MediTab: Scaling Medical Tabular Data Predictors via Data Consolidation, Enrichment, and Refinement

Zifeng Wang[1] , **Chufan Gao**[1] , **Cao Xiao**[2] and **Jimeng Sun**[1,3]
[1]University of Illinois Urbana-Champaign
[2]GE Healthcare
[3]Carle Illinois College of Medicine, University of Illinois Urbana-Champaign
{zifengw2, chufan2, jimeng}@illinois.edu; cao.xiao@ge.com

Abstract

Tabular data prediction has been employed in medical applications such as patient health risk prediction. However, existing methods usually revolve around the algorithm design while overlooking the significance of data engineering. Medical tabular datasets frequently exhibit significant heterogeneity across different sources, with limited sample sizes per source. As such, previous predictors are often trained on manually curated small datasets that struggle to generalize across different tabular datasets during inference. This paper proposes to scale medical tabular data predictors (`MediTab`) to various tabular inputs with varying features. The method uses a data engine that leverages large language models (LLMs) to consolidate tabular samples to overcome the barrier across tables with distinct schema. It also aligns out-domain data with the target task using a "learn, annotate, and refinement" pipeline. The expanded training data then enables the pre-trained `MediTab` to infer for arbitrary tabular input in the domain without finetuning, resulting in significant improvements over supervised baselines: it reaches an average ranking of 1.57 and 1.00 on 7 patient outcome prediction datasets and 3 trial outcome prediction datasets, respectively. In addition, `MediTab` exhibits impressive zero-shot performances: it outperforms supervised XGBoost models by 8.9% and 17.2% on average in two prediction tasks, respectively.

1 Introduction

Tabular data are structured as tables or spreadsheets in a relational database. Each row in the table represents a data sample, while columns represent various feature variables of different types, including categorical, numerical, binary, and textual features. Most previous papers focused on the *model* design of tabular predictors, mainly by (1) augmenting feature interactions via neural networks [Arik and Pfister, 2021], (2) improving tabular data representation learning by self-supervised pre-training [Yin *et al.*, 2020; Yoon *et al.*, 2020; Bahri *et al.*, 2022], and (3) performing cross-tabular pre-training for transfer learning [Wang and Sun, 2022b; Zhu *et al.*, 2023]. Tabular data predictor was also employed in medicine, such as patient health risk prediction [Wang and Sun, 2022b; Ma *et al.*, 2023], clinical trial outcome prediction [Fu *et al.*, 2022], modeling Electronic Health Record (EHR) data [Ma *et al.*, 2020; Gao *et al.*, 2024], and unifying heterogeneous EHRs via text embeddings [Hur *et al.*, 2022]. Additionally, LLMs can sample synthetic and yet highly realistic tabular data as well [Borisov *et al.*, 2022; Theodorou *et al.*, 2023].

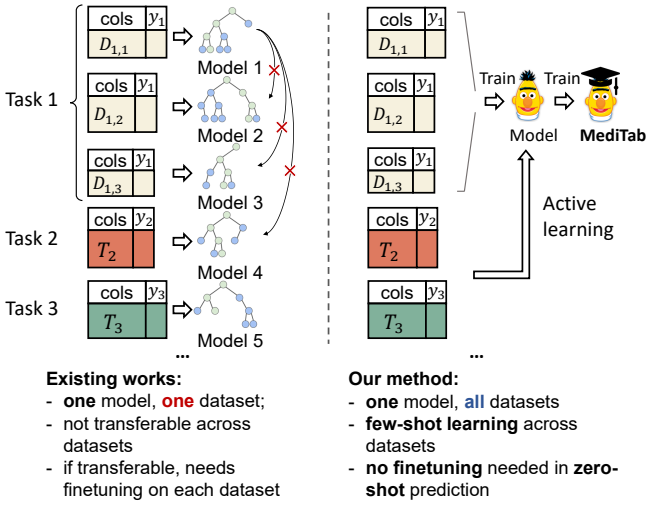

Figure 1: `MediTab` vs. existing tabular prediction methods. Existing methods learn and predict on a per-dataset basis, while `MediTab` can use data from the target task and all other tasks to improve performance.

Despite these significant advances, it is worth noting that the *data-centric* approaches have received comparatively less attention in prior research. Some prominent examples lie in the detection and mitigation of label noise [Wang *et al.*, 2020; Northcutt *et al.*, 2021], but they only address a fraction of the challenges in medical tabular data prediction. As illustrated in Figure 1, there is typically substantial heterogeneity among different data sources in medical data, and within each data source, the available sample sizes are small. Harnessing multi-source data requires extensive manual effort in terms of data cleaning and formatting. As such, current medical tabular prediction methods are often built on small handcrafted

datasets and, hence, do not generalize across tabular datasets.

In this paper, we embrace a data-centric perspective to enhance the scalability of predictive models tailored for medical tabular data. Our core aim revolves around training a single tabular data predictor to accommodate inputs with diverse feature sets. Technically, our framework, namely `MediTab`, encompasses three key components: *data consolidation*, *enrichment*, and *refinement* modules:

- **Data consolidation and enrichment** involves consolidating tabular samples with varying features and schemas using natural language descriptions. We also expand the training data by distilling knowledge from large language models and incorporating external tabular datasets.
- **Data refinement** rectifies errors and hallucinations introduced during the consolidation and enrichment stages. It also aligns a diverse set of tabular samples with the target task through a distantly supervised pipeline.

As illustrated in Figure 1, `MediTab` offers the advantages:

- **Multi-task learning and prediction**: the model can learn from and make predictions for multiple medical tabular datasets without requiring modifications or retraining.
- **Few-shot and zero-shot learning**: the model can quickly adapt to new prediction tasks using only a small amount of training data or even make predictions for any new tabular input when no training data is available.

In Section 2, we provide a detailed description of our approach. We present the experimental results in Section 3, where we demonstrate the effectiveness of our method on 7 patient outcome prediction datasets and 3 trial outcome prediction datasets, achieving an average performance ranking of **1.57** and **1.00**, respectively, across tabular prediction baselines. Furthermore, our method shows impressive few-shot and zero-shot performances that are competitive with supervised baselines: the **zero-shot** `MediTab` outperforms supervised XGBoost by **8.9%** and **17.2%** on average in two prediction tasks, respectively. We discuss related work in Section 4 and conclude our findings in Section 5.

2 Method

2.1 Problem Formulation

We characterize tabular prediction tasks by *dataset* $\mathbf{D}$ and *task* $\mathbf{T}$, where a task $\mathbf{T} = \{\mathbf{D}_1, \mathbf{D}_2, \dots\}$ consists of multiple *in-domain* datasets with varying features and schema but the same target label. For example, the patient mortality prediction task contains samples from many clinical trials (where input features differ between trials). For $\mathbf{T}_1$, the datasets from other tasks $\mathbf{T}_2, \mathbf{T}_3, \dots$ are considered *out-domain* since they differ in prediction objectives. As illustrated by Figure 1, existing methods for tabular prediction fall short in transfer learning across datasets, as each model learns from a single dataset $\mathbf{D}$ and needs to learn from scratch when encountering new datasets. On the contrary, `MediTab` extends the training data to all available tasks $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, \dots\}$, demonstrating its flexibility to encode and predict for arbitrary tabular samples. After training, it serves all $\mathbf{D} \in \mathbf{T}_1$ without further fine-tuning. Our method eliminates the need to keep as many models as datasets, paving the way for the efficient and streamlined deployment of tabular prediction models. Depending on the use case, the problems that our method can handle can be classified into the following categories.

Problem 1 (**Multi-task learning (MTL)**). *MTL is a machine learning technique where a single model is trained to perform multiple tasks simultaneously. Define $f : \mathcal{X} \mapsto \mathcal{Y}$ as a model that takes a consolidated tabular sample x as input and predicts the target label y. The training dataset is formed by combining all the tabular inputs in $\mathbf{D}_* \in \mathbf{T}$. Once trained, the model f is fixed and can be used to make predictions on any new samples $x \sim \mathbf{D}$, $\forall \mathbf{D} \in \mathbf{T}$.*

Problem 2 (**Zero-shot/Few-shot learning**). *The model f is trained on $\mathbf{T} = \{\mathbf{D}_1, \dots, \mathbf{D}_N\}$. Then, it makes predictions for a new dataset $\mathbf{D}_{N+1}$ that has not been included in the training data. Model f performs zero-shot learning if no label is available for all samples in $\mathbf{D}_{N+1}$; Model f performs few-shot learning to predict for $\mathbf{D}_{N+1}$ if a few labeled samples are available.*

2.2 The `MediTab` Framework

As illustrated in Figure 2, our method consists of:

Step 1: Tabular Data Consolidation. The tabular datasets $\mathbf{D}$ differ in their features, schema, and particularly in their target objectives if they are from distinct tasks $\mathbf{T}$. The consolidation is accomplished by converting each row of the table into natural language descriptions that consider the data schema. This conversion process transforms all tabular data into text data that share the same semantic space, enabling them to be utilized in language modeling. Additionally, we can produce diverse consolidated samples by describing one sample in multiple different ways, which allows for data augmentation. To prevent hallucinations that may occur during this transformation, an audit module that utilizes LLMs is employed to perform self-check and self-correction. Our goal of patient survival classification is the same for each dataset; however, we use a diverse number of datasets, so the task is indeed different.

Step 2: Learn, Annotate, & Audit. Our method can benefit from out-of-domain datasets $\mathbf{T}_* \in \mathcal{T}$ through our annotation and data importance pipeline. Once it is trained on $\mathbf{T}_1$, it is used to produce pseudo labels for samples from all other tasks, which yields a big but noisy supplementary dataset $\widetilde{\mathbf{T}}_{1,\text{sup}}$. This dataset is further cleaned by a data audit module based on data Shapley scores, leading to a smaller but cleaner dataset $\mathbf{T}_{1,\text{sup}}$.

Step 3: Learning & Deployment. The final prediction model learns from the combination of the original task 1 data $\mathbf{T}_1$ and the supplementary data $\mathbf{T}_{1,\text{sup}}$. The resulting multi-task model f_{MTL} can be used for all datasets $\mathbf{D}_* \in \mathbf{T}_1$ without any modifications. Furthermore, the model can predict on new datasets $\mathbf{D} \in \mathbf{T}$ in a zero-shot manner and perform few-shot learning for any $\mathbf{D} \in \mathbf{T}$ or $\mathbf{D} \notin \mathbf{T}$. (Note that the primary purpose of the pseudolabels is to facilitate training the zero-shot and few-shot models, and are not meant to improve the performance of the original model.)

We will elaborate on the technical details of these steps in the following sections.

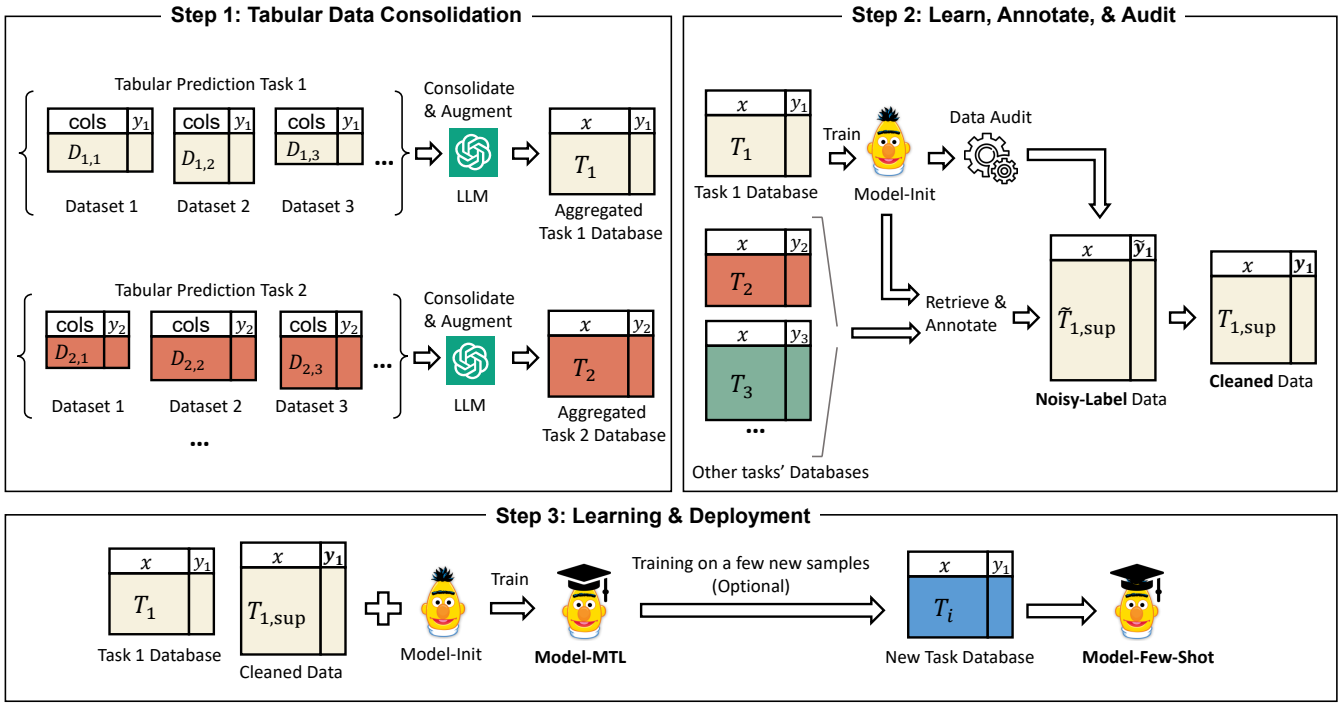

Figure 2: The demonstration of scaling medical tabular data predictors models (`MediTab`). It encompasses three steps: **Step 1** consolidates tabular datasets using LLM; **Step 2** aligns out-domain datasets with the target task; **Step 3** facilitates the predictor with cleaned supplementary data. More details are presented in Section 2.2.

2.3 Tabular Data Consolidation & Sanity Check

Data scarcity is a challenge in scaling medical tabular data predictors. Existing tabular prediction models often struggle on these datasets due to the vague semantic meanings of the varying columns and values. We transform each row into natural language sentences that describe the sample using generative language models like GPT-3.5. Specifically, we combine the *linearization, prompting*, and *santity check* steps to obtain input data that the language model can use to generate coherent and meaningful natural language descriptions.

Linearization. A function $\texttt{linearize}(\mathbf{c}, \mathbf{v})$ takes the column names $\mathbf{c}$ and the corresponding cell values $\mathbf{v}$ from a row as the input, then linearizes the row to a concatenation of paired columns and cells $\{c : v\}$. Notably, we identify sparse tabular datasets that have many binary columns. In linearization, we exclude binary columns that have a cell value of `False` and only include those with positive values. This approach has two key benefits. First, it ensures that the linearized output does not exceed the input limit of language models. Second, it helps in reducing hallucinations arising from failed negation detection during the generation process of the LLMs. An ablation on the different types of tabular-to-text serialization, shown in Table 10 in the Appendix, shows that audited examples improve performance via augmentation. We believe that this performance benefit is useful, and serves to justify our usage of more advanced paraphrasing and auditing techniques.

Prompting. We combine the linearization with prefix p and suffix s to form the LLM prompt as $(p, \texttt{linearize}(\mathbf{c}, \mathbf{v}), s)$. The schema definition is added to p to provide the context for LLM when describing the sample. For each column, we provide the type and explanation as $\{\texttt{column}\}(\{\texttt{type}\}): \ \{\texttt{explanation}\}$ (e.g., "*demo1(numerical): the age of the patient in years.*"). The suffix s represents the instruction that steers LLM to describe the target sample or generate paraphrases for data augmentation. We describe the specific prompt templates we used in Appendix C.1 and display some consolidated examples in Appendix C.2.

The text descriptions $\mathbf{x}$ are hence sampled from LLMs by

$$\mathbf{x} \sim \text{LLM}(p, \texttt{linearize}(\mathbf{c}, \mathbf{v}), s). \quad (1)$$

The paired inputs and target $\{\mathbf{x}, y\}$ will be the training data for the tabular prediction model f. We can adjust the suffix s in Eq. 1 to generate multiple paraphrases of the same sample as a way for instance-level data augmentation. Some instance-level augmentation examples are available in Appendix C.3.

Sanity Check via LLM's Reflection. To reduce low-quality generated samples, a sanity check function evaluates the fidelity of the generated text $\mathbf{x}$ to address potential hallucinations or loss of information that occurs during the translation process $\{\mathbf{c}, \mathbf{v}\} \rightarrow \mathbf{x}$ in Eq. 1, particularly for numerical features. Specifically, we query LLM with the input template "`What is the {column}?` $\{\mathbf{x}\}$" to check if the answer matches the original values in $\{\mathbf{c}, \mathbf{v}\}$. The descriptions are corrected by re-prompting the LLM if the answers do not match. We provide some examples of sanity checks in Ap-

pendix C.4, and the quantitative analysis of this correction is available in Appendix C.5.

2.4 Data Enrichment & Refinement

Through the consolidation and sanity check process, we are able to aggregate all tabular samples $\{\mathbf{x}, y\}$ from the target task $\mathbf{T}_1$ and train a prediction model. We can use the dataset $\mathbf{T}_1$ to train a multi-task learning model, denoted as f_{MTL}, which can be applied to all datasets within the task. Nevertheless, there still lacks a route to leverage data from out-domain tasks $\mathbf{T}_* \sim \mathcal{T},\ \forall \mathbf{T}_* \in \mathcal{T} \setminus \{\mathbf{T}_1\}$ for data enrichment. It is particularly valuable for low-data applications such as healthcare, where there may only be a few dozen data points for each dataset. Specifically, we propose to align out-domain task datasets via a *learn*, *annotate*, and *audit* pipeline for data enrichment.

Learn and Annotate. We train an initial model f_{MTL} on all available training data from $\mathbf{T}_1$ (we will omit the subscript 1 from now on to avoid clutter). The model f_{MTL} then makes pseudo labels for a set of external samples that are retrieved from all other tasks $\mathcal{T} \setminus \{\mathbf{T}\}$, formulating the noisy supplementary data $\widetilde{\mathbf{T}}_{\text{sup}} = \{(\mathbf{x}_i, \tilde{y}_i)\}$: $\mathbf{x}_i$ are consolidated textual description samples and $\tilde{y}_i$ are noisy labels that are aligned with the objective format of the target task.

Quality Audit. It is vital to audit the quality of noisy training data to ensure optimal prediction performance. To this end, we clean $\widetilde{\mathbf{T}}$ by estimating the data Shapley values for each instance [Ghorbani and Zou, 2019]. We denote the value function by $V(\mathbf{S})$, which indicates the performance score evaluated on the target task $\mathbf{T}$ of the predictor trained on training data $\mathbf{S}$. Correspondingly, we have the data Shapley value ϕ_i for any sample $(\mathbf{x}_i, \tilde{y}_i) \in \widetilde{\mathbf{T}}$ defined by

$$\phi_i = C \sum_{\mathbf{S} \subseteq \widetilde{\mathbf{T}} \setminus \{i\}} \frac{V(\mathbf{S} \bigcup \{i\}) - V(\mathbf{S})}{\binom{n-1}{|\mathbf{S}|}}, \tag{2}$$

where the summation is over all subsets of $\widetilde{\mathbf{T}}$ not with sample i; n is the number of samples in $\widetilde{\mathbf{T}}$; $|\cdot|$ implies to the size of the set; C is an arbitrary constant. Intuitively, ϕ_i measures the approximated expected contribution of a data point to the trained model's performance. Therefore, a sample with low ϕ is usually of low quality itself.

Computing the exact data Shapley value with Eq. 2 requires an exponentially large number of computations with respect to the number of train data sources. Instead, we follow [Jia *et al.*, 2021; Jia *et al.*, 2019] to use K-Nearest Neighbors Shapley (KNN-Shapley), which offers an avenue for efficient data Shapley computation. Moreover, we are able to achieve a $10\times$ speedup by parallelizing the algorithm, which completes computing scores for 100K+ samples in minutes. Upon acquiring $\Phi = \{\phi_i\}$, we execute a representative stratified sampling corresponding with the distribution of the sample classes to establish the cleaned supplementary dataset $\mathbf{T}_{\text{sup}}$. Appendix G explores the Shapley value and pseudolabel distributions of different supplemental datasets.

2.5 Learning & Deployment

After the quality check step, we obtain the original task dataset $\mathbf{T}$ and the supplementary dataset $\mathbf{T}_{\text{sup}}$ and have two potential options for model training. The first is to combine both datasets for training, but we have found that this approach results in suboptimal performance. Instead, we employ a two-step training approach: (1) pre-train the model on $\mathbf{T}_{\text{sup}}$, and (2) finetune the model using $\mathbf{T}$. The resulting model will be deployed to provide predictions for any tabular samples belonging to the target task. Because the model trained on the supplementary model has not seen any examples from the original dataset, it can make zero-shot predictions for the test samples from a new dataset $\mathbf{D}' \notin \mathbf{T}$ or adapt to a new task $\mathbf{T}'$ via few-shot learning when a few labeled data are available. Due to the small number of samples in some datasets, we thought it would be best to use all possible training samples in the learning phase, without leaking information in the testing phase. As the label distribution is highly skewed, it may also bias our model if validation samples were chosen randomly, and we did not have the expertise to choose a representative sample. In our case, we chose to simply train the model for 3 epochs on all of the datasets, and then perform a single pass of fine-tuning, without significant hyperparameter optimization, due to the small amount of data and the good performance that it gives.

3 Experiments

We conduct an extensive evaluation of `MediTab`'s performance in **supervised learning** (Q1), **few-shot learning** (Q2), and **zero-shot prediction** (Q3). We also compare the different training strategies for the final deployment of our method (Q4).

3.1 Experimental Setting

Datasets: In our experiments, we introduce the following types of tabular prediction tasks. **Patient Outcome Datasets**. This dataset includes the patient records collected separately from seven oncology clinical trials [1]. These datasets each have their unique schema and contain distinct groups of patients in different conditions. A CTGAN model [Xu *et al.*, 2019] was trained on the raw data to generate the synthetic patient data for the experiments. We train the model to predict the patient's morbidity, which is a binary classification task. The statistics of the datasets are available in Table 1. **Clinical Trial Outcome Datasets**. We use clinical trial data from the `HINT` benchmark [Fu *et al.*, 2022] and `ClinicalTrials.gov` [2]. The `HINT` benchmark contains drug, disease, and eligibility information for 17K clinical trials. The trial database contains 220K clinical trials with information about the trial setup (such as title, phase, enrollment, conditions, etc.). Both datasets cover phases I, II, and III trials, but only the `HINT` benchmark includes the trial outcome labels in {success, failure}. We have also included the MIMIC-IV dataset and PMC-Patients dataset as the external patient database and clinical trial documents as the external trial outcome prediction dataset. Please refer to Appendix D for details.

Implementations: For the patient outcome prediction task, we choose a tree ensemble method (XGBoost) [Chen and

[1] https://data.projectdatasphere.org/projectdatasphere/html/access

[2] https://clinicaltrials.gov/

Trial ID	Trial Name	# Patients	Categorical	Binary	Numerical	Positive Ratio	Train/Test Split
NCT00041119	Breast Cancer 1	3,871	5	8	2	0.07	3096 / 775
NCT00174655	Breast Cancer 2	994	3	31	15	0.02	795 / 199
NCT00312208	Breast Cancer 3	1,651	5	12	6	0.19	1320 / 331
NCT00079274	Colorectal Cancer	2,968	5	8	3	0.12	2374 / 594
NCT00003299	Lung Cancer 1	587	2	11	4	0.94	469 / 118
NCT00694382	Lung Cancer 2	1,604	1	29	11	0.45	1283 / 321
NCT03041311	Lung Cancer 3	53	2	11	13	0.64	42 / 11
External Patient Database							
MIMIC-IV		143,018	4	1	1	N/A	
PMC-Patients		167,034	1	1	1	N/A	

Table 1: The statistics of Patient Outcome Prediction Datasets. # is short for the number of. Categorical, Binary, and Numerical show the number of columns belonging to these types. N/A means no label is available for the target task. We used 20% of the data for testing.

Dataset	# Trials	# Treatments	# Conditions	# Features	Positive Ratio	Train/Test Split
TOP Benchmark Phase I	1,787	2,020	1,392	6	0.56	1136 / 575
TOP Benchmark Phase II	6,102	5,610	2,824	6	0.50	4317 / 1504
TOP Benchmark Phase III	4,576	4,727	1,619	6	0.68	3359 / 1048
ClinicalTrials.gov Database						
Phase I-IV	223,613	244,617	68,697	9	N/A	

Table 2: The statistics of the Clinical Trial Outcome Datasets. # is short for the number of. N/A means no label is available for the target task.

Guestrin, 2016a], Multilayer Perceptron, FT-Transformer [Gorishniy *et al.*, 2021], TransTab [Wang and Sun, 2022b], and TabLLM [Hegselmann *et al.*, 2022] as the baselines. For the trial outcome prediction task, we choose XGBoost, feed-forward neural network (FFNN) [Tranchevent *et al.*, 2019], DeepEnroll [Zhang *et al.*, 2020], COMPOSE [Gao *et al.*, 2020], HINT [Fu *et al.*, 2022], and SPOT [Wang *et al.*, 2023b] as the baselines. We use `PyTrial` [Wang *et al.*, 2023a] to implement most baselines and provide the parameter tuning details of the selected baselines in Appendix E.

We use a pre-trained bidirectional transformer model named BioBERT [Lee *et al.*, 2020] as the classifier for `MediTab`. We utilize GPT-3.5 [Brown *et al.*, 2020] via OpenAI's API [3] for the data consolidation and enhancement. We use UnifiedQA-v2-T5 3B [Khashabi *et al.*, 2020] [4] for the sanity check. The evaluation metrics selected are ROC-AUC and PR-AUC, with the details in Appendix F. Further ablations on base model choice is shown in Appendix H. All experiments were run with 2 RTX-3090 GPUs and AMD Ryzen 3970X 32-Core CPU.

3.2 Results on Patient Outcome Prediction and Trial Outcome Prediction

We report the *supervised* results for patient outcome prediction: the AUROC and PRAUC on the test sets of all clinical trials, in Table 3. Note that we train a single classifier for `MediTab` and predict on all datasets, while the baselines need to be trained on each dataset separately. Our findings demonstrate that a single `MediTab` model achieves the highest ranking in 5 out of 7 datasets, with an overall ranking of 1.57 across all datasets. Conversely, MLP and FT-Transformer fail to converge in certain cases due to imbalanced target labels (e.g., Lung Cancer 1) or limited availability of data (e.g., Lung Cancer 3). This highlights the data-hungry nature of deep learning algorithms and emphasizes the importance of augmenting training data through data consolidation and enrichment. Additionally, we observe that TabLLM fails in both the single-dataset and multi-dataset settings. We see thatonly the text-template serialization performs poorly in this setting, with multiple datasets not converging. The small amount of data and the clinical-specific terms may be too niche for the general-purpose TabLLM. Furthermore, it is not able to generalize across datasets, as the column names are quite diverse (Table 5).

`MediTab` also leads to substantial improvements in trial outcome prediction tasks, as illustrated in Table 4. Notably, our approach outperforms all other methods in every phase of the trials. We observe remarkable improvements of 5.9%, 9.5%, and 3.2% over the previous state-of-the-art baselines in the three phases, respectively. This provides insight into the benefits of increased data availability and the utilization of transfer learning in deep learning-based tabular prediction algorithms.

3.3 Results on Zero-Shot and Few-Shot Learning

We assess the *zero-shot* prediction capability of `MediTab` on two tasks. For the evaluation of the dataset $\mathbf{D}$, we deliberately exclude $\mathbf{D}$ from the training data during step 2, where

[3]Engine `gpt-3.5-turbo-0301`: https://platform.openai.com/docs/models/gpt-3-5

[4]Huggingface: allenai/unifiedqa-v2-t5-large-1363200

Trial Name	Metrics	XGBoost	MLP	FT-Transformer	TransTab	TabLLM (Single Dataset)	TabLLM (Multi-Dataset)	MediTab
Breast Cancer 1	AUROC	0.5430	0.6091	0.5564	0.5409	-	-	**0.6182**
	PRAUC	0.0796	0.0963	0.0803	0.0923	-	-	**0.1064**
Breast Cancer 2	AUROC	0.6827	0.6269	0.6231	0.6000	-	-	**0.8397**
	PRAUC	0.1559	0.1481	0.0520	0.0365	-	-	**0.1849**
Breast Cancer 3	AUROC	0.6489	0.7065	0.6338	0.7100	0.6163	0.6103	**0.7529**
	PRAUC	0.3787	0.4000	0.3145	0.4133	0.3023	0.2977	**0.4567**
Colorectal Cancer	AUROC	0.6704	0.6337	0.5951	0.7096	-	-	**0.7107**
	PRAUC	0.2261	0.1828	0.1541	0.2374	-	-	**0.2402**
Lung Cancer 1	AUROC	-	0.6023	-	0.6499	-	-	**0.7246**
	PRAUC	-	0.9555	-	0.9672	-	-	**0.9707**
Lung Cancer 2	AUROC	**0.6976**	0.5933	0.6093	0.5685	0.6188	0.6279	0.6822
	PRAUC	**0.6865**	0.5662	0.5428	0.4922	0.5619	0.5772	0.6710
Lung Cancer 3	AUROC	0.6976	0.6429	0.5357	0.6786	0.8036	0.6786	**0.8928**
	PRAUC	0.7679	0.8501	0.7250	0.7798	0.8256	0.7338	**0.9478**

Table 3: Test performances on the Patient Outcome Datasets. "-" indicates not converged.

Trial Data	Metrics	XGBoost	FFNN	DeepEnroll	COMPOSE	HINT	SPOT	MediTab
Phase I	AUROC	0.518	0.550	0.575	0.571	0.576	0.660	**0.699**
	PRAUC	0.513	0.547	0.568	0.564	0.567	0.689	**0.726**
Phase II	AUROC	0.600	0.611	0.625	0.628	0.645	0.630	**0.706**
	PRAUC	0.586	0.604	0.600	0.604	0.629	0.685	**0.733**
Phase III	AUROC	0.667	0.681	0.699	0.700	0.723	0.711	**0.734**
	PRAUC	0.697	0.747	0.777	0.782	0.811	0.856	**0.881**

Table 4: Test performances on the Clinical Trial Outcome Datasets.

pseudo labels are generated for the external database. When computing the data Shapley values for out-domain samples during the quality check process, $\mathbf{D}$ is also excluded. Subsequently, we train a model solely on the cleaned supplementary data $\mathbf{T}_{\text{sup}}$ and evaluate its performance on the target dataset $\mathbf{D}$. The results of this evaluation are illustrated in Figure 3. MediTab exhibits impressive zero-shot performances: it wins over supervised XGBoost models in 5 out of 7 datasets in patient outcome prediction and all three datasets in trial outcome prediction by a significant margin. On average, MediTab achieves gains of 8.9% and 17.2% improvements in the two tasks, respectively.

The encouraging zero-shot learning result sheds light on the development of task-specific tabular prediction models that can offer predictions for new datasets even before the label collection stage. This becomes particularly invaluable in scenarios where acquiring training labels is costly. For instance, it enables us to predict the treatment effect of a drug on a group of patients before conducting clinical trials or collecting any trial records. Consequently, it allows us to make informed decisions regarding treatment adjustments or trial discontinuation.

We further visualize the few-shot learning results in Figure 4. We witness consistent performance improvement with more labeled training samples for both methods. Additionally, for all tested cases, XGBoost is unable to surpass the zero-shot score of MediTab.

3.4 Results on Ablations on Different Learning Strategies

Section 2.5 discusses a two-stage training strategy for the final learning & deployment stage. Here, we investigate the

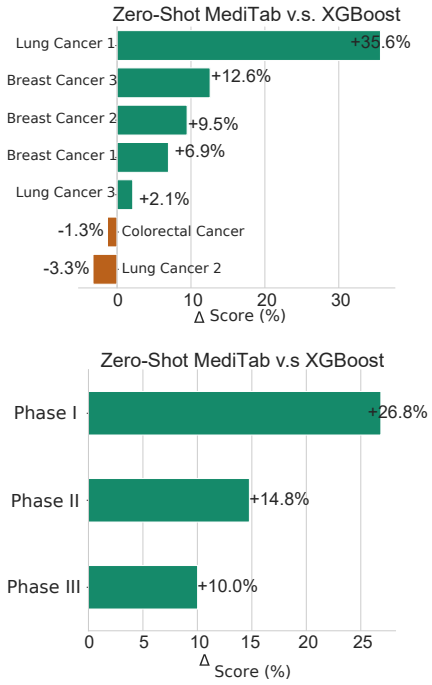

Figure 3: **Zero-shot MediTab** is better than a fully supervised baseline (XGBoost). The evaluation is across 7 patient outcome prediction datasets (left) and 3 trial outcome prediction datasets (right). The compared baseline XGBoost model is fitted on each dataset, respectively.

different training regimens of our method: single-stage training (augment), two-stage training (finetune), training on the original datasets from scratch (scratch), and zero-shot (zeroshot). We list their rankings in Figure 5 and detailed performances across datasets in the Appendix (Tables 7 and 8). Results show that finetune generally performs the best. We conjecture that jointly training on the target task and supplementary data improves the model's overall utility, but it may affect the performance of specific samples in the target task $\mathbf{T}$. Furthermore, we also identify that zeroshot reaches comparable performances with scratch.

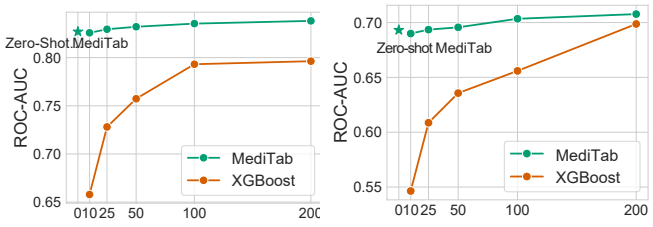

Figure 4: **Few-shot `MediTab`** compared with XGBoost with varying training data sizes. The compared baseline XGBoost model is fitted on each dataset, respectively. x-axis: number of labeled training data per class. Left: patient dataset; Right: trial dataset.

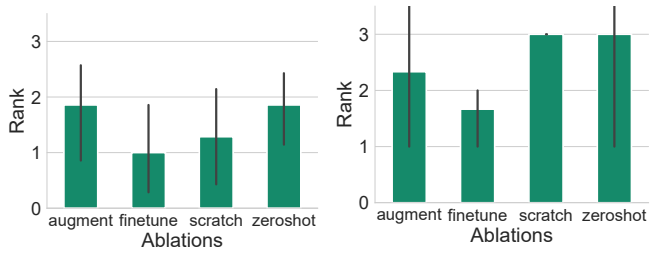

Figure 5: ROC-AUC Ranking (lower is better) of the variations of `MediTab`. Left: patient dataset; Right: Trial dataset.

4 Related Work

Tabular Prediction has traditionally relied on tree ensemble methods [Chen and Guestrin, 2016b; Ke *et al.*, 2017]. In recent years, the powerful representation learning abilities of neural networks have motivated the new design of deep learning algorithms for tabular prediction [Arik and Pfister, 2021; Kadra *et al.*, 2021; Chen *et al.*, 2023; Bertsimas *et al.*, 2022]. They involve using transformer-based architectures [Huang *et al.*, 2020; Gorishniy *et al.*, 2021; Wang and Sun, 2022a] to enhance automatic feature interactions for better prediction performances. In addition, self-supervised learning (SSL) has been extended to tabular prediction tasks. This includes approaches such as generative pretraining objective by masked cell modeling [Yoon *et al.*, 2020; Arik and Pfister, 2021; Nam *et al.*, 2023], and discriminative pretraining objective by self-supervised [Ucar *et al.*, 2021; Somepalli *et al.*, 2022; Bahri *et al.*, 2022] or supervised contrastive learning [Wang and Sun, 2022b]. Moreover, transfer learning was also adapted to tabular prediction, employing prompt learning based on generative language models [Hegselmann *et al.*, 2022] and multi-task learning [Levin *et al.*, 2023]. Multi-task learning and transfer learning were also performed in the medical domain for EHR-based predictive modeling [Hur *et al.*, 2023; Hur *et al.*, 2022]. Nonetheless, these approaches primarily focus on *algorithm* design, including model architecture and objective functions, often overlooking the engineering of the underlying *data*.

Data-Centric AI underscores the importance of data for building advanced machine learning prediction systems [Zha *et al.*, 2023]. Notable progress in the domain of tabular data includes efforts to detect [Wang *et al.*, 2020] and debug the noises in labels [Kong *et al.*, 2021]; automate feature selection [Liu *et al.*, 2023]; and streamline feature generation [Su *et al.*, 2021]. Additionally, LM for finetuning on tabular data has been proposed by [Dinh *et al.*, 2022], but it uses strict templates to create the sentence, which limits its expressivity. These methods were proposed for general tabular data while not covering the challenges of heterogeneity and limited samples in medical tabular data. Though there were efforts in enhancing medical codes in EHRs with text descriptions [Hur *et al.*, 2022], there is no further exploration on augmenting medical tabular data that include more diverse features. In contrast, we present a data engineering framework designed to consolidate diverse tabular datasets, by distilling the knowledge from large language models with hallucination detection and distilling from out-domain datasets with data auditing. `MediTab` is hence able to build a versatile prediction model for the target task.

5 Conclusion

In conclusion, we proposed a novel approach to train universal tabular data predictors for medical data. While there were many efforts in developing new algorithms for tabular prediction, the significance of data engineering has raised much less attention. Specifically, medicine faces challenges in limited data availability, inconsistent dataset structures, and varying prediction targets across domains. To address these challenges, `MediTab` generates large-scale training data for tabular prediction models by utilizing both in-domain tabular datasets and a set of out-domain datasets. The key component of this approach is a data engine that utilizes large language models to consolidate tabular samples by expressing them in natural language, thereby overcoming schema differences across tables. Additionally, the out-domain tabular data is aligned with the target task using a *learn, annotate, and refine* pipeline. By leveraging the expanded training data, `MediTab` can effectively work on any tabular dataset within the domain without requiring further finetuning, achieving significant improvements compared to supervised baselines. Moreover, `MediTab` demonstrates impressive performance even with limited examples (few-shot) or no examples (zero-shot), remaining competitive with supervised approaches across various tabular datasets. A discussion on the ethical implications and limitations can be found in App. A and App. B. Additionally, further ablation on Data Shapley and Auditing can be found in App. H.

Acknowledgments

This work was supported by NSF award SCH-2205289, SCH-2014438, and IIS-2034479.

Contribution Statement

Zifeng Wang and Chufan Gao contributed equally to this work.

References

[Arik and Pfister, 2021] Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6679–6687, 2021.

[Bahri *et al.*, 2022] Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. SCARF: Self-supervised contrastive learning using random feature corruption. In *International Conference on Learning Representations*, 2022.

[Beigi *et al.*, 2022] Mandis Beigi, Afrah Shafquat, Jason Mezey, and Jacob W Aptekar. Synthetic clinical trial data while preserving subject-level privacy. In *NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research*, 2022.

[Bertsimas *et al.*, 2022] Dimitris Bertsimas, Kimberly Villalobos Carballo, Yu Ma, Liangyuan Na, Léonard Boussioux, Cynthia Zeng, Luis R Soenksen, and Ignacio Fuentes. Tabtext: a systematic approach to aggregate knowledge across tabular data structures. *arXiv preprint arXiv:2206.10381*, 2022.

[Borisov *et al.*, 2022] Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. *arXiv preprint arXiv:2210.06280*, 2022.

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

[Chen and Guestrin, 2016a] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[Chen and Guestrin, 2016b] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[Chen *et al.*, 2023] Jintai Chen, KuanLun Liao, Yanwen Fang, Danny Chen, and Jian Wu. Tabcaps: A capsule neural network for tabular data classification with bow routing. In *The Eleventh International Conference on Learning Representations*, 2023.

[Dinh *et al.*, 2022] Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. *Advances in Neural Information Processing Systems*, 35:11763–11784, 2022.

[Fu *et al.*, 2022] Tianfan Fu, Kexin Huang, Cao Xiao, Lucas M Glass, and Jimeng Sun. Hint: Hierarchical interaction network for clinical-trial-outcome predictions. *Patterns*, 3(4):100445, 2022.

[Gao *et al.*, 2020] Junyi Gao, Cao Xiao, Lucas M Glass, and Jimeng Sun. COMPOSE: cross-modal pseudo-siamese network for patient trial matching. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 803–812, 2020.

[Gao *et al.*, 2024] Junyi Gao, Yinghao Zhu, Wenqing Wang, Zixiang Wang, Guiying Dong, Wen Tang, Hao Wang, Yasha Wang, Wen Tang, Ewen M Harrison, and Liantao Ma. A comprehensive benchmark for covid-19 predictive modeling using electronic health records in intensive care. *Patterns*, 5(4), 2024.

[Ghorbani and Zou, 2019] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR, 2019.

[Gorishniy *et al.*, 2021] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943, 2021.

[Hegselmann *et al.*, 2022] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models. *arXiv preprint arXiv:2210.10723*, 2022.

[Huang *et al.*, 2020] Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020.

[Hur *et al.*, 2022] Kyunghoon Hur, Jiyoung Lee, Jungwoo Oh, Wesley Price, Younghak Kim, and Edward Choi. Unifying heterogeneous electronic health records systems via text-based code embedding. In *Conference on Health, Inference, and Learning*, pages 183–203. PMLR, 2022.

[Hur *et al.*, 2023] Kyunghoon Hur, Jungwoo Oh, Junu Kim, Jiyoun Kim, Min Jae Lee, Eunbyeol Cho, Seong-Eun Moon, Young-Hak Kim, Louis Atallah, and Edward Choi. Genhpf: General healthcare predictive framework for multi-task multi-source learning. *IEEE Journal of Biomedical and Health Informatics*, 2023.

[Jia *et al.*, 2019] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas J Spanos, and Dawn Song. Efficient task-specific data valuation for nearest neighbor algorithms. *arXiv preprint arXiv:1908.08619*, 2019.

[Jia *et al.*, 2021] Ruoxi Jia, Fan Wu, Xuehui Sun, Jiacen Xu, David Dao, Bhavya Kailkhura, Ce Zhang, Bo Li, and Dawn Song. Scalability vs. utility: Do we have to sacrifice one for the other in data importance quantification? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8239–8247, 2021.

[Johnson *et al.*, 2023] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV (version 2.2), 2023.

[Kadra *et al.*, 2021] Arlind Kadra, Marius Lindauer, Frank Hutter, and Josif Grabocka. Well-tuned simple nets excel on tabular datasets. *Advances in Neural Information Processing Systems*, 34:23928–23941, 2021.

[Ke *et al.*, 2017] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30, 2017.

[Khashabi *et al.*, 2020] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*, 2020.

[Kong *et al.*, 2021] Shuming Kong, Yanyan Shen, and Linpeng Huang. Resolving training biases via influence-based data relabeling. In *International Conference on Learning Representations*, 2021.

[Lee *et al.*, 2020] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[Levin *et al.*, 2023] Roman Levin, Valeriia Cherepanova, Avi Schwarzschild, Arpit Bansal, C. Bayan Bruss, Tom Goldstein, Andrew Gordon Wilson, and Micah Goldblum. Transfer learning with deep tabular models. In *The Eleventh International Conference on Learning Representations*, 2023.

[Liu *et al.*, 2023] Dugang Liu, Pengxiang Cheng, Hong Zhu, Xing Tang, Yanyu Chen, Xiaoting Wang, Weike Pan, Zhong Ming, and Xiuqiang He. Diwift: Discovering instance-wise influential features for tabular data. In *Proceedings of the ACM Web Conference 2023*, pages 1673–1682, 2023.

[Ma *et al.*, 2020] Liantao Ma, Chaohe Zhang, Yasha Wang, Wenjie Ruan, Jiangtao Wang, Wen Tang, Xinyu Ma, Xin Gao, and Junyi Gao. Concare: Personalized clinical feature embedding via capturing the healthcare context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 833–840, 2020.

[Ma *et al.*, 2023] Liantao Ma, Chaohe Zhang, Junyi Gao, Xianfeng Jiao, Zhihao Yu, Yinghao Zhu, Tianlong Wang, Xinyu Ma, Yasha Wang, Wen Tang, et al. Mortality prediction with adaptive feature importance recalibration for peritoneal dialysis patients. *Patterns*, 4(12), 2023.

[Nam *et al.*, 2023] Jaehyun Nam, Jihoon Tack, Kyungmin Lee, Hankook Lee, and Jinwoo Shin. STUNT: Few-shot tabular learning with self-generated tasks from unlabeled tables. In *ICLR*, 2023.

[Northcutt *et al.*, 2021] Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021.

[Pedregosa *et al.*, 2011] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[Somepalli *et al.*, 2022] Gowthami Somepalli, Avi Schwarzschild, Micah Goldblum, C Bayan Bruss, and Tom Goldstein. SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training. In *NeurIPS 2022 First Table Representation Workshop*, 2022.

[Su *et al.*, 2021] Yixin Su, Rui Zhang, Sarah Erfani, and Zhenghua Xu. Detecting beneficial feature interactions for recommender systems. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4357–4365, 2021.

[Theodorou *et al.*, 2023] Brandon Theodorou, Cao Xiao, and Jimeng Sun. Synthesize high-dimensional longitudinal electronic health records via hierarchical autoregressive language model. *Nature Communications*, 14(1):5305, 2023.

[Tranchevent *et al.*, 2019] Léon-Charles Tranchevent, Francisco Azuaje, and Jagath C Rajapakse. A deep neural network approach to predicting clinical outcomes of neuroblastoma patients. *BMC Medical Genomics*, 12:1–11, 2019.

[Ucar *et al.*, 2021] Talip Ucar, Ehsan Hajiramezanali, and Lindsay Edwards. Subtab: Subsetting features of tabular data for self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34:18853–18865, 2021.

[Wang and Sun, 2022a] Zifeng Wang and Jimeng Sun. Survtrace: Transformers for survival analysis with competing events. In *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 1–9, 2022.

[Wang and Sun, 2022b] Zifeng Wang and Jimeng Sun. TransTab: Learning transferable tabular transformers across tables. In *Advances in Neural Information Processing Systems*, 2022.

[Wang *et al.*, 2020] Zifeng Wang, Hong Zhu, Zhenhua Dong, Xiuqiang He, and Shao-Lun Huang. Less is better: Unweighted data subsampling via influence function. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6340–6347, 2020.

[Wang *et al.*, 2023a] Zifeng Wang, Brandon Theodorou, Tianfan Fu, and Jimeng Sun. PyTrial: A python package for artificial intelligence in drug development, 11 2023.

[Wang *et al.*, 2023b] Zifeng Wang, Cao Xiao, and Jimeng Sun. SPOT: Sequential predictive modeling of clinical trial outcome with meta-learning. *arXiv preprint arXiv:2304.05352*, 2023.

[Xu *et al.*, 2019] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32, 2019.

[Yin *et al.*, 2020] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. Tabert: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*, 2020.

[Yoon *et al.*, 2020] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. VIME: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33:11033–11043, 2020.

[Zha *et al.*, 2023] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. Data-centric artificial intelligence: A survey. *arXiv preprint arXiv:2303.10158*, 2023.

[Zhang *et al.*, 2020] Xingyao Zhang, Cao Xiao, Lucas M Glass, and Jimeng Sun. DeepEnroll: patient-trial matching with deep embedding and entailment prediction. In *Proceedings of The Web Conference 2020*, pages 1029–1037, 2020.

[Zhao *et al.*, 2022] Zhengyun Zhao, Qiao Jin, Fangyuan Chen, Tuorui Peng, and Sheng Yu. Pmc-patients: A large-scale dataset of patient summaries and relations for benchmarking retrieval-based clinical decision support systems. *arXiv preprint arXiv:2202.13876*, 2022.

[Zhu *et al.*, 2023] Bingzhao Zhu, Xingjian Shi, Nick Erickson, Mu Li, George Karypis, and Mahsa Shoaran. XTab: Cross-table pretraining for tabular transformers. *arXiv preprint arXiv:2305.06090*, 2023.