Tabby: Tabular Adaptation for Language Models

Sonia Cromp, Satya Sai Srinath Namburi GNVV, Catherine Cao,
Mohammed Alkhudhayri, Samuel Guo, Nicholas Roberts, Frederic Sala
Department of Computer Sciences
University of Wisconsin-Madison
Madison, WI 53705
{cromp, sgnamburi, ccao35}@wisc.edu
{malkhudhayri, sguo258, nick11roberts, fsala}@wisc.edu

Abstract

While the quality of synthetic text data has greatly improved in recent years, thanks to specialized architectures such as large language models (LLMs), tabular data has received relatively less attention. To address this disparity, we present **Tabby**: a modification to the LLM architecture that enables its use for tabular dataset synthesis. Tabby consists of a novel adaptation of Gated Mixture-of-Expert layers, allowing each data column to be modeled by dedicated parameters within the Transformer multi-layer perceptrons or language modeling head. Applying Tabby to Distilled GPT-2 improves synthetic data quality (measured by machine learning efficacy) by up to 2.7% compared to previous tabular dataset synthesis methods, achieving performance near or equal to that of real data.

1 Introduction

Tabular data is ubiquitous within many domains. Airplane black boxes, website visitor logs and hospital patient records are just a few examples of this versatile modality, which contains rows representing datapoints and columns representing data features. Despite the wide utility of tabular data, this modality has received less attention in modern machine learning than images and text.

The need for improved tabular machine learning techniques has been highlighted by many [9, 23, 30, 6]. Furthermore, it has been speculated that effective tabular data models may have an equal, if not larger value and impact, than text and image models [30]. Humans are capable of interpreting complex images and skimming complicated texts on-the-fly. Meanwhile, humans are reliant on tools such as Pandas [19] or Excel [7] to achieve the same ability of capturing the patterns, ideas and general "big-picture" understanding of tabular data. For this reason, machine learning models may have an easier time exceeding human performance in the tabular modality than in images and text.

Despite the large performance gains that are likely possible in further tabular machine learning research, this modality's progress is slowed by several key challenges. First, tabular data columns may exhibit complex interdependencies, with complexity increasing proportionately to the number of columns in the dataset. Second, many tabular datasets are in fact a combination of various modalities, with text, numerical, and nested datatypes (such as a JSON, dictionary, or other structured object) possible among the columns of one dataset. Third, although the order of items within one column of one row is important, the order of columns with respect to each other is not usually meaningful. The best way to design and train models that are capable of learning arbitrary column modalities without learning spurious correlations caused by the order of dataset columns remains an open question.

There have been notable efforts to extend several preexisting model architectures to this modality. The first works in this direction focused primarily on GANs [11, 32], while more recent works include diffusion models and large language models (LLMs). These deep-learning-based works achieve

Table Representation Learning Workshop at NeurIPS 2024.

higher generation quality, but require significantly more data and compute resources to train than GAN-based methods, a drawback particularly notable for diffusion models. For these reasons, works such as van Breugel and van der Schaar [30] have called for the development of specialized pretrained Large Tabular Models (LTMs), to fill a similar role as Large Language Models (LLMs) or Image Foundation Models including GPT [20] and DALL-E [1]. However, the creation of LTMs would require large and diverse tabular pretraining sets which do not currently exist, a specialized tabular model architecture which has yet to be designed, along with a staggering amount of compute resorces for pretraining.

In this work, we take an initial step towards the development of a LTM by proposing **Tabby**, a modification to the LLM architecture for enabling tabular data synthesis. Tabby's core innovation is the application of Mixture-of-Expert (MoE) layers [27] to tabular synthesis. To our knowledge, Tabby is the first architecture modification to make LLMs better-suited to table generation. Using a pretrained LLM as a starting point allows Tabby to take advantage of its diverse text pretraining, avoiding the logistical challenges of training a LTM entirely from scratch. According to our evaluations using machine learning efficacy with a Decision Tree downstream classifier, Tabby's synthetic data is the first to reach parity with non-synthetic data in two out of three datasets (and achieves the best results out of all works in the third dataset). We summarize our contributions as follows:

- We introduce Tabby, the first architecture modification that allows LLMs to be better-suited to generating tabular data.
- We explore multiple tabular training techniques for LLMs, including our Plain method: a simple, lightweight training technique that may serve as an effective baseline for training future tabular LLM works.
- We demonstrate that Tabby produces higher-quality synthetic data across three datasets, and also allows greater insights into the model's performance and training progress than other tabular synthesis approaches.

After a discussion of this area's preexisting work in Section 2, we provide more details on Tabby in Section 3. Next, we conduct extensive experiments in Section 4.

2 Related Work

Tabular data has played a central role in machine learning since the field's early days. In particular, Decision Trees [29] and their relatives such as Random Forests [5] are well-adapted to classification or regression tasks on tabular datasets.

Classical synthesis: Classical machine learning methods may be used to synthesize tabular data, by modeling each column as a random variable and sampling from the multivariate distribution representing all columns in the dataset. This technique has been successfully applied to decision trees [25] and Bayesian networks [2, 33] . Copulas [10, 14, 4] are another traditional approach, which rely on first modeling each column as a univariate random distribution, then fitting a probabilistic model to the multivariate distribution formed by all columns. However, these approaches are limited in the data types that may be represented among the columns and the varieties of relationships that may be modeled across columns.

Deep learning approaches: For these reasons, several tabular synthesis methods rely on deep learning techniques– in particular, Generative Adversarial Networks (GANs) [11]. CTGAN [32], a top-performing approach, is designed to address the inherent limitations of GANs for tabular data generation. In particular, the distributions of ordinal columns are frequently rather imbalanced, leading less-sophisticated models to undesirable phenomena such as mode collapse. Continuous columns may possess multiple modes and complex interactions with the other columns which GANs struggle to capture. CTGAN employs conditional generation to address these shortcomings. However, the fidelity of CTGAN's synthetic data leaves further improvements to be desired, as demonstrated in Section 4.

LLMs: A small body of work has sought to apply LLMs' demonstrated abilities of modeling complex relationships to tabular data [9]. The landmark work in this area, GReaT [6], details methods

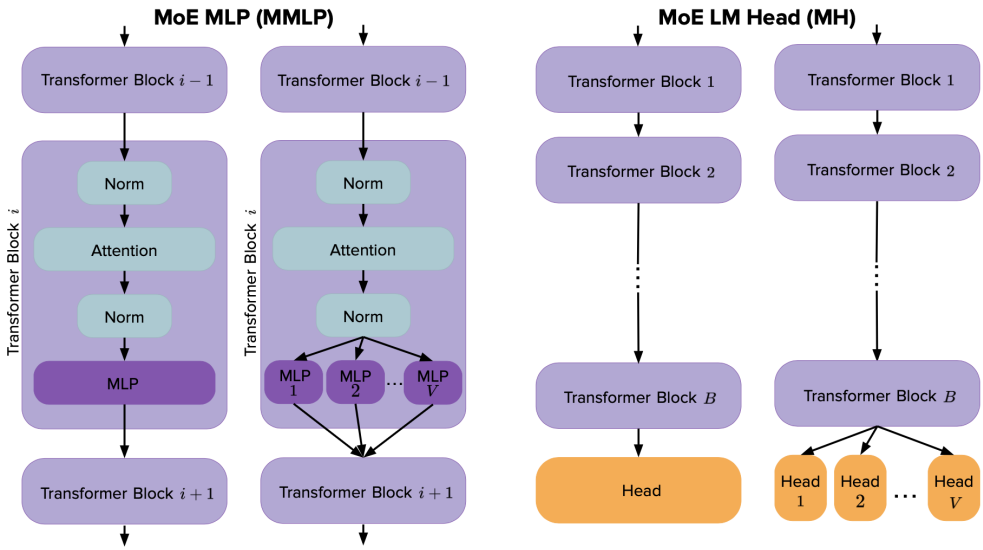

Figure 1: An overview of the Tabby MMLP (left) and MH (right) modifications.

by which to convert tabular data into a sentence format which may be input to LLMs, then proposes a training technique of "shuffling" the order in which columns occur for each row. This randomized ordering allows the LLM to better learn inter-column dependencies.

Two notable works have built off of GReaT to achieve further improved tabular data fidelity: TapTap [34] pretrains full or Distilled GPT-2 [24] on a variety of tabular datasets before fine-tuning on a downstream tabular synthesis task, while Tabula [36] explores methods of preprocessing the training data to be more easily modeled by LLMs. Other LLM-based works have adapted these recent advances to relational tables [28], or used the emergent abilities of very large models such as GPT-4 to generate synthetic data using In-Context Learning in place of fine-tuning [26].

Because many of these prior LLM-based works are training techniques, they may be applied in concert with the Tabby architecture modification. We demonstrate this using GReaT, TapTap and Tabula in Section 4.

MoE Architectures: The key innovation of Tabby is the application of Gated Mixture of Expert (MoE) layers [27, 18] for LLM synthesis of tabular data. MoE layers have enjoyed utility in multitask [17, 12, 13] and multimodal learning [35, 8, 22], by creating sets of model parameters that are dedicated to a specific task or set of similar tasks. We describe our use of MoE layers in Section 3.

3 Method

Tabby is an architecture modification that may be applied to any Transformer-based [31] language model (LM). In Section 3.1, we describe the variations of Tabby. In Section 3.2 , we describe the process for training an arbitrary LM on tabular data, then compare the training process's forward pass and loss calculation of a Tabby model to a non-Tabby model in Section 3.3. Tabby increases the expressivity of LMs, allowing for better modeling of individual columns' distributions and resulting in higher generative fidelity.

3.1 Architecture of Tabby Models

Suppose that a tabular dataset contains V columns and let the order of blocks within an arbitrary Transformer-based LM be represented as $[L_1, L_2, \ldots, L_H]$. We apply the MoE technique by replacing an LM block L_a with a vector $\Lambda_a = [L_{a,1}, L_{a,2}, \ldots L_{a,V}]$ of V blocks. As such, a Tabby model with one MoE block Λ_a is represented as $[L_1, L_2, \ldots, L_{a-1}, [L_{a,1}, L_{a,2}, \ldots L_{a,v}], L_{a+1}, \ldots, L_H]$. The i-th dataset column will be modeled by $L_{a,i}$ within Λ_a.

While this technique may be applied to any set of layers within the model, we focus on the transformer blocks' multi-layer perceptrons (MLPs) and the language modeling (LM) Head. We refer to Tabby models with MoE MLPs as MMLP models, while models with MoE LM Heads are referred to as MH models. Those with both MoE MLPs and MoE LM Heads are MMLP-MH models.

For a visual comparison of Tabby's architecture with that of a general LM, refer to Figure 1.

3.2 Fine-Tuning LLMs on Tabular Data

We now describe the general conventions applied when training or fine-tuning an LM on a tabular dataset. Suppose our training dataset contains N rows and the dataset's column names are denoted by $v_1, v_2, \ldots, v_V$, where the value of the j-th row in the i-th column is denoted as v_i^j. For a given row, the model will train on the columns in order $\ell_1, \ell_2, \ldots, \ell_V$ (for general training we consider this order to be simply $[V]$, while GReaT training allows this order to be arbitrary).

To provide the LM with its expected text modality input, we convert the j-th row as follows, where <EOS> is the end-of-sequence token and <EOC> is a specialized end-of-column token which we introduce to divide the text between columns:

$$\text{``<BOS> } v_{\ell_1} \text{ is } v_{\ell_1}^j \text{ <EOC> } v_{\ell_2} \text{ is } v_{\ell_2}^j \text{ <EOC> } \cdots \; v_{\ell_V} \text{ is } v_{\ell_V}^j \text{ <EOS>"}$$

After converting the tabular dataset in this fashion, an LM is capable of fine-tuning on the dataset in a normal sequence-to-sequence style. The prompt for each row during training is the beginning-of-sequence token <BOS>. During generation, the LM will output text in a similar format to the training data, which can then be parsed into tabular data as desired.

3.3 Tabby Training

At the beginning of fine-tuning a new Tabby model, the weights for each layer in an MoE set Λ_a are initialized to equal the weights of L_a, the corresponding layer in the original LM.

The Tabby training process requires only slight modifications as compared to the training of other LLs on tabular data. Instead of representing each training row as one string, we convert each row into a length-V list of strings as follows:

$$[\text{``}v_{\ell_1} \text{ is } v_{\ell_1}^j \text{ <EOC>", ``}v_{\ell_2} \text{ is } v_{\ell_2}^j \text{ <EOC>", } \cdots \text{, ``}v_{\ell_V} \text{ is } v_{\ell_V}^j \text{ <EOS>"}]$$

Internally, the Tabby model begins by training on column ℓ_1 with prompt <BOS>, attending to tokens 0 through $k-1$ when predicting the k-th token. After computing the loss on column ℓ_1, this column's tokens are appended to the prompt used to train column ℓ_2. As such, the prompt when training on column ℓ_i is

$$\text{``<BOS>}v_{\ell_1} \text{ is } v_{\ell_1}^j \text{ <EOC> } v_{\ell_2} \text{ is } v_{\ell_2}^j \text{ <EOC> } \cdots \; v_{\ell_{i-1}} \text{ is } v_{\ell_{i-1}}^j \text{ <EOS>"}$$

A favorable side-effect of this training style is that we calculate the losses for each column separately, allowing the performances of each column to be monitored separately and compared, as demonstrated in Section 4.3.

4 Evaluation

We conduct a thorough evaluation of Tabby using several standard tabular datasets and metrics as detailed in Section 4.1. In Section 4.2, we compare Tabby to a broad array of prior works, while in Section 4.3, we delve further into the behavior of Tabby and how it adapts to a tabular dataset during training.

4.1 Setup

Calculation of results Throughout the experiments, the reported results for each model and training setup is the average across three training runs. For each of the three trained models, we sample $10,000$ datapoints, compute all evaluation metrics separately for the three resulting synthetic datasets,

Table 1: Summary statistics of datasets. The first three columns list the number of rows in each data split, while the next two columns display the number of categorical versus numerical features, respectively. The rightmost column details whether the dataset is considered a classification or regression task in downstream evaluations.

	N Train	N Validation	N Test	# Cat.	# Num.	Task
Adult [3]	36631	3663	8548	8	6	Classification
Diabetes [15]	576	57	135	0	8	Classification
House [21]	15480	1548	3612	0	8	Regression

then calculate the average metric value across all runs. For LLM approaches, each model is trained for up to 50 epochs, using early stopping when the validation loss (assessed every 5000 steps) fails to improve twice in a row. We perform grid search to select the learning rate with lowest validation loss for each model and training setup. For non-LLM works, we reuse the training setups and hyperparameters selected in their respective publications.

Metrics Multiple standard metrics exist for evaluating the quality of synthetic data, its fidelity to the training data, as well as its ability to preserve the privacy of the examples in the trainset. We focus on machine learning efficacy and detection accuracy.

Machine learning efficacy (MLE) quantifies whether a synthetic dataset would be capable of replacing the original, real data used to train a generative model. Given a real tabular dataset, we form disjoint training and test sets, denoted by R and D respectively. A generative model is trained on R, then generates a synthetic dataset S.

To calculate MLE, a downstream classifier or regressor K_R is trained using R to predict a predetermined label column, using all other columns as features. An additional classifier or regressor K_S is similarly trained on S. Then, the performance of K_S and K_R on the real test dataset D is evaluated: a high-fidelity synthetic dataset S will allow K_S to exhibit similar performance to K_R despite never encountering real datapoints before test-time.

We compare MLE using three downstream models: random forest, decision tree and logistic or linear regression. We compare the accuracy of K_R and K_S on D for classification tasks and the mean squared error on D for regression tasks.

Detection Accuracy: While MLE is useful in assessing whether synthetic data captures patterns in the real dataset, we additionally aim to determine whether the generative model also introduces spurious correlations or other identifying patterns that differentiate the synthetic from the real data. Given the real training dataset R and a synthetic dataset S, we sample the same number of rows from each. Next, we train a random forest classifier C to discriminate between real and synthetic examples. Highest-quality synthetic data will result in 50% discrimination accuracy, indicating that C is unable to distinguish between R and S.

Datasets We evaluate Tabby on three standard tabular datasets, summarized in Table 1.

Adult [3] is a dataset commonly used to benchmark tabular classification algorithms. Each row contains basic information on one American adult, such as their age, years of education and marital status. For each adult, the downstream task is to predict whether their annual income is above or below $\$50,000$. The features are a mix of categorical and numerical columns, with each numerical column taking only integer values.

Another common binary classification dataset, *Diabetes* [15] contains medical information on female hospital patients, including age, number of pregnancies and skin thickness. Downstream models learn to predict whether a given patient suffers from diabetes. Apart from the label, all dataset columns are numerical, with some columns taking only integer values, while others are floats.

House [21] is a standard regression dataset, where each row represents a block of houses in California during the 1990 census. The dataset records the number of households residing in the block, the block's median building age, average number of bedrooms, and other basic information. The dataset's target column is the block's median house value, which is numerical and allows us to assess Tabby's synthetic data in a regression task.

Training techniques As introduced in Sections 2 and 3.3, there are multiple approaches to training LLMs on tabular data which are compatible regardless of whether Tabby is applied.

As a baseline training technique, we implement *Plain* training. While this method has not been described in prior LLM works, it represents a basic method of training the LLM on the columns in the same order as they are found in the training dataset. At sample time, we simply prompt with <BOS> and parse the resulting model output.

Next, we explore the GReaT technique [6] as introduced in Section 2. At each step, the order in which the columns are presented to the model are selected at random, independently of all other steps. During generation, GReaT enforces that the distribution of the label column matches that of the training distribution. Suppose that the label column is v_t and let S_t be the set of all unique values taken by v_t in the trainset (regardless of whether v_t is a categorical or numerical column). GReaT prompts the LLM with <BOS>v_t `is` s_t, where s_t is sampled from S_t with probability proportionate to the frequency of s_t in column v_t.

We additionally explore the use of two more training techniques in conjunction with GReaT. TAPTAP [34] is a checkpoint of Distilled GPT-2 pretrained using GReaT on a large collection of tabular datasets. Meanwhile, Tabula [36] aims to address the challenges encountered by LLMs on categorical columns: Tabula converts each categorical column into an ordinal format by replacing each unique value of the column with a unique integer. In many cases, this technique reduces sequence length, decreasing training and generation time, and helps the LLM during sampling to only generate values that occur within the categorical column's training distribution.

4.2 Experiment 1: Comparison to Related Works

Comparisons We begin our evaluations by comparing Tabby to prior LLM tabular techniques, as well as non-LLM approaches. All LLM approaches use Distilled GPT-2 or the TAPTAP Distilled GPT-2 that is pretrained on tabular data as a base model. In addition to Tabby MMLP, MH and MMLP-MH models, we finetune Non-Tabby (NT) Distilled GPT-2 as a baseline.

To represent non-LLM tabular synthesis techniques, we include Tab-DDPM [16], a diffusion model, as well as CTGAN [32] and TVAE [32], the leading GAN and VAE approaches, respectively.

Results Table 2 summarizes the MLE results for the classification datasets, Adult and Diabetes, by reporting the average accuracy for each model and training technique. Similarly, Table 3 presents the MLE results for the regression dataset, House, using mean squared error (MSE) as the primary metric. The top row of results in both tables corresponds to the "Real" MLE achieved by training the downstream classifier K_R on the original training data R instead of synthetic data. We consider this row as an upper ceiling for synthetic approaches, so any model and training technique that achieves MLE equal to or better than the "Original" row is considered to be a top-performing approach.

We find that Tabby consistently achieves higher MLE than Non-Tabby and non-LLM models. In particular, Plain-trained Tabby models demonstrate the highest MLE across the majority of metrics. While the lower-performing Tabby models experience a boost in performance when trained using GReaT alone or with TapTap and Tabula, the highest LLM performance overall is generally exhibited by Plain-trained Tabby MH models. While Tab-DDPM also achieves high performance on the classification datasets, its scores are often within the margin of error for these datasets. Meanwhile, Plain-trained Non-Tabby and Tabby MH models outperform Tab-DDPM for House's regression task in all three downstream classifiers.

We view similarly-desirable performance in Tabby through the discrimination accuracy metric in Table 4. Where 50% accuracy is the ideal score, implying that the classifier is incapable of distinguishing synthetic from real datapoints, we find that most methods are unreliable: they achieve good metrics on some datasets and struggle for others. Meanwhile, Plain-trained Tabby MH models achieve good scores across all datasets.

4.3 Experiment 2: Tabby behavior

Our final experiment analyzes Tabby's progress of fitting to a tabular dataset during training. For three runs, we train a Tabby MH model on a subset of the House dataset containing 5160 datapoints

Table 2: Machine Learning Efficacy (MLE) for the classification datasets. Reported metric is the accuracy of the downstream classifier, which is a Random Forest Classifier (RF), Decision Tree Classifier (DT) or Logistic Regression model (LR). The row "Original" corresponds to upper-bound performance. In each row, the top result (or any result higher than upper-bound) is presented in bold.

		Adult			Diabetes		
		RF	DT	LR	RF	DT	LR
	Original	84.5	81.4	83.7	73.3	71.1	75.6
	CTGAN	76.2 ± 3.9	68.1 ± 6.4	76.2 ± 3.7	52.1 ± 14.7	50.1 ± 11.2	56.0 ± 14.7
	TVAE	80.4 ± 2.5	77.2 ± 2.9	78.1 ± 3.1	62.2 ± 5.5	61.2 ± 5.1	64.4 ± 6.1
	Tab-DDPM	83.9 ± 0.3	$\mathbf{79.8 \pm 0.0}$	$\mathbf{83.5 \pm 0.1}$	71.9 ± 4.5	69.4 ± 4.7	70.4 ± 7.7
Plain	NT	$\mathbf{84.5 \pm 0.4}$	79.5 ± 1.2	82.6 ± 0.8	$\mathbf{75.3 \pm 1.5}$	$\mathbf{71.4 \pm 5.9}$	$\mathbf{76.5 \pm 1.1}$
	MMLP	77.4 ± 1.4	71.0 ± 2.2	76.8 ± 2.1	$\mathbf{74.8 \pm 3.4}$	67.4 ± 2.0	71.1 ± 3.4
	MH	$\mathbf{84.5 \pm 0.2}$	79.7 ± 0.8	83.1 ± 0.3	$\mathbf{74.3 \pm 0.4}$	70.6 ± 3.0	$\mathbf{77.0 \pm 0.7}$
	MMLP-MH	76.6 ± 0.6	71.3 ± 5.1	74.8 ± 3.1	68.1 ± 0.7	66.2 ± 1.9	70.9 ± 4.8
GReaT	NT	82.9 ± 1.1	77.3 ± 1.9	82.1 ± 0.2	62.2 ± 0.7	54.3 ± 3.4	62.0 ± 0.4
	MMLP	83.2 ± 0.6	76.0 ± 3.5	82.4 ± 0.3	$\mathbf{73.8 \pm 0.9}$	70.1 ± 4.5	74.8 ± 0.7
	MH	83.2 ± 0.1	76.9 ± 0.7	81.5 ± 0.8	63.7 ± 1.3	58.8 ± 2.6	62.5 ± 0.4
	MMLP-MH	83.0 ± 0.2	76.6 ± 1.4	82.0 ± 0.2	69.4 ± 3.7	63.7 ± 7.4	72.6 ± 2.7
TT	NT	82.9 ± 0.8	75.8 ± 0.7	82.2 ± 0.8	71.9 ± 5.9	57.8 ± 5.1	72.3 ± 2.4
	MMLP	83.4 ± 0.3	77.1 ± 1.5	82.1 ± 0.5	69.4 ± 4.3	62.2 ± 6.6	70.6 ± 3.7
	MH	77.1 ± 5.3	73.6 ± 3.4	73.9 ± 7.6	62.5 ± 0.4	49.9 ± 3.8	62.2 ± 0.0
	MMLP-MH	83.0 ± 0.5	77.5 ± 1.3	82.1 ± 0.7	$\mathbf{75.3 \pm 1.5}$	66.7 ± 0.0	74.8 ± 0.0

Table 3: MLE for the regression dataset. Reported metric is the mean squared error (MSE) of the downstream regressor, which is a Random Forest Regressor (RF), Decision Tree Regressor (DT) or Linear Regression model (LR). Unlike the results in Table 2, lower MSE scores are better. As such, the lowest result (or any result lower than the lower-bound presented in the row "Original" is presented in bold.

		House		
		RF	DT	LR
	Original	25.9	39.1	54.6
	CTGAN	143.2 ± 58.7	171.5 ± 66.2	156.7 ± 52.0
	TVAE	133.4 ± 58.8	146.7 ± 55.2	221.9 ± 85.9
	Tab-DDPM	54.8 ± 0.4	73.1 ± 3.2	81.1 ± 0.5
Plain	NT	40.9 ± 14.4	63.0 ± 27.8	567.9 ± 519.8
	MMLP	835414.9 ± 1421184.9	33676.0 ± 29753.3	6908.3 ± 9919.7
	MH	$\mathbf{33.3 \pm 0.2}$	$\mathbf{45.2 \pm 4.7}$	$\mathbf{64.0 \pm 5.2}$
	MMLP-MH	2222.6 ± 1483.1	2594.8 ± 3633.2	230.4 ± 176.9
GReaT	NT	45.1 ± 2.4	52.6 ± 4.1	862.0 ± 362.7
	MMLP	43.0 ± 1.2	51.8 ± 0.8	1032.6 ± 21.1
	MH	43.9 ± 1.0	52.6 ± 1.6	665.6 ± 84.8
	MMLP-MH	42.7 ± 1.9	51.4 ± 0.6	897.8 ± 87.5
TT	NT	43.7 ± 3.0	54.8 ± 6.8	782.5 ± 255.3
	MMLP	44.0 ± 1.1	54.8 ± 1.3	794.9 ± 62.6
	MH	45.5 ± 0.4	55.3 ± 1.1	681.9 ± 86.3
	MMLP-MH	43.2 ± 1.5	52.6 ± 0.1	931.8 ± 120.1

Table 4: Discriminator accuracy for synthetic data generated by each model and training technique. Scores closest to 50% are better, signifying that the discriminator is unable to distinguish between real and synthetic examples.

		Adult	Diabetes	House
	CTGAN	97.8 ± 25.1	88.9 ± 27.3	81.8 ± 18.1
	TVAE	96.4 ± 24.1	95.2 ± 30.8	89.0 ± 21.3
	Tab-DDPM	$\mathbf{50.6 \pm 1.0}$	61.5 ± 0.5	83.2 ± 3.8
Plain	NT	58.2 ± 1.0	56.7 ± 5.7	56.7 ± 5.7
	MMLP	83.6 ± 8.9	68.0 ± 3.1	70.2 ± 4.4
	MH	59.8 ± 0.8	$\mathbf{44.0 \pm 2.1}$	$\mathbf{53.8 \pm 0.6}$
	MMLP-MH	81.4 ± 11.6	68.6 ± 2.0	73.8 ± 1.8
GReaT	NT	70.4 ± 1.0	68.4 ± 4.2	68.4 ± 4.2
	MMLP	68.4 ± 0.9	66.1 ± 0.2	66.1 ± 0.2
	MH	70.1 ± 1.7	79.5 ± 2.1	69.1 ± 0.8
	MMLP-MH	69.6 ± 0.8	77.1 ± 1.6	66.3 ± 0.8
TT	NT	70.5 ± 1.7	68.6 ± 2.0	68.6 ± 2.0
	MMLP	67.8 ± 0.8	68.6 ± 0.7	69.0 ± 0.9
	MH	76.2 ± 6.6	78.1 ± 1.6	69.2 ± 0.7
	MMLP-MH	68.3 ± 0.3	74.2 ± 1.4	65.7 ± 1.0

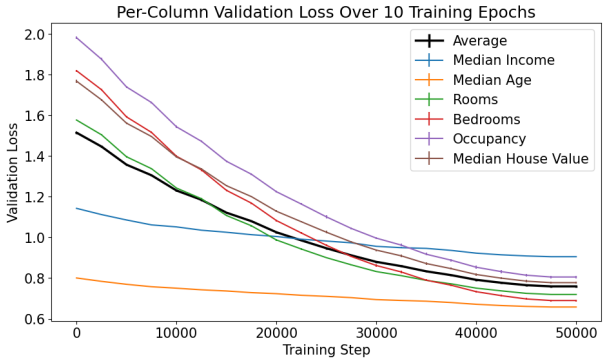

Figure 2: Per-column validation loss across 10 epochs of training Tabby MH Distilled GPT-2 on a subset of House, with average validation loss (black line). While the Occupancy column initially displays the highest loss, Median Income improves little throughout training and becomes the highest-loss column by step 32000.

and six columns. We log the individual columns' losses on the evaluation dataset every 2500 steps while training for 10 epochs, then display the results in Figure 2.

We observe that the Occupancy column is the largest contributor to the model's loss until step 32000. While the Median Income column's loss is initially the second-lowest of all columns, it improves little throughout the training process and exhibits the highest loss of all columns at the end of training. Additionally, we view that convergence occurs around step 40000.

These insights are useful in cases where the model struggles to learn some columns more than others. A machine learning practitioner could use this information to consider implementing better preprocessing for a difficult column, or gathering more datapoints that demonstrate a difficult column's distribution. Additionally, the ability to track each column's loss individually and to determine that the losses are roughly balanced across columns, rather than very low in some columns and very high in others, may improve trust in the model– we can understand that there is a low, aleatoric error in each column as opposed to a sizeable epistemic error in a few columns.

5 Conclusion

We introduce Tabby, a Mixture-of-Experts-based architecture modification that allows LLMs to be better suited to tabular datasets. Tabby reaches parity with non-synthetic data in two out of three evaluated datasets, according to machine learning efficacy with a Decision Tree Classifier. Due to the promising performance of Tabby, we hope to spur future work in this area and further experimentation with architecture modifications that allow LLMs to better fit to tabular data. The concepts behind Tabby may find utility in similar modalities as well, such as geospatial, nested-list, or other highly-structured data.

Acknowledgements

We are grateful for the support of the US Army Research Laboratory (ARL) STRONG program and the Wisconsin Alumni Research Foundation (WARF).

References

[1] DALL·E 2. URL `https://openai.com/index/dall-e-2/`.

[2] L. Aviñó, M. Ruffini, and R. Gavaldà. Generating synthetic but plausible healthcare record datasets. *arXiv preprint arXiv:1807.01514*, 2018.

[3] B. Becker and R. Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[4] F. Benali, D. Bodénès, N. Labroche, and C. de Runz. Mtcopula: Synthetic complex data generation using copula. In *23rd International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP)*, pages 51–60, 2021.

[5] G. Biau and E. Scornet. A random forest guided tour. *Test*, 25:197–227, 2016.

[6] V. Borisov, K. Sessler, T. Leemann, M. Pawelczyk, and G. Kasneci. Language Models are Realistic Tabular Data Generators. Sept. 2022. URL `https://openreview.net/forum?id=cEygmQNOeI`.

[7] D. M. Bourg. *Excel Scientific and Engineering Cookbook: Adding Excel to Your Analysis Arsenal*. "O'Reilly Media, Inc.", Jan. 2006. ISBN 978-0-596-55317-3. Google-Books-ID: vQaGBMGaWI4C.

[8] B. Cao, Y. Sun, P. Zhu, and Q. Hu. Multi-modal gated mixture of local-to-global experts for dynamic image fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23555–23564, 2023.

[9] X. Fang, W. Xu, F. A. Tan, J. Zhang, Z. Hu, Y. Qi, S. Nickleach, D. Socolinsky, S. Sengamedu, and C. Faloutsos. Large Language Models(LLMs) on Tabular Data: Prediction, Generation, and Understanding – A Survey, Feb. 2024. URL `http://arxiv.org/abs/2402.17944`. arXiv:2402.17944 [cs].

[10] E. W. Frees and E. A. Valdez. Understanding relationships using copulas. *North American actuarial journal*, 2(1):1–25, 1998.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. ISSN 0001-0782. doi: 10.1145/3422622. URL `https://dl.acm.org/doi/10.1145/3422622`.

[12] S. Gupta, S. Mukherjee, K. Subudhi, E. Gonzalez, D. Jose, A. H. Awadallah, and J. Gao. Sparsely activated mixture-of-experts are robust multi-task learners. *arXiv preprint arXiv:2204.07689*, 2022.

[13] H. Hazimeh, Z. Zhao, A. Chowdhery, M. Sathiamoorthy, Y. Chen, R. Mazumder, L. Hong, and E. Chi. Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. *Advances in Neural Information Processing Systems*, 34:29335–29347, 2021.

[14] T. Janke, M. Ghanmi, and F. Steinke. Implicit generative copulas. *Advances in Neural Information Processing Systems*, 34:26028–26039, 2021.

[15] M. Kahn. Diabetes. UCI Machine Learning Repository. URL https://www.openml.org/search?type=data&sort=runs&id=37&status=active. DOI: https://doi.org/10.24432/C5T59G.

[16] A. Kotelnikov, D. Baranchuk, I. Rubachev, and A. Babenko. TabDDPM: Modelling Tabular Data with Diffusion Models, Sept. 2022. URL http://arxiv.org/abs/2209.15421. arXiv:2209.15421 [cs].

[17] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1930–1939, 2018.

[18] S. Masoudnia and R. Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42:275–293, 2014.

[19] W. McKinney et al. pandas: a foundational python library for data analysis and statistics. *Python for high performance and scientific computing*, 14(9):1–9, 2011.

[20] OpenAI. GPT-4 Technical Report, Mar. 2023. URL http://arxiv.org/abs/2303.08774. arXiv:2303.08774 [cs].

[21] R. K. Pace and R. Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3): 291–297, 1997.

[22] D. K. Park, S. Yoo, H. Bahng, J. Choo, and N. Park. Megan: Mixture of experts of generative adversarial networks for multimodal image generation. *arXiv preprint arXiv:1805.02481*, 2018.

[23] M. F. D. R., S. Groen, F. Panse, and W. Wingerath. Navigating Tabular Data Synthesis Research: Understanding User Needs and Tool Capabilities, May 2024. URL http://arxiv.org/abs/2405.20959. arXiv:2405.20959 [cs].

[24] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[25] J. P. Reiter. Using cart to generate partially synthetic public use microdata. *Journal of Official Statistics - Stockholm*, 21(3):441, 2005.

[26] N. Seedat, N. Huynh, B. van Breugel, and M. van der Schaar. Curated LLM: Synergy of LLMs and Data Curation for tabular augmentation in ultra low-data regimes, Feb. 2024. URL http://arxiv.org/abs/2312.12112. arXiv:2312.12112 [cs].

[27] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[28] A. V. Solatorio and O. Dupriez. REaLTabFormer: Generating Realistic Relational and Tabular Data using Transformers, Feb. 2023. URL http://arxiv.org/abs/2302.02041. arXiv:2302.02041 [cs].

[29] Y.-y. SONG and Y. LU. Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*, 27(2):130–135, Apr. 2015. ISSN 1002-0829. doi: 10.11919/j.issn.1002-0829.215044. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4466856/.

[30] B. van Breugel and M. van der Schaar. Why Tabular Foundation Models Should Be a Research Priority, May 2024. URL https://arxiv.org/abs/2405.01147v2.

[31] A. Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[32] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni. Modeling Tabular data using Conditional GAN, Oct. 2019. URL http://arxiv.org/abs/1907.00503. arXiv:1907.00503 [cs, stat].

[33] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):1–41, 2017.

[34] T. Zhang, S. Wang, S. Yan, J. Li, and Q. Liu. Generative Table Pre-training Empowers Models for Tabular Prediction, May 2023. URL http://arxiv.org/abs/2305.09696. arXiv:2305.09696 [cs].

[35] X. Zhao, M. Wang, Y. Tan, and X. Wang. Tgmoe: A text guided mixture-of-experts model for multimodal sentiment analysis. *International Journal of Advanced Computer Science & Applications*, 15(8), 2024.

[36] Z. Zhao, R. Birke, and L. Chen. TabuLa: Harnessing Language Models for Tabular Data Synthesis, Oct. 2023. URL `http://arxiv.org/abs/2310.12746`. arXiv:2310.12746 [cs] version: 1.