Lights Out, Tabs On: Advancing Row-Column Encoding for Tabular LLMs

Anonymous Authors[1]

Abstract

Large language models (LLMs) excel in understanding diverse real-world data and achieving cross-domain generalization, but struggle with row-level tabular predictions and table-level QAs. Existing tabular LLMs serialize tables into 1D text using language templates (*e.g.*, *feature name* is *value*), which lack 2D spatial relationships, or structured formats (*e.g.*, *HTML* tables), which disrupt *feature name-value* associations. In this paper, we introduce LoTo: **L**ights **o**ut, **T**abs **o**n, a novel tabular LLM equipped with the axial row-column encoder. Inspired by the "Lights Out" game, LoTo prioritizes attention on cells sharing the same row and column. It incorporates tunable 2D positional encodings to enhance structural awareness, binned embeddings to improve numerical recognition, and a fine-grained cell projector to preserve tabular information. We develop a comprehensive training and evaluation benchmark for general tabular instruction fine-tuning. Experimental results demonstrate that LoTo achieves leading performance across both row-level and table-level tasks, establishing a foundation for general tabular LLMs.

1. Introduction

In recent years, large language models (LLMs) have utilized language to bridge applications across different modalities (Wang et al., 2024; Zhao et al., 2023), domains (Biswas, 2023; Kasneci et al., 2023; Zhang et al., 2023), and types of tasks (Imani et al., 2023; Zhuang et al., 2023). They have become essential tools in several key industries. One promising area for LLMs is tabular learning (Borisov et al., 2022; Sui et al., 2024b), which involves queries with table inputs. Tabular data (van Dijk et al., 2021; Gogas & Papadimitriou, 2021; Hino et al., 2018) is widespread in fields such as natural sciences, finance, and sustainable development. In tables, the first row (header) represents *feature names*, and its column cells contain *values*—paired with discrete text or continuous numbers. The tabular queries hold rich, domain-specific information, combining textual semantics, numerical variations, and more (Jiang et al., 2025).

Unlike vision or language domains, the knowledge gap between different tables can be substantial. This corresponds to the first major challenge for general tabular models: the extreme diversity and heterogeneity inherent in tabular data make cross-table knowledge transfer difficult. Consequently, most tree-based (Chen & Guestrin, 2016; Prokhorenkova et al., 2018; Ke et al., 2017) and deep models (Gorishniy et al., 2021; Wang & Sun, 2022; Somepalli et al., 2022) require retraining on downstream datasets. LLMs, however, offer considerable potential for such transfer (Kim et al., 2024). They can semantically interpret header information and adapt to unseen queries by leveraging context or other tables (Dong et al., 2024).

A further challenge for the general tabular LLMs lies in simultaneously handling diverse task types: **1**) Tabular prediction for row-level classification and regression (Borisov et al., 2022), which focuses on individual row; **2**) Table QA with table-level understanding (Shigarov, 2023), reasoning (Ye et al., 2023), and completion (Sun et al., 2016), which involves queries about the entire table. For most LLM-based tabular models, a common practice is to serialize 2D tables into 1D text sequences, as LLMs are designed to process sequences (Vaswani et al., 2017). This is typically done using either unstructured formats (*e.g.*, *feature name* is *value* (Hegselmann et al., 2023)) or structured formats (*e.g.*, HTML or Markdown) (Wen et al., 2024; Yang et al., 2024). We observe that structured serialization excels for table QA, while unstructured one is more effective for tabular prediction. Existing tabular LLMs often struggle to handle both types of tasks simultaneously, primarily due to inherent limitations in serialization encoding. Structured serialization often disrupts relationships between *feature names* and *values* (Su et al., 2024; Fang et al., 2024). When the context contains multiple entries, LLMs struggle to maintain the long-range dependencies between these elements in tabular prediction tasks. On the other hand, unstructured serialization, though simpler, leads to lengthy sequences as context

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

increases, and repeating *feature* descriptions becomes impractical (Sui et al., 2024a). Worse still, LLMs face additional limitations, such as being naturally not sensitive to numerical data (Yan et al., 2024). Thus, a general tabular encoder with an effective alignment and training strategy is essential to harness the LLMs for table understanding.

To address this, we propose a novel paradigm that integrates a tabular encoder with axial row-column attention (Ho et al., 2019) to make LLMs recognize tabular inputs more clearly. Furthermore, we develop a comprehensive training and evaluation benchmark for general tabular instruction fine-tuning, capable of simultaneously addressing both row-level prediction and table-level QA tasks. The tabular encoder builds on structured serialization with: **1**) Tunable **2D positional encodings** using structure-adaptive prompts for better spatial awareness; **2**) Tree-based auxiliary models to construct **binned embeddings** for robust numerical recognition; **3**) Fine-grained **cell encoder** that integrates *feature names*, positional, magnitude, and other information; **4**) **Axial row-column attention** that captures global information from each cell, which is then concatenated with existing serialization for improved semantic and structural embeddings.

In a tabular encoder, we focus more on rows (same sample) and columns (same header/attribute), which aligns with the "Lights out" game, where controlling the lights in rows and columns turns off the entire panel. We named our method LoTo: **L**ights **o**ut, **T**abs **o**n. "Tabs on" signifies the LLM's understanding of the table by aligning tabular embeddings to its input space, followed by general tabular instruction fine-tuning to adapt the model for diverse downstream tasks. To support this, we design a tabular instruction data engine that gathers data from over 100 datasets, with tasks like table description, understanding, reasoning, and completion, with tabular classification and regression. We employ both template-based and task-intent-driven data generation with the auto engine. We also integrate parts of the TableLLaMA (Zhang et al., 2024) dataset, resulting in a total of $81k$ training samples. LoTo is evaluated on benchmarks for tabular prediction, table QA, and completion across diverse domains, including science, finance, commerce, health, and so on. It achieves leading performance across all task types, establishing a robust foundation for general tabular LLMs. Our main contributions are:

- **LoTo architecture**. A large-scale tabular LLM with axial row-column attention.
- **General tabular instruction tuning**. A data engine and framework for alignment and fine-tuning.
- **Comprehensive tabular instruction benchmarks.** In- and out-of-domain evaluation from authoritative datasets for table QA, tabular prediction, and completion tasks.

2. Preliminary

2.1. Notations

Basic Data Components: Tables consist of *feature names* (defining column semantics) and *values* (discrete text or continuous numbers). We represent *feature names* as a header $\mathbf{t}^{\text{header}} \in \mathbb{R}^{d}$ and values as $\mathbf{x} \in \mathbb{R}^{n \times d}$, forming a table with n data rows and d columns.

Tabular Task Types: **1**) Tabular Prediction: Aims to predict target feature(s) $\mathbf{y}$ (via classification/regression) for individual data rows. A single sample's input includes $\mathbf{t}^{\text{header}}$ and a data row $\mathbf{x}_i \in \mathbb{R}^{d}$. **2**) Table QA: Involves queries about the entire table's content, such as reasoning, filling, summarization, or identifying structural relationships. Input comprises $\mathbf{t}^{\text{header}}$ and the full table data $\mathbf{x}$.

Tables, being inherently 2D, suffer from substantial embedding shifts due to diverse values and heterogeneous feature names. LLMs, benefiting from vast training, offer significant potential for cross-domain generalization. Optionally, relevant context $\mathbf{x}^{\text{context}}$ can further aid LLM decision-making. Subsequent sections will review research on table inputs to language models and tabular LLM advancements, experimentally exploring the rationale for a customized tabular encoder to bridge LLMs and tabular learning.

3. LoTo: Omni-Task Tabular LLMs

3.1. Architecture

Motivation: Our objective is to improve the table recognition capabilities by leveraging a table encoder to align new data characteristics, such as enhancing sensitivity to 2D structures, row-column interactions, numerical data types, and so on. In the LoTo framework, tabular inputs are processed by the encoder and then aligned with textual parts (*e.g.*, contexts, instructions) within the embedding space, supported by pre-training and instruction fine-tuning.

Tabular Encoder with tunable tokens, local aggregators, and global axial row-column attention:

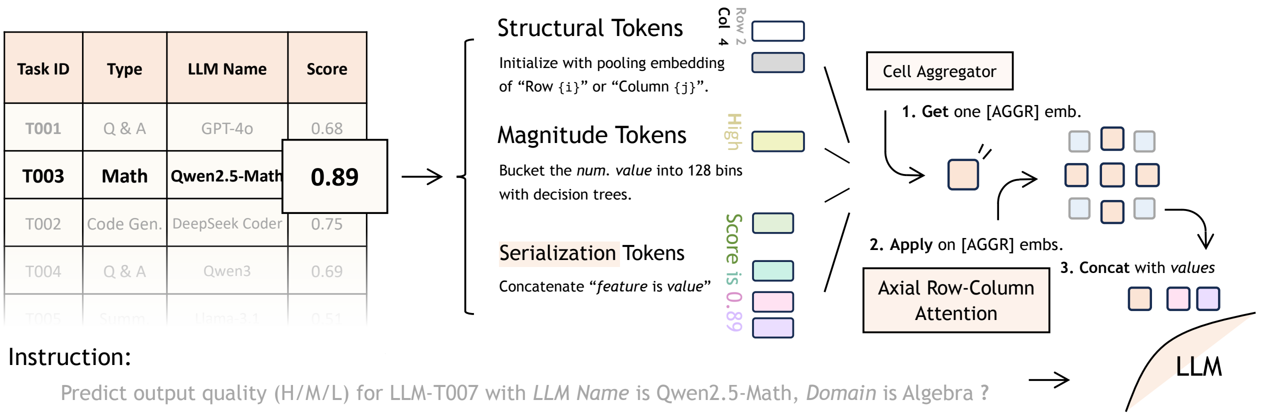
- **Dynamic structural tokens for spatial understanding:** We introduce 2 tunable tokens for each cell to encode row and column. These token embeddings are initialized by:

$$\mathbf{t}^{\texttt{[POS]}} = \text{avg}\left(\text{emb}\left(\text{“Row }\{i\}\text{”}\right)\right), \text{avg}\left(\text{emb}\left(\text{“Column }\{j\}\text{”}\right)\right) . \tag{1}$$

- **Relative magnitude tokenization for numerical values:** Similar to prior works (Yan et al., 2024), we use a tunable binning token divided into 128 quantiles, *i.e.*,

$$\mathbf{t}^{\texttt{[MAG]}} = \begin{cases} \text{C4.5}_{\text{leaf\_index}}(\mathbf{x}^{\text{num}}), & \text{if many-shot annotations} \\ \text{Uniform}_{\text{index}}(\mathbf{x}^{\text{num}}), & \text{if online few-shot context} \end{cases} . \tag{2}$$

For many-shot labeled tables, a decision tree maps numer-



Task ID	Type	LLM Name	Score
T001	Q & A	GPT-4o	0.68
T003	Math	Qwen2.5-Math	0.89
T002	Code Gen.	DeepSeek Coder	0.75
T004	Q & A	Qwen3	0.69
T005	Summ.	Llama-3.1	0.51

Structural Tokens

Initialize with pooling embedding of "Row {i}" or "Column {j}".

Magnitude Tokens

Bucket the *num. value* into 128 bins with decision trees.

Serialization Tokens

Concatenate "*feature* is *value*"

Instruction:

Predict output quality (H/M/L) for LLM-T007 with *LLM Name* is Qwen2.5-Math, *Domain* is Algebra ?

Figure 1: **Architecture of Our Model.** LoTo efficiently aligns embeddings from the tabular encoder to the LLM. The tabular encoder effectively extracts critical features, capturing various numerical types and complex structures. After tabular instruction tuning, LoTo seamlessly integrates context embeddings with the current instruction for enhanced performance.

ical column inputs to annotations, with leaf nodes defining magnitude partitions. For online context tables, numerical ranges are uniformly discretized.

- **Local aggregator for multi-functional information of cell:** Elements associated with each cell include structural token $\mathbf{t}^{\texttt{[POS]}}$, magnitude token $\mathbf{t}^{\texttt{[MAG]}}$, *values*, and their corresponding *feature names*. We employ compact self-attention layers and a reserved $\mathbf{t}^{\texttt{[AGGR]}}$ token to distill representations and reduce redundancy. We have:

$$\mathbf{h}_i^{\texttt{[AGGR]}} = \mathbf{h}_{0,:} \in \mathbb{R}^d \,,$$
$$\text{where } \mathbf{h} = \mathrm{Attn}\left(\left[\mathbf{t}^{\texttt{[AGGR]}};\, \mathbf{t}_i^{\texttt{[POS]}};\, \mathbf{t}_i^{\texttt{[MAG]}};\, \mathbf{t}_i^{\text{name}};\, \mathbf{x}_i\right]\right) \,. \tag{3}$$

- **Axial attention for row-column relationship awareness:** We introduce axial attention to globally encode the above aggregated embeddings $\mathbf{h}^{\texttt{[AGGR]}}$. These are then concatenated with the basic 2D *values* to align with the LLM input embeddings, *i.e.*,

$$\mathbf{h}^{\text{axial}} = \mathrm{Attn}\left(\left\{\mathbf{h}_{i,:}^{\texttt{[AGGR]}}\right\}_{i=1}^{n}\right) + \mathrm{Attn}\left(\left\{\mathbf{h}_{:,j}^{\texttt{[AGGR]}}\right\}_{j=1}^{d}\right)^{\top} , \tag{4}$$

In summary, the tabular encoder represents key information in [AGGR] tokens while aligning with structural values, contexts, and instructions. The final tabular embedding, to be aligned with other text inputs, can be formalized as:

$$\mathbf{h}^{\text{tab}} = \Big(\mathbf{h}_1^{\text{header}}, \ldots, \text{`\textbackslash n'}; \underbrace{\left[\mathbf{h}_{11}^{\text{axial}}, \mathbf{x}_{11}\right]}_{\text{one cell}}, \left[\mathbf{h}_{12}^{\text{axial}}, \mathbf{x}_{12}\right], \ldots, \text{`\textbackslash n'}; \ldots, \left[\mathbf{h}_{nd}^{\text{axial}}, \mathbf{x}_{nd}\right], \text{`\textbackslash n'}; \Big) \,, \tag{5}$$

where $\mathbf{h}_j^{\text{header}}$ represents the embeddings of the *feature names*, and $\mathrm{h}^{\text{instruct}}$ corresponds to the task instruction, which is detailed in the next section. Our framework workflow is illustrated in Figure 1.

3.2. In-context Tabular Instructions

An in-context learning approach is introduced for rapid adaptation using zero- or few-shot examples to connect diverse tabular tasks and leverage flexible language organization. This method categorizes tabular tasks into two components: **1**) a tabular context (table content, potentially with few-shot annotations) and **2**) a task instruction augmented with relevant tabular query information. The tabular context is randomly sampled from training data or via similarity-based methods like k-NN (Peterson, 2009). Task instructions are then composed with a tabular query to create task-oriented prompts (*e.g.*, "What does this table describe?"), providing sufficient context for unseen tasks.

3.3. Cross-Table Language Autoregression

During LoTo's cold start, a gap exists between language and tabular embeddings, hindering LLMs' interpretation of table structures and relationships. To bridge this, we introduce tailored training tasks and strategies focusing on instruction following and table context understanding:

- **Language-supervised**: Make LLMs recognize tabular embeddings by generating full-table descriptions using a "feature is value" format.
- **Autoregression on cells**: Train sequential and reverse autoregressive tasks by predicting the next cell's "feature is value" description.
- **Query-context relation**: Guide LLM in identifying context relevance to queries through interactive tasks mimicking SQL operations. This includes verification, matching, and other context-wide calculations.
- **Domain-oriented understanding**: Equip LoTo with deeper understanding by training it to transfer knowledge and predict beyond existing information.

Table 1: **Performance Comparisons** of LoTo and the baseline models across different tabular **classification**, **regression**, and **table QA** tasks. We present the results of various machine learning models, deep models, and other tabular LLMs. The evaluation datasets span domains like Science, Finance, Commerce, Health, and Others. The best is highlighted in bold, while the second-best is underlined. LoTo achieves leading performance across various domains and shot scenarios.

Dataset \ Model		XGBoost		CatBoost		FTT		TP-BERTa		TabPFN		LoTo (Ours)		
		2-shot	8-shot	2-shot	8-shot	2-shot	8-shot	2-shot	8-shot	2-shot	8-shot	0-shot	2-shot	8-shot
Science	IRI	.50	.68	**.77**	.62	.65	.60	.55	.55	.65	.52	.46	.52	.61
	CUS	.50	.47	.57	.53	.47	.50	.52	.45	**.58**	.55	.54	.50	.55
Finance	DEF	.50	.50	.55	.53	.60	.55	.63	.43	**.65**	.50	.56	.55	.61
	MOB	.25	.37	.38	.47	.35	.42	.25	.27	.35	.50	.32	.38	**.52**
Health	CDC	.50	.45	.65	.52	.68	.57	.65	.54	**.75**	.65	.65	.61	.68
	MAT	.33	.43	.37	.43	.46	.44	.50	.44	.41	.54	.52	.72	**.78**
	OBE	.14	.24	.41	.38	.43	.24	.21	.19	.21	.21	.34	.30	**.46**
Others	GOL	.50	.68	.72	.65	.82	**.85**	.47	.70	.67	.73	.63	.70	.74
	PRE	.33	**.52**	.39	.50	.44	.48	.46	.33	**.52**	.50	.50	.50	.51
	BAS	.50	.50	.50	.47	.45	**.70**	.57	.55	.47	.48	.45	.54	.66

Dataset \ Model		XGBoost		CatBoost		FTT		TP-BERTa		LoTo (Ours)		
		2-shot	8-shot	2-shot	8-shot	2-shot	8-shot	2-shot	8-shot	0-shot	2-shot	8-shot
Science	AIR	.088	.092	.097	.089	.083	.107	.075	.104	.078	.075	**.071**
	DIA	.276	.186	.309	.240	.352	.144	.291	.220	.128	.102	**.095**
Finance	GAR	.218	.182	.179	**.160**	.216	.199	.195	.193	.225	.223	.225
Health	NHA	.258	.239	.240	**.203**	.209	.255	.214	.213	.237	.229	.225
Others	CPM	.194	.172	.175	**.130**	.188	.173	.199	.188	.198	.151	.143
	ALC	.757	.532	.528	.515	.551	**.497**	.704	.532	.640	.539	.506
Mean		.299	.234	.255	.223	.266	.229	.280	.241	.251	.220	**.211**

Dataset \ Model	Qwen3		TableLLM	TableLlama	TableGPT2	LoTo (Ours)		
	0.6B	1.7B	7B	7B	7B	0.6B	1.7B	8B
Relation Extract.	28.9	26.2	3.8	**92.0**	83.4	80.2	80.6	83.5
HiTab	20.8	21.7	0.0	64.7	**70.3**	62.4	63.4	68.1
FetaQA	15.5	16.1	8.7	**39.1**	29.0	25.0	25.3	28.9
FEVEROUS (OoD.)	63.4	63.9	46.9	73.8	**78.1**	70.7	73.5	74.8
Completion	55.9	58.1	37.2	40.0	39.6	63.6	64.1	**66.9**

4. Experiments & Conclusion

Basic architecture & training data. LoTo is built on a Qwen3 backbone (0.6B-8B), utilizing a 2-layer self-attention for local aggregation and axial attention for global feature extraction, paired with a non-reasoning generative prompt. LoTo is trained on a comprehensive dataset of ∼81k samples, comprising tabular alignment instructions, instruction fine-tuning samples for prediction and understanding, and additional data from TableLlama.

Performance Analysis. LoTo consistently demonstrates strong overall performance, excelling in zero-shot transfer and competitive few-shot learning across diverse datasets, often outperforming established baselines like TabPFN.

Discussion. LoTo introduces a novel LLM paradigm for tabular learning, employing a specialized tabular encoder with axial row-column attention, 2D positional encodings, and binned numerical embeddings to address data heterogeneity and diverse task demands. Developed using an 81k-sample instruction tuning framework and comprehensive evaluation benchmarks, LoTo achieves leading performance across various tabular tasks, including row-level prediction and table-level QA. A current limitation is its design for complete, rectangular tables, requiring pre-processing for irregular data, though it lays a robust foundation for future general tabular LLMs.

References

Biswas, S. S. Role of chat gpt in public health. *Annals of biomedical engineering*, 51(5):868–869, 2023.

Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., and Kasneci, G. Deep neural networks and tabular data: A survey. *IEEE transactions on neural networks and learning systems*, 2022.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *KDD*, 2016.

Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Chang, B., Sun, X., and Sui, Z. A survey on in-context learning. In Al-Onaizan, Y., Bansal, M., and Chen, Y. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 1107–1128, 2024.

Fang, X., Xu, W., Tan, F. A., Hu, Z., Zhang, J., Qi, Y., Sengamedu, S. H., and Faloutsos, C. Large language models (llms) on tabular data: Prediction, generation, and understanding - A survey. *Trans. Mach. Learn. Res.*, 2024, 2024.

Gogas, P. and Papadimitriou, T. Machine learning in economics and finance. *Computational Economics*, 57:1–4, 2021.

Gorishniy, Y., Rubachev, I., Khrulkov, V., and Babenko, A. Revisiting deep learning models for tabular data. In *NeurIPS*, 2021.

Hegselmann, S., Buendia, A., Lang, H., Agrawal, M., Jiang, X., and Sontag, D. Tabllm: few-shot classification of tabular data with large language models. In *AISTATS*, 2023.

Hino, M., Benami, E., and Brooks, N. Machine learning for environmental monitoring. *Nature Sustainability*, 1(10): 583–588, 2018.

Ho, J., Kalchbrenner, N., Weissenborn, D., and Salimans, T. Axial attention in multidimensional transformers. *CoRR*, abs/1912.12180, 2019.

Imani, S., Du, L., and Shrivastava, H. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*, 2023.

Jiang, J.-P., Liu, S.-Y., Cai, H.-R., Zhou, Q., and Ye, H.-J. Representation learning for tabular data: A comprehensive survey, 2025. URL `https://arxiv.org/abs/2504.16109`.

Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103: 102274, 2023.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Kim, M. J., Grinsztajn, L., and Varoquaux, G. CARTE: pretraining and transfer for tabular learning. In *ICML*, 2024.

Peterson, L. E. K-nearest neighbor. *Scholarpedia*, 4(2): 1883, 2009.

Prokhorenkova, L. O., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. Catboost: unbiased boosting with categorical features. In *NeurIPS*, 2018.

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

Somepalli, G., Schwarzschild, A., Goldblum, M., Bruss, C. B., and Goldstein, T. SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training. In *NeurIPS Workshop*, 2022.

Su, A., Wang, A., Ye, C., Zhou, C., Zhang, G., Zhu, G., Wang, H., Xu, H., Chen, H., Li, H., et al. Tablegpt2: A large multimodal model with tabular data integration. *arXiv preprint arXiv:2411.02059*, 2024.

Sui, Y., Zhou, M., Zhou, M., Han, S., and Zhang, D. Table meets LLM: can large language models understand structured table data? A benchmark and empirical study. In Caudillo-Mata, L. A., Lattanzi, S., Medina, A. M., Akoglu, L., Gionis, A., and Vassilvitskii, S. (eds.), *ACM WSDM*, pp. 645–654. ACM, 2024a.

Sui, Y., Zhou, M., Zhou, M., Han, S., and Zhang, D. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 2024b.

Sun, H., Ma, H., He, X., Yih, W.-t., Su, Y., and Yan, X. Table cell search for question answering. In *Proceedings of the 25th International Conference on World Wide Web*, pp. 771–782, 2016.

van Dijk, A. D. J., Kootstra, G., Kruijer, W., and de Ridder, D. Machine learning in plant science and plant breeding. *Iscience*, 24(1), 2021.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 2017.

Wang, W., Chen, Z., Chen, X., Wu, J., Zhu, X., Zeng, G., Luo, P., Lu, T., Zhou, J., Qiao, Y., et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *Advances in Neural Information Processing Systems*, 36, 2024.

Wang, Z. and Sun, J. Transtab: Learning transferable tabular transformers across tables. In *NeurIPS*, 2022.

Wen, X., Zhang, H., Zheng, S., Xu, W., and Bian, J. From supervised to generative: A novel paradigm for tabular deep learning with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD, 2024.

Yan, J., Zheng, B., Xu, H., Zhu, Y., Chen, D. Z., Sun, J., Wu, J., and Chen, J. Making pre-trained language models great on tabular prediction. In *ICLR*, 2024.

Yang, Y., Wang, Y., Sen, S., Li, L., and Liu, Q. Unleashing the potential of large language models for predictive tabular tasks in data science. *CoRR*, abs/2403.20208, 2024.

Ye, Y., Hui, B., Yang, M., Li, B., Huang, F., and Li, Y. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pp. 174–184, 2023.

Zhang, H., Li, X., and Bing, L. Video-llama: An instruction-tuned audio-visual language model for video understanding. In Feng, Y. and Lefever, E. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023 - System Demonstrations, Singapore, December 6-10, 2023*, 2023.

Zhang, T., Yue, X., Li, Y., and Sun, H. Tablellama: Towards open large generalist models for tables. In Duh, K., Gómez-Adorno, H., and Bethard, S. (eds.), *NAACL*, pp. 6024–6044. Association for Computational Linguistics, 2024.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

Zhuang, Y., Yu, Y., Wang, K., Sun, H., and Zhang, C. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36:50117–50143, 2023.