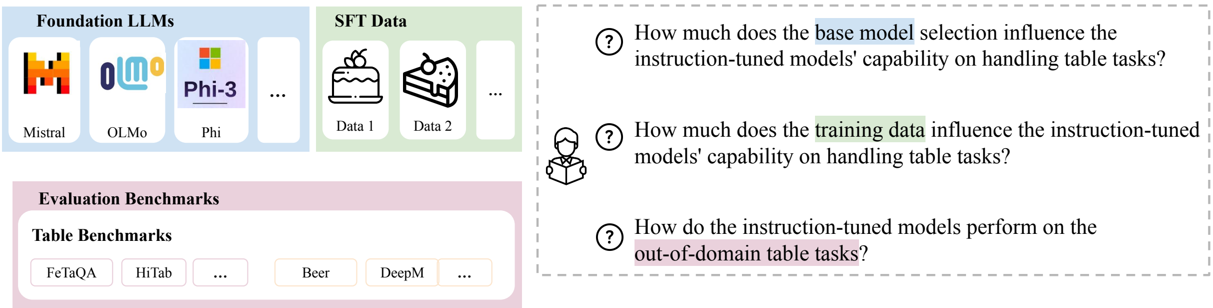


Figure 1: In this paper, we replicate four table LLMs by instruction-tuning three foundation models (OLMo (Groeneveld et al., 2024), Mistral (Jiang et al., 2023), and Phi (Abdin et al., 2024) models all at 7B scale) on four existing training datasets (TableGPT (Li et al., 2023), TableLlama (Zhang et al., 2024a), TableLLM (Zhang et al., 2024b), TableBench (Wu et al., 2025b)), yielding 12 models. We evaluate these models across 16 table benchmarks, trying to address the four research questions listed on the right.

process, we achieve a new state-of-the-art (SOTA) performance on the HiTab dataset.

Beyond benchmarking, our study introduces a new perspective: **disentangling performance sources** through a unified grid of 12 controlled instruction-tuned models. This framework enables Shapley–R^2 attribution analysis, which isolates the effect of foundation quality from that of training data, providing a clearer lens through which to interpret progress in table understanding. Our study (12 replications × 16 benchmarks) serves as a reminder for practitioners and researchers that scaling up evaluation breadth is just as critical as scaling up model size. It highlights the need for broader, standardized comparison practices.

In summary, our contributions are several-fold,

1. We replicate existing table-LLM setups by instruction-tuning three foundation models on four popular table instruction datasets, yielding 12 directly comparable models. To the best of our knowledge, this constitutes the first large-scale, cross-model replication in the context of table LLMs.

2. We conduct a comprehensive evaluation across 16 table benchmarks, spanning diverse table-related tasks and generalization scenarios. This unified evaluation grid provides a consistent basis for comparing instruction-tuning strategies across different foundations.

3. Our analysis highlights the dominant influence of base-model quality on performance — accounting for the majority of observed variance — while showing that current table LLMs continue to struggle with generalization and reasoning. Together, these findings serve as a reminder for practitioners to scale evaluation scope and methodological rigor.

2 Backgrounds and Related Works: Paradigm Shift in Table Modeling

Table-Related Tasks. There has been a long history of table-related tasks. Earlier work has focused on extracting table content from HTML (Chen et al., 2000; Tengli et al., 2004). The deep learning era has seen more diverse table-related tasks such as table question answering (table QA), the task of answering a question given the table and certain context in the format of multiple-choice (Jauhar et al., 2016) and free-form answer (Nan et al., 2022); table fact verification, the task of determining whether a given claim is supported or refuted by the table content (Chen et al., 2020b; Gupta et al., 2020); table-to-text, the task of generating a description given the table or some highlighted table cells (Parikh et al., 2020); text-to-SQL, the task of generating a SQL query given the table schema and a user query (Zhong et al., 2018; Yu et al., 2018b). These proposed benchmarks cover a diverse set of domains, including Wikipedia tables (Parikh et al., 2020), financial tables (Chen et al., 2021b), scientific tables (Moosavi et al., 2021), which serve as invaluable sources for developing and testing general table understanding models.

Paradigm Shift. Researchers have explored various methods for table understanding in the past decades, which can date back to the LUNAR system back in the 1970s (Woods, 1972). We briefly summarize the development of table models into four eras (Figure 2), where researchers develop rule-based (Woods, 1972; Warren and Pereira, 1982) and LSTM-based (Sutskever et al.,

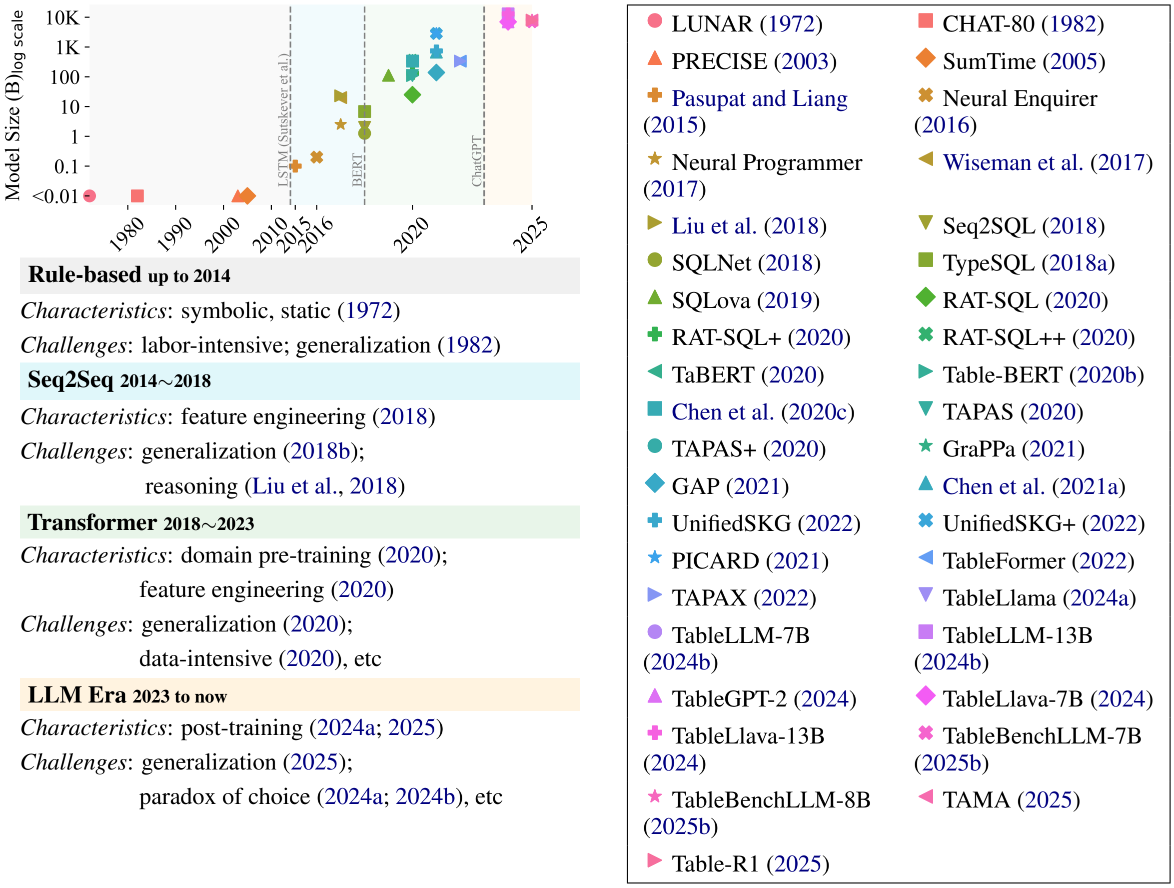


Figure 2: Summarization of different eras for table modeling. We note that the model sizes increase logarithmically with time. When we enter the LLM era, the community has shifted its attention to instruction tune the foundational models (Zhang et al., 2024a). While there are persistent challenges, such as generalization for table models (Warren and Pereira, 1982; Yu et al., 2018b; Suhr et al., 2020; Deng and Mihalcea, 2025) across different eras, new challenges emerge. Section A.1 provides additional details of this plot.

2014) algorithms (Zhong et al., 2018) in the earlier eras. With the rise of transformer (Vaswani et al., 2017) and the success of BERT (Devlin et al., 2019), researchers have started to adapt the transformer for table modeling (Herzig et al., 2020; Yin et al., 2020; Yu et al., 2021; Shi et al., 2021; Yang et al., 2022). With the success of LLMs (Ouyang et al., 2022), the community has shifted its focus on prompting-based methods (Chang and Fosler-Lussier, 2023; Deng et al., 2024)[1] as well as instruction tuning the base LLMs (Li et al., 2023; Zhang et al., 2024a; Zheng et al., 2024; Zhang et al., 2024b). Section A.2 provides additional discussion on the paradigm shifts.

3 Challenges in Table Modeling

There have been challenges for table models in different eras (Warren and Pereira, 1982; Yin et al., 2020). Here, we explain the three challenges we identify for the table LLM era.

Paradox of Choice. As we enter the LLM era, a new challenge emerges as the "paradox of choice", which refers to the difficulty of choosing from the diverse sets of foundation LLMs and training sets (Table 1). We have not seen such a challenge in the previous eras, even in the transformer era, researchers primarily base their models on the BERT model (Yin et al., 2020; Herzig et al., 2020), and fine-tune their models on a single dataset (Yu et al., 2018b; Wang et al., 2020). In contrast, the models in the LLM era adapt different base models (Zhang

[1]Since many of the prompting methods are model-agnostic, and we have no information on the model size of the commercial LLMs such as GPT-4, we do not include these methods in Figure 2.

Model	Base Model	Self-Created Training Data	Evaluation Benchmarks	Open Model?	Open Data?	Compare w. Other Table LLMs?	Train on Multiple Base LLM?
TableGPT (2023)	-	-	-	✗	✗	✗	✗
Table-GPT (2023)	GPT-3.5	✓	CTA (2022), WikiTQ (2015), ...	✗	✓	✗	✓
TableLlama (2024a)	LongLoRA †	✓	FeTaQA (2022), WikiTQ (2015), ...	✓	✓	✗	✗
TableLLM (2024b)	CodeLlama Instruct	✓	WikiTQ_m, TATQA_m, ...	✓	✓	✓	✗
TableBenchLLM (2025b)	Llama 3.1 & others	✓	TableBench (2025b)	✓	✓	✗	✓

Table 1: Information for current table instruction tuned models. †: a variant based on the Llama 2 model. We denote the evaluation datasets with a subscript "m" as they are adapted by Zhang et al. (2024b). We note that these table LLMs are trained from different base LLMs, and each uses its own instruction tuning data, and is tested on a different set of evaluation benchmarks.

et al., 2024a,b; Wu et al., 2025b), some instruction tune these models based on a mix of the existing benchmarks (Zhang et al., 2024a; Deng and Mihalcea, 2025), while others synthesize their training data (Li et al., 2023). Such diversified options make it hard to gauge the contributions of base models versus training data in the LLM era, and open up unanswered questions:

RQ1. How much does the base model selection influence the instruction-tuned models' capability on handling table tasks?

RQ2. How much does the training data influence the instruction-tuned models' capability on handling table tasks?

Generalization. While table LLMs demonstrate competitive performance (Zhang et al., 2024a), whether they pick up the table understanding capabilities or overfit to the dataset-specific patterns is still debatable (Deng and Mihalcea, 2025) and opens up a research question:

RQ3. How do the instruction-tuned models perform on the out-of-domain table tasks?

Section B provides additional discussion.

4 Experimental Setups

Because of the limited computing resources and non-trivial computational costs to train and test LLMs, we cannot exhaust all possible evaluations.

Model Selection. To rigorously study the influences of base model selection and training data, we select three LLMs that are all released in the year of 2023 and 2024 from non-profit organizations or companies, Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), OLMo 7B Instruct (Groeneveld et al., 2024) and Phi 3 Small Instruct (7B) (Abdin et al., 2024) as our base models, detailed in Section C.

Replication. For each base model, we replicate the instruction tuning stage for TableLlama (Zhang et al., 2024a), TableLLM (Zhang et al., 2024b), TableBenchLLM (Wu et al., 2025b), and Table-GPT (Li et al., 2023). Our implementation yields comparable or better results than the performance reported in the existing works (Figure 3, additional details in Section E).

Evaluation. We select eight real-world table understanding datasets, eight synthetic table understanding datasets (details in Section D) for our evaluation. We note that our controlled replication enables an apples-to-apples comparison and allows us to disentangle the respective contributions of base model capabilities and instruction tuning datasets, therefore better answering the research questions we propose in Section 3 (Figure 1).

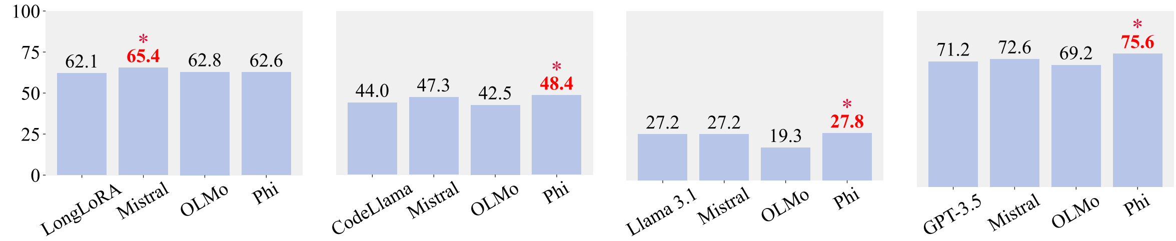
5 Results and Discussions

Figure 3 presents the averaged in-domain (ID) performance. Table 2 presents the out-of-domain (OOD) evaluation on various table understanding benchmarks.

RQ1: How much does the base model selection influence the instruction-tuned models' capability on handling table tasks?

Answer: Large OOD performance variance across base models. Contrary to performance in Figure 3, where we see minimal ID performance variance across different base models, there is a large performance variance across different base models on the OOD table tasks, as shown in Table 2. For instance, when all trained on TableBenchLLM, Phi achieves 83.0 on TabMWP, significantly outperforming Mistral (70.6) and OLMo (62.6).

The base model is crucial, and in some cases, a determinant factor for the OOD performance. In Figure 4, we employ the Shapley R^2 decomposition to decompose the performance contributions of



(a) TableLlama. (b) TableLLM. (c) TableBench. (d) Table-GPT.

Figure 3: Averaged in-domain performance (y-axis) between the models in existing works (the leftmost bar for each plot) versus our replications. Our replicated models achieve better in-domain results than the existing works. The detailed in-domain performance is reported in Section E.

Train Data	*Real*								*Synthesized*							
	Table QA					**Fact Veri.**		**Tab2Text**	**Schema Reasoning**							**Misc.**
	FeT	HiT	TabM	TAT	Wiki	TabF	Inf	ToT	Beer	DeepM	DI	ED	C	CF	CTA	$\text{TabB}_{\text{eval}}$
	BLEU	Acc	Acc	Acc	Acc	Acc	Acc	BLEU	F1	Recall	Acc	F1	F1	Acc	F1	ROUGE-L
Mistral v0.3 7B Instruct																
👑 N/A	**20.0**	35.5	66.9	18.0	27.9	**62.3**	**42.8**	11.5	**97.2**	42.9	27.9	24.1	**30.2**	19.1	63.8	18.9
TableLlama	38.7	70.6	71.2	5.6	23.8	86.8	27.7	**28.5**	25.8	70.0	13.4	25.1	17.4	0.5	34.9	**19.6**
👑 TableLLM	10.2	**44.1**	**75.0**	25.0	32.3	11.9	15.4	6.7	45.0	**78.6**	**33.1**	**43.1**	25.6	15.0	66.9	3.7
TableBench	7.9	**44.1**	70.6	**25.7**	**37.4**	36.5	27.5	3.5	88.5	50.0	32.0	20.3	27.4	13.3	**72.2**	27.2
TableGPT	19.5	35.8	62.2	14.1	25.5	61.4	35.8	4.5	100.0	98.0	46.4	46.0	23.8	**25.3**	68.3	13.1
OLMo 7B Instruct																
N/A	6.0	27.3	54.4	14.3	19.4	38.2	21.4	5.1	**50.5**	35.7	28.9	14.1	15.0	16.2	54.5	7.6
TableLlama	36.8	67.9	**72.9**	9.9	6.7	83.8	15.0	**20.8**	0.0	7.1	21.2	14.6	14.8	10.7	23.5	**17.1**
👑 TableLLM	**9.7**	**35.5**	65.5	**17.7**	26.7	40.6	16.9	8.9	16.5	**42.9**	33.0	**37.6**	13.0	18.7	43.6	6.3
TableBench	3.8	28.3	62.6	15.6	**34.0**	30.9	6.5	7.5	43.4	16.6	**36.6**	28.6	18.1	21.2	46.5	19.3
👑 TableGPT	9.3	27.2	65.6	14.6	16.4	**44.9**	**33.0**	11.4	96.2	100.0	45.4	35.3	**19.9**	**29.3**	**62.5**	13.7
Phi 3 Small Instruct (7B)																
N/A	5.0	39.6	76.1	13.0	29.7	65.3	62.3	1.4	95.0	42.9	31.9	49.7	30.6	43.4	71.5	8.3
TableLlama	38.1	63.6	74.8	18.3	**46.3**	86.2	54.3	**29.6**	**95.6**	35.7	4.3	19.4	27.9	36.5	43.9	22.4
👑 TableLLM	18.2	**45.3**	81.2	**24.1**	37.7	69.6	44.6	8.1	80.2	**50.0**	34.0	41.3	27.9	**49.5**	70.1	**27.2**
👑 TableBench	10.0	3.5	**83.0**	20.5	34.6	68.0	65.3	0.9	95.0	28.6	**35.9**	**53.8**	**31.1**	46.2	**76.7**	27.8
TableGPT	**24.8**	45.1	76.8	15.6	30.0	**71.0**	**67.0**	14.0	98.9	98.8	49.4	55.4	24.8	45.2	68.3	26.1

Table 2: Evaluation for table tasks. Gray indicates that the model is trained on the corresponding training set. Bolded numbers represent the best performance among variants of the same base model, while red is the best overall performance across all models. Mistral v0.3 7B Instruct, OLMo 7B Instruct, and Phi 3 Small Instruct (7B) indicate the base model on which we apply the training data, respectively. "👑" marks the model that has the most number of top performance across all the datasets with respect to the same base model. We note that Phi-based models yield the highest performance scores across most of the out-of-domain table datasets, while TableLLM training data consistently yield the most top performance across different base LLMs.

the base LLM selection versus the different instruction tuning data (additional details in Section F.1). We find that the base LLMs' selection holds an R^2 of 0.816, significantly larger than 0.138, the share of the instruction tuning data. The share for the base LLM selection remains crucial when we consider model pairs in Figure 8 in Section F.1, suggesting that the base model selection is a non-negligible, and sometimes a dominant factor that determines the instruction-tuned model's performance. However, existing works for table instruction tuning (Li et al., 2023; Zhang et al., 2024a,b; Su et al., 2024) barely provide such comparison studies, and typically train their models from a single base LLM, ignoring the crucial factor of base model selection.

Strong base model leads to significantly better OOD performance. In Figure 5, we plot the Pearson r scores for the instruction-tuned model's

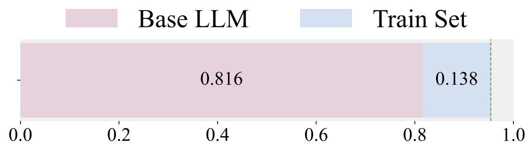


Figure 4: Shapley R^2 decomposition (Shapley et al., 1953; Israeli, 2007) for the contributions of the downstream tasks' performance by the base LLM versus the training set. We can see that the choice of the base LLM is a dominant factor (0.816 compared to 0.138 from the train set) that decides the model's performance on downstream tasks. Figure 8 provides the additional analysis for pair-wise base model comparisons.

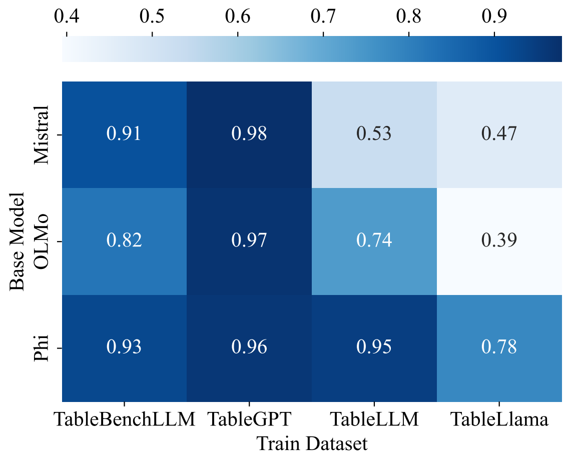


Figure 5: Pearson r scores for the fine-tuned model's performance v.s. the base model's performance on the OOD datasets. We find that in general, there is a strong linear correlation between the two performances, with a Pearson r of around 0.7 to 0.9. Even the lowest Pearson r score, 0.39, indicates a moderate positive correlation.

performance v.s. the base model's performance on the out-of-domain datasets. In general, there is a strong linear correlation between the two performances (Pearson r around 0.7 to 0.9), suggesting that the instruction-tuned model's performance is strongly related to the base model's performance on these table tasks. We notice that in Table 2, the best performance for a single dataset is typically achieved by fine-tuning the Phi model. We note that the Phi model consistently outperforms the other two models even when untuned. For instance, TabMWP's overall best performance is achieved by fine-tuning the Phi model with the TableBench training data, and the original Phi model achieves 76.1, outperforming the original Mistral's 66.9 and the original OLMo's 54.4. TATQA's overall best performance is achieved by fine-tuning the Mistral model with TableBench training data, and the original Mistral model achieves 18.0, outperforming the

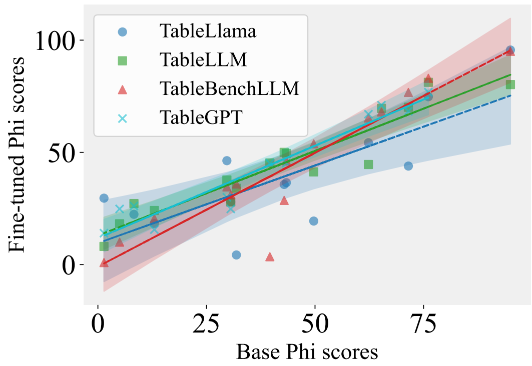


Figure 6: Fine-tuned models' performance (y-axis) with respect to each training dataset v.s. the base Phi model's performance (x-axis) on the OOD table datasets. We find that there is a linear correlation (Pearson r ranges from 0.78 to 0.96) between these two scores.

original OLMo's 14.3 and the original Phi's 13.0. This suggests that while instruction tuning can meaningfully improve a model's performance on table tasks, its effectiveness is still heavily bound by the capabilities of the underlying base model.

RQ2. How much does the training data influence the instruction-tuned models' capability in handling table tasks?

Answer: Instruction tuning yields a significant performance boost for ID datasets. When the dataset is included as part of the training set (e.g., FeTaQA in TableLlama), we observe a significant performance boost compared to the untrained base model (Mistral trained on TableLlama data achieves 38.7 compared to the base's 20.0 on FeTaQA). This echoes with the finding by Zhang et al. (2024a); Deng and Mihalcea (2025) that instruction tuning can significantly boost the ID performance.

Certain training data consistently yield the best OOD performance across different base LLMs. Though in Figure 8, compared to the base LLM selection, the influence of the existing training data remains small in most cases, there is still a linear relation between the training set selection and the instruction-tuned model's performance, as illustrated in Figure 6 In addition, we notice that TableLLM's training data consistently achieves the best (e.g., on HiTab or competitive performance on table QA tasks across all three base models in Table 2. In contrast, though the recipe for TableLlama's training data contains table QA tasks, models trained with the training data from TableLlama underperform those from TableLLM. We attribute the effectiveness of TableLLM's training data on

	Error Types	Description	Example
▶	*Grounding Error*	Fail to properly attend to the correct information.	: Find the column that contains the cell value "348.55". | . . . | BalanceLeftTD | Current Month | . . . | | . . . | 48796.94 | 348.55 | . . . | : BalanceLeftTD
▶	*Math Reasoning Error*	Fail to conduct the math reasoning process correctly.	: . . . the Soviet Union received 29 medals, while East Germany received 25 medals. Therefore, the Soviet Union did not receive 4 more medals than East Germany. . .
▶	*Not Following Instructions*	Generate output while not following the instruction.	: . . . Let's think step by step and show your reasoning before showing the final result . . . : Answer: No
▶	*Hallucination*	Fabricate ungrounded details or facts.	(In the table, Canada has 3 bronze medals; Switzerland has 5.) : . . . According to the table, Switzerland (SUI) and Canada (CAN) both received 3 bronze medals . . .
▶	*Commonsense Errors*	Generate outputs that violate common sense.	: . . . release date is November 11, 2008. However, it does not provide any information about the season in which it was released. Therefore, . . .

Table 3: Types of reasoning errors commonly made by tableLLMs, with their description and example erroneous responses ( ) to questions ( ) from our experiment results on the Phi model trained on TableLLM data.

the table QA task to that when constructing the data, Zhang et al. (2024b) leverage LLMs such as GPT-3.5 to enhance the reasoning process (more in Section F.2). Such an enhanced reasoning path would benefit the model's reasoning process, as suggested by the findings by Guo et al. (2025); Muennighoff et al. (2025).

RQ3. How do the instruction-tuned models perform on the OOD table tasks?

Answer: The best OOD performance is significantly below the ID performance. As shown in Table 2, though there are improvements from the base models on the OOD table tasks, the models' performance is far below that of the ID tuned models. For instance, for the Phi model, if the training set includes HiTab, the model achieves 63.6 (the gray value in Table 2), while the best OOD performance on HiTab is 45.3 (achieved by training the Phi model using TableLLM's training set). Such a large performance gap suggests a large space for improvement.

The instruction-tuned model may yield worse performance than the base model. We note that instruction tuning sometimes leads to decreased OOD performance compared to the base model. For instance, the untuned Mistral model achieves a score of 27.9 on WikiTQ, whereas instruction-tuning it on TableGPT data reduces performance to 25.5. This highlights a potential trade-off introduced by instruction tuning. While it improves alignment on in-domain tasks, it may also cause the model to overfit or overspecialize, leading to reduced generalization on unseen tasks.

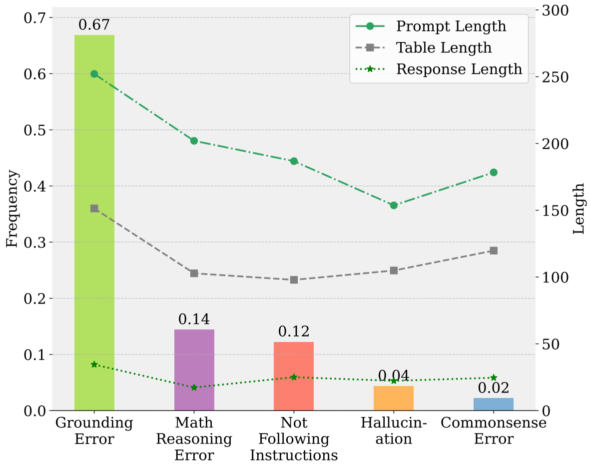


Figure 7: Frequencies of the TableLLM's answers containing the five reasoning errors, and the corresponding prompt, table, and response length.

Instruction-tuned models are still prone to grounding and numerical reasoning errors. We conduct an error analysis on 1,000 samples predicted by the Phi model fine-tuned on TableLLM data to understand its error cases. Representative error cases and their distribution are shown in Table 3 and Figure 7, respectively. We find that grounding errors of failing to correctly associate the question with the relevant table content, are the most frequent, particularly in examples involving longer tables or prompts. This suggests that instruction tuning alone may be insufficient to develop robust table grounding capabilities, highlighting the need for future work focused on improving models' alignment with tabular inputs. In addition, models frequently struggle with basic numerical reasoning, such as subtraction over table entries. This suggests a persistent limitation in integrating arithmetic operations in the context of table understanding. Moreover, we observe instruction-following failures in certain cases, aligning with prior findings that further instruction tuning may degrade the base model's inherent capabilities (Wang et al., 2023). While hallucinations and commonsense errors also occur, they are relatively less frequent in table-based tasks compared to general benchmarks (Clark et al., 2018; Rein et al., 2023).

Additional RQs and Discussions. In addition, we explore several auxiliary research questions and describe our findings here. First, we find that table instruction tuning does not necessarily compromise general capabilities (Section F.3). Second, scaling model size yields diminishing returns under fixed data and compute budgets — larger models improve performance but at significant computational cost (Section F.4). Third, when compared to proprietary LLMs, the best-performing open models (e.g., Phi-based variants) narrow the gap on structured reasoning benchmarks, underscoring the progress of open LLMs (Section G.2). Finally, we include results on additional datasets that confirm the same trends: base model quality remains the dominant explanatory factor across both real and synthetic table tasks (Section G.3).

6 Take-Aways

Base Model Selection Matters Most. Our analysis confirms that base model choice is the single most influential factor driving downstream performance. As shown in Figures 4 and 5, instruction-tuned performance correlates strongly with the base model's intrinsic capability, with base-model factors explaining over 80% of the observed variance.

Instruction Tuning Helps—but Its Impact Is Often Overstated. While instruction tuning substantially improves in-domain performance, its impact on out-of-domain generalization remains limited. Thus, much of the leaderboard gain observed in prior work likely reflects the strength of the underlying foundation model rather than improvements from the new instruction data.

Persistent Weaknesses in Generalization and Reasoning. Despite overall gains, table LLMs continue to exhibit fragile reasoning behaviors, especially in grounding (linking textual queries to table content) and numerical reasoning (handling arithmetic or comparison). Models tend to show sharp performance drops under domain shift.

Toward More Reproducible Evaluation. Taken together, these findings highlight the need for systematic, cross-model evaluation frameworks that can disentangle data, model, and training-driven effects. Scaling evaluation breadth, not just model size, will be essential for ensuring that future progress in table modeling is interpretable, reproducible, and scientifically grounded.

We provide discussions on the future directions for table modeling in Section G.1.

7 Conclusion

In this paper, we revisit the trajectory of table instruction tuning and re-examine what truly drives progress in table understanding. By systematically replicating four popular table LLM setups across three distinct foundation models, we disentangle the intertwined effects of base-model quality and instruction data. Our large-scale experiments, covering twelve instruction-tuned models and sixteen evaluation benchmarks, provide the first controlled analysis that quantifies their relative contributions. The results reveal a striking imbalance: while instruction data plays a meaningful role, base model selection alone explains over eighty percent of performance variance. This finding turns an intuitive belief among practitioners into reproducible evidence and calls for a rethinking of how progress in table modeling is measured.

In revisiting the foundations of table instruction tuning, we highlight that what appears as empirical progress often reflects underlying disparities in model strength. Recognizing and quantifying these effects is not merely diagnostic—it is essential for building a reliable science of table understanding. Future efforts should extend this controlled evaluation framework to broader model scales and modalities, establishing reproducibility as a central norm in table-LLM research.

Limitations

We believe our work presents the first of its kind large-scale controlled analysis that explicitly decouples the effects of base model and instruction tuning data in the table understanding domain. In addition, we want to stress the massive training effort we have invested in, as noted in Section 4. As a side product, we have achieved the new SOTA performance on the HiTab dataset, and provide the first open-source model replication of existing closed-source table LLMs such as Table-GPT. Moreover, we have comprehensively evaluated these twelve models on 16 table understanding benchmarks.

However, there exist other base models, or other datasets proposed by the researchers, which can be used to train the table LLMs and evaluate these models' capabilities, and by no means can we exhaust all of them in this paper. We encourage future efforts in comprehensively evaluating these table LLMs' capabilities, and we believe our work has laid a solid foundation for decoupling the contributions of training data and base models, and further enhancing our understanding of table instruction tuning.

While our study provides the first systematic comparison across multiple table LLMs, we acknowledge that only three base models were examined. Although our results provide strong empirical evidence, further validation across larger models remains valuable." We expect our study to remind practitioners of

Given the computational cost, we limited our replication to three representative 7 B-scale models. Although this covers a diverse spectrum (open, instruction-tuned, and academic), we have no way to obtain results from proprietary LLMs (e.g., commercially closed-source LLMs). We provide additional discussions on comparing these instruction-tuned table LLMs to proprietary LLMs in Section G.2.

Ethical Considerations

In this work, we isolate the contributions of training data proposed by the existing table LLMs by training the same base models and comparing their performance. The base models we have used in this work include Mistral v0.3 7B Instruct model (Jiang et al., 2023), OLMo 7B Instruct (Groeneveld et al., 2024), and Phi 3 Small Instruct (7B) (Abdin et al., 2024). We conduct additional studies on Phi 3 Mini Instruct (4B) in Section F. Foundational models like Mistral v0.3 7B Instruct model are susceptible to jail-breaking instructions (Wei et al., 2024) and may lead to harmful behaviors. Our objective in this work is to understand the limitations of the existing table instruction tuning, and we urge practitioners to stick to the good purpose when developing or using our models. Our replicated models can serve as baseline models for future research on structured data, and we provide a holistic evaluation of these models on both table tasks and how they compromise their general capabilities. Our results lead to various findings on what training data helps the models most on these table tasks, and how to construct LLMs specialized in tables efficiently.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Ziawasch Abedjan, John Morcos, Ihab F Ilyas, Mourad Ouzzani, Paolo Papotti, and Michael Stonebraker. 2016. Dataxformer: A robust transformation discovery system. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 1134–1145. IEEE.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Michael J Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment*, 1(1):538–549.

Shuaichen Chang and Eric Fosler-Lussier. 2023. How to prompt llms for text-to-sql: A study in zero-shot, single-domain, and cross-domain settings. *arXiv preprint arXiv:2305.11853*.

Hsin-Hsi Chen, Shih-Chung Tsai, and Jin-He Tsai. 2000. Mining tables from large scale HTML texts. In *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Wang, and William W Cohen. 2020a. Open question answering over tables and text. *arXiv preprint arXiv:2010.10439*.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W. Cohen. 2021a. Open question answering over tables and text. In *International Conference on Learning Representations*.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020b. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020c. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021b. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, et al. 2021c. Finqa: A dataset of numerical reasoning over financial data. *arXiv preprint arXiv:2109.00122*.

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.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Naihao Deng and Rada Mihalcea. 2025. Rethinking table instruction tuning.

Naihao Deng, Zhenjie Sun, Ruiqi He, Aman Sikka, Yulong Chen, Lin Ma, Yue Zhang, and Rada Mihalcea. 2024. Tables as texts or images: Evaluating the table reasoning ability of LLMs and MLLMs. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 407–426, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Mihail Eric and Christopher D Manning. 2017. Key-value retrieval networks for task-oriented dialogue. *arXiv preprint arXiv:1705.05414*.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. 2024. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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.

Jeff Hawkins. 2021. *A thousand brains: a new theory of intelligence*. Basic Books.

Yeye He, Kris Ganjam, and Xu Chu. 2015. Semajoin: joining semantically-related tables using big table corpora. *Proceedings of the VLDB Endowment*, 8(12):1358–1369.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. 2023. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Wonseok Hwang, Jinyeong Yim, Seunghyun Park, and Minjoon Seo. 2019. A comprehensive exploration on wikisql with table-aware word contextualization. *arXiv preprint arXiv:1902.01069*.

Osnat Israeli. 2007. A shapley-based decomposition of the r-square of a linear regression. *The Journal of Economic Inequality*, 5:199–212.

Sujay Kumar Jauhar, Peter Turney, and Eduard Hovy. 2016. Tabmcq: A dataset of general knowledge tables and multiple-choice questions. *arXiv preprint arXiv:1602.03960*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Pei Ke, Bosi Wen, Andrew Feng, Xiao Liu, Xuanyu Lei, Jiale Cheng, Shengyuan Wang, Aohan Zeng, Yuxiao Dong, Hongning Wang, Jie Tang, and Minlie Huang. 2024. CritiqueLLM: Towards an informative critique generation model for evaluation of large language model generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13034–13054, Bangkok, Thailand. Association for Computational Linguistics.

Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024. Revisiting meta-evaluation for grammatical error correction. *Transactions of the Association for Computational Linguistics*, 12:837–855.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2023. Table-gpt: Table-tuned gpt for diverse table tasks. *arXiv preprint arXiv:2310.09263*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

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*.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2022. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

Nafise Sadat Moosavi, Andreas Rücklé, Dan Roth, and Iryna Gurevych. 2021. Scigen: a dataset for reasoning-aware text generation from scientific tables. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

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, Dragomir Radev, and Dragomir Radev. 2022. FeTaQA: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

Arvind Neelakantan, Quoc V. Le, Martin Abadi, Andrew McCallum, and Dario Amodei. 2017. Learning a natural language interface with neural programmer. In *International Conference on Learning Representations*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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.

Ana-Maria Popescu, Oren Etzioni, and Henry Kautz. 2003. Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 149–157.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167(1-2):137–169.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9895–9901, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Lloyd S Shapley et al. 1953. A value for n-person games.

Peng Shi, Patrick Ng, Zhiguo Wang, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Cicero Nogueira dos Santos, and Bing Xiang. 2021. Learning contextual representations for semantic parsing with generation-augmented pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13806–13814.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE.

Aofeng Su, Aowen Wang, Chao Ye, Chen Zhou, Ga Zhang, Guangcheng Zhu, Haobo Wang, Haokai Xu, Hao Chen, Haoze Li, et al. 2024. Tablegpt2: A large multimodal model with tabular data integration. *arXiv preprint arXiv:2411.02059*.

Alane Suhr, Ming-Wei Chang, Peter Shaw, and Kenton Lee. 2020. Exploring unexplored generalization challenges for cross-database semantic parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8372–8388, Online. Association for Computational Linguistics.

Zhenjie Sun, Naihao Deng, Haofei Yu, and Jiaxuan You. 2025. Table as thought: Exploring structured thoughts in llm reasoning. *arXiv preprint arXiv:2501.02152*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Ashwin Tengli, Yiming Yang, and Nian Li Ma. 2004. Learning table extraction from examples. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 987–993, Geneva, Switzerland. COLING.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jaime Raldua Veuthey, Zainab Ali Majid, Suhas Hariharan, and Jacob Haimes. 2025. Meqa: A meta-evaluation framework for question & answer llm benchmarks. *arXiv preprint arXiv:2504.14039*.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7567–7578, Online. Association for Computational Linguistics.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. 2023. How far can camels go? exploring the state of instruction tuning on open resources. *Advances in Neural Information Processing Systems*, 36:74764–74786.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024a. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Preprint*, arXiv:2406.01574.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. 2024b. Chain-of-table: Evolving tables in the reasoning chain for table understanding. In *The Twelfth International Conference on Learning Representations*.

David H.D. Warren and Fernando C.N. Pereira. 1982. An efficient easily adaptable system for interpreting natural language queries. *American Journal of Computational Linguistics*, 8(3-4):110–122.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2024. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

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.

William Woods. 1972. The lunar sciences natural language information system. *BBN report*.

Jian Wu, Linyi Yang, Dongyuan Li, Yuliang Ji, Manabu Okumura, and Yue Zhang. 2025a. MMQA: Evaluating LLMs with multi-table multi-hop complex questions. In *The Thirteenth International Conference on Learning Representations*.

Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xeron Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, et al. 2025b. Tablebench: A comprehensive and complex benchmark for table question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25497–25506.

Zhenhe Wu, Jian Yang, Jiaheng Liu, Xianjie Wu, Changzai Pan, Jie Zhang, Yu Zhao, Shuangyong Song, Yongxiang Li, and Zhoujun Li. 2025c. Table-r1: Region-based reinforcement learning for table understanding. *arXiv preprint arXiv:2505.12415*.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 602–631, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Junjie Xing, Yeye He, Mengyu Zhou, Haoyu Dong, Shi Han, Lingjiao Chen, Dongmei Zhang, Surajit Chaudhuri, and HV Jagadish. 2025. Mmtu: A massive multi-task table understanding and reasoning benchmark. *arXiv preprint arXiv:2506.05587*.

Xiaojun Xu, Chang Liu, and Dawn Song. 2018. SQLNet: Generating structured queries from natural language without reinforcement learning.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. 2022. TableFormer: Robust transformer modeling for table-text encoding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 528–537, Dublin, Ireland. Association for Computational Linguistics.

Zheyuan Yang, Lyuhao Chen, Arman Cohan, and Yilun Zhao. 2025. Table-r1: Inference-time scaling for table reasoning. *arXiv preprint arXiv:2505.23621*.

Pengcheng Yin, Zhengdong Lu, Hang Li, and Kao Ben. 2016. Neural enquirer: Learning to query tables in natural language. In *Proceedings of the Workshop on Human-Computer Question Answering*, pages 29–35, San Diego, California. Association for Computational Linguistics.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.

Tao Yu, Zifan Li, Zilin Zhang, Rui Zhang, and Dragomir Radev. 2018a. TypeSQL: Knowledge-based type-aware neural text-to-SQL generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 588–594, New Orleans, Louisiana. Association for Computational Linguistics.

Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, bailin wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, richard socher, and Caiming Xiong. 2021. Gra{pp}a: Grammar-augmented pre-training for table semantic parsing. In *International Conference on Learning Representations*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018b. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Liangyu Zha, Junlin Zhou, Liyao Li, Rui Wang, Qingyi Huang, Saisai Yang, Jing Yuan, Changbao Su, Xiang Li, Aofeng Su, et al. 2023. Tablegpt: Towards unifying tables, nature language and commands into one gpt. *arXiv preprint arXiv:2307.08674*.

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 (Volume 1: Long Papers)*, pages 6024–6044, Mexico City, Mexico. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 9102–9124, Bangkok, Thailand. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Seq2SQL: Generating structured queries from natural language using reinforcement learning.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

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 Backgrounds and Related Works: Paradigm Shift in Table Modeling

A.1 Captions of Figure 2

In Figure 2, we use "+" and "++" to denote different sizes of the same model. For instance, TAPAS (Herzig et al., 2020) refers to the model based on the small version of the BERT-base model, while TAPAS+ refers to the model based on the large version of the BERT-base model. For the LSTM models such as Liu et al. (2018)'s model, we estimate the parameter sizes based on the description in the original paper.

A.2 Different Eras for Table Modeling

Here we provide further discussions on different eras for table modeling.

Rule-Based and Seq2Seq Era. The first era is characterized by the symbolic and static nature of the proposed algorithms (Woods, 1972; Warren and Pereira, 1982). Later, with the rise of LSTM in NLP (Sutskever et al., 2014), researchers have incorporated domain-specific features into the models, such as specific components to generate SQL queries to query database tables (Zhong et al., 2018).

Transformer Era. The earlier trend of domain-specific feature engineering from seq2seq era has made its way into the transformer era, where the pre-trained transformer models (Vaswani et al., 2017), such as BERT (Devlin et al., 2019) have taken over most fields in NLP. Herzig et al. (2020) incorporate embeddings designed for rows and columns, Yang et al. (2022) adapt the attention mechanism to better align with table structures. In addition, this era has witnessed a trend of domain-specific pre-training, where researchers collect a large table pre-training corpus (Yin et al., 2020) and design table-specific training objectives (Yu et al., 2021; Shi et al., 2021).

LLM Era. Ever since the successful launch of the ChatGPT system (Ouyang et al., 2022), researchers have increasingly focused on adapting LLMs for table tasks. As LLMs have inherent abilities on table understanding, researchers employ prompt engineering on these LLMs for better performance on tables (Chang and Fosler-Lussier, 2023; Deng et al., 2024)[2]. Another line of re-

[2]Since many of the prompting methods are model-agnostic, and we have no information on the model size of the commer-

search involves instruction-tuning LLMs by adapting existing table-related benchmarks. This leads to various table LLMs such as Table-GPT (Li et al., 2023), TableLlama (Zhang et al., 2024a), TableLlava (Zheng et al., 2024), and TableLLM (Zhang et al., 2024b). Recently, with the rise of reasoning models (Guo et al., 2025), researchers have leveraged reinforcement learning algorithms such as GRPO (Shao et al., 2024) for table modeling (Yang et al., 2025; Wu et al., 2025c). In this paper, we specifically focus on table instruction tuning, a popular post-training paradigm that aligns large language models with structured reasoning tasks over tabular data.

Remarks. We have seen continuous efforts that last several decades where researchers adapt general modeling methods to the domain of table understanding. As a result, much like the trend in the general language models, there has been a logarithmic increase in terms of the table model size in the past decades (Figure 2). While these models have kept pushing the state-of-the-art performance on many benchmarks (Zhang et al., 2024a), the monotonic increase in model sizes is concerning as it limits the access for many research labs where there is no abundant computing resources.

B Challenges in Table Modeling

In the rule-based era, crafting the rules can be labor-intensive (Warren and Pereira, 1982); in the transformer era, crafting a large-scale pre-training corpus is data-intensive (Yin et al., 2020). In addition to the discussion in Section 3, here we further discuss the generalization.

Generalization. The challenge of generalization has shifted across eras. Since the rules in earlier systems are hand-crafted and static, the challenge lies primarily in handling the cases where their rules do not cover (Warren and Pereira, 1982). Such problems are mediated with the appearance of the learning-based models (e.g., LSTM, transformers), where the models may have a chance to conduct compositional reasoning to generalize to unseen examples (Zhong et al., 2018). However, an LSTM model excelled on one domain may fail on other domains (Yu et al., 2018b). This persists in the transformer era, where models perform well on one dataset demonstrate near-zero performance on others (Suhr et al., 2020). While in the table LLM era, there seem to be some promises on generalization to unseen tasks (Zhang et al., 2024a), in our paper, we reveal that generalization challenges remain.

cial LLMs such as GPT-4, we do not include these methods in Figure 2.

C Experimental Setup

Foundation LLM Selections. For the training data from each existing work, we fine-tune Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), OLMo 7B Instruct (Groeneveld et al., 2024) and Phi 3 Small Instruct (7B) (Abdin et al., 2024). Following Zhang et al. (2024a,b); Wu et al. (2025b), we fine-tune all the models through full parameter fine-tuning.

Hyperparameter Selection. To rule out the effects of the learning rate, we train all three models using a set of learning rates: 5e-5, 1e-5, 5e-6, 1e-6, 5e-7, 1e-7, 5e-8, and 1e-8. Empirically, we find that they achieve the best when the learning rate is 5e-7. We do not see significant performance changes as we increase the training steps. For consistency, we fine-tune our models for three epochs across all the experiments.

We run our experiments on 1 server node with 8 A100, each with 48 GB GPU memory. We set the batch size to 16 in our training process.

D Evaluation Setups

D.1 Real-World Table Understanding Benchmarks.

Dataset Description. We evaluate our replicated models on eight existing real-world datasets covering the tasks of table question answering (table QA), table fact verification, and table-to-text generation. **FeTaQA (FeT)** (Nan et al., 2022) is a free-form table QA dataset sourced from Wikipedia-based tables. **HiTab (HiT)** (Cheng et al., 2022) is a table QA dataset sourced from statistical reports and Wikipedia pages on hierarchical tables. **TabMWP (TabM)** (Lu et al., 2022) is an open-domain grade-level table question-answering dataset involving mathematical reasoning. **TATQA (TAT)** (Zhu et al., 2021) is a table QA dataset sourced from real-world financial reports. **WikiTQ (Wiki)** (Pasupat and Liang, 2015) is a table QA dataset sourced from Wikipedia. **TabFact (TabF)** (Chen et al., 2020b) is a table fact verification dataset sourced from Wikipedia. **InfoTabs (Inf)** (Gupta et al., 2020) is a table fact verification dataset with human-written textual hypotheses based on tables extracted from Wikipedia infoboxes. **ToTTo (ToT)** (Parikh et al., 2020) is a table-to-text dataset sourced from Wikipedia tables.

Metrics. For FeTaQA, we use the BLEU4 score following Nan et al. (2022). For ToTTo, we follow Xie et al. (2022) to report the BLEU4 scores over multiple references. We adopt the evaluation script from the original HiTab, TabMWP, TATQA, and WikiTQ repository on GitHub. For these table QA tasks, we notice that since the fine-tuned models may not follow instructions such as "generate in the JSON format", we do not pose any constraints to these models in terms of the generation format. Instead, we use Haiku 3.5[3] to extract the answer entity from the model generation. For TabFact and InfoTabs, we report the accuracy by checking if only the gold answer appears in the prediction.

Data Format. In terms of the test set format, we use the exact same test set for FeTaQA, HiTab, TATQA, and ToTTo as Zhang et al. (2024a) with the Markdown table format. For TabMWP, WikiTQ, and InfoTabs, etc., we follow the original data format. Specifically, TabMWP uses '|' to separate columns, and WikiTQ and InfoTabs use HTML format to represent tables.

D.2 Synthetic Table Understanding Datasets.

In addition, we evaluate these models on eight synthesized datasets including **Beer**, **DeepM**, **Spreadsheet-DI (DI)**, **Spreadsheet-Real (ED)**, **Column-No-Separator (C)**, **Spreadsheet-CF (CF)**, and **Efthymiou (CTA)** (Li et al., 2023) on schema reasoning ability (detailed in our replication for Table-GPT Section E.4), and **TabB$_{\text{eval}}$** (Wu et al., 2025b) on miscellaneous table tasks.

Section H provides examples for these datasets.

E Replicating Existing Table LLMs

Table 1 outlines the base models used in existing table LLMs. These base models, ranging from various Llama models to closed-source models such as GPT-3.5, differ significantly in their architecture designs, model sizes, and training recipes. In addition, each table LLM introduces its own unique training data, making it challenging to disentangle the impact of the training data from that of the base model. Here we report the performance of our fine-tuned models based on Mistral v0.3 7B Instruct, OLMo 7B Instruct, and Phi 3 Small Instruct (7B) versus the original models on the datasets reported in each of the original works.

E.1 Replicating TableLlama

Training Datasets. The original TableLlama (Zhang et al., 2024a) uses 2 million data points in its instruction tuning stage, which can be unnecessarily large. In addition, we do not have enough computing resources to instruction-tune our model on a dataset of such a scale. Therefore, we rule out the table operation datasets and only maintain the training data for FeTaQA (Nan et al., 2022), HiTab (Cheng et al., 2022), and TabFact (Chen et al., 2020b) to fine-tune our model, which results in 107K training instances.

Evaluation Datasets. Following Zhang et al. (2024a), we use the FeTaQA (Nan et al., 2022), HiTab (Cheng et al., 2022), and TabFact (Chen et al., 2020b) as the in-domain evaluation sets. In addition, we compare our fine-tuned models versus the original TableLlama on FEVEROUS (Aly et al., 2021), HybridQA (Chen et al., 2020c), KVRET (Eric and Manning, 2017), ToTTo (Parikh et al., 2020), WikiSQL (Zhong et al., 2018), and WikiTQ (Pasupat and Liang, 2015).

Comparison. Table 4 compares the original TableLlama model (first row) versus our fine-tuned models. Our fine-tuned models yield similar or better performance than the original TableLlama model in most cases. In addition, we achieve the new SOTA performance on HiTab by fine-tuning the Mistral model. As we only use 107K (5% of the 2M data points used by the original TableLlama), our results demonstrate that *with proper instruction-tuning, we can achieve competitive results on table tasks with much fewer data.*

E.2 Replicating TableLLM

Training Datasets. We use the original instruction-tuning set by Zhang et al. (2024b), which includes 80.5K training instances.

Evaluation Datasets. Following Zhang et al. (2024b), we use the modified version of WikiTQ (Pasupat and Liang, 2015), TATQA (Zhu et al., 2021), and FeTaQA (Nan et al., 2022) as the in-domain evaluation sets, and OTT-QA (Chen et al., 2020a) as the out-of-domain evaluation set.

Comparison. Table 5 compares the original TableLLM versus our fine-tuned models. We note

[3]https://www.anthropic.com/claude/haiku

Base Models	FeTaQA (BLEU)	HiTab (Acc)	TabFact (Acc)	FEVEROUS (Acc)	HybridQA (Acc)	KVRET ($F1_{Micro}$)	ToTTo (BLEU)	WikiSQL (Acc)	WikiTQ (Acc)
Original (Zhang et al., 2024a)									
LongLoRA 7B‡	**39.0**	64.7	82.5	73.8	**39.4**	<u>48.7</u>	20.8	50.5	35.0
Ours									
Mistral v0.3 7B Instruct	<u>38.7</u>	**70.6**†	**86.8**	<u>75.9</u>	27.2	46.6	<u>28.5</u>	**64.5**	<u>47.4</u>
OLMo 7B Instruct	36.8	<u>67.9</u>	83.8	69.8	20.3	44.6	20.8	56.9	38.8
Phi 3 Small Instruct (7B)	38.1	63.6	<u>86.2</u>	**78.3**	<u>33.6</u>	**56.0**	**29.6**	<u>63.3</u>	**47.7**

Table 4: Performance comparison between the original TableLlama and our fine-tuned models from different model families on the in-domain tuned (left three columns) and out-of-domain (right six columns) datasets. The number is bold if it is the best among the four, and underscored if it is the second. †: we surpass the previous SOTA performance (64.7 by TableLlama) on HiTab.

Base Models	WikiTQ_m (Acc_p)	TATQA_m (Acc_p)	FeTaQA_m (BLEU)	OTT-QA_m (Acc_p)
Original (Zhang et al., 2024b)				
CodeLlama‡	72.5	51.1	8.4	57.3
Ours				
Mistral	**76.0**	<u>55.4</u>	<u>10.6</u>	**64.3**
OLMo	66.8	50.2	10.5	58.1
Phi	<u>75.4</u>	**57.8**	**12.1**	<u>63.3</u>

Table 5: Performance comparison between the original TableLLM and our fine-tuned models. All four models are 7B and instruction-tuned. We denote the evaluation datasets with a subscript "m" as they are adapted by Zhang et al. (2024b).

Base Models	$\text{TableBench}_\text{eval}$ (R-L)
Original (Wu et al., 2025b)	
Llama 3.1 8B ‡	<u>27.2</u>
Ours	
Mistral v0.3 7B Instruct	<u>27.2</u>
OLMo 7B Instruct	19.3
Phi 3 Small Instruct (7B)	**27.8**

Table 6: Performance comparison between the original TablebBenchLLM based on Llama 3.1 8B and our fine-tuned models. "R-L" denotes the ROUGE-L score.

that our evaluation metrics are distinct from what Zhang et al. (2024b) have used originally. Zhang et al. (2024b) use CritiqueLLM (Ke et al., 2024) as a judge to decide the correctness of the answers. However, the model judgments are made in Chinese[4], a different language from the language in all the training and evaluation datasets. In addition, the scores assigned by the CritiqueLLM is not consistent for a single evaluation example. Therefore, for WikiTQ_m, TATQA_m, and OTT-QA_m, we report the Acc_p scores, where we calculate whether the gold answer entities appear in the model's response. We find that our fine-tuned models based on the Mistral and Phi models consistently outperform the original TableLLM model on these datasets, and we attribute the performance improvement to the stronger base model (Mistral v0.3 7B Instruct and Phi 3 Small Instruct) we have versus theirs (CodeLlama 7B Instruct).

[4] Zhang et al. (2024b)'s inference results are available at `https://github.com/RUCKBReasoning/TableLLM/blob/main/inference/results/TableLLM-7b/Grade_fetaqa.jsonl`

E.3 Replicating TableBenchLLM

Training Datasets. We use the original instruction-tuning set by Wu et al. (2025b), which includes 20K training instances.

Evaluation Datasets. Following Wu et al. (2025b), we only evaluate the model on their constructed test set, which we denote as $\text{TableBench}_\text{eval}$ in Table 6.

Comparison. Following Wu et al. (2025b), we report the ROUGE-L score of our Mistral-TableBenchLLM. In Table 6, we compare our model with the scores reported by Wu et al. (2025b) in the original paper, corresponding to the version of TableBenchLLM fine-tuned based on Llama 3.1 8B model. Our Mistral-TableBenchLLM and Phi-TableBenchLLM achieve similar performance scores of 27.2 and 27.8, respectively, compared to the original TableBenchLLM's 27.2.

E.4 Replicating Table-GPT

Training Dataset. We use the instruction-tuning dataset provided by Li et al. (2023) that contains 66K instances.

Base Models	Beer (F1)	DeepM (Recall)	DI (Acc)	ED (F1)	C (F1)	CF (Acc)	Wiki (Acc)	CTA (F1)
Original (Li et al., 2023)								
GPT-3.5‡	72.7	**100.0**	**55.8**	**56.5**	29.4	**71.3**	**48.6**	**88.6**
Ours								
Mistral	**100.0**	98.0	46.4	46.0	23.8	25.3	25.5	68.3
OLMo	96.2	**100.0**	45.4	35.3	19.9	29.3	16.4	62.5
Phi	98.9	98.8	49.4	55.4	24.8	45.2	30.0	68.3

Table 7: Performance comparison between the original Table-GPT and our fine-tuned models.

	Beer (F1)	DeepM (Recall)	DI (Acc)	ED (F1)	C (F1)	CF (Acc)	Wiki (Acc)	CTA (F1)
13K	98.9	92.9	45.9	43.8	**29.4**	21.2	29.2	66.8
66K	**100.0**	**98.0**	**46.4**	**46.0**	23.8	**25.3**	**29.8**	**68.3**

Table 8: Performance comparison between training Mistral v0.3 7B Instruct on 13K instances versus 66K instances provided by Li et al. (2023).

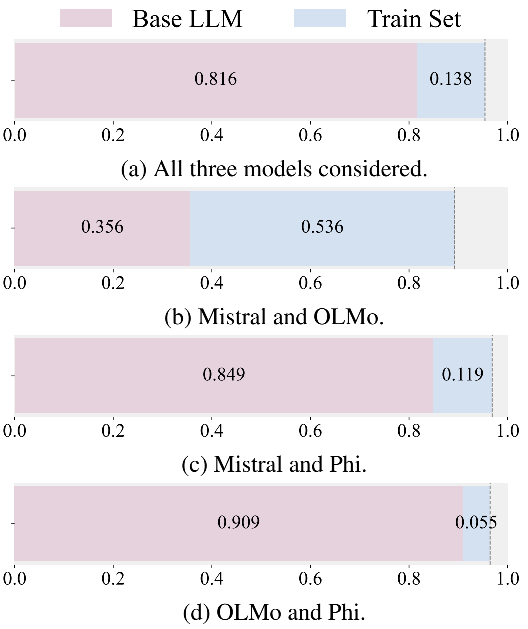


Figure 8: Shapley R^2 decomposition (Shapley et al., 1953; Israeli, 2007) for the contributions of the downstream tasks' performance by the base LLM versus the training set. We can see that the choice of the base LLM is a non-negligible factor, and in many cases, the dominant factor that decides the model's performance on downstream tasks.

Evaluation Datasets. We select four in-domain test sets by Li et al. (2023), Beer for entity matching, DeepM for schema matching, Spreadsheet-DI (DI) for data imputation, and Spreadsheet-Real (ED) for error detection. Furthermore, we report the out-of-domain performance on Column-No-Separator (C) for missing value identification, Spreadsheet-CF (CF) for column finding, WikiTQ (Wiki) for table question answering, and Efthymiou (CTA) for column type annotation.

Comparison. Table 7 reports the results. We note that though the size of our fine-tuned models are all 7B, they achieve better performance than Table-GPT which is based on GPT-3.5 on Beer, and comparable performance on DeepM. However, on the out-of-domain datasets, we can see that Mistral-TableGPT underperforms the original Table-GPT. We attribute such performance differences to the differences between the base models. Since GPT-3.5 is stronger than these open-source 7B models, its innate table understanding ability, as well as its generalization ability, leads to better performance on these out-of-domain table datasets for Table-GPT. This reinforces our motivations of conducting the comparisons using the same base model, as *the performance difference may be because of the base model's capability*, therefore, we need the same base model to conduct an apple-to-apple comparison.

Side Findings. There is a smaller training set provided by Li et al. (2023) containing 13K training instances. We report the performance comparison by training the Mistral v0.3 7B Instruct model on the two sets in Table 8 We do not find a significant performance boost when we use the larger 66K dataset. And on one of the out-of-domain datasets, C, training on 13K instances even yields a better score of 29.4 than training on 66K instances' 23.8. This echoes with the findings by Zhou et al. (2024); Deng and Mihalcea (2025) that limited instruction tuning instances are able to yield a strong model.

F Results and Discussions

F.1 Shapley R^2 Decomposition

To quantify the relative contributions of (i) the base LLM and (ii) the instruction-tuning dataset to downstream table-understanding performance, we apply a Shapley R^2 decomposition. Our implementation follows the standard model-agnostic formulation for linear regression and corresponds directly to our released code.

Setup. For each of the 12 instruction-tuned models, we compute the average performance across the OOD datasets, yielding a target vector

$$y \in \mathbb{R}^{12}.$$

We consider two categorical predictors:

$$\text{Base model} \in \{\text{Mistral-7B}, \text{OLMo-7B}, \text{Phi-3-7B}\},$$

$$\text{Train set} \in \{\text{TableLlama}, \text{TableLLM}, \text{TableBenchLLM}, \text{Table-GPT}\}.$$

Both variables are one-hot encoded, producing the design matrix

$$X = \left[X_{\text{Base}},\ X_{\text{TrainSet}}\right].$$

Shapley R^2 for Two Feature Groups.

For two predictor groups, the Shapley value for group i is

$$\phi_i = \frac{1}{2}\left(R^2(\{i\}) - R^2(\varnothing)\right) + \frac{1}{2}\left(R^2(\{\text{Base}, \text{TrainSet}\}) - R^2(\{j\})\right),$$

where $i \neq j$, and $R^2(S)$ denotes the coefficient of determination of a linear regression model using only the feature subset S.

Concretely, $R^2(\{\text{Base}\})$ uses only base-model indicators, $R^2(\{\text{TrainSet}\})$ uses only instruction-tuning dataset indicators, and $R^2(\{\text{Base}, \text{TrainSet}\})$ is the full model R^2.

Permutation-Based Formulation. More generally, our implementation follows the full permutation-based Shapley formulation:

$$\phi_i = \frac{1}{|\Pi|} \sum_{\pi \in \Pi} \left(R^2\left(S_i^\pi \cup \{i\}\right) - R^2\left(S_i^\pi\right)\right),$$

where Π is the set of all permutations of {Base, TrainSet}, and S_i^π is the set of feature groups that appear before i in permutation π. With two feature groups, there are exactly two permutations (Base $\rightarrow$ TrainSet and TrainSet $\rightarrow$ Base), and we average marginal contributions across both.

Implementation Details. We compute R^2 using ordinary least squares via `sklearn`'s `LinearRegression`. Categorical predictors are one-hot encoded, and Shapley values are accumulated across permutations and normalized such that

$$\phi_{\text{Base}} + \phi_{\text{TrainSet}} = R^2_{\text{Full Model}}.$$

Interpretation. The Shapley values quantify how much variance in table-understanding performance is attributable to (i) base model identity versus (ii) the instruction-tuning dataset. Empirically, we obtain:

$$\phi_{\text{Base}} = 0.816, \qquad \phi_{\text{TrainSet}} = 0.138.$$

These results indicate that base model choice overwhelmingly dominates the contribution of instruction-tuning data in determining the performance of table-tuned LLMs.

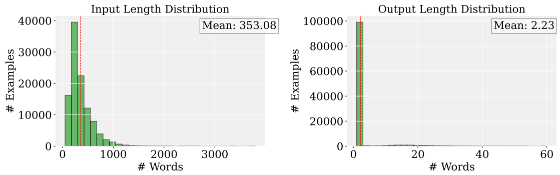
Results and Discussions. Figure 8 provides the Shapley R^2 results for the three models as well as for each pair of models. We note that when we consider model pairs, base model selection is a dominant factor that decides the instruction-tuned models' performance for Mistral and Phi, OLMo and Phi. For models fine-tuned from Mistral and OLMo, base model selection still explains 35.6% of the performance variance. This suggests that the base model selection is a crucial, and in many cases, a dominant factor that determines the instruction-tuned model's performance.

F.2 Training Data Example

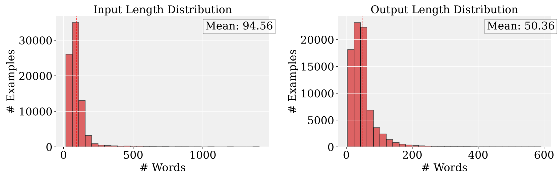
As shown in Table 9, the training instance from TableLLM contains the underlying reasoning process to reach the final answer. Such traces would benefit the model's reasoning process, as suggested by the findings by Guo et al. (2025); Muennighoff et al. (2025). Figure 9 displays the distributions of input and output lengths across training datasets. Notably, TableLlama exhibits significantly shorter output lengths compared to other training datasets. While TableBench has the longest average output length, its distribution possesses a high frequency of single-word answers (the prominent peak in the output distribution in Figure 9c). Furthermore, TableBench outputs may contain irrelevant reasoning elements (the first half of the gold answer is not relevant to the comparison of the performance in Table 9).

F.3 RQ5: How does the table instruction tuning compromise the general capabilities of the foundation LLMs?

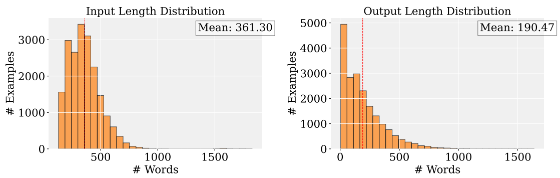
Evaluation Setup. We select five general benchmarks. **MMLU** (Hendrycks et al., 2021) examines the general ability of the model on 57 tasks including elementary mathematics, US history, computer science, etc. We adopt the 5-shot setup. **MMLU$_{\text{Pro}}$**



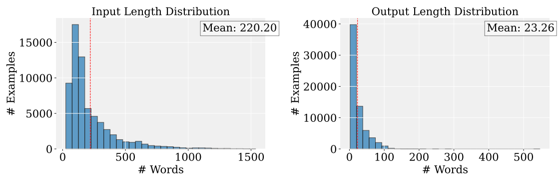
(a) TableLLlama.



(b) TableLLM.



(c) TableBench.



(d) TableGPT.

Figure 9: Distributions of the training data in terms of the input length and output length.

TableLlama (Zhang et al., 2024a)	
Q	What was the percent of restaurants and bars that were frequently the setting of behaviours related to unwanted physical contact or suggested sexual relations that happened off campus?
Gold	49.4.
TableLLM (Zhang et al., 2024b)	
Q	How many works did Leyla Erbil publish in total?
Gold	Leyla Erbil published a total of 11 works. *This can be determined by counting the number of entries in the "Name" column in the provided table.*
TableBench (Wu et al., 2025b)	
Q	Can you compare the performance of the advocates based on their wins, losses, and poll results, and identify which advocate has the most balanced performance across all metrics?
Gold	The table lists various advocates along with their performance metrics in terms of wins, losses, ties, poll wins, and poll losses. Patton Oswalt shows the most balanced performance across all metrics with 3 wins, 2 losses, 1 tie, 3 poll wins, and 3 poll losses.
TableGPT (Li et al., 2023)	
Q	predict the output value for the last row denoted as '[Output Value].'
Gold	6406 m.

Table 9: Training examples from TableLlama, TableLLM, TableBench, and TableGPT. We omit the corresponding table here for readability. The reasoning part is in italics for TableLLM data.

(Wang et al., 2024a) is an enhanced benchmark evaluating the general ability of the model, which contains up to ten options and eliminates the trivial questions in MMLU. We adopt the 5-shot setup. **AI2ARC** (Clark et al., 2018) is a reasoning benchmark containing natural, grade-school questions. We adopt the 0-shot setup and report the accuracy score on the challenging set. **GPQA** (Rein et al., 2023) is a reasoning benchmark containing questions in biology, physics, and chemistry written by domain experts. We adopt a 0-shot setup and report the accuracy score on its main set. **IFEval** (Zhou et al., 2023) is a dataset evaluating the general instruction following ability of the model containing instructions such as "return the answer in JSON format". We report the instance-level strict accuracy defined by Zhou et al. (2023). We include examples from these datasets in Section H.

For MMLU, MMLU_{Pro}, AI2ARC, and GPQA, as they are all multi-choice question-answering datasets, our objective is to select the most appropriate completion among a set of given options based on the provided context. Following Touvron et al. (2023), we select the completion with the highest likelihood given the provided context. As we evaluate the model based on their selection of the letter choice of "A", "B", etc., we do not normalize the likelihood by the number of characters in the completion.

Answer: Table instruction tuning does not necessarily compromise the base models' general capabilities. Figure 10 provides the model's performance on the five general benchmarks, while Table 15 provides the performance in numbers. We find that on MMLU, MMLU_{Pro}, AI2ARC, and GPQA, *our fine-tuned models do not compromise too much of the base models' general capabilities.* On AI2ARC, the score for Mistral-TableGPT is even slightly higher than the base model. Such performance improvement is likely due to the fact that many table tasks involve reasoning over tables, which may enhance the model's general reasoning ability. On IFEval, models fine-tuned from the Mistral model suffer a significant performance drop of over 20 points compared to the original model. However, models fine-tuned from the Phi model even improve the base model's performance. Contrary to the works arguing that tuning would compromise the model's capabilities (Luo et al., 2023), our finding suggests that domain-specific tuning does not necessarily lead to performance decay on general benchmarks, and the base model selection plays a crucial role in maintaining base LLMs' general capabilities.

F.4 RQ6: How does the model size affect performance on table tasks?

Evaluation Setup. We compare Phi 3 Mini Instruct (4B) versus Phi 3 Small Instruct (7B) on the table benchmarks introduced in Section D.

Answer: Model performance increases as the model size becomes larger. Figures 11 and 12 provide a performance comparison between Phi 3 Mini Instruct (4B) versus Phi 3 Small Instruct (7B). Similar to the findings for the general LLMs (Dubey et al., 2024; Wei et al., 2022), we find that the larger-sized model often leads to better performance for both the original model and the model after training on the same set of data.

Additionally, we present **the trade-offs between GPU hours and the model performance** in Table 10. These results demonstrate a consistent trend: the 7B model outperforms the 4B variant in all settings. However, the improvements are modest (e.g., +2.24 points in the TableLlama setting) relative to the increase in computational cost ($\sim$35% more GPU hours).

This highlights the trade-off: while larger models can yield stronger performance, the marginal gains may be insufficient to justify the additional cost in resource-constrained environments.

G Additional Discussions

G.1 Future Directions

Toward better table benchmarks. As LLMs continue to advance rapidly (Ouyang et al., 2022; Touvron et al., 2023; Dubey et al., 2024; Yang et al., 2024), there is a growing need for a comprehensive evaluation of table-related capabilities. Existing benchmarks often focus on narrow domains or specific subtasks (Chen et al., 2020b; Nan et al., 2022), while recent work has begun to explore broader coverage through synthetic datasets (Wu et al., 2025b) and multi-table reasoning setups (Wu et al., 2025a). However, concerns remain regarding the gap between synthetic benchmarks and authentic user needs. Future work shall ground table benchmarks in real-world use cases and build datasets that more accurately reflect user-driven queries and interactions with structured data.

Incorporating prior insights from table modeling. In the era of table LLMs, most efforts have focused on instruction tuning and dataset construction (Zhang et al., 2024a; Zheng et al., 2024). Yet, earlier work in table modeling demonstrates that incorporating table-specific features and structure-aware model architectures can significantly improve performance (Herzig et al., 2020; Yang et al., 2022). We advocate for future research to revisit and integrate these insights into modern table modeling, potentially bridging architecture-level innovations with instruction tuning strategies.

One promising direction is to integrate table-specific inductive biases—originally explored in models like TableFormer (Yang et al., 2022) and TAPAS (Herzig et al., 2020)—into the instruction-tuning pipeline of modern LLMs such as Phi or Mistral. For instance, table-aware attention mechanisms that distinguish between row-wise and column-wise relationships (Yang et al., 2022) can be incorporated during fine-tuning. These architectural modifications can guide the model to better preserve structural information by restricting attention across table rows and columns, thereby encoding relational dependencies more explicitly.

While modifying the core architecture of pretrained models like Phi or Mistral may not be practical, such inductive biases can be *approximated* during instruction tuning by using

1. Input-level encoding schemes (e.g., special

Training Data	Base Model	Model Sizes	GPU Hours	Avg Performance
N/A	Phi 3 Mini Instruct	4B	N/A	39.19
N/A	Phi 3 Small Instruct	7B	N/A	41.42
TableLlama	Phi 3 Mini Instruct	4B	441.2	39.25
TableLlama	Phi 3 Small Instruct	7B	596.0	41.49
TableLLM	Phi 3 Mini Instruct	4B	333.2	42.38
TableLLM	Phi 3 Small Instruct	7B	442.8	42.41

Table 10: Training GPU hours versus the averaged model performance for the Phi models at different sizes. We average the model performance across the OOD table tasks. These results demonstrate a consistent trend: the 7B model outperforms the 4B variant in all settings. However, the improvements are modest (e.g., +2.24 points in the TableLlama setting) relative to the increase in computational cost (∼35% more GPU hours).

tokens or segment IDs to mark rows and columns) (Herzig et al., 2020).

2. Adapter-based approaches where additional lightweight modules are trained to encode structural priors (Hu et al., 2023).

3. Reparameterize attention patterns via learned masks or prompts that encourage the model to attend differently across rows and columns—simulating table-aware attention without altering the model architecture (Lester et al., 2021; Li and Liang, 2021).

Bridging techniques from other fields. Table modeling has a long-standing tradition of adapting techniques from other areas of NLP (Yin et al., 2020). Recent efforts leverage vision-language models (Deng et al., 2024; Zheng et al., 2024). In this paper, we endeavor to leverage meta-evaluation (Kobayashi et al., 2024; Veuthey et al., 2025) to scrutinize the existing table evaluation framework. Here we list two future directions: (1) employing mechanistic interpretability methods (Huben et al., 2024) to better understand how models represent and reason over structured inputs; and (2) leveraging membership inference attacks (Shokri et al., 2017) to probe the potential leakage or memorization of structured data in pretraining corpora.

Bringing structures to the broader NLP. While table modeling often borrows from other subfields, we believe that table research can benefit the broader NLP community. Hawkins (2021) suggest that inherent structures[5] exist in human reasoning, and recent works suggest that LLMs can benefit from reasoning with structures (Sun et al., 2025). Reasoning in structures can potentially lead to more robust, interpretable, and modularized output (Wang et al., 2024b). We encourage future efforts on this and potentially bringing insights into the table research to the broader NLP community.

[5] Hawkins (2021) refer to these structures as “reference frame”.

G.2 Comparing Against Proprietary LLMs

Due to the immense computational cost and lack of public fine-tuning APIs (e.g., for Claude or GPT-4), our study —like most prior table LLM work (Zhang et al., 2024a) — focuses on 7B-scale open-source models such as Mistral, OLMo, and Phi. Despite these constraints, we attempt to contextualize our results by benchmarking our instruction-tuned 7B models against available larger or commercial models on public datasets. Tables 11 and 12 are some of the comparisons (TabFact, WikiTableQuestions) we are able to compile.

In Table 11, our instruction-tuned models match or even outperform commercial models like GPT-4o. For instance, Phi (7B) trained with TableLlama reaches 86.2 on TabFact, compared to GPT-4o’s reported 84.1.

In Table 12, while GPT-4 performs strongly on Wiki, some 7B-scale instruction-tuned models—such as Phi trained on TableLlama—demonstrate competitive performance.

For practitioners and researchers (e.g., start-ups or regulated domains) who lack access to proprietary APIs or massive compute budgets, our findings provide a replicable, useful snapshot of what is possible. For instruction tuning at the 7B scale, our findings explain why some small-model recipes succeed while others fail.

G.3 Additional Datasets

One concern we received is that the 16 benchmarks may not fully represent the scale and messiness of real-world enterprise tables.

To address such a concern, we conduct additional evaluation on six tasks on table QA and real-world table operations, including data integration, semantic enrichment, and schema-level reasoning

Base Model	Size	Training Data	ID?	Reasoning Model?	Acc
o4-mini	Unk	N/A	No	Yes	**94.3**
Deepseek-R1	685B	N/A	No	Yes	92.2
GPT-3.5	Unk	N/A	No	No	67.4
GPT-4	Unk	N/A	No	No	74.4
GPT-4o	Unk	N/A	No	No	**84.1**
Llama 3	70B	N/A	No	No	79.4
Llama 3	8B	N/A	No	No	58.4
Mistral	7B	N/A	No	No	62.3
OLMo	7B	N/A	No	No	38.2
Phi	7B	N/A	No	No	65.3
Phi-Mini	4B	N/A	No	No	41.8
Mistral	7B	Table-GPT	No	No	61.4
OLMo	7B	Table-GPT	No	No	44.9
Phi	7B	Table-GPT	No	No	71.0
Phi-Mini	4B	Table-GPT	No	No	53.2
Mistral	7B	TableLlama	Yes	No	**86.8**
OLMo	7B	TableLlama	Yes	No	83.8
Phi	7B	TableLlama	Yes	No	86.2
Phi-Mini	4B	TableLlama	Yes	No	81.6

Table 11: Model performance on TabFact dataset. Our instruction-tuned models match or even outperform commercial models like GPT-4o. For instance, Phi (7B) trained with TableLlama reaches 86.2 on TabFact, compared to GPT-4o's 84.1.

Base Model	Model Size	Training Data	Acc
GPT-3.5	Unk	N/A	**20.9**
GPT-4	Unk	N/A	**55.9**
Mistral	7B	N/A	27.9
OLMo	7B	N/A	19.4
Phi	7B	N/A	29.7
Phi-Mini	7B	N/A	26.4
GPT-3.5	Unk	Table-GPT	48.6
Mistral	7B	Table-GPT	25.5
OLMo	7B	Table-GPT	16.4
Phi	7B	Table-GPT	30.0
Phi-Mini	4B	Table-GPT	22.7
Mistral	7B	TableLlama	23.8
OLMo	7B	TableLlama	6.7
Phi	7B	TableLlama	**46.3**
Phi-Mini	4B	TableLlama	34.8

Table 12: Model performance on Wiki. While GPT-4 performs strongly on Wiki, some 7B-scale instruction-tuned models—such as Phi trained on TableLlama—demonstrate competitive performance.

Categories	Task Description	Metrics
Table-QA	Answer questions based on tables	Acc
Data-Imputation	Predict missing values in tables	Acc
Semantic-transform	Predict semantic transformations by examples	Acc
Entity-Matching	Match rows refer to the same semantic entity	Acc
Functional-Dependency	Predict functional-dependency in tables	F1
Semantic-join	Predict semantic join between two tables	Acc

Table 13: Additional datasets for testing.

(Xing et al., 2025; Chen et al., 2021c; Cafarella et al., 2008; He et al., 2015; Abedjan et al., 2016) listed in Table 13.

We report the performance of several 7B models (including our instruction-tuned variants) in Table 14. Our experiments reveal substantial performance variance across models, underscoring the central finding of our paper: base model selection plays a critical role. In particular, Phi 3 Small consistently outperforms OLMo across these tasks when trained with comparable instruction data (e.g. TableLLM's training data).

H Dataset Examples

H.1 FeTaQA

Input:

```
[TLE] The Wikipedia page title of this
table is Gerhard Bigalk. The Wikipedia
section title of this table is Ships
attacked. [TAB] | Date | Name |
Nationality | Tonnage (GRT) | Fate | [
SEP] | 14 June 1941 | St. Lindsay |
United Kingdom | 5,370 | Sunk | [SEP] |
21 December 1941 | HMS Audacity | Royal
Navy | 11,000 | Sunk | [SEP] | 2
February 1942 | Corilla | Netherlands |
8,096 | Damaged | [SEP] | 4 February
1942 | Silveray | United Kingdom | 4,535
 | Sunk | [SEP] | 7 February 1942 |
Empire Sun | United Kingdom | 6,952 |
Sunk | [SEP] | 16 May 1942 | Nicarao |
United States | 1,445 | Sunk | [SEP] |
19 May 1942 | Isabela | United States |
3,110 | Sunk |\n\nThe highlighted cells
of the table are: [HIGHLIGHTED_BEGIN]
[11,000], [Sunk], [8,096], [Damaged] [
HIGHLIGHTED_END] What happened to the
two heaviest ships Gerhard Bigalk
attacked?
```

Instruction:

```
This is a free-form table question
answering task. The goal for this task
is to answer the given question based on
 the given table and the highlighted
cells.
```

Output:

```
Gerhard Bigalk damaged one ship of 8,096
 GRT, and sunk one warship of 11,000
tons.
```

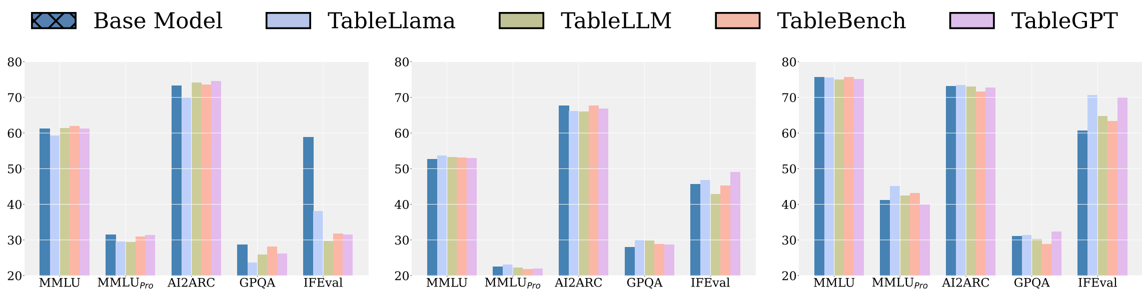
H.2 TabFact

Input:

```
[TLE] The table caption is about tony
lema. [TAB] | tournament | wins | top -
5 | top - 10 | top - 25 | events | cuts
made [SEP] | masters tournament | 0 | 1
```

Dataset	OLMo 7B Instruct	OLMo 7B Instruct	Phi 3 Small	Phi 3 Small	Phi 3 Small	Phi 3 Small	Phi 3 Small
Size	7B	7B	7B	7B	7B	7B	7B
Train Data	N/A	TableLLM	N/A	TableLlama	TableLLM	TableBench	TableGPT
FinQA	0.7	0.1	7.2	**20.0**	6.8	7.2	10.7
WebTable	0.2	4.3	10.5	9.1	14.3	15.9	30.1
SEMA-Join	2.0	17.5	69.4	72.2	67.8	**72.9**	70.6
Fodors-Zagats	49.5	67.2	97.8	61.8	**99.5**	98.9	**99.5**
DBLP-Scholar	77.4	86.2	93.7	43.0	**95.6**	94.4	95.3
Auto-Relate	0.0	0.0	**39.5**	3.0	36.5	28.4	24.2
DataXFormer	0.0	0.0	39.2	31.8	**47.1**	42.0	32.7
AVG	18.5	25.0	51.1	34.4	**52.5**	51.4	51.9

Table 14: Model performance on additional datasets listed in Table 13. Our experiments reveal substantial performance variance across models, underscoring the central finding of our paper: base model selection plays a critical role. In particular, Phi 3 Small consistently outperforms OLMo across these tasks when trained with comparable instruction data (e.g. TableLLM's training data).



(a) Mistral v0.3 7B Instruct.

(b) OLMo 7B Instruct.

(c) Phi 3 Small Instruct (7B).

Figure 10: Performance of fine-tuned models trained on different data (e.g. TableLlama) on general benchmarks. The green and red hatched bars represent performance gains or losses relative to the base model, respectively. As indicated by the similar performance bar heights, table instruction tuning does not necessarily compromise the base model's general capabilities. Table 15 provides the performance in number.

| 2 | 4 | 4 | 4 | [SEP] | us open | 0 | 2 | 3 | 4 | 6 | 5 | [SEP] | the open championship | 1 | 2 | 2 | 2 | 3 | 3 | [SEP] | pga championship | 0 | 0 | 1 | 2 | 5 | 4 | [SEP] | totals | 1 | 5 | 8 | 12 | 18 | 16 |\n\nThe statement is: < tony lema be in the top 5 for the master tournament , the us open , and the open championship>. Is it entailed or refuted by the table above?

Instruction:

This is a table fact verification task. The goal of this task is to distinguish whether the given statement is entailed or refuted by the given table.

Output:

entailed

H.3 ToTTo

Input:

<page_title> List of Governors of South Carolina </page_title> <section_title> Governors under the Constitution of 1868 </section_title> <table> <cell> 76 < col_header> # </col_header> <col_header> 74 </col_header> <col_header> 75 </ col_header> </cell> <cell> Daniel Henry Chamberlain <col_header> Governor </ col_header> <row_header> 76 </row_header > </cell> <cell> December 1, 1874 < col_header> Took Office </col_header> < row_header> 76 </row_header> </cell> </ table>

Instruction:

This is a highlighted cells description task. The goal of this task is to generate the language description given table cells.

Output:

Daniel Henry Chamberlain was the 76th Governor of South Carolina from 1874.

H.4 Beer

Input:

Beer A is:\n|name|factory|\n|---|---|\n| Sierra Amber Ale|Silver Peak Restaurant \& Brewery|\n\nBeer B is:\n|name|factory |\n|---|---|\n|Sierra Andina Alpamayo Amber Ale|Sierra Andina|

Method	MMLU	MMLU_{Pro}	AI2ARC	GPQA	IFEval
	Acc	Acc	Acc	Acc	Acc
M	61.2	31.4	73.3	28.6	58.8
M-TableLlama	59.4	29.5	69.6	23.7	38.0
Δ	↓ 1.9	↓ 1.9	↓ 3.4	↓ 4.9	↓ 20.7
M-TableLLM	61.4	29.3	74.2	25.9	29.6
Δ	↑ 0.2	↓ 2.0	↑ 0.9	↓ 2.7	↓ 29.1
M-TableBenchLLM	62.0	31.0	73.6	28.1	31.8
Δ	↑ 0.7	↓ 0.4	↑ 0.3	↓ 0.5	↓ 27.0
M-TableGPT	61.3	31.3	74.6	26.1	31.4
Δ	↑ 0.1	↓ 0.1	↑ 1.3	↓ 2.4	↓ 27.3
O	52.6	22.5	67.6	27.9	45.6
O-TableLlama	53.7	23.1	66.2	29.7	46.8
Δ	↑ 1.1	↑ 0.6	↓ 1.4	↑ 2.0	↑ 1.2
O-TableLLM	53.3	22.3	66.0	29.0	42.8
Δ	↑ 0.7	↓ 0.3	↓ 1.6	↑ 1.9	↓ 2.8
O-TableBenchLLM	53.1	21.9	67.7	28.6	45.2
Δ	↑ 0.5	↓ 0.7	↑ 0.1	↑ 0.9	↓ 0.4
O-TableGPT	52.9	21.9	66.8	28.8	48.9
Δ	↑ 0.3	↓ 0.6	↓ 0.8	↑ 0.8	↑ 3.4
P	75.7	41.2	73.1	31.0	60.7
P-TableLlama	75.5	45.1	73.5	31.5	70.1
Δ	↓ 0.2	↑ 3.9	↑ 0.3	↑ 0.4	↑ 9.9
P-TableLLM	75.0	42.6	73.1	30.4	64.8
Δ	↓ 0.7	↑ 1.3	↑ 0.0	↓ 0.8	↑ 4.1
P-TableBenchLLM	75.7	43.3	60.8	28.8	63.3
Δ	↑ 0.0	↑ 2.0	↓ 1.5	↓ 2.1	↑ 2.6
P-TableGPT	75.1	40.1	72.6	32.4	70.0
Δ	↓ 0.5	↓ 1.2	↓ 0.3	↑ 1.4	↑ 9.4

Table 15: Evaluation of the models on general benchmarks. "M-", "O-", and "P-" represent Mistral v0.3 7B Instruct, OLMo 7B Instruct, Phi 3 Small Instruct (7B), respectively. "Δ" denotes the performance difference between the instruction-tuned model and its base model.

```
\# Task Description: Please determine
whether Beer A and Beer B refer to the
same entity or not.
```

Instruction:

```
You are a helpful assistant that
specializes in tables.\n Your final
answer should be \'Yes\' or \'No\'.
Return the final result as JSON in the
format \{"answer": "<Yes or No>"\}. Let'
s think step by step and show your
reasoning before showing the final
result.
```

Output:

```
\{"answer": "No"\}
```

H.5 TabB$_{eval}$

Input:

```
Read the table below in JSON format:\n[
TABLE] \n\{"columns": ["index", "
organization", "year", "rank", "out of"],
 "data": [["bribe payers index", "
transparency international", 2011, 19,
28], ["corruption perceptions index", "
transparency international", 2012, 37,
176], ["democracy index", "economist
intelligence unit", 2010, 36, 167], ["
ease of doing business index", "world
bank", 2012, 16, 185], ["economic
freedom index", "fraser institute", 2010,
 15, 144], ["economic freedom index", "
the heritage foundation", 2013, 20, 177],
 ["global competitiveness report", "
world economic forum", 20122013, 13,
144], ["global peace index", "institute
for economics and peace", 2011, 27, 153],
 ["globalization index", "at kearney /
foreign policy magazine", 2006, 35, 62],
 ["press freedom index", "reporters
without borders", 2013, 47, 179], ["
property rights index", "property rights
 alliance", 2008, 28, 115]]\}\n\nLet\'s
get start!\nQuestion: What is the
average rank of the indices published by
 Transparency International?
```

Instruction:

```
You are a helpful assistant that
specializes in tables.\nYou are a table
analyst. Your task is to answer
questions based on the table content.\n\
n\nThe answer should follow the format
below:\n[Answer Format]\nFinal Answer:
AnswerName1, AnswerName2...\n\nEnsure
the final answer format is the last
output line and can only be in the "
Final Answer: AnswerName1, AnswerName2
..." form, no other form. Ensure the "
AnswerName" is a number or entity name,
as short as possible, without any
explanation.\n\n\nGive the final answer
to the question directly without any
explanation.
```

Output:

```
28
```

H.6 MMLU

Input:

```
{5-shot examples}
Find the degree for the given field
extension Q(sqrt(2), sqrt(3), sqrt(18))
over Q.
\nA. 0\nB. 4\nC. 2\nD. 6\nAnswer:
```

Instruction:

```
The following are multiple choice
questions (with answers) about abstract
algebra.\n\n
```

Output:

```
B
```

H.7 IFEval

Input:

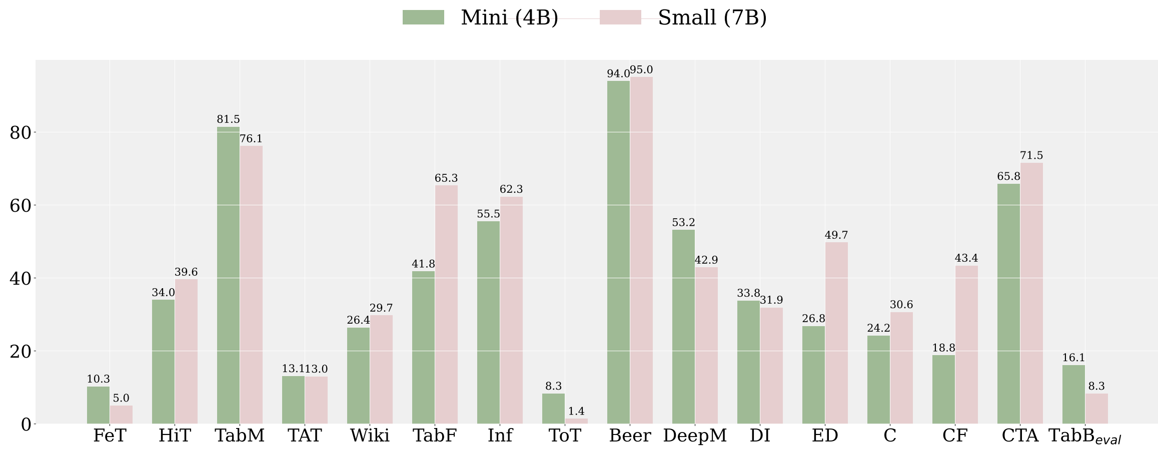
Can you help me make an advertisement for a new product? It’s a diaper that’s designed to be more comfortable for babies and I want the entire output in JSON format.

Instruction:

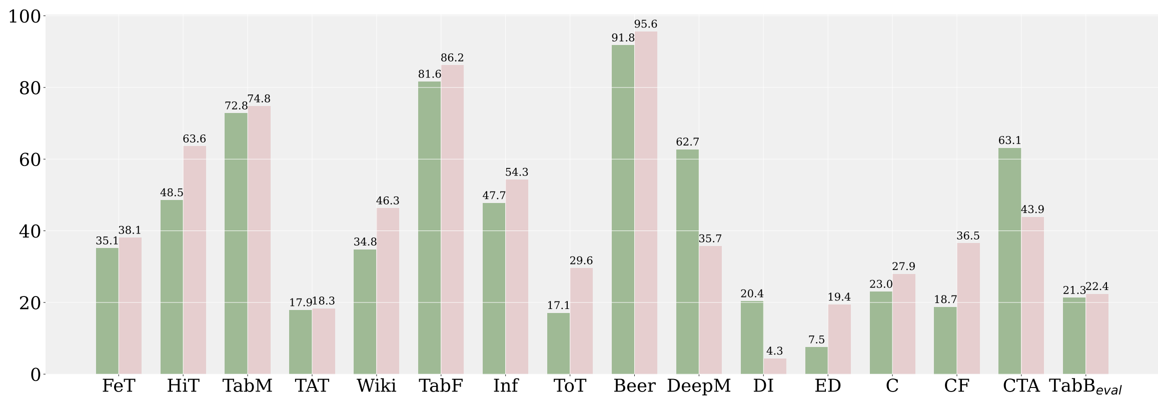
You are a helpful assistant.

Output:

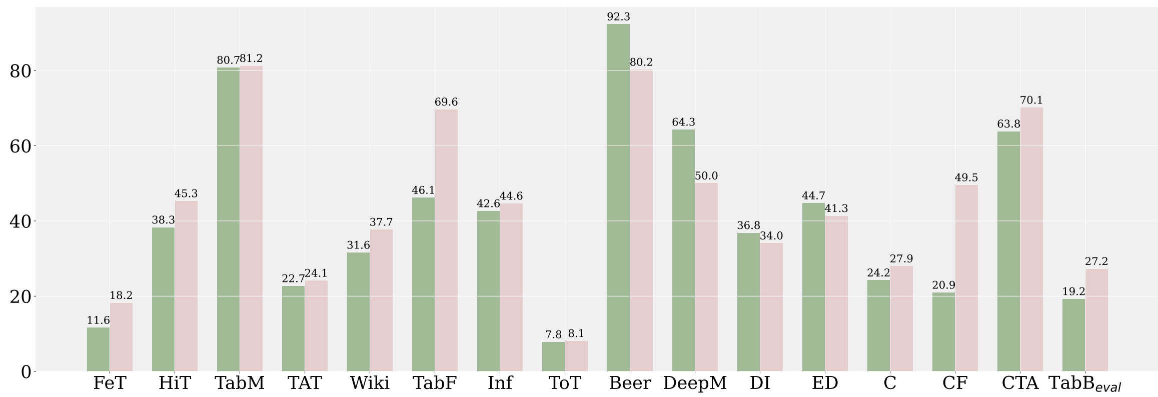
[JSON formatted answer]



(a) No training data, the original model.

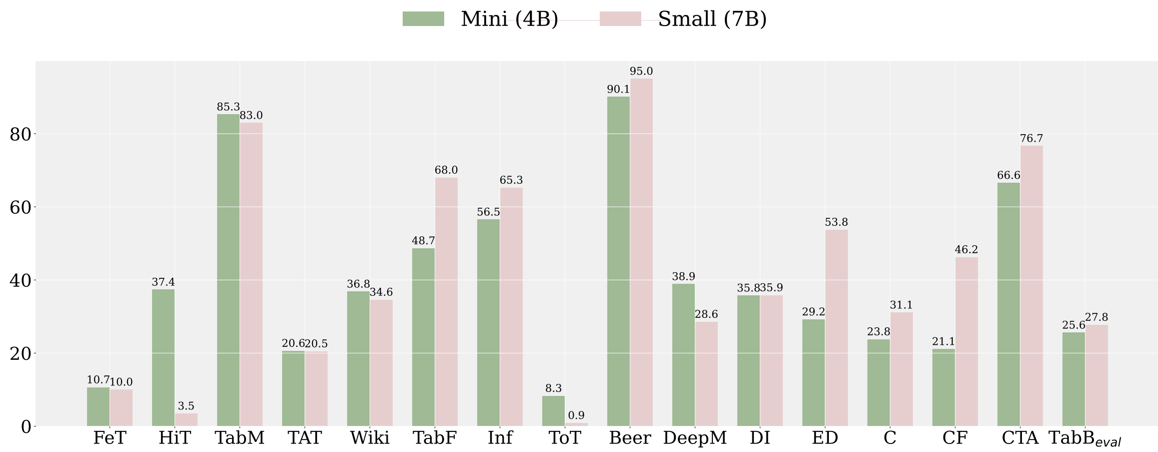


(b) Training data for TableLlama.

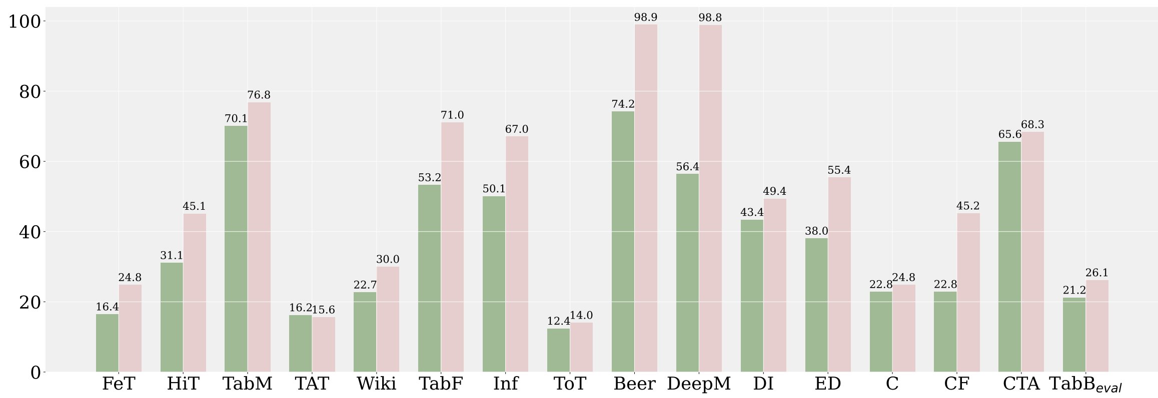


(c) Training data for TableLLM.

Figure 11: Performance of Phi 3 Mini Instruct (4B) versus Phi 3 Small Instruct (7B) model on different table tasks with different training data. In most cases, the 7B model outperforms the 4B model.



(a) Training data for TableBench.



(b) Training data for TableGPT.

Figure 12: Performance of Phi 3 Mini Instruct (4B) versus Phi 3 Small Instruct (7B) model on different table tasks with different training data. In most cases, the 7B model outperforms the 4B model.