Tabby: A Language Model Architecture for Tabular and Structured Data Synthesis

Anonymous authors
Paper under double-blind review

Abstract

Large language models (LLMs) have greatly improved the quality of synthetic text data. We aim to extend these advances to *tabular* data with **Tabby**, a simple but powerful post-training modification to the standard Transformer language model architecture, enabling its use for tabular dataset synthesis. Tabby represents differences across columns using Gated Mixture-of-Experts, with column-specific sets of parameters. Empirically, Tabby results in data quality near or equal to that of real data. Pairing Tabby with **Plain**, our novel tabular training technique, we observe up to a 7% improvement in quality (measured by MLE) over previous methods. Additionally, our approach is *more flexible* than prior strategies and extends beyond tables, to more general structured data. In a structured JSON setting, Tabby outperforms all other methods by 2-3 points and is the only approach with MLE equal to the upper bound of non-synthetic data.

1 Introduction

Modern life is built on tabular data: airplane black boxes, website analytics and hospital patient records are just a few examples of this versatile modality. Despite widespread use of tables and repeated calls for improved table modeling approaches (Fang et al., 2024; Davila et al., 2024), the tabular modality has received less attention in recent deep learning research than images or text (van Breugel & van der Schaar, 2024).

Progress towards realistic tabular data synthesis has encountered several key challenges. *First*, table columns often exhibit complex interdependencies. *Second*, many tabular datasets mix multiple datatypes. A single table might contain free-text fields, numerical features, and even nested JSON or dictionary columns. *Third*, although the order of tokens within one column is important, the order of columns with respect to each other is usually not meaningful and is a potential source of spurious correlations during training. How best to design model architectures and training techniques that address these issues remains an open question.

There have been notable efforts to adapt several model architectures to tabular data, recently focusing on generative adversarial networks (GANs) (Xu et al., 2019), LLMs (Borisov et al., 2022) and diffusion models (Kotelnikov et al., 2022). However, because these architectures were each designed with images or text in mind, significant preprocessing must be made to tabular datasets in order to allow their use.

For these reasons, works including van Breugel & van der Schaar (2024) have called for the development

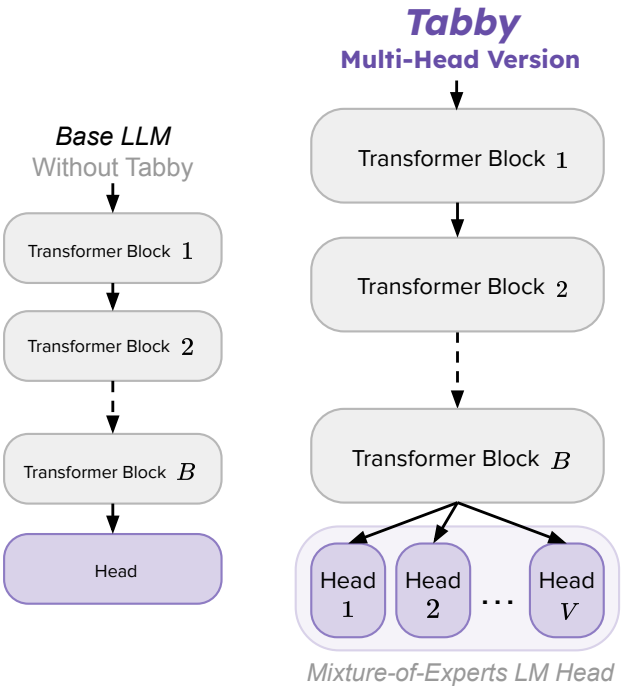

Figure 1: Tabby Multi-Head modifications (right) compared to an original, Non-Tabby LLM on left.

of pretrained *Large Tabular Models (LTMs)* to fill a similar role to text and image foundation models, such as GPT (Achiam et al., 2023) or Stable Diffusion (Blattmann et al., 2023). Unfortunately, the creation of an LTM requires (1) large, diverse tabular pretraining sets which have not yet been curated, (2) a specialized tabular model architecture which has yet to be designed, and (3) substantial compute resources for pretraining. These challenges are even more pronounced in the development of foundation models for structured *non-tabular* modalities, such as JSON and geospatial data.

This work takes an initial step towards LTMs with ***Tabby***, ***a post-training modification to the standard transformer LLM architecture to enable tabular and other structured data synthesis***. After training on text data—but before finetuning on structured data—Tabby replaces select LLM blocks with *Mixture-of-Experts (MoE) layers* (Shazeer et al., 2017), allowing each tabular column, JSON attribute or other structured feature to be modeled by a dedicated set of parameters in the LLM. The greater expressivity afforded by this change results in higher-fidelity synthetic data. Fine-tuning with our novel *Plain* technique results in still higher performance. We show that even small Tabby models are capable of outstripping large non-Tabby LMs with parameter counts orders of magnitude greater.

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 text pretraining, avoiding the logistical challenges of training a LTM fully from scratch. We find that, according to standard metrics, **Tabby produces synthetic data near- or at-parity with real data on 4 out of 6 table datasets**. Additionally, Tabby is not limited to tables and can be easily extended to other structured data. We validate this by **synthesizing nested JSON data at-parity with real data** as well. Our contributions are:

- We introduce *Tabby*, a simple architecture modification that allows transformer-based LLMs to synthesize more realistic tabular and structured non-tabular data.
- We demonstrate that Tabby produces higher-quality synthetic data for 4 out of 6 popular tabular datasets, is compatible with more tabular data than prior high-quality synthesis methods, and can be extended beyond tables to a broader class of *structured data modalities*.
- We introduce our novel tabular LLM training method, *Plain*. Named after its surprising simplicity compared to prior table training approaches for LLMs, Plain increases data quality in 5/6 datasets when used together with or independently of Tabby.

2 Tabby Architecture & Plain Train Method

We now formally introduce our two novel contributions for LLM table synthesis: *Tabby* is an architecture modification that may be applied to any transformer-based language model (LM) (Vaswani, 2017), and *Plain* is a training technique for training any (Tabby or non-Tabby) LM on tabular data. Tabby and Plain are especially powerful together: Tabby increases model expressivity in a way that is specially-suited to tabular data, while Plain allows the model to more effectively fit to the key features of a tabular dataset during training, yielding more realistic data synthesis.

In Section 2.1, we describe Tabby for tabular or other structured data. In Section 2.2 , we outline the process for training an LM on tabular data using our Plain training technique. Then, in Section 2.3, we provide additional insight into how Tabby models are trained (using Plain, or pre-existing LLM table training techniques such as GReaT (Borisov et al., 2022)) by comparing the training process's forward pass and loss calculation for a Tabby model with a non-Tabby model.

2.1 Architecture of Tabby Models

The intuition behind the Tabby modification is simple: we want to allow the model to learn individual columns as distinct–but interdependent–tasks. *The right side of Figure 1 depicts our best-performing Tabby model for tabular data*, where Tabby modifies only the language modeling head.

To provide a general definition of a Tabby model, consider a tabular dataset with V columns. 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. Thus, a Tabby

model with one MoE block Λ_a is represented

$$[L_1, L_2, \ldots, L_{a-1}, [L_{a,1}, L_{a,2}, \ldots, L_{a,V}], L_{a+1}, \ldots, L_H].$$

The dataset's i-th column is modeled by $L_{a,i}$ within Λ_a.

This technique may be applied to any set of layers within the model. While we focus on the language modeling (LM) head [1] in Figure 1 and Section 3 evaluations, we also conduct experiments applying Tabby to the transformer multi-layer perceptrons and attention blocks in Appendix E. We refer to Tabby models with MoE LM Heads as *Tabby Multi-Head (MH)* models.

2.2 The Plain Technique for Fine-Tuning LLMs on Tabular Data

Suppose our trainset contains N rows and column names denoted by $v_1, v_2, \ldots, v_V$, such that v_i^j represents the value of the j-th row in the i-th column. 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:

"<BOS> v_1 is v_1^j <EOC> v_2 is v_2^j <EOC> $\cdots$ v_V is v_V^j <EOS>"

Converting the tabular dataset in this fashion allows an LM to fine-tune on the dataset in a normal sequence-to-sequence style. Because Plain encodes data the same way as prior LLM table training techniques, GReaT and Tabula (Borisov et al., 2022; Zhao et al., 2023), Plain does not require more FLOPs than prior methods.

During inference, the prompt for each row 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. We note that *the simplicity of Plain is particularly impressive given its favorable performance compared to prior LLM table training methods, as we show in Section 3.*

2.3 Tabby Training

Now that we have introduced Tabby and the Plain training method, we are able to provide further insight into aspects of the training process unique to Tabby. Suppose that we construct a Tabby model from a base LM by replacing one of its blocks L_a with an MoE set Λ_a. At the beginning of fine-tuning the Tabby model, weights for each block in Λ_a are initialized to equal the weights of L_a.

The Tabby training process requires only slight modifications compared to other LMs for tabular data. Instead of representing each training row as one string, we convert each row into a list of V strings:

["v_1 is v_1^j <EOC>", "v_2 is v_2^j <EOC>", $\cdots$, "v_V is v_V^j <EOS>"]

Internally, the Tabby model begins by training on column 1 with prompt `<BOS>`, attending to tokens 0 through $k-1$ when predicting the k-th token. After computing the loss on column 1, this column's tokens are appended to the prompt used to train column 2. The prompt when training on column i is

"<BOS>v_1 is v_1^j <EOC> v_2 is v_2^j <EOC> $\cdots$ v_{i-1} is v_{i-1}^j <EOS>"

Because we calculate losses for each column separately, we are able to monitor the performance of each column individually during training. This favorable side-effect is demonstrated in Section 3.4.

2.4 Extensions

We address two additional aspects of Tabby and Plain: (1) generalizations that go beyond tabular data and (2) optimizations for datasets with large numbers of columns.

Synthesis for general structured modalities: The flexibility in Tabby MoE layer design enables extensions to a variety of structured datatypes, such as *hierarchical* data. For example, we create a model for nested JSON data by applying Tabby recursively in Figure 5. The JSON structure is preserved inherently in the model, so that Plain's method of representing data features does not need to be modified to indicate nested features. As we show in Section 3.4, the combination of Plain and Tabby is the only synthesis approach to reach equal performance to real, non-synthetic data.

[1] "LM head" refers here to the language model output layer, distinct from attention heads in the MLP blocks.

High-dimensional data: Because Tabby MoEs contain one block per dataset column, model parameter count is proportionate to the number of data features. In practice, however, techniques such as parameter sharing (Ravanbakhsh et al., 2017) can drastically reduce the number of parameters to represent a Tabby model. Additionally, Tabby may be implemented so that only one block in the MOE layer is in memory at a time, resulting in memory requirements identical to a non-Tabby model.

3 Experimental Results

Our evaluations seek to assess the following claims:

Claim 1: Plain-trained Tabby models generate higher-quality tabular data than prior approaches.
Claim 2: The Tabby architecture modification allows smaller LLMs to achieve similar or better synthetic data fidelity than LLMs with higher parameter counts.
Claim 3: Tabby architecture modifications may also be applied to other structured data beyond tabular data, resulting in higher-quality synthetic data for these modalities as well.
Claim 4: Tabby's loss formulation allows for convenient tracking of per-column performance at training time, leading to better understanding of model behavior.

After providing key evaluation setup details in Section 3.0, we compare Tabby to a broad array of prior works on diverse tabular datasets in Section 3.1 to evaluate Claim 1. As Tabby may be applied to any transformer-based LM, we explore Claim 2 for LMs of varying sizes in Section 3.2. To demonstrate Claim 3, we apply Tabby to a nested (JSON) dataset in Section 3.3. Lastly, in Section 3.4, we investigate how Tabby adapts to individual columns within a dataset during finetuning as a demonstration of Claim 4.

3.0 Setup

We detail here our experiments' essential information, including baselines, evaluation datasets and metrics. Additional details are located in Appendix D.

Baselines and Comparisons: We evaluate a variety of recent tabular synthesis techniques.

LLM Approaches: Prior LLM table synthesis approaches are limited to the development of training techniques. We compare Tabby and Non-Tabby LLMs trained under three different paradigms:

1. Our lightweight and simple ***Plain*** training paradigm, detailed in Section 2.2.
2. **GReaT** (Borisov et al., 2022), which encodes tabular data similarly to Plain, but permutes the orders in which columns are presented in training and imposes some conditional restrictions at sample time. For more details, see Section 4.
3. GReaT combined with TapTap (Zhang et al., 2023) and Tabula (Zhao et al., 2023). We abbreviate this combination as **GTT**. TapTap pretrains the LLM on tabular data, while Tabula encodes each categorical column into an ordinal format by replacing each unique column value with an integer.

To align with the prior works (Borisov et al., 2022; Zhang et al., 2023; Zhao et al., 2023), LLM methods use Distilled-GPT2 (DGPT2) (Radford et al., 2019) as a base model unless otherwise stated.

Non-LLM Approaches: To represent non-LLM tabular synthesis techniques, we include CTGAN (Xu et al., 2019) and TVAE (Xu et al., 2019), the leading GAN and VAE approaches, as well as diffusion models Tab-DDPM (Kotelnikov et al., 2022) and Forest Diffusion (Jolicoeur-Martineau et al., 2024). Although diffusion models are a SOTA approach to achieving high MLE scores, they do so under strong assumptions and are incompatible with many tabular datasets—see Figure 4 and Section 4.

Additional details on how models are trained and sampled are available in Appendix D.

Datasets: We evaluate Tabby on six common tabular datasets, which are summarized in Table 1. The majority of these datasets are standard for the evaluation of tabular synthesis techniques, allowing for easy comparison with prior approaches. For more information on these datasets, see Appendix B.

Metrics: We focus on ***machine learning efficacy (MLE)*** **(Dankar et al., 2022), the standard metric for quantitative evaluation of synthetic tabular data**. In brief, MLE compares the performance of downstream classifiers that were trained using either real or synthetic data.

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 (C) or regression (R) task in downstream evaluations.

	N Train	N Validation	N Test	# Cat.	# Num.	Task
Diabetes (Kahn, 1994)	576	57	135	0	8	C
Travel (Tejashvi, 2023)	715	71	168	4	2	C
Adult (Becker & Kohavi, 1996)	36631	3663	8548	8	6	C
Abalone (Nash et al., 1994)	3132	313	732	1	7	R
Rainfall (Zaman, 2018)	12566	1256	2933	2	1	R
House (Pace & Barry, 1997)	15480	1548	3612	0	8	R

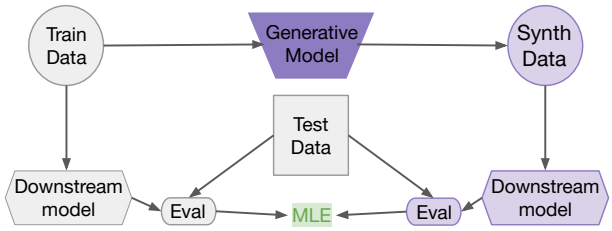

Figure 2: Process for calculating our primary metric, Machine Learning Efficacy (MLE). We train a generative model, which produces a synthetic dataset. Two downstream classifiers are trained: one on the generative model's training data and the other on the synthetic data. Each downstream model is evaluated on real test data. MLE is the difference in downstream models' test-time performance. Higher scores indicate better-quality synthetic data.

Our MLE results in the following sections are interpreted as follows: the downstream classifier that is trained using non-synthetic, real data is considered the upper bound and **any MLE score higher than this "Non-Synthetic" classifier's score is considered the best. If no score surpasses the "Non-Synthetic" score, then any highest score is considered the best**. Figure 2 summarizes the MLE calculation process. In Appendix E, we provide a formal definition of MLE, as well as *several more metrics* including Shape, Trend (Shi et al., 2025) and Distance to Closest Record (DCR). These metrics additionally compare the synthetics' abilities to preserve trainset distributions without memorizing the trainset.

Aggregation of results: Our evaluation involves a comparison between 9 synthesis methods (including Tabby) across 6 datasets. So, while we do report final scores on each task individually, we would also like to understand which method *performs the best across all of the tasks in our evaluation.*

To do so, we aggregate MLE scores using *performance profile* curves (Dolan & Moré, 2002), a robust way to visually compare scores across noisy evaluations in a large number of environments. Performance profiles improve over simpler aggregation techniques, such as averaging scores or computing the average rank of methods across tasks. Specifically, performance profiles are useful when scores for different tasks might be on different scales (which can be an issue with averaging scores), and can take into account methods that are extremely close to the best-performing method on a task without dropping them a full rank (which can be problematic when averaging ranks).

To summarize these curves, we also calculate the *area under the performance profile (AUP) scores* (Roberts et al., 2022), which serve as a final ranking of methods. In short, the performance of a synthesis method across *all* six datasets may be represented as just one performance profile curve. Methods with better performance will have higher curves and, therefore, higher AUP scores. As such, **the method with highest AUP score is considered the best overall method**. Details on performance profiles are in Appendix D.

Table 2: Machine Learning Efficacy (MLE, ↑). The "Non-Synthetic" row is upper-bound performance given by real, non-synthetic data . Top results (or any higher than upper-bound) are **bolded**. The number of datasets that a model achieves top performance on are counted in the "# Best" column. An asterisk indicates that at least one of three runs did not produce valid samples. Tabby models are presented in *italic*. The best-performing Tabby model, *Plain Tabby MH DGPT2* is presented in purple and achieves best performance on 4/6 datasets. Terminology glossary in Appendix A.

	Diabetes	**Travel**	**Adult**	**Abalone**	**Rainfall**	**House**	**# Best**
Non-Synthetic	**0.73**	**0.87**	**0.85**	**0.45**	**0.54**	**0.61**	
CTGAN	0.39 ± 0.00	0.43 ± 0.33	0.76 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0
TVAE	0.62 ± 0.00	0.81 ± 0.00	0.81 ± 0.01	0.07 ± 0.03	0.00 ± 0.00	0.05 ± 0.09	0
Forest Diffusion	$\mathbf{0.76 \pm 0.00}$	0.86 ± 0.01	0.81 ± 0.00	0.35 ± 0.00	0.45 ± 0.02	0.56 ± 0.01	1
Tab-DDPM	$\mathbf{0.75 \pm 0.02}$	$\mathbf{0.87 \pm 0.01}$	0.84 ± 0.00	0.41 ± 0.01	$\mathbf{0.54 \pm 0.01}$	0.43 ± 0.01	3
Plain Base	$\mathbf{0.75 \pm 0.02}$	0.86 ± 0.01	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.44 \pm 0.01}$	0.52 ± 0.03	0.55 ± 0.08	3
Plain Tabby MH	$\mathbf{0.74 \pm 0.00}$	$\mathbf{0.88 \pm 0.01}$	$\mathbf{0.85 \pm 0.00}$	0.43 ± 0.01	0.49 ± 0.00	$\mathbf{0.60 \pm 0.00}$	**4**
GReaT Base	0.62 ± 0.01	0.85 ± 0.02	0.83 ± 0.01	0.41 ± 0.01	N/A*	0.56 ± 0.01	0
GReaT *Tabby MH*	0.64 ± 0.01	0.86 ± 0.01	0.83 ± 0.00	0.40 ± 0.01	0.00 ± 0.00*	0.56 ± 0.01	0
GTT Base DGPT2	0.72 ± 0.06	$\mathbf{0.87 \pm 0.02}$	0.83 ± 0.01	0.40 ± 0.01	0.05 ± 0.01	0.55 ± 0.02	1
GTT *Tabby MH*	0.62 ± 0.00	0.85 ± 0.01	0.76 ± 0.07	0.37 ± 0.02	0.26 ± 0.37	0.55 ± 0.00	0

3.1 Tabby versus Baseline Synthesis Methods

We begin by validating our first claim.
Claim 1: Plain-trained Tabby models generate higher-quality tabular data than prior approaches.

Setup: Table 2 lists MLE for each dataset. For classification datasets (Diabetes, Travel, Adult), the reported metric is the accuracy of the downstream random forest classifier, while for regression datasets (Abalone, Rainfall, House), we report the coefficient of determination R^2 of the downstream random forest regressor.

The "Non-Synthetic" row corresponds to the performance achieved by training the downstream classifier or regressor on real instead of synthetic data. We consider this row to be a performance ceiling for synthetic approaches. Any model and training technique that achieves MLE equal to or better than "Non-Synthetic" is considered to be a top-performing approach and is presented in bold.

Results: We find that ***Plain-trained Tabby models achieve the highest MLE in 4/6 datasets***. Further, Tabby reaches upper-bound performance on Diabetes, Travel and Adult, indicating that *Tabby synthetic data is a capable stand-in for real data* in similar scenarios for these datasets.

We also find that ***Plain is the best-performing technique for training tabular LLMs in almost all cases***: for all six datasets, the highest-scoring LLM is trained using Plain. *Plain-trained Tabby MH models demonstrate the highest MLE among all LLM architectures and training styles.*

For the Rainfall dataset, pre-existing LLM tabular training techinques introduce undesirable effects. Entries marked by an asterisk (*) for this dataset indicate that at least one of three runs were unsuccessful in synthesizing *any* valid samples. Particularly, the Non-Tabby GReaT model is unable to produce valid samples in any of the runs. Meanwhile, each Plain-trained model is successfully sampled and outperforms all GReaT or GTT-trained models in all three runs, indicating that **Plain-trained Tabby models are capable of modeling complexities within the Rainfall dataset that pre-existing LLM-based tabular synthesis works are unable to capture**.

Performance Profile Analysis: The performance profile curves in Figure 3 support our findings. In particular, Plain-trained Tabby MH achieves the highest AUP score. This indicates that **Plain-trained Tabby MH performs the best among all methods** when comparing across all datasets.

Further, we see that the top two synthesis approaches are the two Plain-trained models, which surpass the prior SOTA method of Tab-DDPM. ***Given that these models rely on fewer assumptions than***

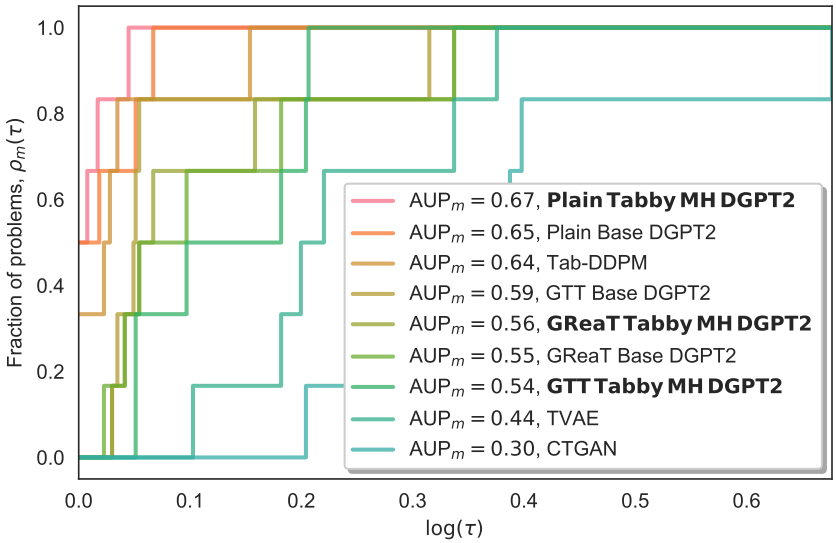

Figure 3: Performance profile curves and AUP scores across computed using the MLE scores on our evaluation tasks. The top performing method is *Tabby MH DGPT2* with plain training.

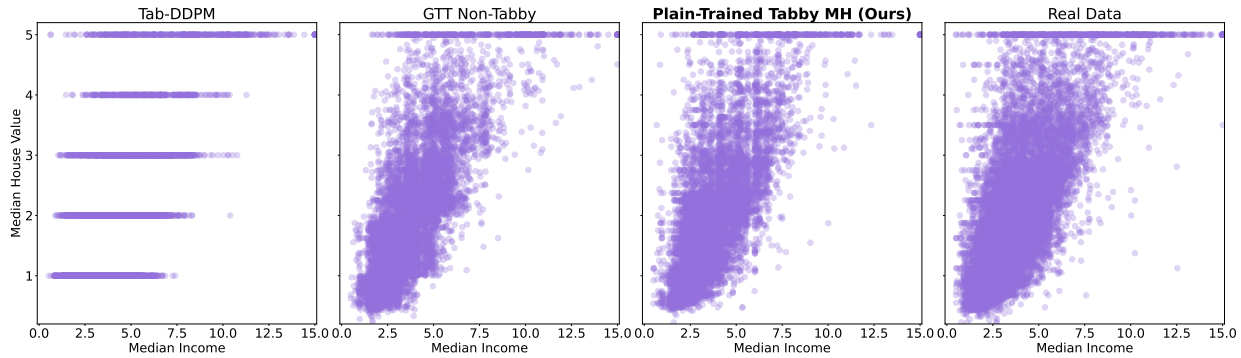

Figure 4: The House dataset's target Median House Value column as a function of its most-predictive feature, Median Income. Left to right: synthetic data from Tab-DDPM, the prior best LLM-based method and Plain Tabby MH, followed by the original data distribution.

Tab-DDPM, and are simpler to train than the GTT or GReaT LLMs, we find that both Tabby MH and Plain training are powerful advancements for the task of tabular data synthesis.

Comparing Multivariate Modeling Capabilities: We further compare the multivariate modeling capabilities of Tab-DDPM, Plain-trained Tabby MH and the prior top-performing LLM-based approach of GReaT-trained Non-Tabby with TapTap and Tabula in Figure 4. We plot the House dataset's target column (Median House Value) as a function of its most predictive feature in the dataset (Median Income), for (left to right) real data, Plain Tabby MH, non-Tabby GTT and Tab-DDPM.

Tab-DDPM's plot (left) differs the most from the real data (right) because the Tab-DDPM model only supports integer-valued regression targets. Accordingly, both LLM-based approaches more accurately capture the target column's distribution than Tab-DDPM.

Meanwhile, GReaT sampling (center left) constrains that the target column distribution in the training dataset is replicated in synthetic data, by prompting the model with target values selected randomly based on their frequency in the training data. Accordingly, GReaT models will not generate target values outside those in the training data, which can be undesirable for datasets with few rows or limited target column coverage. In contrast, Plain training (center right) allows the model to generate previously unseen target

Table 3: MLE for Base and MH versions of 7 LLMs with varying parameter counts, for the Travel dataset. Results higher than Non-Synthetic are presented in **bold**. Tabby improves or maintains upper-bound MLE for 6/7 models. Results for Diabetes and House (and additional metrics) in Appendix E.3.

	MLE (↑)	**Params**
Non-Synthetic (Upper Bound)	**0.87**	
Base Pythia 14M	0.86 ± 0.01	14M
Tabby MH Pythia 14M	0.82 ± 0.02	53M
Base Distilled-GPT2	$\mathbf{0.88 \pm 0.00}$	82M
Tabby MH Distilled-GPT2	$\mathbf{0.89 \pm 0.02}$	310M
Base GPT2	$\mathbf{0.89 \pm 0.01}$	120M
Tabby MH GPT2	$\mathbf{0.87 \pm 0.01}$	360M
Base Pythia 160M	$\mathbf{0.87 \pm 0.01}$	160M
Tabby MH Pythia 160M	0.86 ± 0.00	390M
Base Pythia 410M	0.86 ± 0.02	410M
Tabby MH Pythia 410M	$\mathbf{0.88 \pm 0.03}$	710M
Base Llama 3.2 1B	0.82 ± 0.01	1.2B
Tabby MH Llama 3.2 1B	0.84 ± 0.02	2.8B
Base Llama 3.1 8B	0.84 ± 0.01	8.0B
Tabby MH Llama 3.1 8B	0.86 ± 0.03	11B

values. The improved modeling capacity of Tabby over the Non-Tabby model allows Plain's sampling approach to effectively capture the overall target column distribution.

3.2 Investigating the Choice of Base Model

We now turn to our second claim.
Claim 2: The Tabby architecture modification allows smaller LLMs to achieve similar or better synthetic data fidelity than LLMs with higher parameter counts.

Comparisons: We compare synthesis quality across LLMs of varying sizes. We consider 7 LLMs, listed in Table 3, evaluating Non-Tabby and MH versions of each. Each model is Plain-trained under conditions provided in Section 3.0, then sampled 500 times. Results are averaged across two runs. Llama models use LoRA (Hu et al., 2021) on all linear transformer layers, with the LM head fully fine-tuned.

Results: Table 3 and Figure 7 display results for the Travel dataset, with results for Diabetes and House (plus additional metrics and results for GReaT training) in Appendix E.3.

We find that **Tabby improves MLE or maintains upper-bound MLE for** 6/7 **models**, without necessarily increasing the cost of inference (Section 2.4). Although higher-parameter models are generally correlated with greater generative abilities, Figure 7 demonstrates that this is not always the case: Interestingly, we find that the Llama models (1.2B and 8B parameters each), have lower average MLE than smaller models. Tabby offers favorable performance improvements relative to the scaling curve and **allows even small models to better outperform large, resource-intensive models**.

3.3 Extending Tabby Beyond Tabular Data to General Structured Modalities

While tabular data is frequently overlooked in contemporary machine learning research, related structured modalities such as nested data receive even less attention. While GReaT, TapTap, Tabula, CTGAN and TVAE are focused solely on tabular data and do not clearly extend beyond tables, we demonstrate that Tabby can be generalized to address our third claim.
Claim 3: Tabby architecture modifications may also be applied to other structured data beyond tabular data, resulting in higher-quality synthetic data for these modalities as well.

Comparisons: We plain-train non-Tabby and Tabby MH models on a JSON dataset of patients being evaluated for Glaucoma (Manoj, 2024). Each datapoint has 10 features, organized in 3 groups: a group

Table 4: MLE and Discrimination scores for Plain-trained Base and MH models on a dataset of JSON records. Each record contains diagnostic information of a glaucoma sufferer or a healthy patient.

	MLE (↑)	**Discrim.** (↓)
Non-Synthetic (Upper Bound)	**0.97**	
CTGAN	0.52	0.46
Forest Diffusion	0.95	0.31
Tab-DDPM	0.94	0.45
Base DGPT2	0.93	0.06
Tabby MH DGPT2	**0.97**	**0.01**

of 7 columns representing qualitative aspects of the optic nerve, a group of 2 columns corresponding to measurements between the optic nerve and eye, and a standalone feature for the diagnosis (examples in Box D). The binary classification target is inside the first group and assesses whether the optic nerve is thinning. As with tabular datasets in Section 3.1 and 3.2, we train downstream classifiers to predict the target variable and then present the resulting MLE.

We also consider the *discrimination* metric: Given equal numbers of real and synthetic samples, we measure the accuracy of a discrimination classifier that is trained to distinguish real versus synthetic datapoints. Because 50% accuracy would indicate that the classifier is fully unable to distinguish real from synthetic, we report the accuracy's distance from 50% in Table 4 so that *lower scores indicate higher-quality synthesis.*

Results: Table 4 demonstrates that Tabby MH improves MLE to parity with real data. Tabby MH's lower discrimination score signifies this model's samples are more realistic than non-Tabby samples.

3.4 Tracking the Adaptation to Individual Columns

We address our final claim by examining Tabby's progress while fine-tuning on tabular data.
Claim 4: Tabby's loss formulation allows for convenient tracking of per-column performance at training time, leading to better understanding of model behavior.

Setup: For three runs, we train a Tabby MH model on a subset of the House dataset containing 5160 rows and six columns. We log the individual columns' losses on the evaluation dataset every 2500 steps while training for 10 epochs, then average across the runs.

Results: Individual column losses are shown in Figure 8. This information can be vital to understanding model behavior and training progress, as elaborated in Section E.4.

3.5 Discussion

We find that Tabby models synthesize high-quality data in a variety of settings. In particular, **Plain-trained Tabby MH consistently outperforms all prior LLM-based approaches** and is comparable or better than Tab-DDPM in most settings, despite Tabby enjoying greater flexibility under fewer assumptions than Tab-DDPM. The Tabby architecture modification allows LLMs to better model both univariate column distributions and multivariate relationships across columns.

Unusually, we find that the baseline Plain training technique with Distilled-GPT2 performs quite well on several standard evaluation datasets. **The high performance of the Plain training technique compared to prior LLM works on tabular synthesis, which are more complex, is surprising.** As of this writing, the Adult, House and Diabetes datasets have become quite prevalent for tabular synthesis evaluation. We hope that future research will build off of our evaluation setup by continuing to include more diverse and challenging tabular datasets, along with extensions to other structured modalities.

Limitations: We highlight two additional priorities for future work:

- Privacy preservation is important for trainsets that contain sensitive data, such as patient medical information. While Tabby's privacy preservation is similar to prior works (Appendix E), a method with strong formal privacy guarantees is an important next step for privacy-critical applications.

- Computationally-constrained environments or tasks with particularly large datasets may require deep learning approaches that are specifically designed with efficiency in mind. The Plain training method and insights from Section 3.2 may be useful towards this priority, but we leave further experimentation and the efficient implementations detailed in Section 2.4 to future work.

4 Related Work

Tabular data has played a central role in machine learning since the field's early days. In particular, decision trees and relatives (Song & Lu, 2015) are well-adapted to table classification or regression. Table synthesis is a growing area, though frequently overlooked in favor of image and text synthesis.

Classical synthesis: Classic machine learning models, such as random forest models or Bayesian networks, may be used to synthesize tables (Reiter, 2005; Zhang et al., 2017), but are limited in the data types and distributions that may be represented.

Generative Adversarial Networks (GANs): Many tabular synthesis methods rely on GANs (Goodfellow et al., 2014; Xu et al., 2019), but have encountered several limitations. In particular, distributions of ordinal columns are frequently imbalanced, leading GANs to undesirable phenomena such as mode collapse. Continuous columns may possess multiple modes and complex distributions, which GANs also struggle to capture (Xu et al., 2019).

Diffusion Models: Forest Diffusion (Jolicoeur-Martineau et al., 2024) and Tab-DDPM (Kotelnikov et al., 2022) are state-of-the-art table synthesis approaches based on the diffusion model. Both show top performance on many standard tabular metrics and are reliable for certain applications. Unfortunately, this performance is achieved by strong assumptions on the nature of the data space–for instance, numeric target variables may only assume integer values (see Figure 4) and diffusion models are unable to model non-categorical string columns such as addresses or telephone numbers. The ability to reach table synthesis performance comparable to that of diffusion models, but with fewer assumptions, is as an area of active research.

LLMs: A small, but growing, body of work has applied LLMs' flexible modeling abilities to tables. GReaT (Borisov et al., 2022) is a method to convert tabular data into a sentence format compatible with LLMs, then "shuffling" the order in which columns occur for each row to improve the modeling of inter-column dependencies. TapTap (Zhang et al., 2023) pretrains LMs on a variety of tabular data before fine-tuning on a downstream table synthesis task, while Tabula (Zhao et al., 2023) explores methods of preprocessing the training data to decrease sequence length. Other LLM-based works have adapted these advances to relational tables (Solatorio & Dupriez, 2023), 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 (Seedat et al., 2024). Many of these works can be used in concert with Tabby, as demonstrated in Section 3.

MoE Architectures: The key innovation of Tabby is the application of Gated Mixture of Expert (MoE) layers (Shazeer et al., 2017; Masoudnia & Ebrahimpour, 2014) for LLM table synthesis. MoE layers have enjoyed utility in multitask (Ma et al., 2018; Gupta et al., 2022) and multimodal learning (Zhao et al., 2024; Park et al., 2018), by creating sets of model parameters dedicated to a specific task.

5 Conclusion

Tabby is an MOE-based architecture modification that allows LLMs to generate realistic tabular data. Tabby reaches MLE parity with real data in 3/6 datasets. We hope this promising performance spurs future work on architecture modifications that allow LLMs to represent structured data.

Broader Impact Statement

Tabby is an improvement in the realism of synthetic tabular data, with extensions to non-tabular structured data. As such, Tabby (and its downstream applications) will have positive impacts on tasks that require synthetic structured data, such as low-data downstream tasks or privacy-critical tasks. However, Tabby may also be useful for negative downstream applications, such as the falsification of data—a drawback common to many advancements in the realism of generative modeling.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets, November 2023. URL http://arxiv.org/abs/2311.15127. arXiv:2311.15127 [cs].

Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language Models are Realistic Tabular Data Generators. September 2022. URL https://openreview.net/forum?id=cEygmQNOeI.

Fida K Dankar, Mahmoud K Ibrahim, and Leila Ismail. A multi-dimensional evaluation of synthetic data generators. *IEEE Access*, 10:11147–11158, 2022.

Maria F. Davila, Sven Groen, Fabian Panse, and Wolfram 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].

Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, January 2002. ISSN 1436-4646. doi: 10.1007/s101070100263. URL https://arxiv.org/abs/cs/0102001.

Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. Large Language Models(LLMs) on Tabular Data: Prediction, Generation, and Understanding – A Survey, February 2024. URL http://arxiv.org/abs/2402.17944. arXiv:2402.17944 [cs].

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Shashank Gupta, Subhabrata Mukherjee, Krishan Subudhi, Eduardo Gonzalez, Damien Jose, Ahmed H Awadallah, and Jianfeng Gao. Sparsely activated mixture-of-experts are robust multi-task learners. *arXiv preprint arXiv:2204.07689*, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Alexia Jolicoeur-Martineau, Kilian Fatras, and Tal Kachman. Generating and Imputing Tabular Data via Diffusion and Flow-based Gradient-Boosted Trees, February 2024. URL http://arxiv.org/abs/2309.09968. arXiv:2309.09968 [cs].

Michael Kahn. Diabetes. UCI Machine Learning Repository, 1994. URL https://www.openml.org/search?type=data&sort=runs&id=37&status=active. DOI: https://doi.org/10.24432/C5T59G.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. TabDDPM: Modelling Tabular Data with Diffusion Models, September 2022. URL http://arxiv.org/abs/2209.15421. arXiv:2209.15421 [cs].

Anton D. Lautrup, Tobias Hyrup, Arthur Zimek, and Peter Schneider-Kamp. Syntheval: a framework for detailed utility and privacy evaluation of tabular synthetic data. *Data Mining and Knowledge Discovery*, 39(1):6, December 2024. ISSN 1573-756X. doi: 10.1007/s10618-024-01081-4. URL https://doi.org/10.1007/s10618-024-01081-4.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed 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*, pp. 1930–1939, 2018.

Aswanth Manoj. Glaucoma Diagnosis JSON Analysis Dataset at Hugging Face, 2024. URL `https://huggingface.co/datasets/AswanthCManoj/glaucoma_diagnosis_json_analysis`.

Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42:275–293, 2014.

Warwick J Nash, Tracy L Sellers, Simon R Talbot, Andrew J Cawthorn, and Wes B Ford. The population biology of abalone (haliotis species) in tasmania. i. blacklip abalone (h. rubra) from the north coast and islands of bass strait. *Sea Fisheries Division, Technical Report*, 48:p411, 1994.

R Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3): 291–297, 1997.

David Keetae Park, Seungjoo Yoo, Hyojin Bahng, Jaegul Choo, and Noseong Park. Megan: Mixture of experts of generative adversarial networks for multimodal image generation. *arXiv preprint arXiv:1805.02481*, 2018.

Zhaozhi Qian, Bogdan-Constantin Cebere, and Mihaela van der Schaar. Synthcity: facilitating innovative use cases of synthetic data in different data modalities, 2023. URL `https://arxiv.org/abs/2301.07573`.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Equivariance through parameter-sharing. In *International conference on machine learning*, pp. 2892–2901. PMLR, 2017.

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

Nicholas Roberts, Samuel Guo, Cong Xu, Ameet Talwalkar, David Lander, Lvfang Tao, Linhang Cai, Shuaicheng Niu, Jianyu Heng, Hongyang Qin, Minwen Deng, Johannes Hog, Alexander Pfefferle, Sushil Ammanaghatta Shivakumar, Arjun Krishnakumar, Yubo Wang, Rhea Sukthanker, Frank Hutter, Euxhen Hasanaj, Tien-Dung Le, Mikhail Khodak, Yuriy Nevmyvaka, Kashif Rasul, Frederic Sala, Anderson Schneider, Junhong Shen, and Evan Sparks. Automl decathlon: Diverse tasks, modern methods, and efficiency at scale. In Marco Ciccone, Gustavo Stolovitzky, and Jacob Albrecht (eds.), *Proceedings of the NeurIPS 2022 Competitions Track*, volume 220 of *Proceedings of Machine Learning Research*, pp. 151–170. PMLR, 28 Nov–09 Dec 2022. URL `https://proceedings.mlr.press/v220/roberts23a.html`.

Nabeel Seedat, Nicolas Huynh, Boris van Breugel, and Mihaela van der Schaar. Curated LLM: Synergy of LLMs and Data Curation for tabular augmentation in ultra low-data regimes, February 2024. URL `http://arxiv.org/abs/2312.12112`. arXiv:2312.12112 [cs].

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Juntong Shi, Minkai Xu, Harper Hua, Hengrui Zhang, Stefano Ermon, and Jure Leskovec. TabDiff: a Mixed-type Diffusion Model for Tabular Data Generation, February 2025. URL `http://arxiv.org/abs/2410.20626`. arXiv:2410.20626 [cs].

Aivin V. Solatorio and Olivier Dupriez. REaLTabFormer: Generating Realistic Relational and Tabular Data using Transformers, February 2023. URL `http://arxiv.org/abs/2302.02041`. arXiv:2302.02041 [cs].

Yan-yan Song and Ying Lu. Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*, 27(2):130–135, April 2015. ISSN 1002-0829. doi: 10.11919/j.issn.1002-0829.215044. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4466856/`.

Tejashvi. Tour & Travels Customer Churn Prediction, 2023. URL https://www.kaggle.com/datasets/tejashvi14/tour-travels-customer-churn-prediction.

Boris van Breugel and Mihaela van der Schaar. Why Tabular Foundation Models Should Be a Research Priority, May 2024. URL https://arxiv.org/abs/2405.01147v2.

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

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling Tabular data using Conditional GAN, October 2019. URL http://arxiv.org/abs/1907.00503. arXiv:1907.00503 [cs, stat].

Yousuf Zaman. Machine Learning Model on Rainfall - A Predicted Approach for Bangladesh. 2018.

Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):1–41, 2017.

Tianping Zhang, Shaowen Wang, Shuicheng Yan, Jian Li, and Qian Liu. Generative Table Pre-training Empowers Models for Tabular Prediction, May 2023. URL http://arxiv.org/abs/2305.09696. arXiv:2305.09696 [cs].

Xueliang Zhao, Mingyang Wang, Yingchun Tan, and Xianjie Wang. Tgmoe: A text guided mixture-of-experts model for multimodal sentiment analysis. *International Journal of Advanced Computer Science & Applications*, 15(8), 2024.

Zilong Zhao, Robert Birke, and Lydia Chen. TabuLa: Harnessing Language Models for Tabular Data Synthesis, October 2023. URL http://arxiv.org/abs/2310.12746. arXiv:2310.12746 [cs] version: 1.

A Terminology Glossary

For convenience, in addition to definitions within the main text, we list and define the most frequently-used terms and abbreviations in our paper here:

- **DGPT2**: Distilled-GPT2 (Radford et al., 2019).
- **Distance to Closest Record (DCR)**: Metric for synthetic data quality and privacy, defined in Appendix D.
- **Discrimination**: Metric for synthetic data quality, defined in Appendix D.
- **GReaT** (Borisov et al., 2022): The landmark work on fine-tuning pre-existing LLMs to synthesize tabular data by encoding datapoints as text. Similar to Plain training, but includes train-time complications such as shuffling the order in which columns are encoded and sample-time complications such as the inability to generate label values that do not occur in the training dataset. Discussed in-detail in Section 4.
- **GReaT+Tabula (GT)**: The combination of GReaT training plus Tabula (Zhao et al., 2023) data encoding; see Section 4.
- **GReaT+TapTap+Tabula (GTT)**: The combination of GReaT training plus Tabula encoding and TapTap (Zhang et al., 2023) pre-training on tabular data (which is performed *after* the LLM is pre-trained on text data).
- **Low Rank Adapters (LoRA)**: Parameter-efficient training method from (Hu et al., 2021).
- **Mixture-of-Experts (MoE)**: Architecture technique which replaces one block with a set of specialized blocks; see Section 4.
- **Multi-Head (MH)**: The best-performing variant of Tabby, which replaces the LLM's language model output layer with an MoE layer.
- **Machine Learning Efficacy (MLE)**: Our primary evaluation metric, introduced in Section 3.0 and discussed in-detail in Appendix D.
- **Multi-MMLP (MMLP)**: Tabby modification that applies MoE to the transformer blocks' MLPs.

Table 5: Download links for each dataset.

Dataset	Link
Diabetes	https://www.openml.org/search?type=data&sort=runs&id=37&status=active
Travel	https://www.kaggle.com/datasets/tejashvi14/tour-travels-customer-churn-prediction/data
Adult	https://archive.ics.uci.edu/dataset/2/adult
Abalone	https://www.openml.org/search?type=data&sort=runs&id=183&status=active
Rainfall	https://www.openml.org/search?type=data&status=active&id=41539&sort=runs
House	https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html
Glaucoma	https://huggingface.co/datasets/AswanthCManoj/glaucoma_diagnosis_json_analysis

- **Multi-MLP and LM Head (MMLP+MH)**: Tabby modification that applies MoEs to **both** the transformer blocks' MLPs and to the language model output layers.
- **Non-Synthetic (Upper Bound)**: Used for the MLE metric, this score represents the performance of a downstream classifier trained on real, instead of synthetic, data. See Appendix D for details.
- **Non-Tabby (NT)**: An LLM without the Tabby modification, also referred to as a Base LLM.
- **Plain**: Our simple but high-performing technique for training LLMs on tabular data; introduced in Section 2.
- **Tab-DDPM (TDDPM)**: A state-of-the-art tabular synthesis technique based on the diffusion model architecture, which relies on several important assumptions; see Section 4.

B Additional dataset information

We select a variety of tabular datasets for our evaluations, with two goals in mind. First, the inclusion of the most standard tabular datasets—Diabetes, Adult and House—allows for easy comparison with prior works. Second, we include classification and regression datasets from a variety of domains, such as Earth science (Rainfall), business (Travel) and medicine (Diabetes). This diversity allows us to demonstrate that Tabby models achieve high performance across a variety of real-world data types and distributions. Refer to Table 5 for download links to each dataset.

Diabetes (Kahn, 1994) 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.

Travel (Tejashvi, 2023) was collected by a travel agency wishing to predict customer churn. With the binary variable churn as the target, features include whether the traveler booked a hotel, frequent flyer status and traveler age. While most features are categorical, there are two numerical columns: traveler age and the number of times that the customer has used the travel agency in the past.

Adult (Becker & Kohavi, 1996) 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.

Our first regression dataset is *Abalone* (Nash et al., 1994), which records the basic measurements of abalones, such weight and height. The target variable is the abalone's age.

The *Rainfall* (Zaman, 2018) dataset, while challenging to many LLM-based synthesis methods, contains only four columns which record historical weather data in Bangladesh. Its target variable is the amount of rainfall recorded, and the features are the year, month and weather station location.

House (Pace & Barry, 1997) is a standard regression dataset. 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

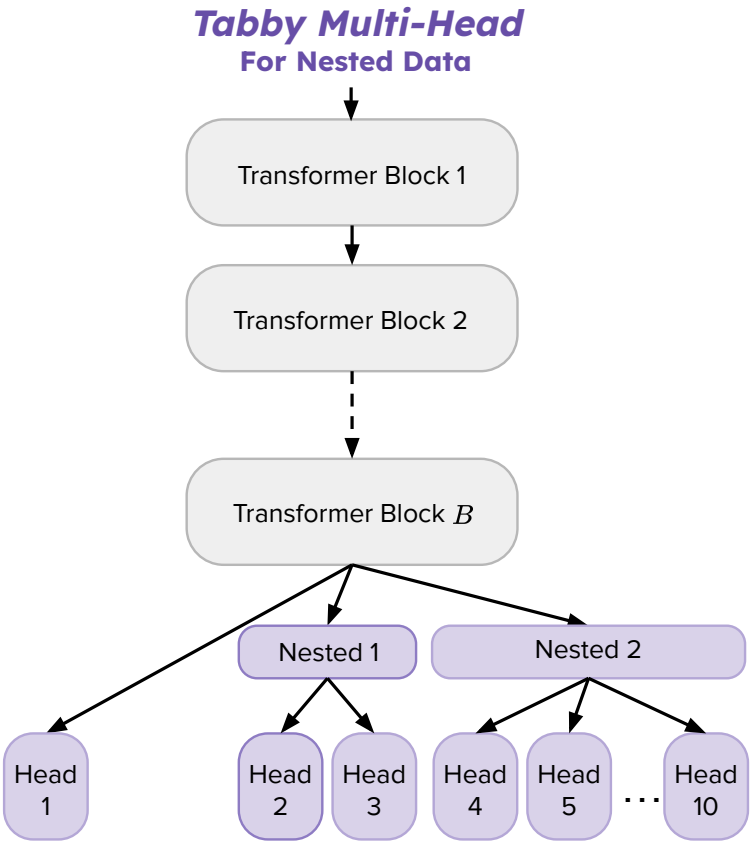

Figure 5: An overview of the Tabby MH modifications for the nested Glaucoma dataset.

column is the block's median house value, which is numerical and allows us to assess Tabby's synthetic data in a regression task.

Glaucoma (Manoj, 2024) dataset consists of JSON records describing ophthalmic patients under consideration for a glaucoma diagnosis. Each record contains various qualitative and quantitative information about the eye, as demonstrated by the examples in Box D.

C Tabby for Nested Data

Figure 5 provides a visualization of the Tabby architecture used in Section E.4 to generate nested JSON data.

D Details on Experimental Setup

Calculation of results: The reported result for each model and training setup is the average across three training runs, where not otherwise stated. For each of the three trained models, we sample $10,000$ datapoints, compute evaluation metrics separately for the three resulting synthetic datasets, 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, with selected learning rates reported in Appendix E.5. For non-LLM works, we follow the procedures detailed in each of these works.

Box D: Representative examples from Glaucoma (Manoj, 2024)


```
[
  {
    "diagnosis": "glaucoma",
    "disc_info": {
      "disc_size": "large",
      "cup_disc_ratio": 0.8
    },
    "rim_info": {
      "rim_pallor": true,
      "rim_color": "pale",
      "bayoneting": true,
      "sharp_edge": true,
      "laminar_dot_sign": true,
      "notching": true,
      "rim_thinning": true
    }
  },
  {
    "diagnosis": "normal",
    "disc_info": {
      "disc_size": "normal",
      "cup_disc_ratio": 0.4
    },
    "rim_info": {
      "rim_pallor": false,
      "rim_color": "pink",
      "bayoneting": false,
      "sharp_edge": false,
      "laminar_dot_sign": false,
      "notching": false,
      "rim_thinning": false
    }
  },
  {
    "diagnosis": "normal",
    "disc_info": {
      "disc_size": "normal",
      "cup_disc_ratio": 0.4
    },
    "rim_info": {
      "rim_pallor": false,
      "rim_color": "pink",
      "bayoneting": false,
      "sharp_edge": false,
      "laminar_dot_sign": false,
      "notching": false,
      "rim_thinning": false
    }
  }
]
```


More detailed definition of Machine Learning Efficacy (MLE): Given a synthetic dataset produced by a generative model, we begin to calculate MLE by training one *downstream classifier* using the synthetic dataset. Then, we evaluate the performance of this downstream classifier on a real test set, drawn from the same distribution as the generative model's train set. We compare this classifier's performance to a *second*

classifier, which is trained on the same training data as the generative model. If the synthetically-trained classifier performs worse than the classifier trained on real data, then (intuitively) the synthetic data is of lower quality than the real data: for instance, the distributions of features in the real data are not well-reflected in the synthetic data.

Put another way: given a real dataset, we form disjoint training and test sets, denoted R and D respectively. A generative model is trained on R, then generates 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 report both K_R and K_S in our results, considering MLE to be the difference in performance between K_R and K_S.

We use a random forest classifier or regressor as our downstream model K. For classification datasets, we compare the accuracy of K_R and K_S, while for regression datasets, we compare the coefficient of determination R^2. We define the coefficient of determination R^2 as $\max(1 - \frac{r}{t}, 0)$, where r and u are the residual sum of squares and total sum of squares, respectively. This formulation means that if a model performs worse than random guessing, its R^2 value will be represented as 0. For both the accuracy and R^2 coefficient metrics, a higher score indicates higher-quality data.

Information on Performance Profiles: For a given method $m \in M$, its performance profile curve is defined as

$$\rho_m(\tau) := \frac{1}{|T|} \left| \left\{ t \in T : \frac{s_{t,m}}{\min_{m' \in M}\{s_{t,m'}\}} \leq \tau \right\} \right|$$

for a set of tasks T and scores $s_{t,m} : t \in T$, where lower values indicate better performance on each task. In order to satisfy the requirement that lower scores are better for the MLE metric, we set $s_{t,m} = 1 - \text{MLE}_{t,m}$. Then for each method, we obtain a final score by taking the area under the curve $\rho_m(\tau)$ to obtain the AUP score as

$$\text{AUP}_m = \int_0^{\tau*} \rho_m(\tau) d\log(\tau).$$

with $\tau*$ being the smallest τ such that $\rho_m(\tau) = 1$ for all methods $m \in M$, and where a higher AUP score indicates better performance.

Discrimination: Discrimination (Qian et al., 2023) quantifies the degree to which the generative model introduces spurious correlations or other patterns that differentiate synthetic from 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. For this reason, our reported discrimination scores are calculated as the absolute difference between 50% and the accuracy of discriminator C. Accordingly, lower discrimination scores represent better performance.

Distance to Closest Record: (DCR) Distance to Closest Record (DCR) (Lautrup et al., 2024) quantifies the distance between each synthetic datapoint and its most-similar example in the training set R. In addition to synthesis quality, this metric is an indication of the degree to which the model memorizes samples during training. Specifically, for each synthetic example $s \in S$, we compute its distance to every training point $r \in R$ (using L0 distance for categorical columns and L1 distance for numerical columns) and take the smallest of these distances. The overall DCR is then reported as the average of these minimum distances across all synthetic examples in SS. Lower DCR is associated with high-quality synthesis, but a DCR score of 0 implies that most synthetic examples are merely copies of training dataset points memorized during training. As such, we consider the best DCR to be the lowest nonzero score.

Table 6: Discrimination metric (↓), defined in Appendix D, for approaches compared in Section 3.1. Tabby produces data with better MLE without worsening the synthetic data's discrimination score, performing competitively with Tab-DDPM.

	Diabetes	**Travel**	**Adult**	**Abalone**	**Rainfall**	**House**
CTGAN	0.42 ± 0.00	0.27 ± 0.01	0.48 ± 0.00	0.46 ± 0.00	0.18 ± 0.05	0.32 ± 0.06
TVAE	0.45 ± 0.02	0.50 ± 0.00	0.46 ± 0.01	0.45 ± 0.02	0.41 ± 0.01	0.39 ± 0.03
Forest Diffusion	0.27 ± 0.00	0.28 ± 0.00	0.50 ± 0.00	0.24 ± 0.00	0.09 ± 0.00	0.16 ± 0.00
Tab-DDPM	0.11 ± 0.00	0.05 ± 0.03	$\mathbf{0.01 \pm 0.01}$	$\mathbf{0.03 \pm 0.01}$	$\mathbf{0.01 \pm 0.02}$	0.33 ± 0.04
Plain Base	$\mathbf{0.04 \pm 0.01}$	0.03 ± 0.02	0.09 ± 0.01	0.06 ± 0.01	0.03 ± 0.01	0.07 ± 0.06
Plain Tabby MH	0.06 ± 0.02	$\mathbf{0.02 \pm 0.01}$	0.10 ± 0.01	0.06 ± 0.00	0.08 ± 0.00	$\mathbf{0.03 \pm 0.01}$
GReaT Base	0.28 ± 0.01	0.06 ± 0.01	0.20 ± 0.01	0.08 ± 0.02	N/A*	0.16 ± 0.01
GReaT *Tabby MH*	0.29 ± 0.02	0.08 ± 0.03	0.20 ± 0.01	0.11 ± 0.03	0.45 ± 0.09*	0.19 ± 0.01
GTT Base	0.27 ± 0.02	0.07 ± 0.01	0.20 ± 0.02	0.05 ± 0.01	0.39 ± 0.11	0.18 ± 0.03
GTT *Tabby MH*	0.28 ± 0.02	0.07 ± 0.02	0.13 ± 0.05	0.16 ± 0.01	0.31 ± 0.21	0.20 ± 0.01

E Further Experimental Results

E.1 Additional Metrics for Main Results Tables

For the experiment presented in Section 3.1, we include five additional metrics:

1. **Discrimination** (Table 6) works similarly to a GAN's discriminator (Goodfellow et al., 2014): the downstream discriminator model receives equal numbers of real and of synthetic datapoints, then learns to distinguish between them. The more difficulty that the discriminator model encounters in distinguishing between real and synthetic, the more we can say that the real dataset's patterns are preserved within the synthetic examples.

2. **Distance to Closest Record** (**DCR**, Table 7) measures how far the average synthetic datapoint lies from its nearest non-synthetic datapoint. If DCR equals zero, it indicates that the model has memorized its trainset, while a very large DCR indicates that the model is not preserving the trainset's patterns very well: small, non-zero, DCR scores are ideal.

3. **Shape** (Shi et al., 2025) (Table 8) measures, for each individual column, how well the synthetic column matches the real column's distribution. We follow the SDMetrics library's implementation, which uses the Kolmogorov-Smirnov statistic for categorical and the complement of Total Variation Distance for numerical columns, then averages the distances across all columns to report a final summary number.

4. **Trend** (Shi et al., 2025) (Table 9)—which is often used in conjunction with Shape—measures relationships *between* columns, by quantifying the degree to which these relationships in the real data are preserved by the synthetic data. We again use the SDMetrics library's implementation, which measures Pearson correlation for relationships between numerical columns and complement of Total Variation Distance for relationships between categorical columns or one categorical and one numerical column (which is first binned to discretize the values).

5. **Wasserstein Distance** (Table 10) computes the distance between the real and synthetic datasets' numeric columns. This metric allows us to capture the similarity of relationships among all numeric columns, as opposed to the pairwise interactions measured by the Trend metric.

These metrics largely corroborate our findings in Section 3.1. In particular, Plain Tabby MH's low DCR and discrimination scores indicate that this model's synthetic data closely resembles that of real data. Additionally, the DCR scores are small but *nonzero*, which indicates that the model is generating novel datapoints rather than simply repeating datapoints memorized during training.

Table 7: Distance to Closest Record (DCR, $\downarrow_{>0}$), defined in Appendix D, for approaches compared in Section 3.1. Tabby MH exhibits low, nonzero scores, indicating that its synthetic examples closely resemble real data without simply copying the training data points.

	Diabetes	**Travel**	**Adult**	**Abalone**	**Rainfall**	**House**
CTGAN	0.82 ± 0.00	0.59 ± 0.03	1.70 ± 0.09	0.76 ± 0.02	0.03 ± 0.01	0.13 ± 0.02
TVAE	0.27 ± 0.01	0.10 ± 0.06	0.16 ± 0.03	0.41 ± 0.01	0.03 ± 0.00	0.07 ± 0.00
Forest Diffusion	0.29 ± 0.00	0.06 ± 0.00	0.35 ± 0.02	0.09 ± 0.01	$\mathbf{0.01 \pm 0.00}$	0.06 ± 0.00
Tab-DDPM	0.63 ± 0.04	0.00 ± 0.00	0.31 ± 0.03	0.12 ± 0.01	$\mathbf{0.01 \pm 0.00}$	0.08 ± 0.00
Plain Base	$\mathbf{0.01 \pm 0.00}$	$\mathbf{0.01 \pm 0.00}$	0.33 ± 0.15	$\mathbf{0.01 \pm 0.00}$	0.00 ± 0.00	$\mathbf{0.03 \pm 0.01}$
Plain Tabby MH	0.02 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.25 ± 0.03	$\mathbf{0.01 \pm 0.01}$	$\mathbf{0.01 \pm 0.00}$	0.04 ± 0.00
GReaT Base	0.33 ± 0.00	0.02 ± 0.01	0.12 ± 0.03	0.10 ± 0.00	N/A*	0.06 ± 0.00
GReaT *Tabby MH*	0.36 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.17 ± 0.08	0.10 ± 0.01	**0.01***	0.06 ± 0.00
GTT Base	0.31 ± 0.01	0.02 ± 0.00	0.14 ± 0.01	0.10 ± 0.00	0.02 ± 0.00	0.06 ± 0.00
GTT *Tabby MH*	0.37 ± 0.01	0.02 ± 0.00	0.16 ± 0.07	0.10 ± 0.00	0.00 ± 0.01	0.05 ± 0.00

Table 8: Shape (Shi et al., 2025) (↑), for diffusion and LLM approaches compared in Section 3.1. Tabby Plain is a best-performing method on 3 out of 6 datasets, which is similar to Tab-DDPM (a best performing method on 4 out of 6 datasets). Tabby Plain reaches similar performance, without the same limiting assumptions on the nature of the dataset that are required by Tab-DDPM.

	Diabetes	**Travel**	**Adult**	**Abalone**	**Rainfall**	**House**
Forest Diffusion	0.91 ± 0.00	0.95 ± 0.00	0.89 ± 0.00	0.96 ± 0.02	0.95 ± 0.00	0.94 ± 0.00
Tab-DDPM	0.89 ± 0.00	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.98 \pm 0.00}$	0.95 ± 0.00
Plain Base	0.96 ± 0.02	$\mathbf{0.99 \pm 0.00}$	0.95 ± 0.00	0.94 ± 0.00	0.96 ± 0.01	0.95 ± 0.03
Plain Tabby MH	$\mathbf{0.98 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.95 ± 0.01	0.93 ± 0.02	0.93 ± 0.00	$\mathbf{0.97 \pm 0.00}$
GReaT Base	0.81 ± 0.01	0.94 ± 0.00	0.89 ± 0.01	0.95 ± 0.01	$N/A*$	0.92 ± 0.00
GReaT *Tabby MH*	0.85 ± 0.00	0.93 ± 0.01	0.90 ± 0.01	0.91 ± 0.03	$0.23 \pm 0.40*$	0.89 ± 0.00
GTT Base	0.80 ± 0.00	0.93 ± 0.00	0.89 ± 0.02	0.95 ± 0.00	$0.50 \pm 0.43*$	0.91 ± 0.02
GTT *Tabby MH*	0.83 ± 0.01	0.94 ± 0.00	0.81 ± 0.09	0.89 ± 0.00	0.47 ± 0.45	0.89 ± 0.00

Table 9: Trend (Shi et al., 2025) (↑), for diffusion and LLM approaches compared in Section 3.1. Tabby Plain is a best-performing method on 3 out of 6 datasets, which is similar to Tab-DDPM (a best performing method on 4 out of 6 datasets). Tabby Plain reaches similar performance, without the same limiting assumptions on the nature of the dataset that are required by Tab-DDPM.

	Diabetes	**Travel**	**Adult**	**Abalone**	**Rainfall**	**House**
Forest Diffusion	0.87 ± 0.00	0.68 ± 0.00	0.60 ± 0.00	0.88 ± 0.01	0.58 ± 0.00	$\mathbf{0.99 \pm 0.00}$
Tab-DDPM	0.88 ± 0.00	0.97 ± 0.00	$\mathbf{0.97 \pm 0.00}$	$\mathbf{0.95 \pm 0.00}$	$\mathbf{0.95 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$
Plain Base	0.95 ± 0.04	0.73 ± 0.06	0.88 ± 0.04	0.91 ± 0.05	0.89 ± 0.01	0.96 ± 0.02
Plain Tabby MH	$\mathbf{0.97 \pm 0.00}$	$\mathbf{0.98 \pm 0.00}$	0.88 ± 0.02	0.94 ± 0.01	0.87 ± 0.00	$\mathbf{0.99 \pm 0.00}$
GReaT Base	0.86 ± 0.01	0.84 ± 0.10	0.77 ± 0.02	0.92 ± 0.00	N/A*	0.95 ± 0.01
GReaT *Tabby MH*	0.85 ± 0.02	0.89 ± 0.01	0.77 ± 0.06	0.92 ± 0.01	$0.47 \pm 0.06*$	0.95 ± 0.01
GTT Base	0.88 ± 0.01	0.84 ± 0.10	0.79 ± 0.03	0.93 ± 0.00	$0.52 \pm 0.17*$	0.95 ± 0.02
GTT *Tabby MH*	0.86 ± 0.01	0.85 ± 0.10	0.56 ± 0.22	0.91 ± 0.01	0.48 ± 0.33	0.96 ± 0.01

Table 10: Wasserstein distance (↓), for diffusion and LLM approaches compared in Section 3.1. Plain Tabby is the best-performing method on Diabetes and House, and performs similarly to Tab-DDPM (best on Adult and Rainfall).

	Diabetes	Travel	Adult	Abalone	Rainfall	House
Forest Diffusion	19.08 ± 0.48	0.35 ± 0.07	$9.9E3 \pm 3.6E3$	0.43 ± 0.10	28.69 ± 5.6	110.40 ± 20.50
Tab-DDPM	80.54 ± 8.09	0.30 ± 0.03	$\mathbf{8.3E3 \pm 1.9E3}$	0.37 ± 0.02	$\mathbf{18.47 \pm 6.88}$	104.47 ± 31.47
Plain Base	20.83 ± 11.04	$\mathbf{0.23 \pm 0.05}$	$1.3E4 \pm 7.6E3$	0.61 ± 0.17	22.56 ± 6.06	116.17 ± 68.63
Plain Tabby MH	$\mathbf{14.75 \pm 1.32}$	0.29 ± 0.05	$1.2E4 \pm 2.9E3$	0.52 ± 0.22	32.95 ± 5.77	$\mathbf{91.48 \pm 9.08}$
GReaT Base	82.75 ± 3.86	0.50 ± 0.12	$2.6E4 \pm 8.5E3$	0.33 ± 0.08	N/A*	387.83 ± 28.45
GReaT *Tabby MH*	82.69 ± 2.09	0.46 ± 0.04	$3.2E4 \pm 1.7E4$	0.34 ± 0.07	$7E6 \pm 6E6*$	304.11 ± 63.08
GTT Base	81.34 ± 3.90	0.48 ± 0.03	$2.2E4 \pm 5.9E3$	$\mathbf{0.30 \pm 0.06}$	$3E6 \pm 6E6*$	309.51 ± 78.00
GTT *Tabby MH*	78.70 ± 1.73	0.37 ± 0.01	$4.9E4 \pm 1.8E4$	0.45 ± 0.07	$3E6 \pm 6E6$	375.05 ± 20.07

Table 11: Machine Learning Efficacy (MLE, ↑), defined in Section 3.0, for Base non-Tabby GPT2 models, as well as Tabby models with MoE layers applied to the transformer MLPs, language modeling head, or both (notated as MMLP, MH, and MMLP+MH respectively).

	Diabetes	Travel	Adult	Abalone	Rainfall	House
Non-Synthetic	**0.73**	**0.87**	**0.85**	**0.45**	**0.54**	**0.61**
Plain Non-Tabby	$\mathbf{0.75 \pm 0.02}$	0.86 ± 0.01	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.44 \pm 0.01}$	0.52 ± 0.03	0.55 ± 0.08
Plain Tabby MMLP	$\mathbf{0.75 \pm 0.03}$	0.83 ± 0.02	0.77 ± 0.01	0.32 ± 0.03	0.35 ± 0.04	0.00 ± 0.00
Plain Tabby MH	$\mathbf{0.74 \pm 0.00}$	$\mathbf{0.88 \pm 0.01}$	$\mathbf{0.85 \pm 0.00}$	0.43 ± 0.01	0.49 ± 0.00	$\mathbf{0.60 \pm 0.00}$
Plain Tabby MMLP+MH	0.68 ± 0.02	0.83 ± 0.01	0.76 ± 0.01	0.33 ± 0.03	0.36 ± 0.19	0.02 ± 0.03
GReaT Non-Tabby	0.62 ± 0.01	0.85 ± 0.02	0.83 ± 0.01	0.41 ± 0.01	N/A*	0.56 ± 0.01
GReaT *Tabby MMLP*	$\mathbf{0.74 \pm 0.01}$	0.85 ± 0.03	0.84 ± 0.01	0.38 ± 0.01	0.24 ± 0.25	0.56 ± 0.02
GReaT *Tabby MH*	0.64 ± 0.01	0.86 ± 0.01	0.83 ± 0.00	0.40 ± 0.01	0.00 ± 0.00*	0.56 ± 0.01
GReaT *Tabby MMLP+MH*	0.69 ± 0.04	0.83 ± 0.02	0.83 ± 0.01	0.38 ± 0.03	0.17 ± 0.30	0.57 ± 0.01
GTT Non-Tabby	0.72 ± 0.06	$\mathbf{0.87 \pm 0.02}$	0.83 ± 0.01	0.40 ± 0.01	0.05 ± 0.01	0.55 ± 0.02
GTT *Tabby MMLP*	0.69 ± 0.04	$\mathbf{0.87 \pm 0.01}$	0.84 ± 0.00	0.36 ± 0.01	0.03 ± 0.00*	0.56 ± 0.01
GTT *Tabby MH*	0.62 ± 0.00	0.85 ± 0.01	0.76 ± 0.07	0.37 ± 0.02	0.26 ± 0.37	0.55 ± 0.00
GTT *Tabby MMLP+MH*	0.70 ± 0.04	0.85 ± 0.02	0.84 ± 0.00	0.38 ± 0.02	0.09 ± 0.13	0.57 ± 0.00

E.2 Applying Tabby to Transformer MLPs or Attention Blocks

We examine in-detail the performance of Tabby models with MoE applied to the transformer MLPs or attention blocks. We use the following terminology to refer to these architectures, visualized in Figure 6:

- **Multi-MLP** when each transformer's MLP block is replaced with an MoE layer,
- **Multi-MLP and Multi-Head (MMLP+MH)** when each transformer's MLP block is replaced with an MoE layer *and* the LM head is replaced with an MoE layer,
- **Multi-Attention (MA)** when each transformer's attention block is replaced with an MoE layer.

We focus on Tabby MH in Sections 3.1-3.4 because it demonstrates top performance in most settings. We display results for the MMLP and MMLP+MH architectures across all six datasets for MLE, discrimination and DCR in Tables 11, 12 and 13, respectively. All three metrics are displayed for the MA architecture on two datasets in Table 14.

E.3 Additional Metrics for Experiment Applying Tabby MH to Models of Varying Sizes

The results in Section 3.2 compare the MLE scores of Plain-trained models of varying sizes on the Travel dataset. Table 15 incorporates the results for the Diabetes and House datasets as well. Similarly, Table 16 presents results for models trained using GReaT and Tabula (TapTap is not included here, because TapTap-pretrained checkpoints are available only for Distill-GPT2 and GPT2).

Table 12: Discrimination metric (↓), defined in Appendix D, for Base non-Tabby GPT2 models, as well as Tabby models with MoE layers applied to the transformer MLPs, language modeling head, or both (notated as MMLP, MH, and MMLP+MH respectively).

	Diabetes	**Travel**	**Adult**	**Abalone**	**Rainfall**	**House**
Plain Non-Tabby	$\mathbf{0.04 \pm 0.01}$	0.03 ± 0.02	0.09 ± 0.01	0.06 ± 0.01	0.03 ± 0.01	0.07 ± 0.06
Plain Tabby MMLP	0.22 ± 0.03	$\mathbf{0.02 \pm 0.02}$	0.22 ± 0.06	0.19 ± 0.04	0.12 ± 0.00	0.19 ± 0.06
Plain Tabby MH	0.06 ± 0.02	$\mathbf{0.02 \pm 0.01}$	0.10 ± 0.01	0.06 ± 0.00	0.08 ± 0.00	$\mathbf{0.03 \pm 0.01}$
Plain Tabby MMLP+MH	0.19 ± 0.02	0.03 ± 0.02	0.25 ± 0.11	0.22 ± 0.03	0.12 ± 0.01	0.23 ± 0.03
GReaT Non-Tabby	0.28 ± 0.01	0.06 ± 0.01	0.20 ± 0.01	0.08 ± 0.02	N/A*	0.16 ± 0.01
GReaT *Tabby MMLP*	0.23 ± 0.01	0.10 ± 0.02	0.19 ± 0.00	0.08 ± 0.01	0.27 ± 0.17	0.16 ± 0.01
GReaT *Tabby MH*	0.29 ± 0.02	0.08 ± 0.03	0.20 ± 0.01	0.11 ± 0.03	0.45 ± 0.09*	0.19 ± 0.01
GReaT *Tabby MMLP+MH*	0.24 ± 0.01	0.09 ± 0.01	0.21 ± 0.01	0.07 ± 0.00	0.24 ± 0.17	0.16 ± 0.00
GTT Non-Tabby	0.27 ± 0.02	0.07 ± 0.01	0.20 ± 0.02	0.05 ± 0.01	0.39 ± 0.11	0.18 ± 0.03
GTT *Tabby MMLP*	0.28 ± 0.01	0.09 ± 0.01	0.18 ± 0.01	0.14 ± 0.02	0.46 ± 0.07*	0.18 ± 0.01
GTT *Tabby MH*	0.28 ± 0.02	0.07 ± 0.02	0.13 ± 0.05	0.16 ± 0.01	0.31 ± 0.21	0.20 ± 0.01
GTT *Tabby MMLP+MH*	0.24 ± 0.01	0.08 ± 0.01	0.18 ± 0.00	0.14 ± 0.02	0.24 ± 0.24	0.16 ± 0.01

Table 13: Distance to Closest Record (DCR, $\downarrow_{>0}$), defined in Appendix D, for Base non-Tabby GPT2 models, as well as Tabby models with MoE layers applied to the transformer MLPs, language modeling head, or both (notated as MMLP, MH, and MMLP+MH respectively).

	Diabetes	**Travel**	**Adult**	**Abalone**	**Rainfall**	**House**
Plain Non-Tabby	$\mathbf{0.01 \pm 0.00}$	$\mathbf{0.01 \pm 0.00}$	0.33 ± 0.15	$\mathbf{0.01 \pm 0.00}$	0.00 ± 0.00	$\mathbf{0.03 \pm 0.01}$
Plain Tabby MMLP	0.35 ± 0.03	0.08 ± 0.00	0.55 ± 0.09	0.21 ± 0.02	0.03 ± 0.00	$1.7e12 \pm 2.7e12$
Plain Tabby MH	0.02 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.25 ± 0.03	$\mathbf{0.01 \pm 0.01}$	$\mathbf{0.01 \pm 0.00}$	0.04 ± 0.00
Plain Tabby MMLP+MH	0.34 ± 0.02	0.07 ± 0.00	0.39 ± 0.15	0.20 ± 0.03	0.03 ± 0.01	$2.3e12 \pm 4.1e12$
GReaT Non-Tabby	0.33 ± 0.00	0.02 ± 0.01	0.12 ± 0.03	0.10 ± 0.00	N/A*	0.06 ± 0.00
GReaT *Tabby MMLP*	0.34 ± 0.01	0.02 ± 0.00	0.12 ± 0.01	0.10 ± 0.00	0.00 ± 0.01	0.06 ± 0.00
GReaT *Tabby MH*	0.36 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.17 ± 0.08	0.10 ± 0.01	$\mathbf{0.01}$*	0.06 ± 0.00
GReaT *Tabby MMLP+MH*	0.33 ± 0.02	0.02 ± 0.00	$\mathbf{0.11 \pm 0.01}$	0.10 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.06 ± 0.00
GTT Non-Tabby	0.31 ± 0.01	0.02 ± 0.00	0.14 ± 0.01	0.10 ± 0.00	0.02 ± 0.00	0.06 ± 0.00
GTT *Tabby MMLP*	0.31 ± 0.02	0.02 ± 0.00	0.14 ± 0.03	0.10 ± 0.00	$\mathbf{0.01}$*	0.06 ± 0.00
GTT *Tabby MH*	0.37 ± 0.01	0.02 ± 0.00	0.16 ± 0.07	0.10 ± 0.00	0.00 ± 0.01	0.05 ± 0.00
GTT *Tabby MMLP+MH*	0.31 ± 0.00	0.02 ± 0.00	$\mathbf{0.11 \pm 0.02}$	0.11 ± 0.00	$\mathbf{0.01 \pm 0.01}$	0.06 ± 0.00

Table 14: All evaluation metrics, for non-Tabby models and Tabby models with MoE applied to the transformer attention blocks (abbreviated as Tabby MA). Base LLM is DGPT2.

	MLE (↑)		**Discrimination (↓)**		**DCR ($\downarrow_{>0}$)**	
	Diabetes	**House**	**Diabetes**	**House**	**Diabetes**	**House**
Non-Synthetic (Upper Bound)	**0.73**	**0.61**				
Plain Non-Tabby	**0.75**	0.55	**0.04**	0.07	**0.01**	0.03
Plain Tabby MA	0.62	0.08	0.23	0.28	0.41	0.08
GTT Non-Tabby	0.72	0.55	0.27	0.18	0.31	0.06
GTT *Tabby MA*	0.62	0.56	0.31	0.17	0.36	0.06

Table 15: Results using Plain training for all three datasets of the experiment in Section 3.2, which compares non-Tabby and Tabby MH models across base LLMs of varying sizes.

	Travel		Diabetes		House	
	MLE (↑)	**Params**	**MLE (↑)**	**Params**	**MLE (↑)**	**Params**
Non-Synthetic (Upper Bound)	**0.87**		**0.73**		**0.61**	
Base Pythia 14m	0.86 ± 0.01	14M	$\mathbf{0.76 \pm 0.02}$	14M	0.52 ± 0.07	14M
Tabby MH Pythia 14m	0.82 ± 0.02	53M	$\mathbf{0.77 \pm 0.00}$	66M	0.54 ± 0.01	66M
Base Distilled-GPT2	$\mathbf{0.88 \pm 0.00}$	82M	0.73 ± 0.02	82M	0.53 ± 0.10	82M
Tabby MH Distilled-GPT2	$\mathbf{0.89 \pm 0.02}$	310M	0.73 ± 0.01	390M	$\mathbf{0.61 \pm 0.01}$	390M
Base GPT2	$\mathbf{0.89 \pm 0.01}$	120M	$\mathbf{0.76 \pm 0.01}$	120M	0.60 ± 0.00	120M
Tabby MH GPT2	$\mathbf{0.87 \pm 0.01}$	360M	0.73 ± 0.03	430M	0.53 ± 0.11	430M
Base Pythia 160M	$\mathbf{0.87 \pm 0.01}$	160M	$\mathbf{0.75 \pm 0.04}$	160M	0.52 ± 0.11	160M
Tabby MH Pythia 160M	0.86 ± 0.00	390M	0.73 ± 0.02	470M	0.54 ± 0.02	470M
Base Pythia 410M	0.86 ± 0.02	410M	$\mathbf{0.74 \pm 0.03}$	410M	0.28 ± 0.40	410M
Tabby MH Pythia 410M	$\mathbf{0.88 \pm 0.03}$	710M	0.72 ± 0.05	820M	0.54 ± 0.02	820M
Base Llama 3.2 1B	0.82 ± 0.01	1.2B	0.73 ± 0.01	1.2B	0.29 ± 0.01	1.2B
Tabby MH Llama 3.2 1B	0.84 ± 0.02	2.8B	0.68 ± 0.09	3.3B	0.18 ± 0.26	3.3B
Base Llama 3.1 8B	0.84 ± 0.01	8.0B	$\mathbf{0.75 \pm 0.01}$	8.0B	0.35 ± 0.01	8.0B
Tabby MH Llama 3.1 8B	0.86 ± 0.03	11B	0.72 ± 0.01	12B	0.30 ± 0.01	12B

Table 16: Results using GReaT and Tabula training for all three datasets of the experiment in Section 3.2, which compares non-Tabby and Tabby MH models across base LLMs of varying sizes.

	Travel		Diabetes		House	
	MLE (↑)	**Params**	**MLE (↑)**	**Params**	**MLE (↑)**	**Params**
Non-Synthetic (Upper Bound)	**0.87**		**0.73**		**0.61**	
Base Pythia 14m	0.81 ± 0.00	14M	0.60 ± 0.04	14M	0.46 ± 0.06	14M
Tabby MH Pythia 14m	0.81 ± 0.00	53M	0.67 ± 0.01	66M	0.51 ± 0.03	66M
Base Distilled-GPT2	0.86 ± 0.00	82M	0.62 ± 0.00	82M	0.57 ± 0.00	82M
Tabby MH Distilled-GPT2	0.84 ± 0.00	310M	0.70 ± 0.06	390M	0.56 ± 0.01	390M
Base GPT2	0.85 ± 0.02	120M	0.64 ± 0.02	120M	0.55 ± 0.00	120M
Tabby MH GPT2	$\mathbf{0.87 \pm 0.03}$	360M	$\mathbf{0.74 \pm 0.03}$	430M	0.58 ± 0.01	430M
Base Pythia 160M	0.81 ± 0.01	160M	0.70 ± 0.01	160M	0.00 ± 0.00	160M
Tabby MH Pythia 160M	0.82 ± 0.02	390M	0.73 ± 0.03	470M	0.54 ± 0.02	470M
Base Pythia 410M	0.85 ± 0.01	410M	0.73 ± 0.03	410M	0.53 ± 0.02	410M
Tabby MH Pythia 410M	0.83 ± 0.01	710M	$\mathbf{0.74 \pm 0.04}$	820M	0.58 ± 0.01	820M
Base Llama 3.2 1B	0.82 ± 0.01	1.2B	0.70 ± 0.08	1.2B	0.53 ± 0.01	1.2B
Tabby MH Llama 3.2 1B	0.78 ± 0.03	2.8B	0.71 ± 0.03	3.3B	0.43 ± 0.08	3.3B
Base Llama 3.1 8B	0.78 ± 0.04	8.0B	0.67 ± 0.01	8.0B	0.53 ± 0.01	8.0B
Tabby MH Llama 3.1 8B	0.83 ± 0.03	11B	0.73 ± 0.02	12B	0.45 ± 0.00	12B

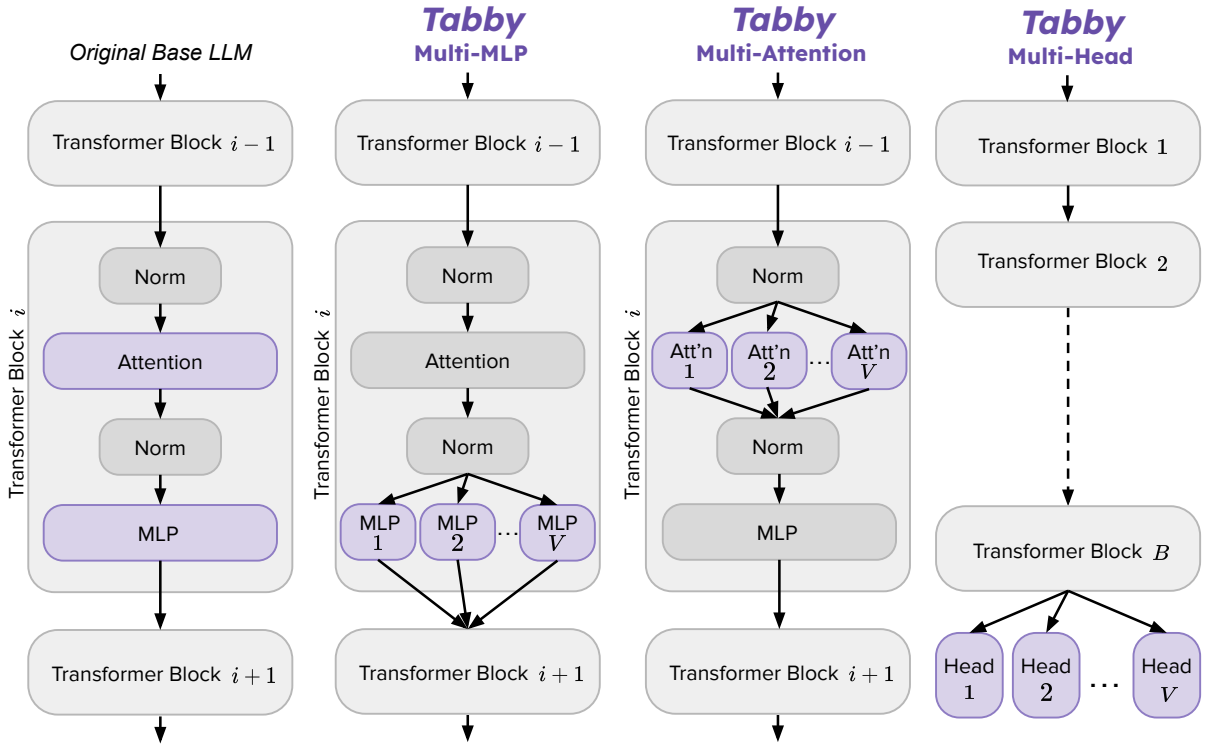

Figure 6: An overview of the Tabby MH modifications that can occur inside the LLM transformer blocks. Left to right: an original, non-Tabby LLM, a Tabby LLM with MoE MLP block, a Tabby LLM with MoE attention block, and a Tabby LLM with both MoE MLP and attention blocks. Tabby is very flexible, so as to accommodate a wide variety of tabular datasets.

E.4 Analysis from Tracking the Adaptation to Individual Columns

Individual column losses are shown in Figure 8. We observe that Occupancy is the largest contributor to the model's loss until step 32000. While Median Income's loss is initially the second-lowest, it improves little throughout the training process and exhibits the highest loss of all columns at the end of training. Additionally, convergence occurs across most columns around step 40000.

These insights are useful in cases where the model struggles to learn some columns more than others. Such information may indicate a need for better preprocessing for a difficult column, or gathering more datapoints to demonstrate the 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.

E.5 Hyperparameters for All Experiments

We list the learning rates chosen for Section 3.1 in Table 17, Section 3.2 in Table 18 and Section 20 in Table 20. We select the learning rate that yields lowest training loss from the set $\{1e-3, 1e-4, 1e-6, 1e-8\}$. For non-LLM methods in our experiments, we use the hyperparameters recommended by their respective papers.

Table 17: Learning rates for LLM results presented in Section 3.1.

	Diabetes	**Travel**	**Adult**	**Abalone**	**Rainfall**	**House**
Plain Non-Tabby	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$
Plain Tabby MMLP	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$
Plain Tabby MH	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$
Plain Tabby MMLP+MH	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$
GReaT Non-Tabby	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$
GReaT *Tabby MMLP*	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$
GReaT *Tabby MH*	$1e-6$	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$
GReaT *Tabby MMLP+MH*	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$
GTT Non-Tabby	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$
GTT *Tabby MMLP*	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$
GTT *Tabby MH*	$1e-6$	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$
GTT *Tabby MMLP+MH*	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$	$1e-4$

Table 18: Learning rates for *Plain-trained* LLMs of varying sizes in Section 3.2.

	Plain Training		
	Travel	**Diabetes**	**House**
Base Pythia 14M	$1e-4$	$1e-4$	$1e-4$
Tabby MH Pythia 14M	$1e-6$	$1e-4$	$1e-4$
Base Distilled-GPT2	$1e-4$	$1e-4$	$1e-4$
Tabby MH Distilled-GPT2	$1e-4$	$1e-4$	$1e-4$
Base GPT2	$1e-4$	$1e-4$	$1e-4$
Tabby MH GPT2	$1e-4$	$1e-4$	$1e-4$
Base Pythia 160M	$1e-4$	$1e-4$	$1e-6$
Tabby MH Pythia 160M	$1e-4$	$1e-4$	$1e-4$
Base Pythia 410M	$1e-6$	$1e-4$	$1e-6$
Tabby MH Pythia 410M	$1e-6$	$1e-6$	$1e-4$
Base Llama 3.2 1B	$1e-6$	$1e-4$	$1e-6$
Tabby MH Llama 3.2 1B	$1e-6$	$1e-4$	$1e-6$
Base Llama 3.1 8B	$1e-6$	$1e-6$	$1e-6$
Tabby MH Llama 3.1 8B	$1e-6$	$1e-6$	$1e-6$

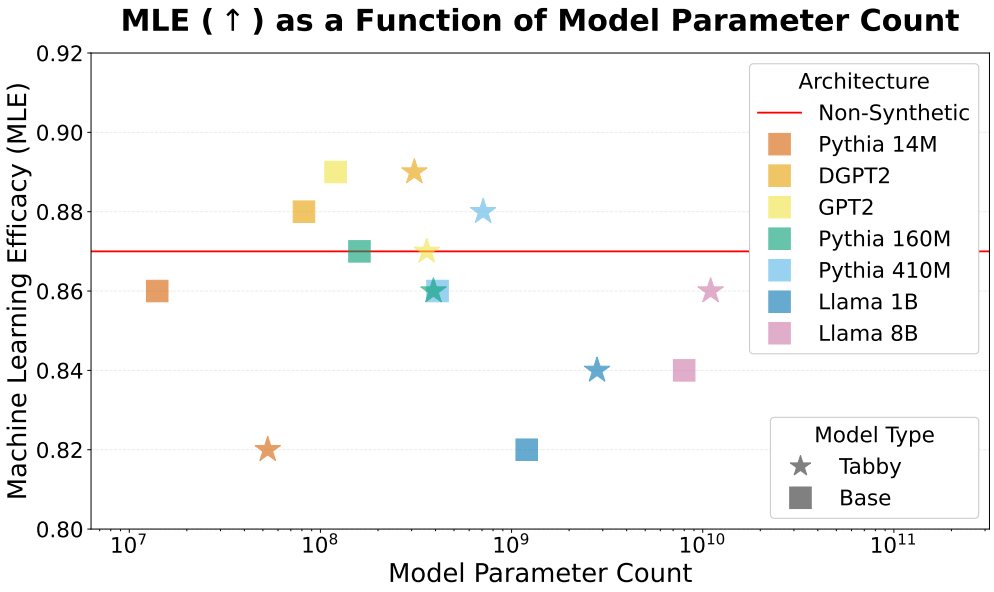

Figure 7: Machine Learning Efficacy (MLE) as a function of parameter count for 7 base LLMs, using Non-Tabby or Tabby MH architectures. Non-Tabby points displayed in blue; MH points in purple. Red line represents Non-Synthetic, upper-bound performance.

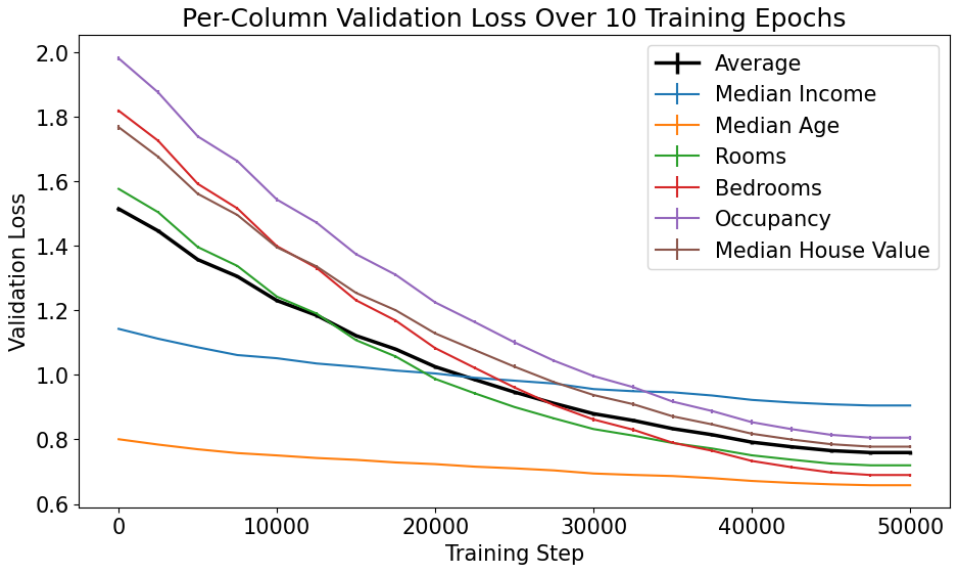

Figure 8: Per-column validation loss across 10 epochs of training Tabby MH Distilled-GPT2 on a subset of House, with average validation loss (black line). While the Occupancy column initially displays the highest loss, Median Income improves little throughout training and becomes the highest-loss column by step 32000.

Table 19: Learning rates for *GReaT (plus TapTap)-trained* LLMs of varying sizes in Section 3.2.

	GReaT + TapTap Training		
	Travel	**Diabetes**	**House**
Base Pythia 14M	$1e-4$	$1e-6$	$1e-4$
Tabby MH Pythia 14M	$1e-6$	$1e-6$	$1e-4$
Base Distilled-GPT2	$1e-4$	$1e-4$	$1e-4$
Tabby MH Distilled-GPT2	$1e-4$	$1e-4$	$1e-4$
Base GPT2	$1e-4$	$1e-4$	$1e-4$
Tabby MH GPT2	$1e-4$	$1e-4$	$1e-4$
Base Pythia 160M	$1e-4$	$1e-6$	$1e-4$
Tabby MH Pythia 160M	$1e-6$	$1e-6$	$1e-6$
Base Pythia 410M	$1e-6$	$1e-6$	$1e-6$
Tabby MH Pythia 410M	$1e-4$	$1e-6$	$1e-6$
Base Llama 3.2 1B	$1e-4$	$1e-4$	$1e-6$
Tabby MH Llama 3.2 1B	$1e-4$	$1e-4$	$1e-4$
Base Llama 3.1 8B	$1e-4$	$1e-4$	$1e-6$
Tabby MH Llama 3.1 8B	$1e-4$	$1e-4$	$1e-6$

Table 20: Learning rates for JSON Glaucoma (Manoj, 2024) experiment presented in Section 3.4.

	Glaucoma
Base DGPT2	$1e-4$
Tabby MH DGPT2	$1e-4$