Introducing the OBSERVATORY Library for End-to-End Table Embedding Inference

Tianji Cong
University of Michigan
congtj@umich.edu

Zhenjie Sun*
University of Michigan
zjsun@umich.edu

Paul Groth
University of Amsterdam
p.t.groth@uva.nl

H. V. Jagadish
University of Michigan
jag@umich.edu

Madelon Hulsebos
University of Amsterdam
m.hulsebos@uva.nl

Abstract

Transformer-based table embedding models have become prevalent for a wide range of applications involving tabular data. Such models require the serialization of a table as a sequence of tokens for model ingestion and embedding inference. Different downstream tasks require different kinds or levels of embeddings such as column or entity embeddings. Hence, various serialization and encoding methods have been proposed and implemented. Surprisingly, this conceptually simple process of creating table embeddings is not straightforward in practice for a few reasons: 1) a model may not natively expose a certain level of embedding; 2) choosing the correct table serialization and input preprocessing methods is difficult because there are many available; and 3) tables with a massive number of rows and columns cannot fit the input limit of models. In this work, we extend OBSERVATORY, a framework for characterizing embeddings of relational tables, by streamlining end-to-end inference of table embeddings, which eases the use of table embedding models in practice. The codebase of OBSERVATORY is publicly available at `https://github.com/superctj/observatory`.

1 Introduction

Table embedding models have attracted significant interests from both the natural language processing (NLP) and data management communities (Dong et al., 2022; Badaro et al., 2023). Transformer-based models (Vaswani et al., 2017) manifest appealing performance on a diverse range of downstream tasks including but not limited to semantic parsing (Yin et al., 2020), table question answering (Herzig et al., 2020), and table fact verification (Liu et al., 2022) in NLP, as well as entity matching, semantic column type detection, and data integration and augmentation in data management (Li et al., 2020; Deng et al., 2020; Suhara et al., 2022; Cong et al., 2023c).

As various applications may need different kinds of embeddings (e.g., semantic column type detection is based on column embeddings whereas entity matching requires entity embeddings), many serialization and encoding methods have been proposed and implemented to serialize a table to a sequence of tokens for model ingestion and embedding inference. Given that these embeddings look at different *levels of aggregation* of the table structure, we refer to these kinds of embeddings as levels of embeddings. Despite being conceptually straightforward, we identify three barriers in practice that hinder the application of table embeddings:

*Co-first author.

Table Representation Learning Workshop at NeurIPS 2023.

1) A model may not natively expose a certain level of embeddings. Table embedding models are usually fine-tuned for specific downstream tasks and expose embeddings corresponding to those tasks. On the other hand, table embedding models are pretrained on a large corpus with downstream-task-agnostic objectives like cell value recovery to capture generic semantics in tabular data. For this reason, researchers and practitioners are keen to leverage table embedding models in new application contexts that may require entity-level embeddings instead of column-level embeddings, as natively provided by the models.
2) Plenty of table serialization and input preprocessing methods exist. Each table embedding model designs and implements their own table serialization and input preprocessing methods, many of which have subtle differences. There is a need to collect such methods so that users can easily switch from one method to another for their experiments.
3) Tables with a massive number of rows and columns cannot fit the input limit of models. Unlike Wikipedia tables with small sizes on average, tables from open repositories such as Data.gov and OpenML can have millions of rows and dozens of columns. How to handle such large tables remains a practical challenge.

In this work, we build on top of OBSERVATORY (Cong et al., 2023b), a framework for characterizing embeddings of relational tables. We enrich and streamline the end-to-end table embedding inference by extending OBSERVATORY with modularized implementations of common practices that ease the obstacles of using table embedding models as previously described.

2 The OBSERVATORY Library

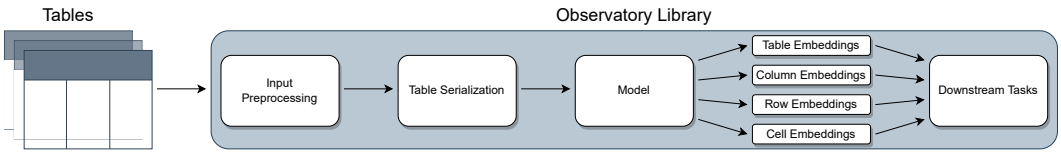

Figure 1: Overview of the table embedding inference pipeline in OBSERVATORY.

Motivated by the need for understanding the strengths and weaknesses of table embedding models and the representations they generate, OBSERVATORY (Cong et al., 2023b) contributes a formal framework that systematically analyzes embedding representations of relational tables. Based on invariants of the relational data model and on statistical considerations regarding data distributions, OBSERVATORY defines eight primitive properties and corresponding measures to quantitatively characterize table embeddings (e.g., column embeddings) for these properties. Over a suite of relational table datasets, OBSERVATORY evaluates and analyzes nine popular language and table embedding models. We find, for example, that some models are sensitive to table structure such as column order, that functional dependencies are rarely reflected in embeddings, and that specialized table embedding models have relatively lower sample fidelity. Such insights help researchers and practitioners better anticipate model behaviors and select appropriate models for their downstream tasks, while guiding researchers in the development of new models. In this work, we primarily focus on the pipeline of table embedding inference and describe features supported in each component of the inference pipeline (see Figure 1 for an overview). With the OBSERVATORY library, we introduce and contribute a standardized package that researchers and practitioners can easily employ for end-to-end inference of table embeddings and customize for their use cases.

2.1 Table Serialization

Many table embedding models are adapted from Transformer-based language models (Vaswani et al., 2017; Devlin et al., 2019). Due to the sequence nature of natural languages, the models require serializing row-column structured relational tables to sequences of tokens before feeding to these embedding models. Moreover, Transformer models natively expose token-level embeddings, but for downstream tasks involving tabular data, different levels of embedding outputs are usually needed (e.g., cell and column level). To obtain the desired level of output embeddings, it is common to leverage special tokens and/or positional embeddings. In practice, table serialization is often tightly

Bank Name	City	State	Cert	Acquiring Institution	Closing Date
Heartland Tri-State Bank	Elkhart	KS	25851	"Dream First Bank, N.A."	28-July-23
First Republic Bank	San Francisco	CA	59017	"JPMorgan Chase Bank N.A."	1-May-23
Signature Bank	New York	NY	57053	"Flagstar Bank, N.A."	12-Mar-23

Figure 2: A sample table listing failed banks since October 1, 2000.

coupled with the embedding models (e.g., in HuggingFace Transformers library (Wolf et al., 2020)). In other words, prepared sequences from a row-wise serialization method cannot be feed to a column embedding model. We describe below two major types of serialization methods as well as additional customization that are implemented in the OBSERVATORY library.

Row-wise Serialization. Tables are parsed in a row-by-row manner in which rows are concatenated with special tokens inserted as delimiters, and are optionally prepended with table headers and data types. For instance, the table in Figure 2 will be serialized in TaBERT (Yin et al., 2020) as

```
[SEP] Bank Name | text | Heartland Tri-State Bank [SEP] City ... [SEP]
... [SEP] Cert | real | 25851 [SEP] ... [SEP] Bank Name ...
```

and in TaPEx (Liu et al., 2022) as

```
[HEAD] Bank Name | City | State | ... | Closing Date [Row] 1 Heartland ...
[Row] 2 First Republic Bank | San Francisco | CA | 59017 ...
```

The nuance is that the serialization method of TaBERT concatenates the header (and data types) to each row whereas that of TaPEx only prepends the header once at the beginning of the sequence following the special token [HEAD].

Column-wise Serialization. As to column-oriented downstream tasks such as semantic column type prediction and join discovery, tables are serialized by column so that column representations can be conveniently extracted from special tokens prepended before each column. For example, the table in Figure 2 will be serialized in DODUO (Suhara et al., 2022) as

```
[CLS] Heartland Tri-State Bank, First Republic Bank, Signature Bank
[CLS] Elkhart, ..., New York [CLS] ... [CLS] 28-July-23, ..., 12-Mar-23 [SEP]
```

and in Pylon (Cong et al., 2023c) as

```
[CLS] Bank Name | Heartland Tri-State Bank, ..., Signature Bank
[CLS] City | Elkhart, ... [CLS] ... [CLS] Closing Date | ..., 12-Mar-23 [SEP]
```

The only difference is that DODUO assumes meaningful table headers are lacking in spreadsheets and data frames, and therefore only takes in column data values as inputs. In contrast, Pylon experiments with the effect of adding headers in column representation learning based on SimCLR, a simple yet effective contrastive learning technique (Chen et al., 2020). An alternative to using special tokens like `[CLS]` to retrieve the desired level of output embeddings, is to keep track of row and column indices of each token and aggregate token embeddings to, for example, column embeddings or entity embeddings (Cong et al., 2023b). Badaro et al. (2023) also summarizes another two types of serialization methods, i.e., the combination of row/column-wise serialization and the text templates to represent the tabular data as sentences. We plan to investigate these methods and may integrate them in future releases of the OBSERVATORY library.

2.2 Input Preprocessing

Large relational tables can easily have millions of rows and a few dozens of columns, which exceeds the input limitation of Transformer-based embedding models due to memory constraints. We

implement all the input selection methods surveyed in Dong et al. (2022); Badaro et al. (2023) and briefly describe them below.

Random Sampling. Randomly selecting table elements (e.g., rows or cells) is a straightforward workaround to limit the input size. Despite being simple, random sampling has been shown to be effective in various tasks such as semantic parsing (Yin et al., 2020), representation learning (Herzig et al., 2020; Wang et al., 2021; Liu et al., 2022; Cong et al., 2023c), semantic column type detection (Suhara et al., 2022), and join discovery (Cong et al., 2023a).

Content Snapshot. When additional inputs such as natural language questions are available, top-k relevant rows / columns (known as content snapshot) can be selected by n-gram overlap (Yin et al., 2020) or the Jaccard coefficient (Eisenschlos et al., 2020) relative to natural language sentences.

TF-IDF. Term Frequency-Inverse Document Frequency (TF-IDF) scores are also used to downsample input tables and select relevant contents. For instance, DITTO (Li et al., 2020) truncates long entities by retaining only non-stopword tokens with high TF-IDF scores.

Input Splitting. Alternative to sampling, a large table can be split row-wise (Wang et al., 2021; Deng et al., 2020) and/or column-wise (Chen et al., 2021; Cong et al., 2023b). Note that column-wise splitting is necessary when a single wide row cannot fit into the input sequence, which is common for tables with a few dozens of columns like those in Open Data Lakes.

It is also worth mentioning that besides truncating inputs, there are efforts in increasing the input size of embedding models. We refer readers to Dong et al. (2022) for reference of sparse attention.

2.3 Embedding Aggregation

Although a model may not natively expose certain levels of embeddings for out-of-the-box use, we observe that it is common for a model to give access to token-level embeddings. Additionally, table embedding models usually have auxiliary mask embeddings or positional embeddings that indicate to which row and column a token belongs. Hence, we aggregate token embeddings (by averaging them for example) to embeddings on a level (e.g. row or column) as needed. For vanilla language models, we exploit different serialization methods and use special tokens to retrieve row or column embeddings. Alternatively, we keep track of token positions in the table and aggregate them accordingly.

2.4 Supported Models

The OBSERVATORY library currently supports three language models including BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and T5 (Raffel et al., 2020); and six table embedding models comprising TAPAS (Herzig et al., 2020), TaBERT (Yin et al., 2020), TURL (Deng et al., 2020), DODUO (Suhara et al., 2022), TaPEx (Liu et al., 2022), and TapTap (Zhang et al., 2023). Technically, any encoder or encoder-decoder or decoder-only architecture model on HuggingFace that exposes token representations can be seamlessly integrated within OBSERVATORY by simply following our example use cases of BERT, RoBERTa and T5.

2.5 Supported Datasets

The OBSERVATORY library also supports loading and inference over several datasets of relational tables across multiple domains (e.g, Web tables and tables from open repositories such as Kaggle). Table 1 gives a summary of supported datasets. These datasets from heterogeneous sources have various dimensions with the average number of rows ranging from a few dozens to over twelve millions and the average number of columns up to 56.

3 Example Usage

As demonstrated in Cong et al. (2023b), researchers and practitioners can use the OBSERVATORY library to assess models and gain insights into the strengths and weaknesses of these models through the embeddings they generate over eight proposed properties. For instance, we find that some models are sensitive to table structure (i.e., row and column order). We also connect such observations to model behaviors on downstream tasks. An an example, DODUO, characterized as sensitive

Table 1: Supported datasets in the OBSERVATORY library.

Dataset	# Tables	Avg. # Rows	Avg. # Columns	Domain/Source
WikiTables-TURL (Test Set) (Deng et al., 2020)	4,964	21	4	Wikipedia
Spider (Yu et al., 2018) & Dr.Spider (Chang et al., 2023)	70	7,632	6	Mixed (e.g., college DB courses and Web)
NextiaJD-XS / S / M / L (Flores et al., 2021)	28 46 46 19	1,938 209,646 3,175,904 12,288,165	9 56 23 28	Open repositories (e.g., Kaggle and OpenML)
SOTAB (Korini et al., 2022)	59,548	33	3	Web Data Commons
GitTables (Hulsebos et al., 2023)	1,000,000	142	12	GitHub

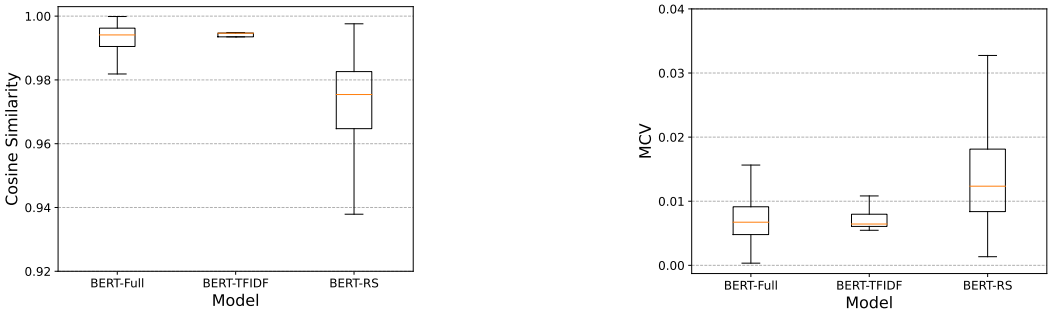

Figure 3: Cosine similarity and MCV value distributions of BERT column embeddings. The TF-IDF score-based input selection method results in the least variance in both measures.

to row/column order shuffling and random sampling, is shown to make unstable predictions over shuffled data in the downstream task of semantic column type prediction. For more such insights and experimental findings, we refer readers to Cong et al. (2023b).

In addition to experiments and analyses in Cong et al. (2023b), the OBSERVATORY library now supports various input preprocessing methods. Figure 3 shows the distributions of two measures of column embeddings from BERT over the WikiTables-TURL dataset. For each table, we shuffle the row order at most 1000 times and compute cosine similary and multivariate coefficient of variation (MCV) (Albert and Zhang, 2010) for each column and their variants. BERT-Full refers to column embeddings using all rows; BERT-TFIDF refers to column embeddings generated from 75% of rows selected by TF-IDF scores; and BERT-RS refers to column embeddings based on 75% of rows randomly sampled. The figure clearly demonstrates that the TF-IDF score-based input selection method results in the least variance of BERT column embeddings over the WikiTables-TURL dataset. This suggests that appropriate preprocessing can render embeddings more robust to table structure changes. In this regard, the OBSERVATORY library can facilitate researchers and practitioners to conduct more such interesting analyses.

4 Conclusion

To the best of our knowledge, OBSERVATORY is the first library that streamlines end-to-end inference of table embeddings at different levels and integrates many common practices of table serialization and input preprocessing. We envision OBSERVATORY to ease the obstacles of using table embedding models, allow customization of the inference pipeline, and accelerate the development of new models.

Acknowledgments and Disclosure of Funding

This research is supported in part by NSF grants 1946932 and 2312931, by Dutch Research Council (NWO) through grant MVI.19.032, and through computational resources and services provided by Advanced Research Computing at the University of Michigan, Ann Arbor.

References

Adelin Albert and Lixin Zhang. 2010. A novel definition of the multivariate coefficient of variation. *Biometrical Journal* 52, 5 (2010), 667–675.

Gilbert Badaro, Mohammed Saeed, and Paolo Papotti. 2023. Transformers for Tabular Data Representation: A Survey of Models and Applications. *Transactions of the Association for Computational Linguistics* 11 (2023), 227–249.

Shuaichen Chang, Jun Wang, Mingwen Dong, Lin Pan, Henghui Zhu, Alexander Hanbo Li, Wuwei Lan, Sheng Zhang, Jiarong Jiang, Joseph Lilien, Steve Ash, William Yang Wang, Zhiguo Wang, Vittorio Castelli, Patrick Ng, and Bing Xiang. 2023. Dr.Spider: A Diagnostic Evaluation Benchmark towards Text-to-SQL Robustness. *CoRR* abs/2301.08881 (2023).

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

Xinyun Chen, Petros Maniatis, Rishabh Singh, Charles Sutton, Hanjun Dai, Max Lin, and Denny Zhou. 2021. SpreadsheetCoder: Formula Prediction from Semi-structured Context. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research)*, Marina Meila and Tong Zhang (Eds.), Vol. 139. PMLR, 1661–1672.

Tianji Cong, James Gale, Jason Frantz, H. V. Jagadish, and Çagatay Demiralp. 2023a. WarpGate: A Semantic Join Discovery System for Cloud Data Warehouses. In *13th Conference on Innovative Data Systems Research, CIDR 2023, Amsterdam, The Netherlands, January 8-11, 2023*. www.cidrdb.org.

Tianji Cong, Madelon Hulsebos, Zhenjie Sun, Paul Groth, and H. V. Jagadish. 2023b. Observatory: Characterizing Embeddings of Relational Tables. *CoRR* abs/2310.07736 (2023).

Tianji Cong, Fatemeh Nargesian, and H. V. Jagadish. 2023c. Pylon: Semantic Table Union Search in Data Lakes. *CoRR* abs/2301.04901 (2023).

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. TURL: Table Understanding through Representation Learning. *Proc. VLDB Endow.* (2020), 307–319.

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 Conference on the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*. 4171–4186.

Haoyu Dong, Zhoujun Cheng, Xinyi He, Mengyu Zhou, Anda Zhou, Fan Zhou, Ao Liu, Shi Han, and Dongmei Zhang. 2022. Table Pre-training: A Survey on Model Architectures, Pre-training Objectives, and Downstream Tasks. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, Luc De Raedt (Ed.). ijcai.org, 5426–5435.

Julian Martin Eisenschlos, Syrine Krichene, and Thomas Müller. 2020. Understanding tables with intermediate pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020 (Findings of ACL)*, Trevor Cohn, Yulan He, and Yang Liu (Eds.), Vol. EMNLP 2020. Association for Computational Linguistics, 281–296.

Javier Flores, Sergi Nadal, and Oscar Romero. 2021. Towards Scalable Data Discovery. In *Proceedings of the 24th International Conference on Extending Database Technology, EDBT 2021, Nicosia, Cyprus, March 23 - 26, 2021*. OpenProceedings.org, 433–438.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. 2020. TaPas: Weakly Supervised Table Parsing via Pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, 4320–4333.

Madelon Hulsebos, Çagatay Demiralp, and Paul Groth. 2023. GitTables: A Large-Scale Corpus of Relational Tables. *Proc. ACM Manag. Data* 1, 1 (2023), 30:1–30:17.

Keti Korini, Ralph Peeters, and Christian Bizer. 2022. SOTAB: The WDC Schema.org Table Annotation Benchmark. In *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching, SemTab 2021, co-located with the 21st International Semantic Web Conference, ISWC 2022, Virtual conference, October 23-27, 2022 (CEUR Workshop Proceedings)*.

Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep Entity Matching with Pre-Trained Language Models. *Proc. VLDB Endow.* (2020), 50–60.

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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67.

Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çagatay Demiralp, Chen Chen, and Wang-Chiew Tan. 2022. Annotating Columns with Pre-trained Language Models. In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*. 1493–1503.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of the Conference on Neural Information Processing Systems*. 5998–6008.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021. TUTA: Tree-based Transformers for Generally Structured Table Pre-training. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*. ACM, 1780–1790.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*. Association for Computational Linguistics, 38–45.

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, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, 8413–8426.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. 2018. 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, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics, 3911–3921.

Tianping Zhang, Shaowen Wang, Shuicheng Yan, Jian Li, and Qian Liu. 2023. Generative Table Pre-training Empowers Models for Tabular Prediction. *CoRR* abs/2305.09696 (2023).