AnyPredict: Foundation Model for Tabular Prediction

Zifeng Wang[1], **Chufan Gao**[1], **Cao Xiao**[2], **Jimeng Sun**[1]
[1]University of Illinois Urbana-Champaign, [2]Relativity
{zifengw2,chufan2,jimeng}@illinois.edu, cao.xiao@relativity.com

Abstract

Foundation models are pre-trained on massive data to perform well across many downstream tasks. They have demonstrated significant success in natural language processing and computer vision. Nonetheless, the use of such models in tabular prediction tasks has been limited, with the main hurdles consisting of (1) the lack of large-scale and diverse tabular datasets with standardized labels and (2) the schema mismatch and predictive target heterogeneity across domains.

This paper proposes a method for building training data at scale for tabular prediction foundation models (AnyPredict) using both in-domain and a wide range of out-domain datasets. The method uses a data engine that leverages large language models (LLMs) to consolidate tabular samples to overcome the barrier across tables with varying schema and align out-domain data with the target task using a "learn, annotate, and audit" pipeline. The expanded training data enables the pre-trained AnyPredict to support every tabular dataset in the domain without fine-tuning, 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, AnyPredict 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

The rise of foundation models has enabled major breakthroughs in areas such as natural language processing (NLP) and computer vision (CV). These models are pre-trained on massive amounts of data and can perform well across many downstream tasks with zero-shot learning or minimal fine-tuning [1]. For instance, BERT [2] and GPT-3.5 [3] are two examples of foundation models that have significantly improved NLP tasks. Similarly, foundation models like SAM [4] and ImageBind [5] have demonstrated their effectiveness in CV tasks such as image segmentation and retrieval.

Despite the success of foundation models in NLP and CV, the use of such models in tabular prediction tasks has been limited. Tabular prediction is a fundamental task in many industries, including finance, healthcare, and marketing, among others [6, 7, 8, 9, 10]. Tabular data are typically 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. Despite its widespread use, current tabular prediction methods often rely on small, domain-specific datasets and do not generalize well across tabular datasets. This limitation can result in reduced prediction accuracy and increased development time, hindering the scalability and efficiency of tabular prediction tasks. This lack of large-scale, diverse tabular datasets with standardized labels also makes it challenging to pre-train foundation models specifically for tabular prediction. As a result, the field of tabular prediction has largely relied on traditional machine learning methods, such as tree ensembles (e.g., XGBoost [11]), although researchers have recently

Preprint. Under review.

began to explore the potential of deep learning [12, 13, 14, 15, 16]. We hypothesize that a foundation model for tabular prediction would bridge this gap, and it would furthermore unlock the potential of deep learning for this important class of problems.

We identify two major challenges in building tabular prediction foundation models:

- **Model**: Existing foundation models, particularly those used in NLP, are not equipped to handle numerical predictions or tabular inputs. Meanwhile, while there are some tabular prediction models available, they are often limited in size and restricted to a fixed set of columns for making predictions. Consequently, whenever a new tabular dataset is encountered, a new tabular prediction model must be trained from scratch, resulting in significant time and resource costs.
- **Data**: Scaling the size of training data is crucial for foundation models. However, tabular datasets pose a significant challenge for training foundation models due to the diversity and heterogeneity of feature types, schemas, and target labels across different data sources.

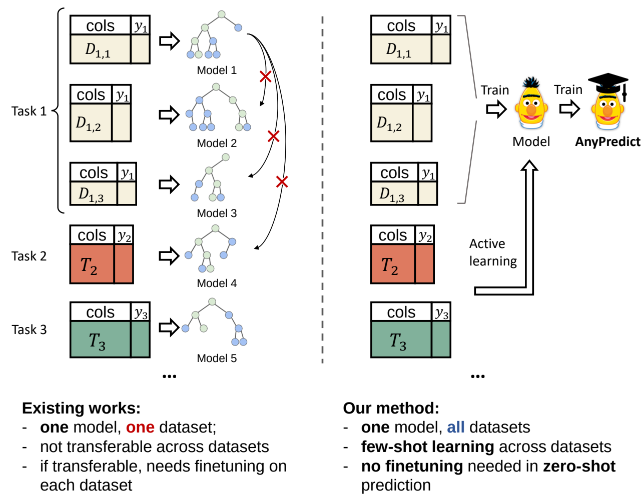

Figure 1: `AnyPredict` vs. existing tabular prediction methods. Existing methods learn and predict on a per-dataset basis, while `AnyPredict` can use data from the target task and all other tasks to improve performance.

In this paper, we introduce `AnyPredict`, a new approach for constructing foundation models for tabular prediction. Unlike existing methods that require multiple models for different input types or tasks, our method can predict any tabular input using a single foundation model. This is enabled by the following technical contributions:

- **Data consolidation and enrichment** involves consolidating tabular samples with varying features and schemas using natural language descriptions, while also expanding the training data through distilling knowledge from LLMs and incorporating diverse tabular datasets.
- **Data quality audit process** circumvents 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 active learning pipeline.

Consequently, `AnyPredict` offers the following key advantages that are typical of foundation models (demonstrated by Fig. 1):

- **Multi-task learning and prediction**: the model can learn from and make predictions for multiple 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** `AnyPredict` 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.

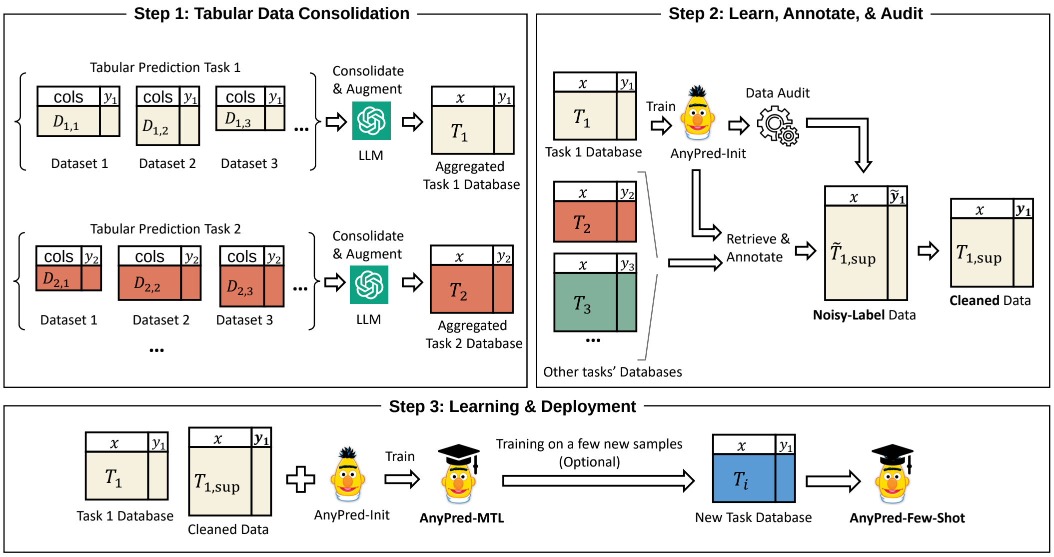

Figure 2: The demonstration of the proposed framework for building foundation models for tabular predictions (`AnyPredict`). It encompasses three steps: **Step 1** consolidates tabular datasets using LLM; **Step 2** aligns out-domain datasets with the target task; **Step 3** facilitates the predictor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

2 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 Fig. 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, `AnyPredict` 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 `AnyPredict` Framework

Our proposed framework is illustrated in Fig. 2, and consists of three main components in the pipeline.

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.

Step 2: Learn, Annotate, & Audit. Our method can benefit from wide-range out-domain datasets $\mathbf{T}_* \in \mathcal{T}$ through an active learning 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 for 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 $\mathbf{D} \notin \mathbf{T}$.

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

2.3 Tabular Data Consolidation & Sanity Check

The primary challenge in developing foundation models for tabular predictions is the scarcity of large datasets with standardized schemas, e.g., two clinical trial datasets have different schemas, as reflected in the sample data below.

age	gender	height	weight	…	mortality
18	f	1.7	60	…	0

demo1	demo2	demo3	ae1	ae2	…	target
25	160	0	0	1	…	14

Existing tabular prediction models often struggle on these datasets due to the vague semantic meanings of the varying columns and values. Our approach is to 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.

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 `{column}({type}):{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

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

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 $\{\mathbf{c}, \mathbf{v}\} \to \mathbf{x}$ in Eq. (1), particularly for numerical features. Specifically, we query LLM with the input template "`What is the {column}? {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 Appendix C.4, and the quantitative analysis of this correction is available in Appendix C.5.

2.4 Data Enrichment & Quality Audit

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 [17]. 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 [18, 19] 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 F explores the Shapley value and pseudo label 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 simply combine both datasets for training, but we have found that this approach results in suboptimal performance. Instead, we propose 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. The model can also 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.

3 Experiments

We conduct an extensive evaluation of `AnyPredict`'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.

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

Trial ID	Trial Name	# Patients	Categorical	Binary	Numerical	Positive Ratio
NCT00041119	Breast Cancer 1	3,871	5	8	2	0.07
NCT00174655	Breast Cancer 2	994	3	31	15	0.02
NCT00312208	Breast Cancer 3	1,651	5	12	6	0.19
NCT00079274	Colorectal Cancer	2,968	5	8	3	0.12
NCT00003299	Lung Cancer 1	587	2	11	4	0.94
NCT00694382	Lung Cancer 2	1,604	1	29	11	0.45
NCT03041311	Lung Cancer 3	53	2	11	13	0.64
External Patient Database						
	MIMIC-IV	143,018	2	1	1	N/A
	PMC-Patients	167,034	1	1	1	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.

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

(1) Patient Outcome Datasets. This dataset includes the patient records collected separately from seven oncology clinical trials [1]. These datasets each have their own unique schema and contain distinct groups of patients in different conditions. 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.

(2) Clinical Trial Outcome Datasets. We use clinical trial data from the `HINT` benchmark [20] 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}.

Implementations: For the patient outcome prediction task, we choose a tree ensemble method (XGBoost) [21], Multilayer Perceptron, FT-Transformer [12], and TransTab [14] as the baselines. For the trial outcome prediction task, we choose XGBoost, feed-forward neural network (FFNN) [22], DeepEnroll [23], COMPOSE [24], HINT [20], and SPOT [25] as the baselines. We use `PyTrial` [26] to implement most baselines, ane provide the details of the selected baselines in Appendix D.

We use a pre-trained bidirectional transformer model named BioBERT [27] as the classifier for `AnyPredict`. We utilize GPT-3.5 [28] via OpenAI's API [3] for the data consolidation and enhancement. We use UnifiedQA-v2-T5 [29] [4] for the sanity check. The evaluation metrics selected are ROC-AUC and PR-AUC, with the details in Appendix E. All experiments were run with 2 RTX-3090 GPUs and AMD Ryzen 3970X 32-Core CPU.

[1] https://data.projectdatasphere.org/projectdatasphere/html/access
[2] https://clinicaltrials.gov/
[3] Engine `gpt-3.5-turbo-0301`: https://platform.openai.com/docs/models/gpt-3-5
[4] Huggingface: allenai/unifiedqa-v2-t5-large-1363200

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

Trial Name	Metrics	XGBoost	MLP	FT-Transformer	TransTab	`AnyPredict`
Breast Cancer 1	AUROC	0.5526	0.6091	-	0.6088	**0.6262**
	PRAUC	0.0797	0.0963	-	0.1064	**0.1064**
Breast Cancer 2	AUROC	0.6827	0.6269	**0.8423**	0.7359	0.8038
	PRAUC	0.1559	0.1481	**0.1440**	0.0544	0.0775
Breast Cancer 3	AUROC	0.6489	0.7233	0.6532	0.7100	**0.7596**
	PRAUC	0.3847	0.4124	0.3221	0.4133	**0.4914**
Colorectal Cancer	AUROC	0.6711	0.6337	0.6386	**0.7096**	0.7004
	PRAUC	0.2314	0.1828	0.1990	0.2374	**0.2594**
Lung Cancer 1	AUROC	-	0.7465	-	0.6499	**0.8649**
	PRAUC	-	0.9789	-	0.9672	**0.9911**
Lung Cancer 2	AUROC	0.6969	0.6547	**0.7197**	0.5685	0.6802
	PRAUC	0.6865	0.5800	**0.7106**	0.4922	0.6687
Lung Cancer 3	AUROC	0.8393	-	-	0.6786	**0.9286**
	PRAUC	0.9276	-	-	0.7798	**0.9683**
Avg Rank		3.5714	3.7143	3.5714	2.8571	**1.5714**

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

Trial Data	Metrics	XGBoost	FFNN	DeepEnroll	COMPOSE	HINT	SPOT	`AnyPredict`
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**
Avg Rank		7.0000	6.0000	4.6667	4.3333	2.3333	2.6667	**1.0000**

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 `AnyPredict` and predict on all datasets, while the baselines need to be trained on each dataset separately. Our findings demonstrate that a single foundation model often 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.

`AnyPredict` 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 `AnyPredict` on two tasks. For the evaluation of the dataset $\mathbf{D}$, we deliberately exclude $\mathbf{D}$ from the training data during step 2, where 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 Fig. 3. `AnyPredict` exhibits impressive zero-shot performances: it wins over supervised XGBoost models in 5 out of 7 datasets in patient outcome prediction and all

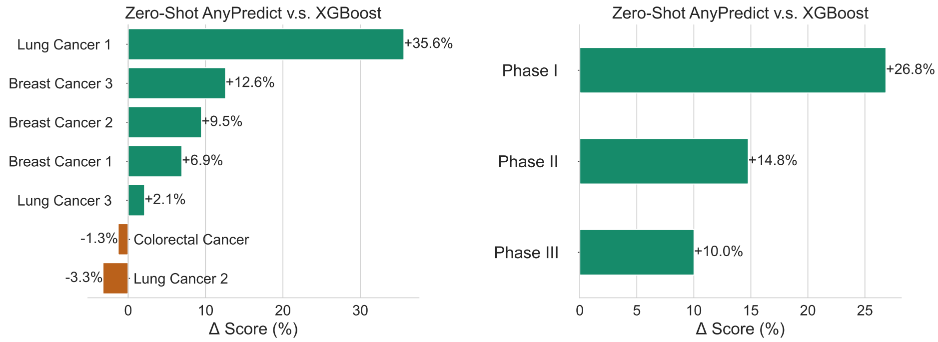

Figure 3: **Zero-shot** `AnyPredict` is better than a fully supervised baseline (XGBoost). The evaluation is across 7 patient outcome prediction datasets (left) and 3 trial outcome prediction datasets (right). The compared baseline XGBoost model is fitted on each dataset, respectively.

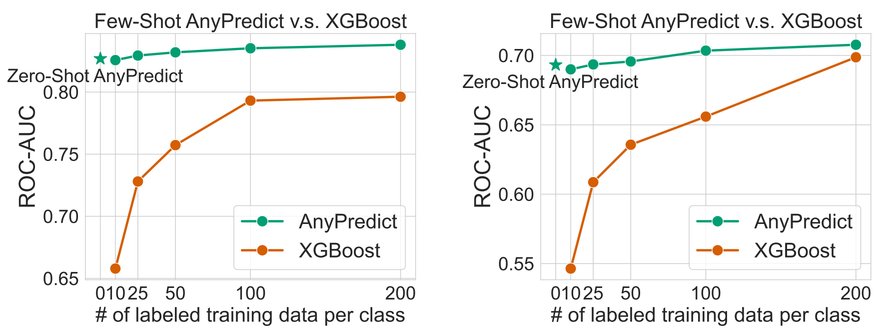

Figure 4: **Few-shot** `AnyPredict` compared with XGBoost with varying training data sizes. The compared baseline XGBoost model is fitted on each dataset, respectively.

three datasets in trial outcome prediction by a significant margin. On average, `AnyPredict` 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 foundation 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 Fig. 4. We are able to 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 `AnyPredict`.

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 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 Fig. 5 and detailed performances across datasets in Tables 6 and 7. Results show that `finetune` generally performs the best. We conjecture that jointly training on the target task and supplementary data improves

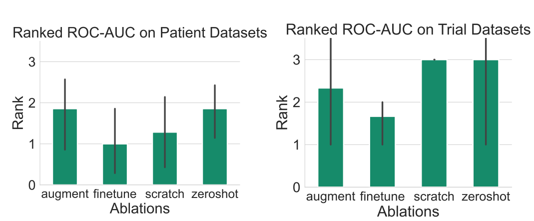

Figure 5: ROC-AUC Ranking (lower is better) of the variations of `AnyPredict`.

the model's overall utility, but it may affect the performance on specific samples in the target task $\mathbf{T}$. Furthermore, we also identify that `zeroshot` reaches comparable performances with `scratch`.

4 Related Work

Tabular Prediction has traditionally relied on tree ensemble methods [11, 30]. In recent years, the powerful representation learning abilities of neural networks have motivated the application of deep learning to tabular prediction [31, 13, 32]. Furthermore, recent research has proposed using transformer-based architectures [33, 12, 34] to enhance automatic feature interactions for better prediction performances. Nonetheless, there is still no definitive consensus on whether tree ensemble methods or deep learning approaches are superior for tabular prediction [12, 13, 14, 15, 16].

Data scarcity is a prevalent issue for tabular prediction. To address this challenge, self-supervised learning (SSL) has been extended to tabular prediction tasks. This includes approaches such as generative pretraining objective by masked cell modeling [7, 31, 35], and discriminative pretraining objective by self-supervised [36, 37, 38] or supervised contrastive learning [14]. Moreover, transfer learning was also adapted to tabular prediction, employing prompt learning based on generative language models [39] and multi-task learning [40]. Specifically, TransTab [14] designed a processor to generate dense embeddings from arbitrary tabular samples, enabling tabular sample encoding and transfer learning across tables via supervised learning. While these techniques show potential benefits of transfer learning in tabular predictions, no attempt was made to build foundation models that can perform predictions for multiple tabular datasets. In this paper, we provide a novel recipe for building `AnyPredict` that can make predictions for diverse tabular datasets without task-specific finetuning.

Foundation Models are trained on a broad set of data for many downstream tasks, with zero-shot learning, or minimal fine-tuning [1]. This conception emerged with the success of self-supervised transformers like BERT [2], and was followed by efforts in building foundation models for language [28, 3], vision [41, 4], and vision-language modeling [42, 43]. Foundation models have led to enormous gains for extensive domains when task-specific data is limited through transfer learning, weak supervision, or even zero-shot learning [42, 44]. However, tabular prediction tasks have not benefited from the development of foundation models as much because tabular data is uniquely domain-specific and do not share common features such as basic shape features or grammatical structure. While there have been efforts to build tabular pretraining models, they only work for fixed-column tabular data [7, 38] or do not utilize large pre-trained models and data at scale [14]. On the other hand, existing NLP/CV foundation models do not directly apply to tabular prediction tasks because they are either not optimized for numerical predictions or not amenable to tabular inputs.

5 Conclusion

In conclusion, we proposed a novel approach to training foundation models for tabular prediction tasks. While foundation models excel in various tasks, they face challenges in tabular predictions due to limited data availability, inconsistent dataset structures, and varying prediction targets across domains. To address these challenges, `AnyPredict` generates large-scale training data for tabular prediction foundation 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 audit* pipeline. By leveraging the expanded training data, `AnyPredict` can effectively work on any tabular dataset within the domain without requiring further fine-tuning, achieving significant improvements compared to supervised baselines. Moreover, `AnyPredict` demonstrates impressive performance even with limited examples (few-shot) or no examples (zero-shot), remaining competitive with supervised approaches across various tabular datasets.

References

[1] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[2] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[3] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[5] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. ImageBind: One embedding space to bind them all, 2023.

[6] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, pages 1–7. 2017.

[7] 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.

[8] Yixuan Zhang, Jialiang Tong, Ziyi Wang, and Fengqiang Gao. Customer transaction fraud detection using xgboost model. In *International Conference on Computer Engineering and Application*, pages 554–558. IEEE, 2020.

[9] Zifeng Wang and Suzhen Li. Data-driven risk assessment on urban pipeline network based on a cluster model. *Reliability Engineering & System Safety*, 196:106781, 2020.

[10] Zifeng Wang, Xi Chen, Rui Wen, Shao-Lun Huang, Ercan Kuruoglu, and Yefeng Zheng. Information theoretic counterfactual learning from missing-not-at-random feedback. *Advances in Neural Information Processing Systems*, 33:1854–1864, 2020.

[11] 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.

[12] 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.

[13] 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.

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

[15] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

[16] Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Ganesh Ramakrishnan, Micah Goldblum, Colin White, et al. When do neural nets outperform boosted trees on tabular data? *arXiv preprint arXiv:2305.02997*, 2023.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

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

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

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

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[43] Zifeng Wang, Zhenbang Wu, Dinesh Agarwal, and Jimeng Sun. MedCLIP: Contrastive learning from unpaired medical images and text. In *EMNLP*, 2022.

[44] Chufan Gao, Mononito Goswami, Jieshi Chen, and Artur Dubrawski. Classifying unstructured clinical notes via automatic weak supervision. In *Machine Learning for Healthcare Conference*, 2022.

[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

A Broader Impact

Tabular data, which is commonly used in various fields such as healthcare, finance, systems monitoring, and more, has received less attention compared to other domains like vision, language, and audio in terms of investigating the generalizability of its models. Non-deep algorithms, particularly tree-based methods such as XGBoost and Random Forest, still dominate the analysis of tabular data but generally be less flexible in terms of their input features and the target task. This research paper introduces `AnyPredict`, a framework that creates a foundational tabular model in a particular domain by leveraging knowledge transfer from pre-trained LLMs.

`AnyPredict` excels in handling input tables with varying column numbers and leveraging publicly available unstructured text in the domain, saving significant time and resources in data engineering and cleaning. It enables knowledge transfer from models trained on sensitive data and facilitates the use of zero-shot models that match fully supervised baselines. While further research is needed to generalize `AnyPredict` to more application domains, it is a significant step in creating general tabular models comparable to foundational models in fields like CV and NLP.

B Limitations

However, the current framework has drawbacks. Data privacy concerns arise if patient data is handled through an unsecured server vulnerable to exploitation by malicious third parties. To ensure safety, proper security procedures must be followed. If resources permit, training a private model specifically for paraphrasing would yield the safest outcome. Another issue is the tendency of language models to generate hallucinated text. While metrics exist to measure the presence of original information, measuring the creation of new, false information remains an open problem in NLP. Although some hallucinations may aid downstream models in interpreting table data, future research should include sanity checks to minimize irrelevant hallucinations. Additionally, the exploration of using external, supplemental datasets as vehicles of knowledge transfer and / or distantly supervised data via data Shapley is highly interesting, and requires more exploration. Future use cases may even reduce the risk of privacy leaks by creating datasets from publicly available, open data, that preserve performance of models trained on more sensitive data.

C Data Consolidation and Augmentation: More Details

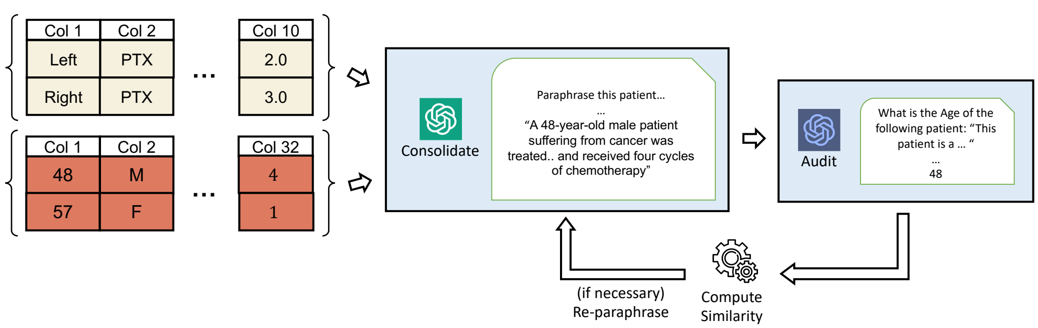

Figure 6: The paraphrasing pipeline is showing an example of a specific patient. The raw tabular data (of varying schema) is converted to unstructured natural language text by a first LLM and then audited by a secondary LLM with QA. Errors are then used to correct the initial conversion. Patient features have been changed to preserve anonymity.

C.1 Task Prompt Templates

In order to generate the desired text, we start by generating primitive sentences that capture the essential information from the original row of data. For categorical and numerical features, we combine the feature name and its corresponding values into a single sentence. For binary features, we include the feature name in the primitive sentence only if the value is `True` in order to avoid generating false information. We then use GPT-3.5 to paraphrase these primitive sentences. Furthermore, these texts are audited and corrected, resulting in a final text that accurately represents the original data. An illustration of this process can be found in Figure 6.

The prompt and suffix that we leverage to consolidate a row are as follows:

Listing 1: prompt for data consolidation

```
prompt = '''
Here is the schema definition of the table:

{schema_definition}

This is a sample from the table:

{linearization}

Please describe the sample using natural language.
'''
```

If we want to augment the original samples by paraphrasing, we use the prompt as follows:

Listing 2: prompt for instance-level augmentation

```
prompt = '''
Here is the schema definition of the table:

{schema_definition}

This is a sample from the table:

{linearization}

Please paraphrase the sample in 5 different ways in natural language.
'''
```

C.2 Example Consolidations

Here are 7 examples of consolidations as given after the linearization of the samples in the trial datasets (patient details changed for clarity).

Breast Cancer 1

race White; treatment paclitaxel; tumor laterality left; cancer histologic grade Low; biopsy type incisional; post-menopause ; estrogen receptor positive ; progesterone receptor positive ; prior hormonal therapy ; number of positive axillary nodes 0; tumor size 3.0	The patient has been confirmed to be positive for human epidermal growth factor receptor 2. She is of White race and is currently undergoing treatment using paclitaxel. The tumor is on the patient's left side and has a low histologic grade. The biopsy type used was incisional, and the patient is post-menopausal. Additionally, the patient is estrogen receptor positive, progesterone receptor positive, and has undergone prior hormonal therapy. The tumor size is approximately 3.0 units.

Breast Cancer 2

sex Female; histopathologic grade Moderately differentiated; histopathologic type Infiltrating ductal carcinoma; adverse effect: diarrhea ; surgery: mastectomy ; multifocal tumor ; estrogen receptor positive ; medical condition: history of tobacco use ; drug: zofran ; drug: tamoxifen ; age 48.0; tumor size 2.5; number of positive axillary lymph nodes 3.0; number of resected axillary lymph nodes 19.0; lab test: hemoglobin 7.59; lab test: neutrophils 3.03; lab test: platelets 327.65; lab test: white blood cells 3.78; lab test: asat (sgot) 30.846; lab test: alkaline phosphatase 172.385; lab test: alat (sgpt) 38.0; lab test: total bilirubin 7.462; lab test: creatinine 64.9; height 166.0; weight 84.6	The patient, a 48-year-old female, had moderately differentiated infiltrating ductal carcinoma and underwent lumpectomy and mastectomy surgeries for a multifocal tumor. She has a medical history of tobacco use and experienced adverse effects from penicillins. Her lab tests indicate a hemoglobin of 7.59, slightly low neutrophils at 3.03, high platelets at 327.65, and a low white blood cell count of 3.78. She also has elevated asat (sgot) levels at 30.846, slightly high alkaline phosphatase at 172.385, alat (sgpt) at 38.0, total bilirubin at 7.462, and a creatinine of 64.9. She also suffered from vomiting and diarrhea, which were side effects of her medication, though she was given zofran to alleviate those symptoms.

Breast Cancer 3

estrogen receptor positive ; progesterone receptor positive ; age 64.0; number of resected axillary node 11.0; numer of positive axillary node 7.0; primary tumor size 3.0; weight 113.0; height 162.0	The patient, a 64-year-old with estrogen and progesterone receptor positive breast cancer, underwent surgery to remove 11 axillary nodes, with 7 testing positive. She was prescribed anti-tumor therapies including Xeloda, Taxotere, Arimidex, Zoladex, and Cyclophosphamide. Unfortunately, she experienced the adverse effect of febrile neutropenia.

Colorectal Cancer

arms Oxaliplatin + 5-fluorouracil/Leucovorin + Cetuximab; histology well differentiated; race white; sex female; biomarker kras wild-type; adherence ; age 45.0; ecog performance score 1; bmi 24.883	The patient experienced a negative reaction of blood clotting, specifically thrombosis, as well as a harmful effect called infarction in their arms as a result of taking Oxaliplatin + 5-fluorouracil/Leucovorin + Cetuximab.

Lung Cancer 1

gender male; treatment arm paclitaxel, cisplatin, etoposide,G-CDF; adverse effects: granulocytes/bands ; adverse effects: wbc ; adverse effects: lymphocytes ; adverse effects: platelets ; adverse effects: dyspnea ; adverse effects: nausea ; adverse effects: other miscellaneous 2 ; age 76.0; number of metastatic sites 1.0; number of chemotherapy cycles 2.0; ecog performance status 0.0	A 76-year-old man receiving paclitaxel, cisplatin, etoposide, G-CDF for his single metastatic site experienced a variety of adverse effects, including dyspnea, nausea, and other miscellaneous symptoms. He also had decreased levels of granulocytes/bands, white blood cells, lymphocytes, and platelets.

Lung Cancer 2

sex M; aderse effect: severe malignant neoplasm progression ; aderse effect: severe neutropenia ; aderse effect: severe thrombocytopenia ; aderse effect: severe anaemia ; aderse effect: severe vomiting ; aderse effect: severe pneumonia ; aderse effect: severe diarrhoea ; aderse effect: severe abdominal pain ; aderse effect: severe sepsis ; aderse effect: severe leukopenia ; medication: dexamethasone ; medication: ondansetron ; medication: heparin ; medication: ranitidine ; medication: cisplatin ; medication: metoclopramide ; medication: carboplatin ; medication: furosemide ; age 67; lab test: hemoglobin 134.0; lab test: leukocytes 4.3; lab test: creatinine clearance 92.489; lab test: creatinine 84.0; lab test: platelet 256.0; lab test: bilirubin 18.5; lab test: alanine aminotransferase 36.1; lab test: aspartate aminotransferase 17.9; lab test: neutrophils 2.58; lab test: alkaline phosphatase 556.0	The patient, a 67-year-old male, has undergone lab tests revealing that their hemoglobin is 134.0, leukocytes are 4.3, creatinine clearance is 92.489, creatinine is 84.0, platelet count is 256.0, bilirubin is 18.5, alanine aminotransferase is 36.1, aspartate aminotransferase is 17.9, neutrophils are 2.58, and alkaline phosphatase is 556.0. They have a medical history of antiandrogen therapy, chronic cardiac failure, and venous insufficiency. However, the patient has experienced adverse effects, including severe malignant neoplasm progression, neutropenia, thrombocytopenia, anemia, vomiting, pneumonia, diarrhea, abdominal pain, sepsis, and leukopenia, due to the medications dexamethasone, ondansetron, heparin, ranitidine, cisplatin, metoclopramide, carboplatin, and furosemide.

Lung Cancer 3

sex M; smoke status Former Smoker; lactate dehydrogenase isoenzymes test abnormal ; adverse effect: neutropenia ; adverse effect: anaemia ; adverse effect: thrombocytopenia ; drug: ondansetron ; drug: prednisone ; age 0.671; ecog performance score 1; height 0.805; weight 0.517; lab test: leukocytes (10^9/l) 4.996; lab test: hemoglobin (g/l) 108.816; lab test: platelets (10^9/l) 175.895; lab test: lymphocytes (10^9/l) 1.039; lab test: platelets/lymphocytes 176.287; lab test: neutrophils (10^9/l) 3.367; lab test: neutrophils/lymphocytes 3.221; lab test: monocytes (10^9/l) 0.509; lab test: eosinophils (10^9/l) 0.068	The patient's lab results show a leukocyte count of 4.996 x 10^9/L, hemoglobin level of 108.816 g/L, platelet count of 175.895 x 10^9/L, platelet/lymphocyte ratio of 176.287, neutrophil count of 3.367 x 10^9/L, neutrophil/lymphocyte ratio of 3.221, and eosinophil count of 0.068 x 10^9/L. The patient has experienced adverse effects of decreased neutrophil count, platelet count, neutropenia, anemia, and thrombocytopenia. They were prescribed ondansetron and prednisone. The patient is male, a former smoker, and has an ecog performance score of 1. They are 0.671 years old and have a height of 0.805 and weight of 0.517.

C.3 Data Augmentation

We show a couple of examples of data augmentation as obtained by paraphrasing 5 times:

```
1.  The patient is White, and receiving paclitaxel for a
left-sided tumor with Low histologic grade.  An incisional biopsy
was performed on the tumor.  The patient is post-menopausal and
has estrogen and progesterone receptor-positive cancer.  They had
prior hormonal therapy and no positive axillary nodes.  The tumor
size is 3.0.
2.  A left-sided tumor with Low histologic grade is being treated
with paclitaxel for a White patient who had an incisional biopsy.
The patient is post-menopausal and has estrogen and progesterone
receptor-positive cancer, as well as prior hormonal therapy.
There are zero positive axillary nodes and the tumor size is 3.0.
3.  For a White patient, paclitaxel is the treatment for a
left-sided tumor with Low histologic grade.  An incisional biopsy
was performed, and the patient is post-menopausal with estrogen
and progesterone receptor-positive cancer.  They had prior
hormonal therapy, zero positive axillary nodes, and the tumor size
is 3.0.
4.  The tumor laterality is left for a White patient receiving
paclitaxel, with a Low histologic grade.  The biopsy type was
incisional, and the patient is post-menopausal with estrogen and
progesterone receptor-positive cancer.  They had prior hormonal
therapy and no positive axillary nodes, with a tumor size of 3.0.
5.  An incisional biopsy was performed on a left-sided, Low
histologic grade tumor for a White patient receiving paclitaxel.
They are post-menopausal with estrogen and progesterone
receptor-positive cancer, having had prior hormonal therapy.
There are zero positive axillary nodes, and the tumor size is 3.0.
```

```
1.  The patient is a 62-year-old white woman with
well-differentiated histology.  She is taking arms Oxaliplatin,
5-fluorouracil/Leucovorin, and Cetuximab.  She has Kras wild-type
biomarkers and an ECOG performance score of 1.  Her BMI is 24.883
and she is adhering to her treatment plan.
2.  A white female patient with well-differentiated
histology is being treated with a combination of Oxaliplatin,
5-fluorouracil/Leucovorin, and Cetuximab.  She is 62 years old,
has Kras wild-type biomarkers, and an Ecog performance score of 1.
She is maintaining a BMI of 24.883 and adhering to her medication.
3.  An adherence patient, who is a 62-year-old white woman, has
well-differentiated histology and is taking arms Oxaliplatin,
5-fluorouracil/Leucovorin, and Cetuximab.  Her Kras biomarkers
are wild-type and she has an Ecog performance score of 1.  She is
maintaining a BMI of 24.883.
4.  The patient is a well-differentiated white female with
Kras wild-type biomarkers.  She is being treated with arms
Oxaliplatin, 5-fluorouracil/Leucovorin, and Cetuximab and has an
ECOG performance score of 1.  She is 62 years old, adhering to her
medication, and has a BMI of 24.883.
5.  A 62-year-old white woman with good histology is taking arms
Oxaliplatin, 5-fluorouracil/Leucovorin, and Cetuximab.  She has
Kras wild-type biomarkers, an Ecog performance score of 1, and is
adhering to her medication.  Her BMI is 24.883.
```

C.4 Sanity Check with LLM's Relection

The generated text must undergo auditing to ensure practical suitability for downstream tasks and human interpretability. This is especially critical when dealing with tabular data, as accurate paraphrasing is essential for maintaining important information that can significantly impact model performance. Thus, verifying faithful paraphrasing of the data is of utmost importance.

To evaluate the fidelity of the paraphrased text, we employ a cutting-edge Question-Answering model to test the paraphrased text. For different features, we query the model using the prompts below.

Listing 3: prompt for sanity check of categorical features

```
prompt = '''
{consolidated_output}

What is the value of {feature_name}?
'''
```

Listing 4: prompt for sanity check of binary features

```
prompt = '''
{consolidated_output}

Is {feature_name} present in the above paragraph? (a) yes (b) no.
'''
```

We then compare the QA model's answers with the actual feature values and calculate a normalized edit distance score (edit distance of longest string) as a measure of the paraphrase's accuracy:

$$\hat{D} = 1 - \frac{D}{\max(l_1, l_2)}, \tag{3}$$

where D represents the edit distance of two strings; l_1 and l_2 are length of two input strings.

If the matching metric is too low, we add it to the list of features we wish to preserve. Then, we rerun the paraphrasing with a secondary prompt as shown here:

Listing 5: prompt for correcting a generated text

```
prompt = '''
Here is the schema definition of the table:

{schema_definition}

Please paraphrase the following in natural language.

{previous_paraphrase} + {linearization_of_missed_features}
'''
```

C.4.1 Examples

In this case, the first paraphrase completely left out the severe adverse effects.

```
A male patient, aged 70.0
years, whose race is white
having poorly differentiated
histology and a wild-type Kras
biomarker, is taking Oxaliplatin
and 5-fluorouracil/Leucovorin
along with Cetuximab.  His ECOG
performance score is 0, and BMI
is 27.554.
```

```
A male patient who is white
and 70 years old is currently
being treated with Oxaliplatin
+ 5-fluorouracil/Leucovorin
+ Cetuximab for poorly
differentiated histology with
kras wild-type biomarkers.  He
has a BMI of 27.554 and a ECOG
performance score of 0.  There
have been severe adverse effects
of thrombosis, hypersensitivity,
infarction, and diarrhea,
and there is a risk of bowel
perforation.
```

In this case, the first paraphrase did not preserve the lab results:

```
This male patient has a history
of smoking but is now a former
smoker.  Unfortunately, he has
developed brain metastasis and an
abnormal lactate dehydrogenase
isoenzymes test result.  He
experienced neutropenia as
a side effect of treatment.
Additionally, he has a relatively
young age, high ecog performance
score, and low height and weight.
His lab tests show elevated
leukocytes and platelets, low
lymphocytes and hemoglobin, and
abnormal neutrophils, monocytes,
and eosinophils.
```

```
The male patient, who is a
former smoker, has a brain
metastasis and abnormal lactate
dehydrogenase isoenzymes test
result.  He is 0.429 years old
and has an ecog performance score
of 2.  With a height of 0.732
and weight of 0.517, his lab
tests reveal a leukocyte count
of 8.371 (10^9/l), hemoglobin
level of 109.0 (g/l), platelet
count of 337.286 (10^9/l),
lymphocyte level of 0.917
(10^9/l), platelet to lymphocyte
ratio of 815.19, neutrophil
count of 6.923 (10^9/l),
neutrophil to lymphocyte ratio
of 6.187, monocyte count of
0.424 (10^9/l), and eosinophil
count of 0.039 (10^9/l).  The
patient experienced adverse
effects of anaemia, decreased
neutrophil and platelet count,
and neutropenia, while being
treated with dexamethasone,
ondansetron, prednisone, and
sodium chloride.
```

C.5 Evaluation of Hallucinations Produced in Consolidation

Table 5: The Mean Normalized Edit Distance (MNED). Higher is better, the range is [0,1]. For each dataset as calculated on the feature values retrieved by the QA model vs the actual feature value.

Dataset	MNED	MNED (After Correction)
Breast Cancer 1	0.7197	0.8319
Breast Cancer 2	0.5007	0.5199
Breast Cancer 3	0.3463	0.7247
Colorectal Cancer	0.5498	0.7366
Lung Cancer 1	0.4287	0.6266
Lung Cancer 2	0.5059	0.5624
Lung Cancer 3	0.3216	0.3308

We show the quantitative analysis of the hallucinations during the sanity check in Table 5. We see that the normalized Edit Distance (Eq. (3)) shows that the paraphrasing may not preserve all of the original values of the row, but that also, re-paraphrasing does significantly help. However, this is not as bad as of an issue as it may appear. Since the performance with data augmentation is higher than without. Mean Normalized Edit Distance only measures the character differences between the strings. Upon manual inspection, there are many cases where feature values are paraphrased into a similar meaning. In other cases, the GPT-3.5 model does fail to summarize the patient completely. For example, in the Lung Cancer datasets, there are many lab numerical values that the generator fails to preserve within a reasonable max length of generated text, even with multiple tries. Further work is required to generate better paraphrases as well as improved metrics.

D Baseline Models

The baselines for **patient outcome prediction** datasets:

- XGBoost [11]: This is a tree ensemble method augmented by gradient-boosting. We use its official implementation of Python interface [5] in our experiments. We use ordinal encoding for categorical and binary features and standardize numerical features via `scikit-learn` [45]. We encode textual features, e.g., patient notes, via a pre-trained BioBERT [27] model. The encoded embeddings are fed to XGBoost as the input. We tune the model using the hyperparameters: *max_depth* in $\{4, 6, 8\}$; *n_estimator* in $\{50, 100, 200\}$; *learning_rate* in $\{0.1, 0.2\}$; We take early-stopping with patience of 5 rounds.
- Multilayer Perceptron (MLP): This is a simple neural network built with multiple fully-connected layers. We use the implementation from the individual outcome prediction module of `PyTrial` [6]. The model is with 2 dense layers where each layer has 128 hidden units. We tune the model using the hyperparameters: *learning_rate* in $\{$1e-4,5e-4,1e-3$\}$; *batch_size* in $\{32, 64\}$; We take the max training *epochs* of 10; *weight_decay* of 1e-4.
- FT-Transformer [12]: This is a transformer-based tabular prediction model. We use the implementation from the individual outcome prediction module of `PyTrial`. The model is with 2 transformer modules where each module has 128 hidden units in the attention layer and 256 hidden units in the feed-forward layer. We use multi-head attention with 8 heads. We tune the model using the hyperparameters: *learning_rate* in $\{$1e-4,5e-4,1e-3$\}$; *batch_size* in $\{32, 64\}$; We take the max training *epochs* of 10 and *weight_decay* of 1e-4.
- TransTab [14]: This is a transformer-based tabular prediction model that is able to learn from multiple tabular datasets. Following the transfer learning setup of this method, we take a two-stage training strategy: first, train it on all datasets in the task, then fine-tune it on each dataset and report the evaluation performances. We use the implementation from the individual outcome prediction module of `PyTrial`. The model is with 2 transformer modules where each module has 128 hidden units in the attention layer and 256 hidden units in the feed-forward layer. We use

[5] DMLC XGBoost: https://xgboost.readthedocs.io/en/stable/

[6] `pytrial.indiv_outcome`: https://pytrial.readthedocs.io/en/latest/pytrial.tasks.indiv_outcome.html

multi-head attention with 8 heads. We tune the model using the hyperparameters: *learning_rate* in {1e-4,5e-4,1e-3}; *batch_size* in $\{32, 64\}$; We take the max training *epochs* of 10 and *weight_decay* of 1e-4.

The baselines for **clinical trial outcome prediction** datasets:

- XGBoost [11]: This is a tree ensemble method augmented by gradient-boosting. We follow the setup used in [20].
- FFNN [22]: It is a feed-forward neural network, which has 3 fully-connected layers with the dimensions of dim-of-input-feature, 500, and 100, and ReLU activations. We follow the setup used in [20].
- DeepEnroll [23]: It was originally aimed at facilitating patient-trial matching, encompassing a hierarchical embedding network designed to encode disease ontology. Additionally, an alignment model was incorporated to capture the interconnections between eligibility criteria and disease information. We follow the setup used in [20].
- COMPOSE [24]: Initially, its application revolved around patient-trial matching, employing a combination of a convolutional neural network and a memory network. The convolutional neural network encoded eligibility criteria, while the memory network encoded diseases. To enhance trial outcome prediction, the model's embedding was concatenated with a molecule embedding using MPNN (Message Passing Neural Network). We follow the setup used in [20].
- HINT [20]: It integrates several key components. Firstly, there is a drug molecule encoder utilizing MPNN (Message Passing Neural Network). Secondly, a disease ontology encoder based on GRAM is incorporated. Thirdly, a trial eligibility criteria encoder leveraging BERT is utilized. Additionally, there is a drug molecule pharmacokinetic encoder, and a graph neural network is employed to capture feature interactions. Subsequently, the model feeds the interacted embeddings into a prediction model for accurate outcome predictions. We follow the setup used in [20].
- SPOT [25]: The Sequential Predictive Modeling of Clinical Trial Outcome (SPOT) is an innovative approach that follows a sequential process. Initially, it identifies trial topics to cluster the diverse trial data from multiple sources into relevant trial topics. Next, it generates trial embeddings and organizes them based on topic and timestamp, creating structured clinical trial sequences. Treating each trial sequence as an individual task, SPOT employs a meta-learning strategy, enabling the model to adapt to new tasks with minimal updates swiftly. We follow the setup used in [25].

E Evaluation Metrics

We consider the following performance metrics: (1) **AUROC**: the area under the Receiver Operating Characteristic curve summarizes the trade-off between the true positive rate (TPR) and the false positive rate (FPR) with the varying threshold of FPR. In theory, it is equivalent to calculating the ranking quality by the model predictions to identify the true positive samples. However, better AUROC does not necessarily indicate better outputting of well-calibrated probability predictions. (2) **PRAUC**: the area under the Precision-Recall curve summarizes the trade-off between the precision (PPV) and the recall (TPR) with the varying threshold of recall. It is equivalent to the average of precision scores calculated for each recall threshold and is more sensitive to the detection quality of true positives from the data, e.g., identifying which trial is going to succeed.

F Data Shapley Distributions

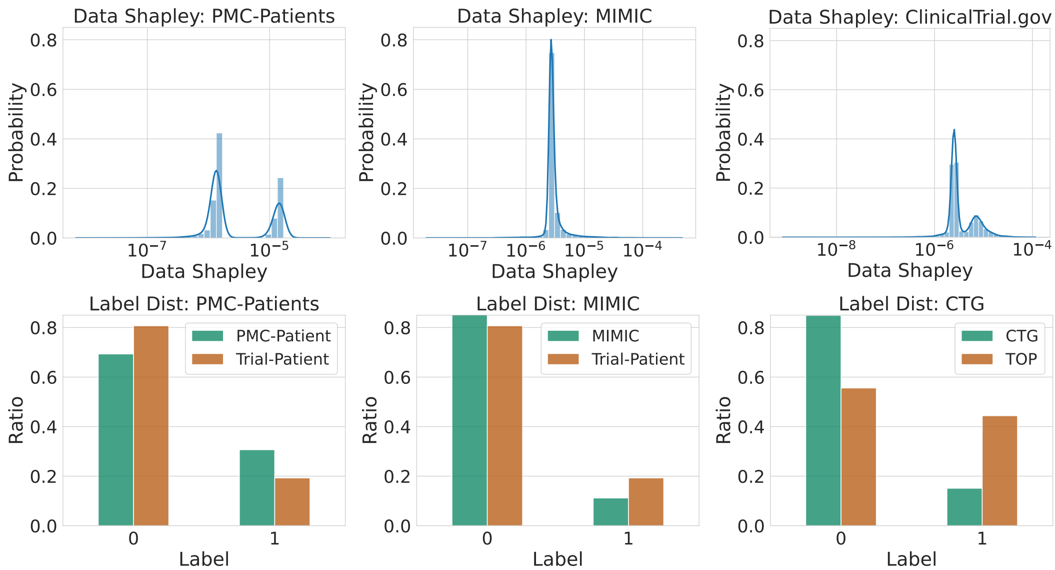

Figure 7: A plot of the distributions of the data Shapley values as well as the label ratio of 3 datasets. We see that in both the PMC-Patients and ClinicalTrial.gov datasets, there exist a bimodal distribution of shapley values, indicating that there exists a subset of data subset that is more relevant than the rest. The MIMIC distribution is more unimodal, albeit it is slightly right tailed. We also see that the predicted labels generally match the original distribution's labels, which indicates that the generated supplemental dataset does indeed preserve the true label imbalance.

G Ablation Results

Table 6: Ablations of `AnyPredict` with its variants. The evaluation is made on patient outcome prediction datasets.

Trial Name	Metrics	Augment	Finetune	Scratch	Zero-Shot
Breast Cancer 1	ROC-AUC	**0.624**	0.617	0.591	0.591
	PR-AUC	**0.111**	0.103	0.094	0.102
Breast Cancer 2	ROC-AUC	0.713	**0.841**	0.803	0.742
	PR-AUC	0.049	**0.071**	0.060	0.051
Breast Cancer 3	ROC-AUC	0.734	**0.741**	0.721	0.731
	PR-AUC	0.456	**0.486**	0.437	0.391
Colorectal Cancer	ROC-AUC	0.677	0.697	**0.705**	0.662
	PR-AUC	0.236	0.244	**0.267**	0.207
Lung Cancer 1	ROC-AUC	0.623	**0.822**	0.699	0.678
	PR-AUC	0.963	**0.987**	0.971	0.969
Lung Cancer 2	ROC-AUC	0.682	0.677	**0.711**	0.677
	PR-AUC	0.673	0.669	**0.691**	0.671
Lung Cancer 3	ROC-AUC	**0.893**	**0.893**	**0.893**	**0.893**
	PR-AUC	0.948	0.948	**0.957**	0.938

Table 7: Ablations of `AnyPredict` with its variants. The evaluation is made on clinical trial outcome prediction datasets.

Trial Data	Metrics	Augment	Finetune	Scratch	Zero-Shot
Phase I	ROC-AUC	**0.706**	0.701	0.699	0.657
	PR-AUC	0.725	0.722	**0.726**	0.704
Phase II	ROC-AUC	0.718	**0.726**	0.706	0.689
	PR-AUC	**0.747**	0.743	0.733	0.728
Phase III	ROC-AUC	0.724	0.729	0.726	**0.734**
	PR-AUC	0.863	0.876	**0.881**	0.877