# Language Modeling on Tabular Data: A Survey of Foundations, Techniques and Evolution

Yucheng Ruan<sup>1,2†</sup>, Xiang Lan<sup>1,2†</sup>, Jingying Ma<sup>1</sup>, Yizhi Dong<sup>1</sup>, Kai He<sup>1</sup>, Mengling Feng<sup>1,2\*</sup>

<sup>1</sup>Saw Swee Hock School of Public Health, National University of Singapore, Singapore.

<sup>2</sup>Institute of Data Science, National University of Singapore, Singapore.

\*Corresponding author(s). E-mail(s): ephfm@nus.edu.sg;
Contributing authors: yuchengruan@u.nus.edu; ephlanx@nus.edu.sg;
jingyingma@u.nus.edu; yizhi@nus.edu.sg; kai\_he@nus.edu.sg;
†These authors contributed equally to this work.

#### Abstract

Tabular data, a prevalent data type across various domains, presents unique challenges due to its heterogeneous nature and complex structural relationships. Achieving high predictive performance and robustness in tabular data analysis holds significant promise for numerous applications. Influenced by recent advancements in natural language processing, particularly transformer architectures, new methods for tabular data modeling have emerged. Early techniques concentrated on pre-training transformers from scratch, often encountering scalability issues. Subsequently, methods leveraging pre-trained language models like BERT have been developed, which require less data and yield enhanced performance. The recent advent of large language models, such as GPT and LLaMA, has further revolutionized the field, facilitating more advanced and diverse applications with minimal fine-tuning. Despite the growing interest, a comprehensive survey of language modeling techniques for tabular data remains absent. This paper fills this gap by providing a systematic review of the development of language modeling for tabular data, encompassing: (1) a categorization of different tabular data structures and data types; (2) a review of key datasets used in model training and tasks used for evaluation; (3) a summary of modeling techniques including widely-adopted data processing methods, popular architectures, and training objectives; (4) the evolution from adapting traditional Pre-training/Pre-trained language models to the utilization of large language models; (5) an identification of persistent challenges and potential future research directions in language modeling for tabular data analysis. GitHub page associated with this survey is available at: https://github.com/lanxiang1017/Language-Modeling-on-Tabular-Data-Survey.git.

**Keywords:** Language modeling, Tabular data, Pre-training language model, Large language model

# 1 Introduction

Tabular data, consisting of rows with a consistent set of features, is one of the most prevalent data type in real-world and has been wildly used in different domains [1, 2]. In some crucial areas [3–5], a good predictive performance and robustness can provide significant benefits. However, effectively analyzing tabular data is challenging due to its complex structure. For example, a sample from tabular data could be either a single row of a table (1D tabular data), or a complete table from a set of tables (2D tabular data). Additionally, tabular data typically features a wide range of heterogeneous characteristics [6], such as various data types including numerical, categorical, and textual elements. Furthermore, tables often exhibit complex relationship between both columns and rows.

Over the past few decades, the field of natural language processing (NLP) has witnessed significant evolution in language modeling, particularly with the advent of transformer architecture. In the context of tabular modeling, early researches mainly focus on processing tabular data with NLP techniques such as embedding mechanisms, pre-training methods, and architectural modifications. These works mainly involve pre-training transformer-based models from scratch specifically for tabular data, which demands a substantial amount of data and can be impractical in certain fields, such as healthcare [7, 8]. While effective in certain scenarios, these approaches often face challenges in scalability and efficiency. Meanwhile, some researchers leverage pre-trained language models (PLMs) (e.g. BERT [9]) to model tabular data. These models, built on top of PLMs, require less training data while delivering superior predictive performance. It shows the effectiveness of adapting and reusing pre-trained LMs on task-specific tabular datasets [10].

More recently, the emergence of large language models (LLMs) has further transformed the landscape. Models such as GPT [11] and LLaMA [12] have demonstrated remarkable capabilities, achieving state-of-the-art results across a variety of tasks with minimal fine-tuning. These models excel in few-shot and zero-shot learning scenarios, where they can perform complex tasks with little to no additional training data. This development has opened new avenues for utilizing LLMs in more advanced and diverse applications for tabular data [13]. Strong evidence of this trend is the dramatic increase in the volume of research on tabular modeling with LLMs. This evolution from training models from scratch or using PLMs to adopting LLMs marks a significant paradigm shift in language modeling for tabular data.

Despite significant interest in extracting extensive knowledge from tabular data, there is a notable lack of a comprehensive survey in the research community that clearly sorts out existing language modeling methods on tabular data, outlines technical trends, identifies challenges, and suggests future research directions. In this work, we bridge this gap by conducting a systematic review of language modeling for tabular data.

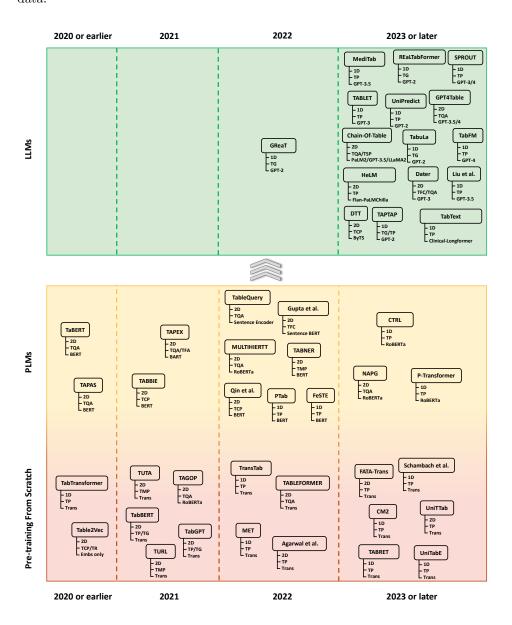
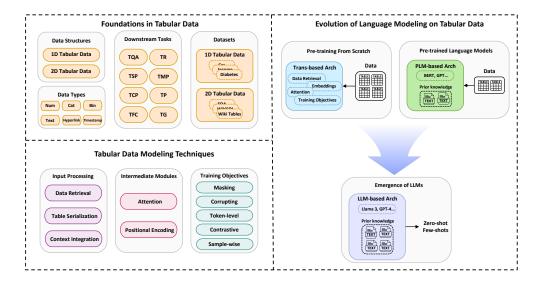


Fig. 1 The timeline of the evolution of language modeling on tabular data. Each model includes the following information from top to bottom: type of tabular data (1D or 2D), evaluation tasks, and the backbone model.

This survey paper aims to provide a thorough overview of the evolution in language modeling for tabular data (Figure 1) at this pivotal moment of paradigm shift, offering a comprehensive summary and categorization of existing studies to present a clear big picture of this promising research field.

In summary, the main contributions of this survey are threefold. First, it firstly categorizes tabular data into 1D and 2D data formats. Unlike existing surveys either focus on 1D tabular data [14, 15] for traditional tasks like inference and data generation, or concentrate on 2D tabular data [16, 17] for more complex tasks such as information retrieval and table understanding, this work is the first one to provide a systematic review of tasks and datasets for both types of tabular data. Second, it reviews upto-date progress of language modeling techniques on tabular data and provides an exhaustive taxonomic categorization. Third, it highlights various research challenges and potential avenues for exploration in language modeling for tabular data.



 $\textbf{Fig. 2} \ \ \text{The structure of survey paper. It includes three main parts: foundations in tabular data, tabular data modelling techniques and evolution of language modelling on tabular data.$ 

In this paper, as illustrated in Figure 2, we firstly introduce the foundations of tabular data in Section 2, offering a comprehensive overview of four main parts: data structures (Section 2.1), data types (Section 2.2), downstream tasks (Section 2.3), and datasets (Section 2.4). We explain the two primary tabular data structures that recent research focuses on: 1D and 2D tabular data. We also discuss the different data types found in the tabular domain. Following this, we provide a detailed description of eight major downstream tasks: table question answering (Section 2.3.1), table retrieval (Section 2.3.2), table semantic parsing (Section 2.3.3), table metadata prediction (Section 2.3.4), table content population (Section 2.3.5), table prediction (Section 2.3.6), table fact-checking (Section 2.3.7) and table generation (Section 2.3.8).

Subsequently, we outline some commonly used datasets for each data type with key characteristics, which are linked to different downstream tasks.

Next, we present a taxonomy of recent research that maps out the language modeling techniques on tabular data, categorizing it into three key domains: input processing (Section 3.1), intermediate modules (Section 3.2), and training objectives (Section 3.3). Specifically, input processing focuses on transforming the raw tabular data into the format suitable for LMs. We further examine the input processing techniques by breaking them into specific sub-categories: data retrieval (Section 3.1.1), table serialization (Section 3.1.2) and context integration (Section 3.1.3). In intermediate modules, we discuss two components: positional encoding (3.2.1) and attention mechanisms (3.2.2), which are modified to achieve better predictive performance in tabular domain. Furthermore, we discuss the training objectives, which plays a critical role in helping LMs to learn semantic information.

Following that, we analyze the evolution of how language models are adapted in tabular domain (Section 4). Firstly, we describe the adaptions and benefits of pre-training from scratch and using PLMs early on, particularly with the introduction of transformers (Section 4.1). We then review the recent advance of LLMs in tabular data modeling and highlight how their adaptions differ from previous methods (Section 4.2).

Finally, we identify several challenges and future opportunities in language modeling for tabular data (Section 5), and conclude our paper in Section 6.

#### 2 Foundations in Tabular Data

In this section, we discuss the foundations of tabular data, including data structures, related downstream tasks, and common datasets. Specifically, we provide the definition of 1D and 2D tabular data, highlighting the relevant downstream tasks and datasets for each type.

#### 2.1 Data Structures

In this survey, we categorize tabular datasets based on their structural characteristics into two main categories. Firstly, 1D tabular datasets usually contain single tables with multiple rows and columns, annotated with task-specific labels. The primary objective of analyzing 1D tabular data is row-level prediction within a single table. In contrast, 2D tabular datasets comprise multiple tables, which may not necessarily be annotated. They are typically leveraged for pre-training or fine-tuning downstream tasks conducted at the table level. Subsequently, we present an exploration of various data types for tabular data.

#### 2.1.1 1D Tabular Data

1D tabular data is a common form of structured data. As illustrated in Figure 3, it is organized in a specific and well-defined schema, making it easy to query, analyze, and process. It is widely employed for machine learning analysis due to its simplicity in structure. This type of data has a clear and fixed schema, where each data element is organized into rows and columns within a table. A 1D tabular data is defined as

#### 

Time					ВР
Time	Glucose	Temp	SBP	DBP	
114520	65	36.7	114	77	
121135	78	37.1	103	64	
154911	71	35.8	90	78	
162346	102	36.9	118	88	
183456	98	36.2	98	61	
211745	84	37.3	91	68	
234519	99	36.9	120	79	

Fig. 3 The illustration of 1D tabular data (left) and 2D tabular data (right). One row represents a sample in 1D tabular data while one tabular table corresponds to a sample in 2D tabular data.

 $X \in \mathbb{R}^{N \times D} = \{x_1, x_2, \dots, x_D\}$ , where D is the number of columns and N is the number of rows. Each column  $x_d \in \mathbb{R}^N$  typically represents a specific attribute or field, and each row  $X_i \in \mathbb{R}^D$  represents a sample record or entry. Each column adheres to a specific data type constraint, maintaining a straightforward format. Each row within a 1D table incorporates a label according to the specific task, typically a row-level classification or regression based on the sample. Given its simple structure, it is extensively used in classical machine learning models. We define the sample size of 1D tabular data as the number of rows N and feature size as the number of features used as independent variables D.

#### 2.1.2 2D Tabular Data

In the modern machine learning era, more complicated tasks have been proposed such as question answering and information retrieval based on table. These tasks involve understanding information from tables and performing table-level analysis. The data used for these tasks are referred to as 2D tabular data. As shown in Figure 3, such data comprises a collection of tables.

Data types in 2D tabular data are more complex (discussed in Section 2.2). Unlike structured tables where data adheres to column data type constraints, the cells within the 2D tabular data can include diverse types, making the tables to be semi-structured. A 2D tabular data is defined as  $X \in \mathbb{R}^{N \times R \times C} = \{x_1, x_2, \dots, x_N\}$ , where each table  $x_i \in \mathbb{R}^{R \times C}$  comprises R rows and C columns, listing information related to a specific topic. In addition, the dataset often incorporates data from other modalities such as questions, metadata, and associated free-text. The sample size of 2D tabular data is defined as the number of tables N.

Furthermore, there is a special type of 2D tabular data, particularly significant for analysis in fields such as healthcare and finance. This data incorporates dynamic temporal information, with timestamps capturing changes over time, thus adding an additional layer of complexity to the dataset's structure. Meantime, information about individuals is recorded across multiple tables, linked by individual identifiers, forming an interconnected 2D data structure.

#### 2.2 Data Types

The values of individual cells within tabular data are governed by the constraints of column data types, ensuring regularity and consistency. Commonly, 1D datasets include simple data types such as numerical, categorical, and binary data. In contrast, the data types in 2D tabular data are characterized by greater flexibility and complexity. In the following section, we elaborate on the diverse data types that emerge in tabular data.

- Numerical data. Numerical data consists of values that can be expressed as numbers. The values can either be discrete, such as integers denoting whole numbers, or continuous, such as floating-point numbers with decimal precision. Numerical data type is a fundamental data type, allowing direct mathematical calculation.
- Categorical data. Categorical data, also known as qualitative data, represents data that can be grouped into a finite set of distinct categories. These categories can be nominal without any order information, or ordinal, exhibiting sequential rankings, albeit inter-category distances are unknown.
- Binary data. Binary data is a special case of categorical data where each observation falls into one of two distinct categories. These categories are often labeled as "Yes" and "No", represented numerically as 1 and 0, or expressed as Boolean values, representing logical outcomes denoted by "True" and "False".
- Text data. Text data includes both structured texts and free-texts. Structured texts follow a predefined format and a structured schema, such as name and address. On the other hand, free-texts refer to unstructured textual data that lacks a standardized structure. They comprise collections of text without format limitations, such as comments and descriptions.
- Hyperlinks. Hyperlinks contained in tabular data serve as connections to external sources, such as websites or documents, providing the dataset with supplementary context. The presence of hyperlinks in tabular data necessitates multi-hop reasoning over both the tables and the hyperlinked contents.
- Timestamps. Timestamp data denote the date and time information when an event occurs. By providing information on the sequence and intervals of events, it plays a pivotal role in tracking events over time. Capturing temporal information enables the analysis of time-series data, offering valuable insights into trends and patterns.

# 2.3 Downstream Tasks

With the advent and widespread deployment of language models, particularly LLMs, there has been a expansion in the spectrum of downstream tasks related to tabular data modeling using language models. These tasks are also executed with higher accuracy using LLMs compared to previous models. In this section, we'll elaborate 8 distinct categories of downstream tasks for which language models are employed. A summary

of downstream tasks are provided in Table 1. We provide a comprehensive overview of these tasks, detailing their nature, the inputs they require, the outputs they generate and specific cases. Also, we show the evaluation metric for each task in Table 2.

#### 2.3.1 Table Question Answering (TQA)

In the Table Question Answering (TQA) task, the system answers questions based on information contained in structured tables. Since TQA task requires understanding both the question (natural language) and the table's schema to provide accurate answers, language models could help. TQA is divided into simple TQA and complex TQA [17]. Simple TQA involves looking for answers within the table's cells, while complex TQA involves aggregation or reasoning based on the table's information to get the answer. The input of a TQA task is a question and a table, while the output is the answer to the question.

The evaluation metrics for the TQA task mainly include Execution Accuracy, Exact Match (EM), and F1. Execution Accuracy measures the model's ability to generate correct answers given a table and a question. It calculates accuracy by executing the query generated by the model and comparing the results with the expected query results. This metric was used by [18] [19] for evaluation in their paper. EM evaluates the extent to which the model's generated answer matches the standard answer exactly. Specifically, Exact Match requires that the content and format of the model's generated answer be identical to the standard answer. This metric was used by [19] [20] in their paper.

#### 2.3.2 Table Retrieval (TR)

Table Retrieval (TR) involves extracting relevant information from multiple tables based on a given query. The input is a question, and the output is the most relevant cells or tables. Language models can significantly aid this task by understanding the context of the query and the semantics of the data, enabling more accurate and efficient retrieval.

The evaluation metrics for the Table Retrieval task mainly include Normalized Discounted Cumulative Gain (NDCG). NDCG is a commonly used metric to measure the ranking effectiveness of information retrieval systems. It calculates the cumulative gain of the retrieval results, applies a discount to the gain, and normalizes it to evaluate the system's ability to return relevant results at various positions. NDCG was first introduced by [21] and has been widely used. For example, Table2vec [22], RIM [23], PET [24], Table2Graph [25] used NDCG for evaluation in their proposed methods.

#### 2.3.3 Table Semantic Parsing (TSP)

Table Semantic Parsing (TSP), mainly includes Text to SQL, which translates natural language queries into SQL commands. The input is a user's textual query, and the output is an executable SQL query. Language models could assist by understanding the textual query first, and then mapping it accurately to SQL syntax.

The evaluation metrics for the Table Semantic Parsing task mainly include Accuracy and Logical Form Accuracy (AccLF). Accuracy measures whether the answers

generated by the model match the standard answers. For example, TAPEX [26] and StructGPT [27] used Accuracy for evaluation in their paper. AccLF evaluates whether the logical forms generated by the model match the standard logical forms. For example, Seq2SQL [28] detailed discussed the application of AccLF in SQL query generation.

#### 2.3.4 Table Metadata Prediction (TMP)

Table Metadata Prediction (TMP) aims at inferring various types of metadata about tables, which includes subtasks such as Column Type Annotation. The input to this task is typically a row or a semi-structured table, and the output is enriched metadata information: Column Type Prediction categorizes each column by their data types (e.g., integer, text, date); Table Type Classification determines the inherent property of the table (e.g., transactional, relational, or summary tables). TMP could also involves identifying and understanding the relationships within and between tables and their entities (typically refers to the cell value and elements within a table, which could be specific data in a cell value such as names, places, dates, numbers, etc., or could also refer to elements of the table such as rows or columns), encompassing subtasks such as Table Relation Prediction, Table Relation Extraction, Table Entity Linking, Entity-based Relation Retrieval, and Cell Entity Recognition. The input for this task includes one or more tables along with potential external knowledge sources such as knowledge graph. The output ranges from identified relationships between tables to linked entities within cells to external knowledge sources, and evaluations of how entities and their relationships are represented within a table. Language models could significantly aid in these tasks by leveraging their ability to to process and interpret natural language and to understand context and semantics from textual data. For example, in column type annotation task, language models understand linguistic nuances and naming conventions in table headers. For instance, a language model can discern that a header containing "Date Of Birth" likely refers to a date type, whereas "Total Amount" or "Quantity" suggests numeric data. This nuanced understanding stems from the model's training on a vast corpus of text, enabling it to capture a broader range of expressions and context than typical data mining approaches.

The common evaluation metrics for the sub-task Column Type Annotation are Precision, Recall, F1, and Macro-F1. Additionally, the common evaluation metric for the sub-tasks Table Type Classification and Cell Entity Recognition is F1. Furthermore, the common evaluation metrics for the tasks Table Relation Extraction and Table Entity Linking are F1, Precision, and Recall. Finally, the common evaluation metrics for the sub-tasks Table Relation Prediction and Entity-based Relation Retrieval are Accuracy at 1, 3, and 5 (Precision at 1, 3, 5, and 10). Accuracy/Precision at n is a metric used to evaluate the performance of a model in recommendation or retrieval tasks. Accuracy at n indicates whether the model includes the target answer within the top n recommended positions. It measures the accuracy of the target answer being within the top 1 to n positions of the recommendation list. Precision at n refers to the frequency with which the model accurately recommends the target answer within the top n positions of the recommendation list. For example, Qin et al [29] used Accuracy at n for evaluation.

#### 2.3.5 Table Content Population (TCP)

The aim of Table Content Population (TCP) is enriching and completing tables by addressing subtasks such as Column Population, Row Population, and Cell Filling. The input for this task is a table that may have missing or incomplete information in its columns, rows, or cells. The output is a more complete table, with all missing elements accurately filled in. Language models can assist with these tasks by leveraging their knowledge base and understanding of context to infer missing information accurately.

The evaluation metrics for the sub-tasks Column Population and Row Population are Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), Re-call, Normalized Discounted Cumulative Gain at 10 (NDCG-10), Normalized Discounted Cumulative Gain at 20 (NDCG-20). MAP measures the average precision of a model across multiple queries. It calculates the average precision (AP) for each query and then takes the mean of these APs. MAP effectively evaluates the overall performance of a model in handling multiple queries. For example, Table 2Vec [22] used MAP for evaluation. MRR assesses the rank of the first correct answer returned by the model. It calculates the reciprocal rank for each query and then averages these values. A higher MRR indicates that the model returns the correct answer earlier. For exapmle, RIM [23] used MRR for evaluation. Recall measures the proportion of all relevant answers that the model can identify. NDCG-n evaluates the ranking effectiveness of a model in the top n recommended positions. By applying a discount to the cumulative gain at the top n positions and normalizing it, NDCG-n assesses the model's ability to return relevant results at higher ranks. For example, PET [24] used NDCG-5 and NDCG-10 for evaluation. The common evaluation metrics for the sub-task Cell Filling is Precision at n.

# 2.3.6 Table Prediction (TP)

Table Prediction (TP) is the most common task in the field of tabular modeling, incorporating subtasks like regression and classification. The input is a dataset organized in table format, featuring rows of instances and columns representing features or variables. For regression, the output is a continuous value, whereas for classification, the output categorizes each instance into discrete classes or labels. Since tables can contain cells with free-text, language models could significantly aid in these tasks by leveraging their vast pre-trained knowledge and contextual understanding.

The evaluation metrics for classification include F1 score, AUROC, and Accuracy, while for prediction tasks, they encompass Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and  $R^2$ . RMSE quantifies the average magnitude of errors between predicted and actual values. MAE similarly measures the average of absolute errors in predictions. The  $R^2$  statistic, also known as the coefficient of determination, tells us what percentage of the variation in the outcome can be explained by the model.

#### 2.3.7 Table Fact-checking (TFC)

Table Fact-checking (TFC) is to verify the accuracy and truthfulness of a statement based on knowledge presented in tables. The input is a table, alongside claims or statements. The output is a quantification of the veracity of these claims, categorizing

them as true, false, or partially true/false based on the evidence within the table. Language models can be helpful in this task due to their ability to process natural language and their knowledge in pre-trained models. The common evaluation metric for TFC is Accuracy.

#### 2.3.8 Table Generation (TG)

Table Generation is the process of creating structured tables from given inputs, which can range from specific data points to natural language descriptions or queries. The output is a well-structured table that organizes the input data into columns and rows, effectively presenting the information in a clear, concise, and accessible format. This task was born with the emergence of large language models.

The evaluation metrics for the Table Generation task mainly include Bilingual Evaluation Understudy (BLEU) and Average Normalized Edit Distance (ANED). BLEU is a metric used to assess the quality of generated text, initially designed for machine translation tasks. It calculates scores by comparing the n-gram matches between the generated text and multiple reference texts. The score ranges from 0 to 1, with a score closer to 1 indicating a higher similarity between the generated text and the reference texts. This metric was first proposed by Papineni et al. [30] in their paper. UniTabE [31], GPT4Table [32] and DATER [33] used BLEU for evaluation in their paper. ANED is a metric used to evaluate the edit distance between the generated text and reference texts. It calculates the number of edit operations (such as insertions, deletions, and substitutions) needed to transform the generated text into the reference text, then normalizes the score to a range from 0 to 1. A lower score indicates a higher similarity between the generated text and the reference text. This metric was first proposed by Marzal et al. [34] in their paper. DTT [35] used ANED for evaluation in their paper.

#### 2.4 Datasets

In the scope of this survey, 1D tabular data were extensively employed during the fine-tuning stage of language models with supervised TP downstream tasks, which enables the adjustment of the learned representation according to the specific task. They can also be used to evaluate the performance of tabular data mining models. Table 3 presents an overview of the most prevalent 1D datasets identified in the surveved literature. The 1D dataset spans various domains including census, financial, and healthcare sectors, including regression, classification, and top-n ranking subtasks. The sample size varies from a few hundred to 11 million, with feature sizes typically remaining under 50, although notable exceptions such as MNIST [49] (784 features) and BlogFeedback [50] (280 features). One popular example is the UCI adult income (Income) dataset [51], it is a well-known dataset that contains census variables such as age, work class, and education level. This dataset constitutes a binary classification dataset, with labels denoting whether income exceeds \$50K/vr. Additionally, the diabetes dataset [52] from OpenML presents another binary classification challenge, aimed at predicting the onset of diabetes mellitus using demographic and clinical attributes.

Table 1: Summary of downstream tasks related to tabular data.

Task	Sub-tasks	Input	Output	Exapmle
	Simple QA	Table and con-	Cell value	Tabert [18]
TQA	Complex QA	text	Aggregation Prediction	TableQAKit [36]
TR	-	Table and query	Tables or cells with ranking scores	Table2vec [22]
TSP	Text to SQL	Table and query	SQL sentence	TAPEX [26]
	Column Type Annotation	Table and a set of semantic types	Table type	TABBIE [37]
	Table Type Classification	Table	Table type	TUTA [37]
	Table Relation Prediction	2 colum without header	Relations	Qin el al [29]
TMP	Table Relation Extraction	Table and a set of relations R in knowledge base	Relations	TURL [38]
	Table Entity Linking	Table and Knowledge base	Entities Candidates	TURL [38]
	Entity-based Relation Retrieval	2 cells and Knowledge Graph	Relations	Qin el al [29]
	Cell Entity Recogonization	2 cells	Relations	TABNER [39]
	Column Population	The first N columns of a table	Cells value	Table2vec [22], TABBIE [37]
TCP	Row Population	Table and set of entities	Entities	Table2vec [22], TURL [38]
	Cell Filling	Partial Tables	Object entity	TURL [38]
TP	Classification	Table	Category	CTRL [40], CT-BERT [41], MediTab [42], UniPredict [43]
	Regression	Table	Probability	DANets [44], <b>TAPTAP</b> [45]
TFC	-	Table and state- ment	Binary decision	TAPEX [26], gpt4table [46]
TG	-	Text description	Table	GReaT [46], TabuLa [47], REaLTab- Former [48]

Note: In the **Example** column, the bolded entries are Large Language Models, and the non-bolded entries are Language Models.

Table 2: Summary of metrics for tabular downstream tasks.

Task	Sub-tasks	Metrics
TQA	Simple QA	Execution accuracy, Exact Match (EM), F1
1QA	Complex QA	Execution accuracy, Exact Match (EM), F1
TR	-	Normalized Discounted Cumulative Gain (NDCG)
TSP	Text to SQL	Accuracy, Logical Form Accuracy (AccLF)
	Column Type Annotation	Precision, Recall, F1, Macro-F1
	Table Type Classification	F1
	Cell Entity Recogonization	FI
TMP	Table Relation Extraction	F1, Precision, Recall
	Table Entity Linking	F1, Freeision, recan
	Table Relation Prediction	
	Entity-based Relation Retrieval	Accuracy at $n$ (Precision at $n$ )
-	Column Population	Mean Average Precision (MAP), Mean Reciprocal Rank
TCP	Row Population	(MRR), Recall, Normalized Discounted Cumulative Gain at $n$ (NDCG- $n$ )
	Cell Filling	Precision at n
	Classification	F1, AUROC, Accuracy
TP	Regression	Root Mean Square Error (RMSE), Mean Absolute Error (MAE), $\mathbb{R}^2$
TFC	-	Accuracy
TG	-	Bilingual Evaluation Understudy (BLEU), Average Normalized Edit Distance (ANED)

Table 4 summarizes the downstream tasks and sample sizes of the most prevalent 2D tabular datasets identified, where 'x' indicates the tasks applicable to each dataset. The majority of 2D datasets are applicable or designed for TQA. Table 5 demonstrates the table metadata, context information, and task related context associated with each dataset. 2D tabular datasets can be either unlabeled or labeled. Unlabeled datasets such as Wikipedia Tables and WDC Web Table Corpus [80] are vast collections of tables, each containing information on a specific topic. They are often used for pre-training to facilitate the language model's understanding of semantics and knowledge of tabular data. In contrast, labeled datasets with task-specific annotations are used during the fine-tuning stage. For instance, the WikiTableQuestion dataset [81] comprises table-associated questions and corresponding answers. Data types within 2D tabular data can be more complicated, which requires more nuanced reasoning. For example, in MULTIHIERTT [20], each document contains multiple tables, which often exhibit hierarchical structures with multi-level headers. Tables in HybridQA [82] incorporate hyperlinks leading to Wikipedia passages, requiring multi-hop reasoning. Furthermore, tables within the dataset can be inter-related. In Spider [83], the task involves joining multiple tables within a database by keys.

 $\begin{tabular}{l} Table 3: Summary of 1D tabular datasets including prediction sub-tasks, sample size, feature size and models using the datasets. \end{tabular}$ 

Dataset	Prediction Sub-Task	Sample Size	Feature Size	Example
Income [51]	Classification	48,842	14	STab [53], TABLET [54], TabuLa [47]
Diabetes [52]	Classification	768	8	STUNT [55], TABRET [56], TAP- TAP [45]
Forest Cover Type [57]	Classification	581,012	52	DANets [44], TabNet [58], DANets, MET [59]
California Housing [60]	Regression	20,640	8	Gorishniy et al. [61], GReaT [62], REaLTabFormer [48]
MNIST (tabular) [49]	Classification	7,000	784	SDAT [63], MET [59], Contrastive Mixup [64]
Qsar Bio [65]	Classification	1,055	41	PTab [66], TabPFN [67], Schambach et al. [68]
Credit-g [69]	Classification	1,000	20	UniTab E $[31],$ TabPFN $[67],$ Liu et al. $[70]$
Higgs Boson [71]	Classification	11M	28	TabNet [58], CT-BERT [41], Schambach et al. [68]
BlogFeedback [50]	Regression	60,021	280	VIME [72], SubTab [73], SDAT [63]
Bank Marketing [74]	Classification	45,211	16	TabTransformer [75], TabLLM [76], PTab [66]
BlastChar <sup>1</sup>	Classification	7,043	20	TabTransformer [75], SAINT [77], PTab [66]
MovieLens-1M [78]	Top-n rank- ing	1M	7	RIM [23], PET [24], Table2Graph [25]
Car [79]	Classification	1,728	6	TabLLM [76], SPROUT [46], CT-BERT [41]

 $<sup>^{1}\;</sup> https://www.kaggle.com/datasets/blastchar/telco-customer-churn$ 

Furthermore, a special type of 2D tabular dataset incorporates dynamic temporal information, with some entries recorded over time along with their corresponding timestamps. Tables within the dataset covering various domains are linked by unique identifiers. For instance, MIMIC is a large publicly available electronic health record dataset, with its latest version being MIMIC-IV [94]. It originates from the deidentified records of the Beth Israel Deaconess Medical Center, covering admissions from 2008 to 2019. MIMIC-IV adopts a modular structure, featuring three tabular schemas adopted relational structure: Hospital, ICU, and Emergency Department. These database schemas contain comprehensive patient information including demographics, laboratory measurements, medications, vital signs and more. Additionally,

Table 4: Properties of 2D tabular datasets including downstream tasks and sample size.

Dataset		Ι	Oownstr	eam Ta	sk		Sample
Dataset	TFC	$\mathbf{TQA}$	TSP	$\mathbf{TR}$	TMP	TCP	Size
Wikipedia Tables <sup>1</sup>		x	x	x	x	x	~2.62M
WDC Web Table Corpus [80]		x	x		x		~233M
WikiTableQuestion [81]		X	x				2,108
WikiSQL $[84]$		x	x	x			24,241
Spider [83]			x	x			1,020
SQA [85]		x	x				982
MULTIHIERTT [20]		x					~9.8K
FeTaQA [86]		x	x				10,330
TAT-QA [87]		x					2,757
HybridQA [82]		x					13,000
ToTTo [88]		x					83,141
TabFact [89]	x						16,573

 $<sup>^1</sup>$  WikiTable: https://github.com/bfetahu/wiki\_tables

MIMIC-IV includes modules covering other medical data modalities: Note, Diagnostic Electrocardiogram, and Chest X-ray. UK Biobank is a large-scale prospective biomedical database comprising 500,000 individuals of middle and old age, recruited between 2016 and 2010 across the United Kingdom. It is a multimodal dataset containing an extensive amount of clinical information, such as patient demographics, detailed questionnaires, and various physical measurements. Additionally, the dataset also collects multimodal imaging data and genome-wide genotyping. The dataset is linked to electronic health records allowing for longitudinal follow-up to track health outcomes.

# 3 Language Modeling Techniques on Tabular Data

In this section, we analyze language model techniques and their adaptation to tabular data. Key components of tabular data modeling include input processing techniques, attention techniques, and training objectives.

#### 3.1 Input Processing

For transformer-based language models, it is typical to use text sequences as inputs. Prior to the standard embedding process, tabular data must be preprocessed through several steps to ensure compatibility with LM-based tabular models. These steps include data retrieval, table serialization, and context integration. Data retrieval plays a key role in certain frameworks, helping to maintain compliance with the language model's input size limits and boosting training efficiency. Table serialization involves

Table 5: Summary of context of 2D tabular datasets with models using the datasets.

Dataset	Other Modalidities	Example
Wikipedia Tables <sup>1</sup>	Table Metadata: titles, captions, and NL contexts	Table 2Vec [22], TURL [38], TAB-BIE [37]
WDC Web Table Corpus [80]	Table Metadata: table orientation, header rows, key columns Context Information: the title of the HTML page, the caption of the table, the text before and after the table, and timestamps from the page	TaBERT [18], TUTA [90]
WikiTableQuestion [81]	Task Related Content: 22,033 question-answer pairs	TABLEFORMER [91], Table-QAKit [36], StructGPT [27]
WikiSQL [84]	Task Related Content: 80,654 questions, answers and SQL queries	TAPEX [26], TableQAKit [36], TAPAS [92]
Spider [83]	Task Related Content: 10,181 questions and 5,693 SQL queries	Tabert [18], TUTA [90], Struct-GPT [27]
SQA [85]	Task Related Content:         6,066 question sequences containing           17,553 questions-answer pairs	TABLEFORMER [91], GPT4Table [32], UniTabPT [93]
MULTIHIERTT [20]	Table Metadata: hierarchical column headers and row headers Context Information: unstructured text Task Related Content: 10,440 question-answer pairs, annotations of reasoning processes and supporting facts	MULTIHIERTT [20], NAPG [19], TableQAKit [36]
FeTaQA [86]	Table Metadata: page title, section title  Task Related Content: 10,330 question, free-form answer, and supporting table cells	DATER [33], UniTabPT [93]
TAT-QA [87]	Context Information: 2,757 associated paragraphs  Task Related Content: 16,552 question-answer pairs with scale, derivation to arrive at the answer, and answer source	TAT-QA [87], TableQAKit [36]
HybridQA [82]	Context Information: 293,269 hyperlinked passages Task Related Content: 69,611 question-answer pairs	GPT4Table [32], TableQAKit [36]
ToTTo [88]	Table Metadata: page title, section title, section text Task Related Content: highlighted cells, final text (annotation)	GPT4Table [32], UniTabPT [93]
TabFact [89]	Task Related Content: 117,854 annotated statements	TAPEX [26], TABLEFORMER [91], DATER [33]

 $<sup>^{1}\</sup> WikiTable:\ https://github.com/bfetahu/wiki_tables$ 

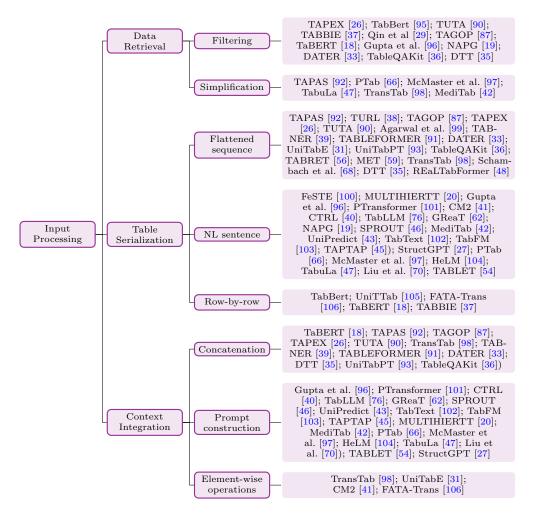


Fig. 4 The taxonomy of input processing. It contains data retrieval, table serialization and content integration.

modifying the original tabular data to fit the input specifications of language models. Context integration is also critical for effectively modeling tabular data, particularly when using pre-trained language models. The taxonomic overview of input processing is illustrated in Figure 4.

# 3.1.1 Data Retrieval

The primary objective of the data retrieval module in LM-based tabular models is to expedite the training process, thereby cutting down on computational demands and training duration, especially when dealing with large tabular datasets. This module also plays a crucial role in ensuring that tabular inputs adhere to the token limits of language models. Typically, data retrieval strategies fall into two main categories.

Filtering. Filtering aims to reduce the number of rows or columns to be processed in large tables. Several straightforward techniques have been introduced for this purpose. For example, TAPEX [26] employs random selection of rows to manage input size. TabBert [95] suggest using a sliding window to sample consecutive rows. TUTA [90] split large tables into non-overlapping rows with same header, while TABBIE [37] and Qin et al [29] truncate tables to a preset row and column limit to prevent memory issues. TAGOP [87] retains only those tables with number of rows and columns below a fixed threshold. Other methods, such as TaBERT [18], Gupta et al. [96], NAPG [19], DATER [33], TableQAKit [36], DTT [35], incorporate specialized modules for extracting the most relevant top-n rows from tables.

Simplification. Simplification entails condensing the data by streamlining table contents. Some approaches simplify the content and context information such as TAPAS [92], PTab [66], McMaster et al. [97], TabuLa [47]. This might involve abbreviating or replacing column names and categorical values with synonymous terms. Additional methods, such as TransTab [98], MediTab [42], opt to omit negative binary features. This strategy is especially effective in reducing computational and memory requirements when dealing with inputs that have high-dimensional, sparse one-hot features.

#### 3.1.2 Table Serialization

Table serialization involves converting tabular data into a format suitable for LM-based tabular systems. This process can be categorized into three different types.

Flattened sequence. The table is converted into flattened sequence, which transforms tabular data, either 1D or 2D, into a predefined flattened sequence format. This method is widely adopted due to its simplicity and directness. Figure 5 provides an overview of this method for table serialization. More precisely, for 2D tables, the prevalent methodology involves reformatting these tables into a linear sequence of words pieces. This process integrates specific separators, such as [CLS] for representation generation, and [SEP] and [ROW] for row separation, to efficiently structure the data. Significant research in this domain includes TAPAS [92], TURL [38], TAGOP [87], TAPEX [26], TUTA [90], Agarwal et al. [99], TABNER [39], TABLEFORMER [91], DATER [33], UniTabE [31], UniTabPT [93], and TableQAKit [36]. For tasks that require additional information besides tabular data, such as QA tasks, the tokens from the additional information (e.g., query or question) are usually positioned ahead of the flattened table-derived tokens. In contrast, models designed for 1D tables often rely solely on representation generator to facilitate a streamlined sequence, as seen in TABRET [56], MET [59], TransTab [98], Schambach et al. [68], DTT [35], and REalTabFormer [48].

Natural language (NL) sentence. It is typically used with pre-trained language models to make full use of the textual information. This adaptation is essential for training and using pre-trained models, particularly in tasks such as masked language modeling, where understanding language senmantics is crucial. Many systems have designed the prompts for each feature to be transformed into natural sentences, and then concatenate them into a whole NL sentences for finetuning (FeSTE [100], MULTIHIERTT [20], Gupta et al. [96], PTransformer [101], CM2 [41], CTRL [40], TabLLM

# 2D Tabular Data 1D Tabular Data OR CLS Feature tokens in one cell CLS Token for sample representation [SEP/ROW] Token for row separation

Fig. 5 The illustration of flattened sequence for 1D (down) and 2D (up) data in table serialization.

[76], GReaT [62], NAPG [19], SPROUT [46], MediTab [42]), UniPredict [43], TabText [102], TabFM [103], TAPTAP [45]), and StructGPT [27]. Alternative methodologies have been explored to achieve this objective by simplifying language prompts with separators (PTab [66], McMaster et al. [97], HeLM [104], TabuLa [47], Liu et al. [70], TABLET [54]). These transformed formats, despite their modifications, continue to align with the domain of NL sentences.

Row-by-row. It involves processing data on a row-by-row basis, typically employed for handling data where each row represents a essential entity, with an emphasis on exploring the interactions among various rows. This approach is predominantly utilized in longitudinal tabular data, where each row symbolizes the attributes at a specific timestamp. Processing longitudinal tabular data in this manner aids in uncovering both intra and inter-temporal information. For example, TabBert [95] introduced a field transformer to discern local associations within a single record at a given time step, while its sequence transformer captures broader relationships across different time steps, thus addressing the temporal aspects of the data. UniTTab [105] builds upon TabBert's framework by incorporating a linear projection layer, enabling the processing of rows with varying internal structures and types within time series. Based on TabBert, FATA-Trans [106] designs a unique approach to separately process static and dynamic fields, integrating time interval information through a time-aware position embedding that operates on a row-by-row basis.

Moreover, some models have adopted row-by-row table serialization to align with their architectural designs, extending beyond just longitudinal tabular data application. For instance, TaBERT [18] introduces vertical self-attention, which operates across vertically aligned encoded vectors from different rows. Similarly, TABBIE [37] employs a row Transformer to encode cells along each row of the table, complemented by a column Transformer that performs an analogous function across columns, thereby facilitating the generation of contextualized cell embeddings.

#### 3.1.3 Context Integration

In LM-based tabular data modeling, the integration of table context with its content is a pivotal step.

Concatenation. The most straightforward technique involves the sequential combination of the context with table data, typically achieved through serial concatenation. A notable example is TaBERT [18], which merges column names, types, and cell values in table serialization with the [SEP] symbol as a separator. Alternatively, some approaches prefer concatenating all column names prior to all cell values as illustrated in Figure (TAPAS [92], TAGOP [87], TAPEX [26], TUTA [90], TransTab [98], TABNER [39], TABLEFORMER [91], DATER [33], DTT [35], UniTabPT [93], TableQAKit [36]).

Prompt construction. Another commonly adopted technique of context integration utilizes prompt construction, often in conjunction with pre-trained language models on tabular datasets. This approach often entails the transformation of context data into prompts, augmented with additional words or separators. Some methods implements the combination of column names with connective words to NL sentences (Gupta et al. [96], PTransformer [101], CTRL [40], TabLLM [76], GReaT [62], SPROUT [46], UniPredict [43], TabText [102], TabFM [103], TAPTAP [45]). Additionally, MULTIHIERTT [20] employs a Facts Retrieving Module to convert table metadata into NL context sentences, while MediTab [42] leverages LLMs to transform tabular data into NL sentences, utilizing column names as the contextual information. For simplicity, some models such as PTab [66], McMaster et al. [97], HeLM [104], TabuLa [47], Liu et al. [70]), TABLET [54], StructGPT [27], directly concatenate column names with cell values, using colons or spaces as contextual separators during the pre-training phase of the model.

Element-wise operations. Furthermore, there are approaches that incorporate context through element-wise operations such as addition and multiplication. For instance, TransTab [98] implements element-wise product between column name embeddings and cell value embeddings for binary and numerical features. UniTabE [31] introduces a unique linking layer, designed to intricately fuse information from column names with their corresponding cell values. Similarly, CM2 [41] utilizes an element-wise multiplication approach, combining normalized numerical values with their respective header embeddings. FATA-Trans [106] adopts an element-wise summation technique, combining record embeddings, field type embeddings, and time-aware position embeddings to generate sequence embeddings with the sequential encoding transformer.

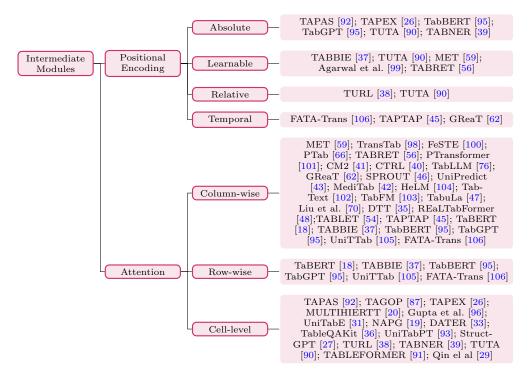


Fig. 6 The taxonomy of intermediate modules. It contains positional encoding and attention mechanism.

Apart from these methods, some models have developed specialized modules for context integration. For example, TABBIE [37] concatenates column names with cell values using row and column transformers. Qin et al. [29] enhance large-scale tabular pre-training models by integrating a common-sense knowledge graph, providing a flexible and easily integrable solution.

#### 3.2 Intermediate Modules

In addition to input processing techniques, researchers are also interested in modifying intermediate modules within transformer architectures to better adapt them to the tabular domain. As illustrated in Figure 6, this includes two major components: positional encoding and attention mechanisms.

#### 3.2.1 Positional Encoding

Positional encoding serves as a fundamental technique within deep learning methodologies, particularly evident in the architecture of transformers, facilitating the model's ability to capture crucial ordering information in input sequences. This necessity arises from the distinct processing mechanism of transformers compared to recurrent neural networks (RNNs) or long short-term memory networks (LSTMs), which inherently process data sequentially. It's worthy noting that transformers require a method to integrate positional information into the frameworks. Therefore, some adaptations have been proposed to incorporate additional positional encoding for rows or columns to explicitly capture the underlying table structure.

Absolute. The first form of positional encoding, known as absolute positional encoding, was initially employed in the original transformer model architecture to inject information about token positions within a sequence. It comprises sinusoidal functions of different frequencies to encode each position, enabling the model to understand the sequence's order. This approach has also been adapted for tabular data modeling to retain information concerning the tabular structure, as observed in works such as TAPAS [92], TAPEX [26], TabBERT, and TabGPT [95]. TUTA [90] utilized tree-based positional encodings to better capture hierarchical information. However, in TABNER [39], the authors noted that, positional encoding across the entire table could blur the Named Entity Recognition (NER) training signal for BERT.

Learnable. Unlike the fixed mathematical formula used in absolute encoding, learned positional encodings comprise learnable parameters that the model can update during training. This enables the model to learn an embedding for each position, potentially capturing more intricate patterns than the absolute encoding. For instance, TABBIE [37] and TUTA [90] introduce various tree-based positional embedding approaches tailored to the hierarchical nature of tabular data, with parameters initialized randomly and tuned through training. MET [59], Agarwal et al. [99], and TABRET [56] develop learnable positional encodings to encode column-specific information.

Relative. Relative positional encoding differs from absolute encoding by encoding the relative positions of tokens in input sequences rather than their absolute positions. It allows the model to focus on the distance between tokens in each modality, where the relative positioning of elements holds greater significance than their absolute position. Techniques such as those employed in TURL [38], which provides a positional embedding vector containing relative position information for a token within captions or headers, and TUTA [90], which incorporates in-cell position encoding to encode single tokens relative to their cell positions, exemplify the application of relative positional encoding within various contexts.

**Temporal.** In practical settings involving the modeling of longitudinal tabular data, temporal information such as timestamps is available and informative. Despite their informational richness, these temporal information is often underutilized in conventional approaches. In addressing this limitation, FATA-Trans [106] introduces a novel approach: the exploration of learnable time-aware position embedding. This innovative technique considers both the sequence order and time intervals between rows, enabling the model to effectively understand underlying temporal patterns within a sequence.

Notably, tabular data exhibits permutation invariance, implying that its structure remains unchanged despite arbitrary rearrangements of columns. Consequently, certain frameworks in the tabular domain opt to forgo positional encoding to mitigate the introduction of positional bias, particularly when modeling 1D tabular data [41, 75]. For some models that utilize natural language sentences as a table serialization method, permutation functions are proposed to shuffle the order of features.

This approach aims to mitigate the impact of implicit injection on spurious positional relationships within textual encoding [45, 62].

#### 3.2.2 Attention

Attention mechanisms in language modeling have significantly advanced the field of NLP by allowing models to dynamically focus on different parts of the input data when producing an output. This capability has been particularly impactful in dealing with sequential data, such as text, where understanding the context and the relationships between words or tokens is crucial for tasks like translation, summarization, and question-answering.

Tabular data, characterized by its structured format in rows and columns, presents a challenge in applying attention mechanisms, compared to sequential text data. However, the concept of attention can still be beneficial in this context, especially for tasks that involve understanding relationships between different rows or columns within the table.

Column-wise. Column-wise attention mechanisms are dominantly used in the analysis of tabular datasets as it helps to understand the data structure by capturing the relationship between columns. This could be useful in tasks where the output depends on understanding complex interactions between features. In the domain of 1D tabular data analysis, researchers have proposed to directly use stacked transformer to incorporate attention mechanisms on tabular data, as illustrated in the works MET [59], TransTab [98], FeSTE [100], PTab [66], TABRET [56], PTransformer [101], CM2 [41], CTRL [40]. These models leverage the strengths of transformer architectures to capture complex feature interactions effectively. Further innovations have emerged through the integration of LLMs, focusing on serialized tabular data and NL prompts. Key advancements include TabLLM [76], GReaT [62], SPROUT [46], UniPredict [43], MediTab [42],HeLM [104], TabText [102], TabFM [103], TabuLa [47], Liu et al. [70], DTT [35], REaLTabFormer [48], TABLET [54], TAPTAP [45]. These studies highlight the potential of leveraging attention mechanisms on tabular data, opening new avenues for column-wise feature interaction and processing of tabular information.

For 2D tabular data, particularly with complex tasks such as question answering and information retrieval, the deployment of attention mechanisms also plays a pivotal role in facilitating the aggregation of information across column-wise features and, when applicable, across different modalities. For example, TaBERT [18] leverages attention in transformer to to bridge the gap in relationship exploration between questions and relevant rows. Similarly, TABBIE [37] utilizes row transformers to compute row embeddings, thereby enabling the generation of column-wise contextualized cell embeddings. Moreover, in handling longitudinal tabular data, the advent of field transformers marks a significant advancement by being attentive on column-wise features at individual timestamps to extract valuable information, as demonstrated in TabBERT, TabGPT [95] and UniTTab [105]. FATA-Trans [106] introduces both static and dynamic field transformers to cater to varying column-wise data types, enhancing its utility across a broad spectrum of tasks using longitudinal tabular data.

Row-wise. To explore relationships between multiple rows within tabular data, particularly for tasks like time series prediction, the multi-head attention mechanism

emerges as a crucial tool by weighting the importance of different rows. A notable implementation of this is the vertical attention mechanism proposed in TaBERT [18], which aggregates information across diverse rows in a content snapshot, enabling TaBERT to effectively capture cross-row dependencies on cell values. Furthermore, TABBIE [37] introduces a column transformer with a row transformer to produce row-wise contextualized representations that enrich the model's interpretability. For longitudinal tabular data, sequence encoding transformer serves as a fundamental framework on the top of field transformer to effectively integrate time-sensitive information across multiple rows through the attention mechanisms ( e.g., TabBERT, TabGPT [95], UniTTab [105], and FATA-Trans [106].)

Cell-level. Attention can also be applied at the cell level, where the model learns to focus on specific cells within the table that are most relevant for the task at hand. This approach can be particularly useful for dealing with heterogeneous data or data with high-dimensional features for complex downstream tasks such as question answering, information retrieval. Many methodologies leverage BERT [9] or its variants [107–109] to implement attention on table cells within 2D tabular data through pre-training (TAPAS [92], TAGOP [87], TAPEX [26], MULTIHIERTT [20], Gupta et al. [96], UniTabE [31], NAPG [19]) or via PLM/LLM (DATER [33], Table-QAKit [36], UniTabPT [93], StructGPT [27]). Notably, beyond employing multi-head self-attention, TURL [38] and TABNER [39] introduce a visibility matrix as an attention mask, enabling the information aggregation only on structurally related tokens during self-attention calculation. Further advancements include TUTA [90], which incorporates structure-aware tree-based attention to capture spatial and hierarchical information in tables, and TABLEFORMER [91], which enhances table understanding and alignment with text by introducing task-independent relative attention biases.

In addition to the previous attention mechanisms on tabular data, some work has developed cross-attention mechanism that integrate information from table and other modalities. For instance, Qin et al [29] have proposed path-wise attention layer to align the cross-domain representation with the weighted contribution on tabular data and external knowledge graph.

#### 3.3 Training Objectives

Training objectives in language modeling are designed to help the model learn the semantics of natural language, enabling it to perform well on various downstream tasks. In tabular data modeling, the training objectives are typically modified to be adapted into the structure of tabular data. An overview of these training objectives for language modeling on tabular data is illustrated in Figure 7.

#### 3.3.1 Masking

The Masked Language Model (MLM) is firstly introduced in the pre-training phase of BERT [9]. In the MLM, some tokens within a sequence of text are randomly masked, and the model is trained to predict these masked tokens based on the surrounding context. This task helps the model understand the contextual relationships between tokens.

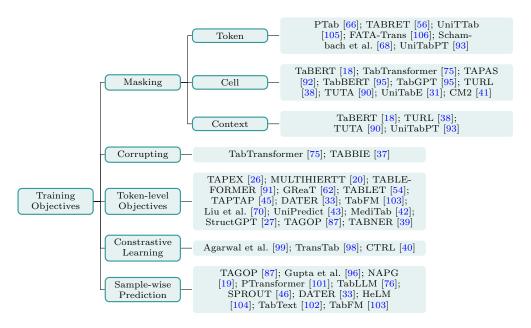


Fig. 7 The taxonomy of training objectives. It includes masking, corrupting, token-level objectives, contrastive learning and sample-wise prediction.

**Token.** Several studies [56, 66, 68, 93, 105, 106] have adopted the token-level masking as their main training objective, which involves randomly masking a specific percentage of tokens with [MASK] to facilitate the acquisition of contextual representations from text-based tabular data.

Cell. In contrast to conventional NLP pre-training by masking a random subset of tokens, several studies [18, 38, 75, 90, 92, 95] have embraced the approach of masking whole cells and training models to predict these masked cells. Building upon this strategy of whole cell masking, CM2 [41] introduces the utilization of column names as prompts to enhance whole cell masking effectiveness in cross-table pre-training.

Context. Beyond the focus on understanding table contents through token-level and cell-level masking strategies, table context—including metadata, surrounding texts, and even query questions—plays a critical role in tabular data modeling for complex tasks alongside table contents. For example, TaBERT [18] incorporates an MLM objective with a 15% masking rate for sub-tokens in NL contexts and a Masked Column Prediction objective to facilitate model learning of column names and types. TURL [38] and TUTA [90] utilize an MLM objective to enable models to grasp the lexical, semantic, and contextual information within table metadata for cell understanding. UniTabPT [93] also employs the MLM objective on NL/SQL texts and table headers. It's noted that instead of masking several tokens in header, the model masks all tokens in single header to bolster the model's capability in recovering column information from context, thereby deepening its understanding of the relationship between table column values and their headers.

#### 3.3.2 Corrupting

Apart from the conventional MLM-based training objective, corrupting emerges as another potent strategy for tabular representation learning. This approach is particularly beneficial for tabular data, where detecting corrupted cells is crucial for accurate table structure decomposition tasks, in which incorrectly identified row/column separators or cell boundaries can result in corrupted cell contents. For example, TabTransformer [75] implements this by replacing features with random values and utilizing a binary classifier to determine feature corruption. TABBIE [37] further introduces two corruption methods: frequency-based cell sampling and intra-table cell swapping, with the former showing significant effectiveness across most downstream table-based tasks.

### 3.3.3 Token-level Objectives

Token-level objectives include two main categories, the foremost being seq2seq (sequence-to-sequence) learning. This approach involves training models to produce text sequences based on given inputs, proving especially beneficial for machine translation, summarization, and question answering applications. Existing studies [20, 26, 27, 33, 42, 43, 45, 54, 62, 70, 91, 103] have explored the application of seq2seq learning objectives within the tabular domain. This exploration involves reinterpreting traditional tasks, such as classification, as question answering task on tables. In addition to predicting the tokens conditioned on previous input, models can further be tailored to predict a variety of token-level characteristics, including but not limited to part-of-speech tags, named entities, syntactic configurations, and classifications at the token level. An illustrative case is TABNER [39], which validates the utility of language models for addressing a unique named entity recognition challenge within industrial spreadsheets. Similarly, TAGOP [87] adeptly extracts supporting evidence from a hybrid context, containing both tables and related texts, through sequence tagging. This approach also facilitates the introduction of a Number Order Classifier [87], aimed at determining the sequential positioning of two numerals in the final outcome.

#### 3.3.4 Contrastive Learning

Contrastive learning in language models brings representations of similar text sequences closer and pushes dissimilar ones apart. In tabular data modeling, this approach is adapted for various purposes. Agarwal et al. [99] have introduced an additional contrastive loss component to mask prediction loss, treating embeddings from different masked instances of the identical user as positive examples. Furthermore, TransTab [98] presents a novel vertical-partition contrastive learning approach, which substantially increases the scope of positive and negative samples for learning more informative sample representation. CTRL [40] takes tabular and corresponding textual data as two modalities, employing contrastive learning for knowledge alignment and integration.

#### 3.3.5 Sample-wise Prediction

Sample-wise prediction involves the training of models through the utilization of sample representations, commonly the [CLS] token, for the predictions of various tasks. A plethora of studies [33, 46, 76, 96, 101–104] that leveraged PLMs/LLMs, which do not require extensive training corpora, have adopted this approach for training their models across diverse downstream tasks. Moreover, TAGOP [87] introduces an Operator Classifier that accounts for the correct scale in addition to the accurate number. Concurrently, NAPG [19] has unveiled a non-autoregressive program generation framework equipped with five predictors based on sample representations for numerical reasoning. This innovation enables the parallel generation of programs, significantly enhancing efficiency in numerical reasoning tasks.

# 4 Evolution of How Language Models are Adapted

In addition to providing foundational concepts and modeling techniques, this survey emphasizes the evolution of language model adaptation for tabular data and explores potential future research directions. As illustrated in Figure 1, tabular modeling initially benefited from language modeling in two major ways: pre-training from scratch, where researchers focused on critical elements (e.g., embeddings, attention) within language models to improve decision-making, and using PLMs, where some researchers integrated PLMs as modules within larger models to leverage prior semantic knowledge for addressing more complex tasks. In recent years, the field of NLP has advanced into the era of LLMs, marking a series of paradigm shifts that have introduced novel capabilities and enhanced the efficiency of tabular data processing.

#### 4.1 Pre-training From Scratch and PLMs

Prior to the emergence of LLMs, tabular modeling predominantly followed two distinct approaches based on the objectives.

#### 4.1.1 Pre-training From Scratch

The first approach includes models trained from scratch, primary designed for tabular prediction tasks. To improve their ability to extract relevant features and correlations from tables, researchers predominantly adopt transformer architectures, integrating fundamental language modeling techniques. This included modifications to embedding methods and training objectives to better handle tabular data. For instance, Table2Vec [22] learns meaningful embeddings for table elements such as captions, column headings, and cells, aiding tasks such as row population and table retrieval. TabTransformer [75] uses Transformer architectures with a self-attention mechanism to convert categorical feature embeddings into strong contextual embeddings, which are combined with continuous features to boost prediction accuracy. This model is pre-trained with multiple objectives, including masked language modeling and replaced token detection, making it robust against noisy and missing data. CM2 [41] introduces an efficient crosstable pre-training framework featuring a semantic-aware tabular model that uniformly encodes heterogeneous tables with minimal restrictions. It utilizes a novel pre-training

objective called prompt masked table modeling, enabling scalable pre-training on diverse tables. Schambach et al. [68] propose a Transformer-based architecture for cross-table representation learning, employing table-specific tokenizers and a shared Transformer backbone to minimize inductive biases. This approach includes both single-table and cross-table models, trained using a self-supervised masked cell recovery objective, improving the model's ability to learn representations across different tables.

Moreover, researchers are adapting transformer-based architecture to different applications including table understanding, time series analysis and tabular prediction. Table understanding represents an advanced task in tabular data modeling, with several models leveraging the transformer architecture to enhance comprehension of table structures. For example, TAGOP [87] introduces an innovative questionanswering model based on RoBERTa that reasons over both tables and text. It uses sequence tagging to extract relevant cells from tables and text spans, applying symbolic reasoning with aggregation operators to derive answers. On the other hand, TUTA [90] provides a unified pre-training architecture for structured tables, utilizing tree-based attention and positional embeddings to effectively capture both spatial and hierarchical information. The model incorporates three pre-training objectives: masked language modeling, cell-level cloze tasks, and table context retrieval, which collectively facilitate token, cell, and table-level representations. Pre-trained on a vast array of diverse web and spreadsheet tables, TUTA is subsequently fine-tuned for tasks such as cell and table type classification. TURL [38] introduces a structure-aware Transformer encoder that models relational tables using a Masked Entity Recovery objective, effectively capturing semantics and knowledge from large-scale unlabeled data, and demonstrating strong performance across six table understanding tasks. Similarly, TABLEFORMER [91] presents a robust structure-aware table-text encoding architecture that incorporates learnable attention biases to mitigate biass from table linearization, outperforming strong baselines in numerical experiments on datasets like SQA, WTQ, and TABFACT.

In addition to aforementioned tasks on 2D tabular data, time series data is special type of 2D tabular data, on which recent advancements in transformer-based architectures have revolutionized the modeling. TabBERT and TabGPT [95] are pioneers in this field, with TabBERT enabling end-to-end pre-training for classification or regression tasks, and TabGPT generating realistic synthetic tabular sequences while preserving patient privacy. FATA-Trans [106] innovatively utilizes two field transformers to distinguish static and dynamic fields and employs time-aware position embeddings to capture temporal patterns. Agarwal et al. [99] introduce a self-supervised transformer model that learns from both sequential and tabular features, excelling in tasks such as supervised classification and click bot detection without labels. UniTTab [105], on the other hand, focuses on heterogeneous time-dependent data, using continuous embedding vectors for numerical and categorical features, and is uniformly trained with a masked token task, showing robust performance across various tasks. Additionally, UniTabE [31] presents a versatile framework for handling tables, addressing pre-training challenges on large-scale tabular data using free-form

prompts to encode inputs into a Transformer encoder, with a curated dataset of approximately 13 billion samples from Kaggle aiding its pre-training phase.

Tabular prediction utilizing 1D tabular data has significantly benefited from language modeling techniques, resulting in enhanced prediction performance. For instance, MET [59] proposes a reconstruction-based approach for tabular representation learning using masked encoding to reconstruct original input data through a series of stacked transformers. TransTab [98] introduces a transferable tabular Transformer architecture that converts each data sample into a generalizable embedding vector, processed through stacked Transformers for feature encoding, incorporating column descriptions and table cells into a gated transformer model. It uses both supervised and self-supervised pre-training methods to boost performance. TABRET [56], a pre-trainable Transformer-based model, adapts to unseen columns in downstream tasks with an extra retokenizing step before fine-tuning, calibrating feature embeddings based on masked autoencoding loss.

While these methods have demonstrated effectiveness in specific contexts, they often required considerable amounts of tabular data for pre-training from scratch in order to perform well for specific tasks. This issue is especially pronounced in data-sensitive fields such as healthcare, where the availability of extensive, high-quality data is often limited, and privacy concerns are critical.

#### 4.1.2 Pre-trained Language Models

The second approach involves the use of PLMs within tabular modeling. These models are especially suited for tasks requiring a deeper semantic understanding, such as TQA. PLMs like BERT are favored in this approach due to their foundational pre-trained architectures, which require less training data and yield superior predictive performance compared to earlier methods. Additionally, these pre-trained models allow for fine-tuning on task-specific datasets, enhancing both the efficiency and effectiveness of the modeling process.

For the TQA task, TaBERT [18] proposes pre-trained language model built on top of Bert to simultaneously process textual and tabular data. It uses content snapshots and vertical self-attention mechanisms to capture the association between table content and natural language. TAPAS [92] is a weakly supervised question answering model that extends BERT's masked language model objective to structured data. It answers questions by selecting relevant cells and performing aggregation operations, bypassing the need for generating logical forms. Employing a unique pre-training strategy, TAPAS was further trained on millions of tables and associated text segments sourced from Wikipedia. TAPEX [26] develops an innovative table pre-training method that simulates a neural SQL executor to learn structured tabular data effectively. It overcomes the scarcity of high-quality tabular data by utilizing synthetic SQL queries and their execution results as a pre-training corpus. MULTIHIERTT [20] is a benchmark specifically crafted for complex numerical reasoning tasks involving documents with multiple hierarchical tables and textual content, primarily derived from financial reports. The dataset is distinguished by several features: it includes documents with multiple tables and extensive unstructured texts, predominantly hierarchical tables, and necessitates intricate reasoning more challenging than current benchmarks. It also

provides detailed annotations of reasoning paths and supporting facts. To address the challenges in MULTIHIERTT, a novel question-answering model, MT2NET is proposed. It uses the RoBERTa model as encoder and is the first one that conducts fact retrieval to pinpoint relevant facts before engaging a reasoning module for symbolic processing. TableQuery [110] introduces a tool for querying tabular data with natural language, overcoming limitations of existing deep learning methods for question answering on tabular data. TableQuery utilizes pre-trained LM for free-text question answering to convert natural language queries into structured queries, avoiding the need to load large datasets into memory or serialize databases. It supports various column types and does not require re-training, allowing for easy integration of better-performing models. NAPG [19] proposes a non-autoregressive program generation framework based on the RoBERTa encoder, specifically designed for numerical reasoning in mixed tabular-textual question answering scenarios. It uniquely generates complete program tuples, including both operators and operands, independently, which effectively mitigates the exposure bias problem seen in traditional autoregressive models and drastically increases generation speed.

PLMs are also used for other tasks such as TP, for example, FeSTE [100] is a Transformer-based framework based on Bert and designed to enrich tabular datasets by leveraging unstructured data. It trains on various datasets to create versatile models suitable for additional datasets using few-shot learning. FeSTE introduces a fine-tuning method that converts dataset tuples into sentences and uses pre-trained LM to generate valuable features from external data sources effectively. PTab [66] proposes a framework that utilizes pre-trained LM (Bert) to enhance the contextual representation of tabular data, enabling training on mixed datasets. The framework processes tabular data in three stages: Modality Transformation (MT), Masked Language Fine-Tuning (MF), and Classification Fine-Tuning (CF). P-Transformer [101] devises a prompt-based multimodal transformer architecture tailored for medical tabular data. This framework comprises two key components: a tabular cell embedding generator based on RoBERTa and a tabular transformer. The embedding generator effectively encodes inputs from both structured and unstructured tabular data into a unified language semantic space, utilizing a pre-trained sentence encoder and specialized medical prompts. The transformer component then integrates these cell representations to create comprehensive patient embeddings, which are applied across diverse medical tasks. CTRL [40] is a framework that enhances Click-Through Rate (CTR) prediction by integrating collaborative and language models. It transforms tabular data into text format, which is then processed using both a collaborative model and a pre-trained LM (RoBERTa). This approach allows for the alignment of collaborative and semantic signals via cross-modal learning.

In addition to TQA and TP, researchers have also demonstrated that PLMs enhance performance in other semantically demanding tasks. For example, TABNER [39] proposes a table transformer model tailored for the industrial Named Entity Recognition (NER) challenge, designed to identify entities in complex, structured spreadsheets. It incorporates a domain-specific data augmentation technique that uses knowledge graphs to enhance performance in low-resource environments and address the technical complexities of industrial data. The research underscores the importance

of tabular inductive bias for model convergence and shows that this data augmentation method markedly boosts performance compared to sequential models. TABBIE [37] introduces a self-supervised model tailored for tabular data, employing a corrupt cell detection objective to learn table structure and semantics. It initializes cell embeddings using BERT and utilizes two distinct Transformers to separately encode rows and columns. TABBIE leverages the self-supervised ELECTRA objective for pre-training and excels in various downstream tasks such as column population, row population, and column type prediction. Qin et al. [29] proposes dual-adapters within a pre-trained tabular Transformer model to bridge domain discrepancies between external knowledge sources and tabular data. These dual-adapters operate in parallel: one adapter is trained on knowledge graph triplets, while the other processes semantically enhanced tables. Furthermore, a path-wise attention layer is integrated to enhance the cross-domain representation. Gupta et al. [96] explores enhancing NLP systems' interpretability and reliability through Trustworthy Tabular Inference. This task ensures systems substantiate their predictions with clear evidence. Employing a two-stage approach, the methodology begins by extracting evidence from tables, then uses this data to predict inference labels.

# 4.2 Emergence of Large Language Models

Large language models have recently achieved remarkable success in the domain of natural language understanding. These LLMs often have billions of parameters and pre-trained on enormous text corpus, leading to strong zero-shot or few-shot capability on a variety of tasks such as reasoning, question answering, and code generation. Hence, it naturally raises the question of how can we leverage the advantage of LLMs to push the boundaries of tabular data modeling.

In contrast to most **Pre-training From Scratch** and **PLMs** which are designed for specific tasks, researchers have begun to explore the use of a single unified model for a broader and more challenging range of tabular tasks, such as few-shot classification and table generation.

In the Table Prediction domain, TabLLM [76] explore LLMs' proficiency in zero-shot and few-shot classification tasks specifically applied to tabular data. The performance of LLMs is notably sensitive to the specific details of the natural language input. Therefore, TabLLM studied nine different serialization techniques and found that simple text template (e.g., the [column\_name] is [value]) perform best in almost all datasets for zero-shot and few-shot classifications. Following serialization, TabLLM uses a short task description as a prompt to obtain output probabilities from the LLM and then finetune the LLM. TAPTAP [111] proposes table pre-training for tabular prediction, where the model was pre-trained on 450 tabular datasets with a total of nearly 2 million samples. TAPTAP pre-train the model by prediction the subsequent token using serialized tables. With pre-training, TAPTAP is capable of generating high-quality synthetic tables, which can support various applications involving tabular data. SPROUT [112] extends the application of LLMs to semi-supervised tabular data modeling. It first provides a LLM with a selection of labeled samples and prompts it to identify the most important feature for the downstream task. In order to maintain the

relevance of the constructed examples and the real downstream task, SPROUT generates prompts that predict the selected important features based upon the remaining column features. The generated prompts are then combined with descriptions of a few labeled samples to further bolster the in-context-learning performance of the LLM. UniPredict [43] presents an universal predictor for tabular data classification tasks. It aimed to create a more adaptable tabular model, wherein a single set of parameters could be applied universally across datasets from any domain. To achieve this goal, UniPredict integrates the prompts for serialized tables and target confidence from external predictors to finetune its backbone LLM. MediTab [42] focuses on medical tabular data where existing models are often trained on a single dataset for a specific task, leading to poor generalizibility. It uses a LLM to consolidate the tabular data: by describing one sample in a variety of ways, MediTab can generates diverse consolidated samples as data augmentation. Meanwhile, MediTab employed an audit module to prevent potential hallucinations from LLM. To benefit from the out-domain datasets, MediTab expands the available data for one task by aligning a trained model on all other different tasks to obtain supplementary data. HeLM [104] presents a multimodal LLM for health domain with the use of individual specific data. Unlike other tabular models in health domain, HeLM integragtes non-tabular data such as medical time series into the serialization process. Specifically, it trains separate encoders for non-textal data modality on top of LLM to learn their mapping to the same representation space as text. HeLM has shown effectiveness for zero/few-shot disease prediction on UK biobank dataset. However, it was also observed that the tuned model degraded in conversational ability. TabText [102] leverages the text embeddings of the serialized table as additional contexutal information to improve the performance of standard machine learning performance. An significant advantage of TabText is that the LLM is flexible to replace when new models available. TABLET[54] is a benchmark designed to assess the ability of LLMs to learn from instructions for tabular prediction tasks. It encompasses 20 different tabular datasets, each annotated with instructions that differ in phrasing, granularity, and technicality. TABLET enables researchers to gauge model performance in tabular predictions based on in-context instructions alone (the zero-shot setting) or with a limited number of labeled examples (the few-shot setting). Interestingly, the findings indicate that while instructions generally enhance LLM performance, LLMs exhibit a strong bias against accurately classifying certain instances and do not consistently follow the provided in-context instructions. Liu et al. [70] investigates the fairness issues associated with employing pre-trained LLMs for tabular prediction tasks. They found that LLMs, specifically GPT-3.5 turbo, often depend on social biases present in their pre-training data for making predictions in tabular tasks. While few-shot in-context learning can somewhat reduce these biases, it does not completely remove them. In addition, the gap in fairness metrics across different subgroups remains larger compared to traditional machine learning models, such as Random Forest and shallow Neural Networks. TabFM [103] introduces a method for training a LLM based foundation model specifically for tabular data. It involves incorporating examples from 115 tabular datasets, which are serialized into text for the training process. TabFM aims to create a model that effectively generalizes to new

tabular datasets in both zero and few-shot settings. The results reveal that their training approach lead to better performance on tabular tasks compared to other models like GPT4 and TabLLM. Moreover, when trained with an increased number of data points, TabFM achieves performance comparable to traditional tabular models such as XGBoost.

Meanwhile, by leveraging the powerful language generation capabilities of LLMs, researchers have used these models to generate more complex and realistic tables. GReaT [62] introduces a novel approach for the modeling and generation of realistic, heterogeneous tabular data, leveraging the capabilities of LLMs. Contrary to conventional tabular data generation techniques which convert the data into a purely numerical format, GReaT encodes tabular data into textual representations, effectively capturing the underlying semantics. GReaT implements a random feature order permutation on the serialized tabular data and finetunes the pretrained LLM on samples devoid of order dependencies. This approach enables arbitrary conditioning in the generation of tabular data. In the generation phase, GReaT feeds the LLMs a text description, prompting them to proceed with completion. REaLTabFormer [48] proposes a generative model for relational tabular data generation. REaLTabFormer contains a partent table model and a child table model to generate synthetic observations and synthetic related observations, respectively. It uses a GPT-2 model to generate a parent table, and subsequently produce the child tables that conditioned on the parent table using a GPT-2 decoder. TabuLa [47] proposes several approaches to enhance tabular data synthesis. Unlike traditional methods that rely on pre-trained models, TabuLa utilizes a randomly initialized model, enabling quicker adaptation to the specific requirements of tabular data synthesis tasks. Furthermore, TabuLa compresses all column names and categorical values into a single token each, effectively reducing sequence length. It also implements a middle padding strategy, ensuring that features within the same data column retain their absolute positions in the encoded token sequence, thus preserving the integrity of the original data structure. DTT [35] explores the challenge of converting tabular data from one source format to another target format using only a few examples. The approach begins by decomposing the problem into smaller subtasks and serializing the input. It then employs a LLM (ByT5) to predict outcomes for each subtask. An aggregator combines these individual predictions to produce a final output. Experimental evaluations indicate that the performance of DTT matches or surpasses that of other LLMs such as GPT3, despite notable differences in size. Furthermore, DTT also enhances the performance of LLMs in the table transformation task by integrating them into it.

Additionally, researchers have focused on harnessing the strong reasoning capabilities of LLMs for table understanding tasks such as TQA. For example, Dater [113] explores the application of LLMs for table-based reasoning tasks. It focuses on addressing two main issues associated with table-based reasoning: (i) LLMs are unable to directly encoding large tables due to input token limits; (ii) direct decompose complex questions could easily fall into a hallucination dilemma. Dater tackles the two main challenges by leveraging the in-context learning in LLMs. It first decompose a large table into a small table that relevant to the question using LLM alongside a handful of prompting examples. Following this, Dater decompose a complex question

into simpler step-by-step sub-questions. Notably, Dater operates without task-specific finetuning that might diminish the LLMs' in-context capabilities. GPT4Table [32] establishes a benchmark aimed at studying which factors of input design most significantly influence LLMs' ability to understand tabular data. The proposed Structural Understanding Capabilities (SUC) benchmark includes seven tasks, each designed to compare different input configurations. The findings indicate that while LLMs possess fundamental abilities in interpreting tabular data, correctly choosing the input design could be as a key element in enhancing the SUC. Therefore, authors proposed a self-augmented prompting method to provide additional knowledge and constraints, thus enhancing the LLM's performance in downstream tasks. Chain-Of-Table [114] has further augmented the reasoning abilities of LLMs by utilizing the tabular structure to generate intermediate thought processes during table-based reasoning tasks. It guides LLMs to dynamically create a sequence of operations in response to a given table and its related question, offering more accurate and reliable predictions.

The emergence of these LLM-based tabular modeling methods represents a paradigm shift from models pre-trained from scratch or PLMs designed for specific tasks to LLM-based methods capable of handling various tasks. This shift opens up new opportunities and challenges for future exploration.

# 5 Challenges and Future Opportunities

As discussed in Section 4.2, LLMs has already been used in many tabular data applications, such as predictions, data synthesis, question answering and table understanding. Here we outline some practical limitations and considerations for future research. Specially, we discuss four challenges for language modeling on tabular data, consisting of computation efficiency, interpretability, biases, and data types. Computation efficiency and interpretability are general challenges in AI application scenarios with PLMs and LLMs. Biases is also a general concern, but this problem has different meaning when focus on tabular data. Data types is a particular challenge for tabular data. For other LLMs/PLMs application scenarios, it is rare to simultaneously face many different data type, such as numerical, categorical, binary, text, timestamps, and even nested data.

# 5.1 Computation Efficiency

Compared to methods based on data mining or PLMs, using LLMs significantly require more computational resources. This increase in computational resources is not only a cost issue but also a practical challenge in scenarios where sufficient hardware support is not available, such as when mobile devices are required. To address this issue, there are two directions worth exploring. The first solution is that how to improve the efficiency of training and inference for large models. There are numerous studies focused on enhancing the efficiency of LLMs, such as the use of Adapter modules [115–117]. This technique involves integrating smaller neural network modules into the intermediate layers of PLMs or LLMs. Another method is Prefix Tuning [118, 119], where a trainable prefix is added either to the input sequence or to the hidden layers to facilitate more efficient training. LoRA [120] represents an advanced strategy for achieving

parameter-efficient fine-tuning while circumventing common issues associated with other methods. The fundamental principle of LoRA is to approximate the parameter updates of a full-rank weight matrix using a low-rank matrix. However, these methods, while speeding up the training process, do not provide help when perform inference. Meanwhile, deploying such a large model remains challenging.

The second direction is that leveraging more powerful LLMs to enhance the performance of smaller models to approach larger ones. For instance, the studies [121–123] let a LLM act as a teacher and distil knowledge into a small model. Besides, some studies [124, 125] employ LLMs to generate synthetic data to further enhance smaller models. However, knowledge distillation faces several challenges, including the inability to precisely control the information being distilled, a tendency for overfitting, and high sensitivity to the quality of data. When using synthetic data, while it can enhance the scale of model training, it also carries the risk of introducing biases which may skew the results. Furthermore, evaluating synthetic data presents additional complexities. For such reason, addressing these issues requires further exploration of more effective methods to enhance computational efficiency.

# 5.2 Interpretability

Over the last decade, interest in interpretability has surged, driven by the proliferation of large datasets and the advancement of deep neural networks. Despite LLMs demonstrating exceptional capabilities across diverse applications, the depth of interpretability research remains relatively superficial. Research on neural network interpretability can be categorized into two distinct groups. The first focuses on result interpretability. For instance, when a neural model is asked a question such as, "There are 5 apples on the table. If Aria used 2 to make dinner and then added 10 more apples on the table, how many apples are on the table?" a typical model might simply respond, "There are 13 apples on the table". This response lacks an explanation of the reasoning process. However, with the introduction of Chain of Thought (CoT) technology [126], models are now capable of explaining their reasoning in a step-by-step manner, such as "Starting with 5 apples, 2 were used, leaving 3. Adding the 10 apples gives a total of 13." This approach significantly enhances interpretability by allowing models to articulate the thought process behind their conclusions, essentially using the models themselves to explain their reasoning. Recent advancements in retrieval argument generation (RAG) and the integration of knowledge graph represent similar efforts to bolster interpretability.

The second research stream theoretically explores the internal mechanisms of LLMs to shed light on their operations. For example, the study by Ansuini et al.[127] utilizes the concept of intrinsic dimensionality to analyze network layers. It reveals that the intrinsic dimensionality of each layer in a trained network is significantly smaller than the number of units in each layer, compared to an untrained network. This phenomenon aids in assessing the generalization performance of the network and enhances our understanding of neural network interpretability. Another study[128] employs various attribution methods to determine the relevance or contribution of each input variable to the final output. Despite these efforts, most studies in this area face

strong constraints or focus only on specific aspects, making a generalized approach to interpreting neural networks a continuing challenge.

#### 5.3 Biases

LLMs often inherit biases from their training data, affecting their fairness in tasks like TQA, TR, and TMP. For instance, Liu et al. [70] assessed the fairness of GPT-3.5 in tabular predictions using few-shot learning. The findings confirmed that biases in training data significantly contribute to model biases. Additionally, Mao et al. [129] showed that PLMs and LLMs might produce biased outcomes based on task design, prompt crafting, and label word selection. The study highlighted how variations in label definitions and arrangements can affect model effectiveness and introduce biases.

For tabular data modeling, serialization introduces specific biases. Common methods like flattening the table sequence, converting tables into natural language sentences, or employing row-by-row serialization all result in some loss of table structure information, and make tables into plain input sequences. However, according to Liu et al. [130], LLMs tend to focus more on the beginning and end of the input sequence, often overlooking the middle sections, which complicates the retrieval of relevant information. Each serialization approach also impacts how much attention each cell receives, introducing potential biases.

Given these issues, exploring more effective serialization methods is crucial. Moreover, addressing the inherent biases of LLMs deserves more focused attention to enhance model fairness and efficacy.

# 5.4 Data Types

Diverse data types is one of the most challenging issue in language modeling tabular data. In other fields and tasks, it is rare to simultaneously handle numerical, categorical, binary, text, hyperlinks, timestamps data, and even these data exist nested scenario. Traditionally, data mining-based approaches perform well in handling multiple data types, by developing specific modeling methods for each type. However, these approaches significantly lag behind LLM-based methods in tasks involving text understanding, generation, and reasoning. Consequently, more research is shifting towards LLM-based methods for addressing tabular data issues. However, LLMs are not naturally suited for handling multiple data types, especially numerical data. While some studies [20, 41, 76] have circumvented this issue by converting numeric data into text, this solution significantly increases the model's input length, introducing new problems.

Currently, there are LLMs studies [104, 111] specifically trained on table data, which somewhat enhance the model's ability to understand various data types. Nevertheless, the following issues remain worthy of further exploration: 1) During training, how to balance the proportion of different types of data in large-scale table data, maintaining data integrity while ensuring the diversity and scale of training data. 2) How to design more appropriate unsupervised pre-training tasks. Unlike traditional language models, table data may involve different levels and granularities of attention

interactions between rows, columns, and the entire table, and sometime even existed nested tables. Under such condition, various data types present additional challenges.

# 6 Conclusion

Language models for tabular data capitalize on innovative techniques to harness the intricacies of heterogeneous data structures, thereby enabling analytics across diverse applications. This comprehensive survey delves into the core aspects of language modeling for tabular data from various dimensions, including foundational structures, methodologies, and the evolution of modeling techniques. A detailed taxonomy and analysis of techniques, ranging from data retrieval to the strategic use of intermediate modules and training objectives, offer a granular view of the field's current state. The evolution of language modeling is critically reviewed in two pivotal stages: (i) Initial explorations involving bespoke pre-training adjustments for tabular contexts and the integration of pre-trained models like BERT for enhanced semantic understanding. (ii) The recent shift towards leveraging LLMs represents a significant paradigm shift, broadening the scope for addressing more complex tabular data challenges. This paper not only highlights the substantial progress made but also articulates four major challenges and proposes potential avenues for future research in this promising and rapidly evolving area, making it a clear roadmap for navigating future explorations in language modeling for tabular data.

### **Declarations**

### **Funding**

This work is supported by the Cisco-NUS Accelerated Digital Economy Corporate Laboratory (Award I21001E0002, funded by A\*STAR, CISCO Systems (USA) Pte. Ltd, and National University of Singapore).

#### Competing interests

The authors declare no conflict of interest.

#### Data availability

No datasets were generated or analysed during the current study.

#### Author contribution

Y.R. & X.L.: Conceptualization, Methodology, Resources, Data Acquisition, Writing(Original Draft Preparation), Writing(Review & Editing), Visualization; J.M. & Y.D.: Resources, Data Acquisition, Writing(Original Draft Preparation); K.H.: Writing(Original Draft Preparation); M.F.: Writing(Review & Editing), Supervision, Funding Acquisition

## References

- [1] Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: Deepfm: a factorization-machine based neural network for ctr prediction. arXiv preprint arXiv:1703.04247 (2017)
- [2] Clements, J.M., Xu, D., Yousefi, N., Efimov, D.: Sequential deep learning for credit risk monitoring with tabular financial data. arXiv preprint arXiv:2012.15330 (2020)
- [3] Somani, S., Russak, A.J., Richter, F., Zhao, S., Vaid, A., Chaudhry, F., De Freitas, J.K., Naik, N., Miotto, R., Nadkarni, G.N., et al.: Deep learning and the electrocardiogram: review of the current state-of-the-art. EP Europace 23(8), 1179–1191 (2021)
- [4] Bonacin, R., Vechi, S.M., Dametto, M., Ruppert, G.C.S.: Exploring deep learning techniques in the prediction of cancer relapse using an open brazilian tabular database. In: International Conference on Information Technology-New Generations, pp. 331–338 (2024). Springer
- [5] Gandhar, A., Gupta, K., Pandey, A.K., Raj, D.: Fraud detection using machine learning and deep learning. SN Computer Science 5(5), 1–10 (2024)
- [6] Shwartz-Ziv, R., Armon, A.: Tabular data: Deep learning is not all you need. Information Fusion 81, 84–90 (2022)
- [7] Gianfrancesco, M.A., Goldstein, N.D.: A narrative review on the validity of electronic health record-based research in epidemiology. BMC medical research methodology **21**(1), 234 (2021)
- [8] Ghebrehiwet, I., Zaki, N., Damseh, R., Mohamad, M.S.: Revolutionizing personalized medicine with generative ai: a systematic review. Artificial Intelligence Review **57**(5), 1–41 (2024)
- [9] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
- [10] Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146 (2018)
- [11] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems 33, 1877–1901 (2020)
- [12] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient

- foundation language models. arXiv preprint arXiv:2302.13971 (2023)
- [13] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog 1(8), 9 (2019)
- [14] Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., Kasneci, G.: Deep neural networks and tabular data: A survey. IEEE Transactions on Neural Networks and Learning Systems (2022)
- [15] Singh, R., Bedathur, S.: Embeddings for tabular data: A survey. arXiv preprint arXiv:2302.11777 (2023)
- [16] Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., Sun, L.: Transformers in time series: A survey. arXiv preprint arXiv:2202.07125 (2022)
- [17] Badaro, G., Saeed, M., Papotti, P.: Transformers for tabular data representation: A survey of models and applications. Transactions of the Association for Computational Linguistics 11, 227–249 (2023)
- [18] Yin, P., Neubig, G., Yih, W.-t., Riedel, S.: TaBERT: Pretraining for joint understanding of textual and tabular data. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8413–8426. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.acl-main.745 . https://aclanthology.org/2020.acl-main.745
- [19] Zhang, T., Xu, H., Genabith, J., Xiong, D., Zan, H.: Napg: Non-autoregressive program generation for hybrid tabular-textual question answering. In: CCF International Conference on Natural Language Processing and Chinese Computing, pp. 591–603 (2023). Springer
- [20] Zhao, Y., Li, Y., Li, C., Zhang, R.: Multihiertt: Numerical reasoning over multi hierarchical tabular and textual data. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 6588–6600 (2022)
- [21] Webber, W., Moffat, A., Zobel, J.: A similarity measure for indefinite rankings. ACM Trans. Inf. Syst. 28(4), 20–12038 (2010) https://doi.org/10.1145/1852102. 1852106
- [22] Zhang, L., Zhang, S., Balog, K.: Table2vec: Neural word and entity embeddings for table population and retrieval. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1029–1032 (2019)
- [23] Qin, J., Zhang, W., Su, R., Liu, Z., Liu, W., Tang, R., He, X., Yu, Y.: Retrieval &

- interaction machine for tabular data prediction. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 1379–1389 (2021)
- [24] Du, K., Zhang, W., Zhou, R., Wang, Y., Zhao, X., Jin, J., Gan, Q., Zhang, Z., Wipf, D.P.: Learning enhanced representation for tabular data via neighborhood propagation. Advances in Neural Information Processing Systems 35, 16373–16384 (2022)
- [25] Zhou, K., Liu, Z., Chen, R., Li, L., Choi, S.-H., Hu, X.: Table2graph: Transforming tabular data to unified weighted graph. In: IJCAI, pp. 2420–2426 (2022)
- [26] Liu, Q., Chen, B., Guo, J., Ziyadi, M., Lin, Z., Chen, W., Lou, J.-G.: Tapex: Table pre-training via learning a neural sql executor. In: International Conference on Learning Representations (2021)
- [27] Jiang, J., Zhou, K., Dong, Z., Ye, K., Zhao, W.X., Wen, J.-R.: Structgpt: A general framework for large language model to reason over structured data. arXiv preprint arXiv:2305.09645 (2023)
- [28] Zhong, V., Xiong, C., Socher, R.: Seq2sql: Generating structured queries from natural language using reinforcement learning. CoRR abs/1709.00103 (2017) 1709.00103
- [29] Qin, C., Kim, S., Zhao, H., Yu, T., Rossi, R.A., Fu, Y.: External knowledge infusion for tabular pre-training models with dual-adapters. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1401–1409 (2022)
- [30] Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA, pp. 311–318. ACL, ??? (2002). https://doi.org/10.3115/1073083.1073135. https://aclanthology.org/P02-1040/
- [31] Yang, Y., Wang, Y., Liu, G., Wu, L., Liu, Q.: Unitabe: Pretraining a unified tabular encoder for heterogeneous tabular data. arXiv preprint arXiv:2307.09249 (2023)
- [32] Sui, Y., Zhou, M., Zhou, M., Han, S., Zhang, D.: GPT4Table: Can Large Language Models Understand Structured Table Data? A Benchmark and Empirical Study (2023)
- [33] Ye, Y., Hui, B., Yang, M., Li, B., Huang, F., 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)
- [34] Marzal, A., Vidal, E.: Computation of normalized edit distance and applications. IEEE Trans. Pattern Anal. Mach. Intell. 15(9), 926–932 (1993) https://doi.org/ 10.1109/34.232078
- [35] Nobari, A.D., Rafiei, D.: Dtt: An example-driven tabular transformer by leveraging large language models. arXiv preprint arXiv:2303.06748 (2023)
- [36] Lei, F., Luo, T., Yang, P., Liu, W., Liu, H., Lei, J., Huang, Y., Wei, Y., He, S., Zhao, J., et al.: Tableqakit: A comprehensive and practical toolkit for table-based question answering. arXiv preprint arXiv:2310.15075 (2023)
- [37] Iida, H., Thai, D., Manjunatha, V., Iyyer, M.: Tabbie: Pretrained representations of tabular data. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 3446–3456 (2021)
- [38] Deng, X., Sun, H., Lees, A., Wu, Y., Yu, C.: Turl: Table understanding through representation learning. ACM SIGMOD Record **51**(1), 33–40 (2022)
- [39] Koleva, A., Ringsquandl, M., Buckley, M., Hasan, R., Tresp, V.: Named entity recognition in industrial tables using tabular language models. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track, pp. 348–356 (2022)
- [40] Li, X., Chen, B., Hou, L., Tang, R.: Ctrl: Connect tabular and language model for ctr prediction. arXiv preprint arXiv:2306.02841 (2023)
- [41] Ye, C., Lu, G., Wang, H., Li, L., Wu, S., Chen, G., Zhao, J.: Ct-bert: learning better tabular representations through cross-table pre-training. arXiv preprint arXiv:2307.04308 (2023)
- [42] Wang, Z., Gao, C., Xiao, C., Sun, J.: MediTab: Scaling Medical Tabular Data Predictors via Data Consolidation, Enrichment, and Refinement (2023)
- [43] Wang, R., Wang, Z., Sun, J.: Unipredict: Large language models are universal tabular predictors. arXiv preprint arXiv:2310.03266 (2023)
- [44] Chen, J., Liao, K., Wan, Y., Chen, D.Z., Wu, J.: Danets: Deep abstract networks for tabular data classification and regression. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 3930–3938 (2022)
- [45] Zhang, T., Wang, S., Yan, S., Li, J., Liu, Q.: Generative table pre-training empowers models for tabular prediction. arXiv preprint arXiv:2305.09696 (2023)
- [46] Nam, J., Song, W., Park, S.H., Tack, J., Yun, S., Kim, J., Shin, J.: Semi-supervised tabular classification via in-context learning of large language models.

- In: Workshop on Efficient Systems for Foundation Models@ ICML2023 (2023)
- [47] Zhao, Z., Birke, R., Chen, L.: Tabula: Harnessing language models for tabular data synthesis. arXiv preprint arXiv:2310.12746 (2023)
- [48] Solatorio, A.V., Dupriez, O.: Realtabformer: Generating realistic relational and tabular data using transformers. arXiv preprint arXiv:2302.02041 (2023)
- [49] Deng, L.: The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE signal processing magazine **29**(6), 141–142 (2012)
- [50] Buza, K.: BlogFeedback. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C58S3F (2014)
- [51] Becker, B., Kohavi, R.: Adult. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20 (1996)
- [52] Smith, J.W., Everhart, J.E., Dickson, W., Knowler, W.C., Johannes, R.S.: Using the adap learning algorithm to forecast the onset of diabetes mellitus. In: Proceedings of the Annual Symposium on Computer Application in Medical Care, p. 261 (1988). American Medical Informatics Association
- [53] Hajiramezanali, E., Diamant, N.L., Scalia, G., Shen, M.W.: Stab: Self-supervised learning for tabular data. In: NeurIPS 2022 First Table Representation Workshop (2022)
- [54] Slack, D., Singh, S.: Tablet: Learning from instructions for tabular data. arXiv preprint arXiv:2304.13188 (2023)
- [55] Nam, J., Tack, J., Lee, K., Lee, H., Shin, J.: Stunt: Few-shot tabular learning with self-generated tasks from unlabeled tables. arXiv preprint arXiv:2303.00918 (2023)
- [56] Onishi, S., Oono, K., Hayashi, K.: Tabret: Pre-training transformer-based tabular models for unseen columns. In: ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models (2023)
- [57] Blackard, J.: Covertype. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C50K5N (1998)
- [58] Arik, S.Ö., Pfister, T.: Tabnet: Attentive interpretable tabular learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 6679–6687 (2021)
- [59] Majmundar, K.A., Goyal, S., Netrapalli, P., Jain, P.: Met: Masked encoding for tabular data. In: NeurIPS 2022 First Table Representation Workshop (2022)

- [60] Pace, R.K., Barry, R.: Sparse spatial autoregressions. Statistics & Probability Letters 33(3), 291–297 (1997)
- [61] Gorishniy, Y., Rubachev, I., Babenko, A.: On embeddings for numerical features in tabular deep learning. Advances in Neural Information Processing Systems 35, 24991–25004 (2022)
- [62] Borisov, V., Sessler, K., Leemann, T., Pawelczyk, M., Kasneci, G.: Language models are realistic tabular data generators. In: The Eleventh International Conference on Learning Representations (2022)
- [63] Fang, J., Tang, C., Cui, Q., Zhu, F., Li, L., Zhou, J., Zhu, W.: Semi-supervised learning with data augmentation for tabular data. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, pp. 3928–3932 (2022)
- [64] Darabi, S., Fazeli, S., Pazoki, A., Sankararaman, S., Sarrafzadeh, M.: Contrastive mixup: Self-and semi-supervised learning for tabular domain. arXiv preprint arXiv:2108.12296 (2021)
- [65] Mansouri, K., Ringsted, T., Ballabio, D., Todeschini, R., Consonni, V.: QSAR biodegradation. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5H60M (2013)
- [66] Liu, G., Yang, J., Wu, L.: Ptab: Using the pre-trained language model for modeling tabular data. arXiv preprint arXiv:2209.08060 (2022)
- [67] Hollmann, N., Müller, S., Eggensperger, K., Hutter, F.: Tabpfn: A transformer that solves small tabular classification problems in a second. arXiv preprint arXiv:2207.01848 (2022)
- [68] Schambach, M., Paul, D., Otterbach, J.: Scaling experiments in self-supervised cross-table representation learning. In: NeurIPS 2023 Second Table Representation Learning Workshop (2023)
- [69] Hofmann, H.: Statlog (German Credit Data). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5NC77 (1994)
- [70] Liu, Y., Gautam, S., Ma, J., Lakkaraju, H.: Investigating the fairness of large language models for predictions on tabular data. In: Socially Responsible Language Modelling Research (2023)
- [71] Whiteson, D.: HIGGS. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5V312 (2014)
- [72] Yoon, J., Zhang, Y., Jordon, J., Schaar, M.: Vime: Extending the success of selfand semi-supervised learning to tabular domain. Advances in Neural Information

- Processing Systems 33, 11033–11043 (2020)
- [73] Ucar, T., Hajiramezanali, E., Edwards, L.: Subtab: Subsetting features of tabular data for self-supervised representation learning. Advances in Neural Information Processing Systems **34**, 18853–18865 (2021)
- [74] Moro, S., Rita, P., Cortez, P.: Bank Marketing. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5K306 (2012)
- [75] Huang, X., Khetan, A., Cvitkovic, M., Karnin, Z.: Tabtransformer: Tabular data modeling using contextual embeddings. arXiv preprint arXiv:2012.06678 (2020)
- [76] Hegselmann, S., Buendia, A., Lang, H., Agrawal, M., Jiang, X., Sontag, D.: Tabllm: Few-shot classification of tabular data with large language models. In: International Conference on Artificial Intelligence and Statistics, pp. 5549–5581 (2023). PMLR
- [77] Somepalli, G., Goldblum, M., Schwarzschild, A., Bruss, C.B., Goldstein, T.: Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. arXiv preprint arXiv:2106.01342 (2021)
- [78] Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis) **5**(4), 1–19 (2015)
- [79] Bohanec, M.: Car Evaluation. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5JP48 (1997)
- [80] Lehmberg, O., Ritze, D., Meusel, R., Bizer, C.: A large public corpus of web tables containing time and context metadata. In: Proceedings of the 25th International Conference Companion on World Wide Web, pp. 75–76 (2016)
- [81] Pasupat, P., Liang, P.: Compositional semantic parsing on semi-structured tables. arXiv preprint arXiv:1508.00305 (2015)
- [82] Chen, W., Zha, H., Chen, Z., Xiong, W., Wang, H., Wang, W.: Hybridqa: A dataset of multi-hop question answering over tabular and textual data. arXiv preprint arXiv:2004.07347 (2020)
- [83] Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., et al.: Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. arXiv preprint arXiv:1809.08887 (2018)
- [84] Zhong, V., Xiong, C., Socher, R.: Seq2sql: Generating structured queries from natural language using reinforcement learning. arXiv preprint arXiv:1709.00103 (2017)
- [85] Iyyer, M., Yih, W.-t., Chang, M.-W.: Search-based neural structured learning

- for sequential question answering. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1821–1831 (2017)
- [86] Nan, L., Hsieh, C., Mao, Z., Lin, X.V., Verma, N., Zhang, R., Kryściński, W., Schoelkopf, H., Kong, R., Tang, X., et al.: Fetaqa: Free-form table question answering. Transactions of the Association for Computational Linguistics 10, 35–49 (2022)
- [87] Zhu, F., Lei, W., Huang, Y., Wang, C., Zhang, S., Lv, J., Feng, F., Chua, T.-S.: 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), pp. 3277–3287 (2021)
- [88] Parikh, A.P., Wang, X., Gehrmann, S., Faruqui, M., Dhingra, B., Yang, D., Das, D.: Totto: A controlled table-to-text generation dataset. arXiv preprint arXiv:2004.14373 (2020)
- [89] Chen, W., Wang, H., Chen, J., Zhang, Y., Wang, H., Li, S., Zhou, X., Wang, W.Y.: Tabfact: A large-scale dataset for table-based fact verification. arXiv preprint arXiv:1909.02164 (2019)
- [90] Wang, Z., Dong, H., Jia, R., Li, J., Fu, Z., Han, S., Zhang, D.: Tuta: Tree-based transformers for generally structured table pre-training. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 1780–1790 (2021)
- [91] Yang, J., Gupta, A., Upadhyay, S., He, L., Goel, R., Paul, S.: 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), pp. 528–537 (2022)
- [92] Herzig, J., Nowak, P.K., Mueller, T., Piccinno, F., Eisenschlos, J.: Tapas: Weakly supervised table parsing via pre-training. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4320–4333 (2020)
- [93] Sarkar, S., Lausen, L.: Testing the limits of unified sequence to sequence llm pretraining on diverse table data tasks. In: NeurIPS 2023 Second Table Representation Learning Workshop (2023)
- [94] Johnson, A., Bulgarelli, L., Pollard, T., Horng, S., Celi, L.A., Mark, R.: Mimic-iv. PhysioNet. Available online at: https://physionet.org/content/mimiciv/1.0/(accessed August 23, 2021), 49–55 (2020)
- [95] Padhi, I., Schiff, Y., Melnyk, I., Rigotti, M., Mroueh, Y., Dognin, P., Ross, J., Nair, R., Altman, E.: Tabular transformers for modeling multivariate time series.

- In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3565–3569 (2021). IEEE
- [96] Gupta, V., Zhang, S., Vempala, A., He, Y., Choji, T., Srikumar, V.: Right for the right reason: Evidence extraction for trustworthy tabular reasoning. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 3268–3283 (2022)
- [97] McMaster, C., Liew, D.F., Pires, D.E.: Adapting pretrained language models for solving tabular prediction problems in the electronic health record. arXiv preprint arXiv:2303.14920 (2023)
- [98] Wang, Z., Sun, J.: Transtab: Learning transferable tabular transformers across tables. Advances in Neural Information Processing Systems 35, 2902–2915 (2022)
- [99] Agarwal, R., Muralidhar, A., Som, A., Kowshik, H.: Self-supervised representation learning across sequential and tabular features using transformers. In: NeurIPS 2022 First Table Representation Workshop (2022)
- [100] Harari, A., Katz, G.: Few-shot tabular data enrichment using fine-tuned transformer architectures. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1577–1591 (2022)
- [101] Ruan, Y., Lan, X., Tan, D.J., Abdullah, H.R., Feng, M.: P-Transformer: A Prompt-based Multimodal Transformer Architecture For Medical Tabular Data (2024)
- [102] Carballo, K.V., Na, L., Ma, Y., Boussioux, L., Zeng, C., Soenksen, L.R., Bertsimas, D.: TabText: A Flexible and Contextual Approach to Tabular Data Representation. Jul (2023)
- [103] Zhang, H., Wen, X., Zheng, S., Xu, W., Bian, J.: Towards foundation models for learning on tabular data. arXiv preprint arXiv:2310.07338 (2023)
- [104] Belyaeva, A., Cosentino, J., Hormozdiari, F., Eswaran, K., Shetty, S., Corrado, G., Carroll, A., McLean, C.Y., Furlotte, N.A.: Multimodal llms for health grounded in individual-specific data. In: Workshop on Machine Learning for Multimodal Healthcare Data, pp. 86–102 (2023). Springer
- [105] Luetto, S., Garuti, F., Sangineto, E., Forni, L., Cucchiara, R.: One transformer for all time series: Representing and training with time-dependent heterogeneous tabular data. arXiv preprint arXiv:2302.06375 (2023)
- [106] Zhang, D., Wang, L., Dai, X., Jain, S., Wang, J., Fan, Y., Yeh, C.-C.M., Zheng, Y., Zhuang, Z., Zhang, W.: Fata-trans: Field and time-aware transformer

- for sequential tabular data. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pp. 3247–3256 (2023)
- [107] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
- [108] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 (2019)
- [109] Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (2019). Association for Computational Linguistics
- [110] Abraham, A.N., Rahman, F., Kaur, D.: Tablequery: Querying tabular data with natural language. arXiv preprint arXiv:2202.00454 (2022)
- [111] Zhang, T., Wang, S., Yan, S., Jian, L., Liu, Q.: Generative table pre-training empowers models for tabular prediction. In: Bouamor, H., Pino, J., Bali, K. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 14836–14854. Association for Computational Linguistics, Singapore (2023). https://doi.org/10.18653/v1/2023.emnlp-main.917. https://aclanthology.org/2023.emnlp-main.917
- [112] Nam, J., Song, W., Park, S.H., Tack, J., Yun, S., Kim, J., Shin, J.: Semi-supervised tabular classification via in-context learning of large language models. In: Workshop on Efficient Systems for Foundation Models @ ICML2023 (2023). https://openreview.net/forum?id=r77CeOBO0L
- [113] Ye, Y., Hui, B., Yang, M., Li, B., Huang, F., 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. SIGIR '23, pp. 174–184. Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3539618.3591708
- [114] Wang, Z., Zhang, H., Li, C.-L., Eisenschlos, J.M., Perot, V., Wang, Z., Miculicich, L., Fujii, Y., Shang, J., Lee, C.-Y., Pfister, T.: Chain-of-table: Evolving tables in the reasoning chain for table understanding. In: The Twelfth International Conference on Learning Representations (2024). https://openreview.net/forum?id=4L0xnS4GQM
- [115] He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., Neubig, G.: Towards a unified

- view of parameter-efficient transfer learning. arXiv preprint arXiv:2110.04366 (2021)
- [116] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: International Conference on Machine Learning, pp. 2790–2799 (2019). PMLR
- [117] Hu, Z., Lan, Y., Wang, L., Xu, W., Lim, E.-P., Lee, R.K.-W., Bing, L., Poria, S.: Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. arXiv preprint arXiv:2304.01933 (2023)
- [118] Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021)
- [119] He, K., Huang, Y., Mao, R., Gong, T., Li, C., Cambria, E.: Virtual prompt pretraining for prototype-based few-shot relation extraction. Expert Systems with Applications 213, 118927 (2023)
- [120] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
- [121] Ho, N., Schmid, L., Yun, S.-Y.: Large language models are reasoning teachers. arXiv preprint arXiv:2212.10071 (2022)
- [122] Hsieh, C.-Y., Li, C.-L., YEH, C.-K., Nakhost, H., Fujii, Y., Ratner, A.J., Krishna, R., Lee, C.-Y., Pfister, T.: Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In: The 61st Annual Meeting Of The Association For Computational Linguistics (2023)
- [123] Li, S., Chen, J., Shen, Y., Chen, Z., Zhang, X., Li, Z., Wang, H., Qian, J., Peng, B., Mao, Y., et al.: Explanations from large language models make small reasoners better. arXiv preprint arXiv:2210.06726 (2022)
- [124] Abdullin, Y., Molla-Aliod, D., Ofoghi, B., Yearwood, J., Li, Q.: Synthetic dialogue dataset generation using llm agents. arXiv preprint arXiv:2401.17461 (2024)
- [125] Schmidhuber, M., Kruschwitz, U.: Llm-based synthetic datasets: Applications and limitations in toxicity detection. LREC-COLING 2024, 37 (2024)
- [126] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems 35, 24824–24837 (2022)

- [127] Ansuini, A., Laio, A., Macke, J.H., Zoccolan, D.: Intrinsic dimension of data representations in deep neural networks. Advances in Neural Information Processing Systems 32 (2019)
- [128] Deng, H., Zou, N., Du, M., Chen, W., Feng, G., Yang, Z., Li, Z., Zhang, Q.: Understanding and unifying fourteen attribution methods with taylor interactions. arXiv preprint arXiv:2303.01506 (2023)
- [129] Mao, R., Liu, Q., He, K., Li, W., Cambria, E.: The biases of pre-trained language models: An empirical study on prompt-based sentiment analysis and emotion detection. IEEE Transactions on Affective Computing (2022)
- [130] Liu, N.F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., Liang, P.: Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics 12 (2024)