Curated LLM: Synergy of LLMs and Data Curation for tabular augmentation in low-data regimes

Nabeel Seedat [*1] **Nicolas Huynh** [*1] **Boris van Breugel** [1] **Mihaela van der Schaar** [1]

Abstract

Machine Learning (ML) in low-data settings remains an underappreciated yet crucial problem. Hence, data augmentation methods to increase the sample size of datasets needed for ML are key to unlocking the transformative potential of ML in data-deprived regions and domains. Unfortunately, the limited training set constrains traditional tabular synthetic data generators in their ability to generate a large and diverse augmented dataset needed for ML tasks. To address this challenge, we introduce `CLLM`, which leverages the prior knowledge of Large Language Models (LLMs) for data augmentation in the low-data regime. However, not all the data generated by LLMs will improve downstream utility, as for any generative model. Consequently, we introduce a principled curation mechanism, leveraging learning dynamics, coupled with confidence and uncertainty metrics, to obtain a high-quality dataset. Empirically, on multiple real-world datasets, we demonstrate the superior performance of `CLLM` in the low-data regime compared to conventional generators. Additionally, we provide insights into the LLM generation and curation mechanism, shedding light on the features that enable them to output high-quality augmented datasets.

1. Introduction

No data, No Machine Learning. Machine learning (ML) has transformed numerous industries, but its wider adoption is hindered by a pervasive roadblock: insufficient data. Specifically, the use of ML algorithms presumes the availability and access to large datasets for training, be it labeled or unlabeled. Unfortunately, real-world domains are often data scarce: (i) in healthcare and finance, collecting annotations can be expensive or practically impossible; (ii) in developing and low-to-middle income countries (LMICs), digital infrastructure (such as electronic healthcare records (EHRs)) can be limited or nonexistent (Ade-Ibijola & Okonkwo, 2023; Asiedu et al., 2023; Owoyemi et al., 2020; Mollura et al., 2020; Alami et al., 2020; Ciecierski-Holmes et al., 2022) and (iii) within large datasets, there can be (ethnic) minorities that are underrepresented. This lack of data has serious consequences: to sideline these settings to the peripheries of ML advancements and prevent the development of accurate models. How can we build a reliable ML model in this *low-data regime*, with so few samples? Solving this problem is a major opportunity that would unlock the potential of ML across society, domains, and regions.

Aim. To address this important yet undervalued low-data problem, we aim to augment the *small labeled dataset* ($n < 100$) with synthetic samples. We focus on tabular data, as defining augmentations is non-trivial and can easily result in nonsensical or invalid samples. Moreover, tabular domains like healthcare are often where data scarcity is acute.

Related work. Data augmentation is a widely used and different approach to address data scarcity in tabular data contexts. Methods are either based on generative models (Ghosheh et al., 2023; Biswas et al., 2023; Wang & Pai, 2023; Machado et al., 2022; Tanaka & Aranha, 2019) such as GANs (Xu et al., 2019), VAEs (Xu et al., 2019), Normalizing Flows (Papamakarios et al., 2021), Score-based models (Kotelnikov et al., 2022; Kim et al., 2022), or alternatively traditional methods such as SMOTE (Chawla et al., 2002; Wang & Pai, 2023; Machado et al., 2022). However, in low-data regimes ($n < 100$), the training data may not describe the full data distribution well, despite it being i.i.d. draws. Consequently, this harms conventional methods since the augmented data may not be sufficiently diverse and accurate, restricting the generalizability of predictive models trained on such data. Recent work has shown the potential of fine-tuning Large Language Models (LLMs) for tabular data generation (Borisov et al., 2023). While LLMs offer some degree of prior knowledge, there are two challenges in our setting. First, it is computationally expensive to fine-tune LLMs, while needing specialized hardware —luxuries

[*]Equal contribution [1]University of Cambridge. Correspondence to: Nabeel Seedat <ns741@cam.ac.uk>, Nicolas Huynh <nvth2@cam.ac.uk>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

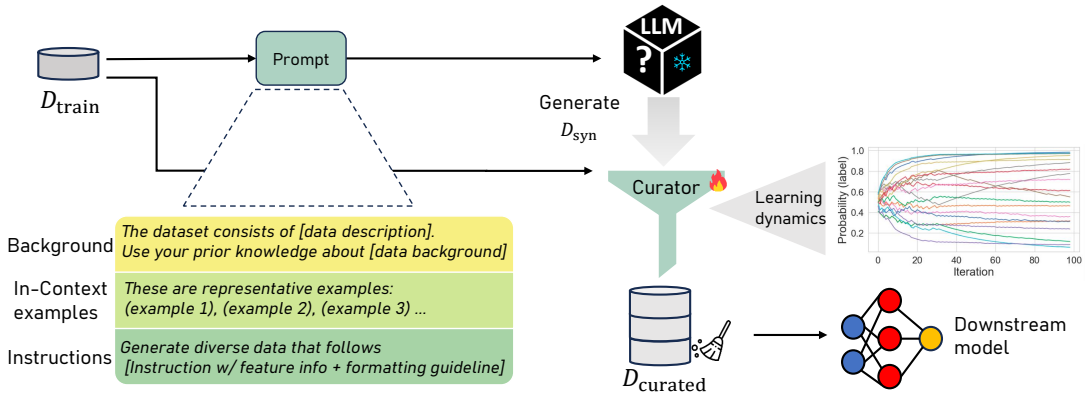

Figure 1: CLLM uses a small dataset D_{train} and a frozen black-box LLM to generate a larger synthetic set D_{syn}. The curator computes the learning dynamics of samples in D_{syn}, assessing samples based on their aleatoric uncertainty and predictive confidence, then curates D_{syn} with the goal that a downstream model trained on the curated D_{curated} will have improved performance.

often not available in LMICs, thereby limiting applicability in such settings. Second, fine-tuning often assumes a large number of samples. In our low-data setting, it could lead to overfitting and low-quality generated samples, and hence poor downstream models—as we show for (Borisov et al., 2023) in Sec. 3. Tangential to data augmentation, prior work has tackled data scarcity in the tabular setting via the lens of transfer learning or few-short learning, by using a knowledge graph (Margeloiu et al., 2022; Ruiz et al., 2023) (which might not be available) or a pretrained model (Levin et al., 2022; Jin & Ucar, 2023; Hegselmann et al., 2023). However, unlike data augmentation, these approaches are not flexible, as they tie the data customer to use a certain downstream predictor. We provide an extended discussion on this point in Appendix A.

Curated LLMs. To address the shortcomings of the aforementioned augmentation approaches, we propose Curated LLM (CLLM). First, CLLM leverages the in-context capabilities of LLMs for generation, thereby reducing the computational burden compared to fine-tuning. We also posit for the low-data regime; the diverse pretraining corpus of LLMs carries valuable prior knowledge, which may offer more diversity in their generation compared to other conventional tabular generators. *Of course, LLMs are not perfect*. Consequently, balancing the utility of LLMs against the risk of noisy, irrelevant data is important to ensure reliable downstream performance. Hence, this necessitates systematic assessment of the generated data. In fact, this issue is vital for *any* generative model.

This motivates the second key aspect of CLLM, i.e. a post-generation data curation mechanism. This addresses the *overlooked* aspect that not all of the synthetic samples are useful to downstream model performance, with some samples even harmful. We anchor our approach with ideas from learning theory that show the behavior of individual data samples during training, called learning dynamics, provides a salient signal about the value of samples to a learner (Arpit et al., 2017; Arora et al., 2019; Li et al., 2020). To provide intuition, samples with variable predictions might be considered ambiguous or other samples might never be learned correctly and could harm a model. In CLLM, we study the learning dynamics of the synthetic data samples, with respect to a model trained on the small real dataset. We then analyze these dynamics by computing two key metrics: confidence and aleatoric (data) uncertainty. These metrics form the basis for curating the synthetic samples. We then aim to enable a highly performant downstream model when trained on the curated dataset.

Contributions: CLLM is a novel data augmentation approach allying the strengths of LLMs with a robust data curation mechanism to improve data augmentation in the *low-data regime* ($n < 100$), bringing several contributions: ① **Improved performance:** we empirically demonstrate on 7 real-world datasets that CLLM enables superior downstream performance compared to 6 widely used tabular data generative models and data augmentation techniques. ② **Value of curation:** we show the *overlooked* aspect of synthetic data curation improves downstream performance across the generative models. This highlights the flexibility and broad utility of our curation mechanism for data augmentation. ③ **Insights:** we dissect the two aspects of CLLM (LLM and data curation) along a variety of dimensions, providing insights and understanding into why the approach is beneficial. We show the largest gains are for underrepresented subgroups and in low-data settings. These contributions, which address the data logjam, pave the way towards wider usage of ML across society, domains and regions.

2. CLLM: Synergy of LLM Generation and Data Curation

Set-up. Given feature space $\mathcal{X}$, and label space $\mathcal{Y} = \{1, ..., k\}$, we assume that we only have a small labeled dataset $D_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{n}$, with $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$ and $n < 100$ (low data setting). Assume D_{train} is drawn i.i.d. from the real distribution $p_R(X, Y)$. We also assume access to a pretrained LLM to generate samples. We denote the output distribution of the LLM as $p_\Phi(X, Y)$, with Φ containing parameters that we control (e.g., input prompts). Our goal is to generate a dataset to augment the small D_{train}, and subsequently use it to train a classifier $f : \mathcal{X} \to \mathcal{Y}$. Successful augmentation will provide a better classifier f, than if we had trained f on the small D_{train} itself. We measure downstream performance on a separate held-out dataset of real data, D_{test}.

Our Approach. To address this challenge, we introduce CLLM, an approach for data augmentation in low-data regimes. As shown in Figure 1, CLLM leverages LLMs to **generate** a synthetic dataset D_{syn} using a small dataset D_{train} (Sec. 2.1). It exploits the LLMs' prior knowledge via in-context learning (ICL) and contextual information. CLLM then **curates** D_{syn} by analyzing the learning dynamics of samples in D_{syn} based on predictive confidence and aleatoric (data) uncertainty. These metrics are obtained by training a supervised model on D_{train}. We leverage them to define a curated dataset D_{curated}, which is used to train a downstream classifier (Sec. 2.2).

In each sub-section we describe and motivate the design of the different aspects of CLLM (LLM and curation mechanism). Furthermore, we provide insights and understanding into their role in improving data utility, which we later quantify on multiple real-world datasets in Sec. 3.

2.1. Data generation with LLMs based on a small D_{train}

As outlined in Sec. 1, in the low-data regime, conventional tabular generative models (e.g. CTGAN, TVAE) are constrained by the limited D_{train} and may not generate sufficiently diverse and/or accurate synthetic data. To address this, we propose to leverage LLMs, building on their large-scale pretraining. We first outline the appealing properties of LLMs for tabular data generation when we have very few samples, then describe design choices to exploit these.

- **Prior knowledge.** LLMs are pretrained with a vast corpus of information (Chowdhery et al., 2022; Singhal et al., 2023). When prompted to generate samples with limited real data, LLMs can leverage this encoded prior information about similar problems and feature-label relationships to enhance both accuracy and diversity of generation.
- **Contextual understanding.** LLMs can process background and contextual information about the problem via natural language (Yang et al., 2023). For example, a high-level description of the task, features and their meanings can be conveniently described through natural language. Such information is unavailable to conventional generators that only utilize numerical examples.
- **Few-shot capabilities.** LLMs have demonstrated proficiency in generalizing to tasks with just a few examples (Brown et al., 2020; Wei et al., 2023; Mirchandani et al., 2023). In the context of generation, we envision the idea of in-context generation using limited real examples.

To benefit from these capabilities, we craft the LLM prompt with three different parts (see Fig. 1): (1) *Background*: text description of the dataset and task (e.g. predict Covid mortality). Additionally, we include a description of what each feature means, explicitly prompting the LLM to use prior knowledge about these features. (2) *Examples*: we serialize the samples in D_{train} as example demonstrations and provide both the features and the label in text format. (3) *Instructions*: To generate a synthetic dataset D_{syn}, we instruct the LLM to leverage the contextual information and provided examples as an i.i.d. draw from the distribution. We instruct the LLM to identify structural and feature-label relationships in the data and generate diverse data following the structure and format of the provided examples. We provide more details on the prompts in Appendix B.

Motivation for a frozen LLM. Using a frozen black-box LLM (e.g. GPT-4 or GPT-3.5) is computationally cheaper and requires less specialized hardware (i.e. GPUs) compared to fine-tuning. This relates to settings described in Sec. 1, such as LMICs, where we may not have the computational resources to fine-tune an LLM. Even in settings where fine-tuning is possible, we show empirically in Sec. 3 that LLM fine-tuning (e.g. GReaT baseline) is suboptimal in low-data settings ($n < 100$) compared to providing in-context examples coupled with curation.

Dissecting the LLM's generative features. We now investigate various dimensions to understand and illustrate empirically the appealing features of LLMs as data generators in the low-data regime, and how our design choices unlock them. We use the Brazil *Covid-19* dataset (Baqui et al., 2020) as a running example and focus on GPT-4 as the LLM.

► **Extrapolation to unseen regions of the manifold.** We compare the samples generated by GPT-4 to TVAE, a widely used tabular data generator. We consider D_{oracle}, a held-out dataset from the same distribution as D_{train}, such that $|D_{\text{oracle}}| \gg |D_{\text{train}}|$, thereby providing an approximation for the true manifold. The t-SNE plots in Fig. 2 shows, when D_{train} is very small ($n = 20$ samples), that its samples do

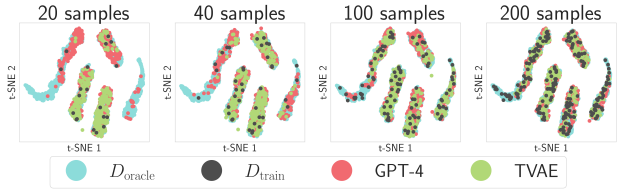

Figure 2: GPT-4 is able to extrapolate to regions of the oracle (true manifold) even where there is no training data covering them, as can be seen by the overlap with the turquoise dots, with the effect more pronounced when D_{train} is small

not cover all regions of D_{oracle}. For example, D_{train} does not contain samples from specific demographic subgroups (e.g. people with age 40 or below). As expected, TVAE only generates samples constrained by the limited D_{train}. In contrast, GPT-4 is able to extrapolate and generate samples even in unseen regions of D_{train}, thereby better covering D_{oracle}. This stems from its *contextual understanding* of the features, unlocking the use of its *prior knowledge*. It leads to better coverage in the low-data regime, consequently aiding in superior downstream performance, as shown in Table 3. As n increases ($\geq$ 100), D_{train} provides better coverage, which naturally benefits both GPT-4 and TVAE. Overall, this result shows how prior knowledge encoded in LLMs addresses shortcomings of conventional generative approaches (e.g. TVAE) in the low-data regime.

► **GPT-4 benefits underrepresented groups the most.** Having illustrated the extrapolation capabilities of GPT-4, we now ask: *where does augmentation benefit downstream performance the most?* We evaluate performance gains for different demographic subgroups, such as age groups and ethnic groups (Amarela, Prada). Fig. 3 shows the performance gain obtained by training a classifier on data generated by GPT-4 compared to training on the small D_{train}. The greatest gains, on average, are for subgroups for which we have *no data* in D_{train}, yet GPT-4 can extrapolate and generate samples for these subgroups. This further validates the rationale of extrapolation via prior knowledge as a key source of gain for GPT-4.

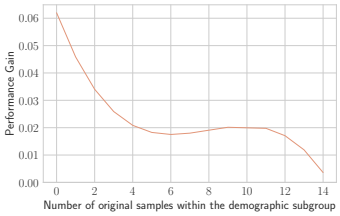

Figure 3: Subgroups with fewest samples in D_{train} benefit the most from data augmentation, on average.

Table 1 shows fine-grained results (across 10 different seeds) for the 5 subgroups that benefit the most from data augmentation, which are small-sized demographic subgroups. This finding has real-world implications for equity, showing we can improve performance for underrepresented subgroups even when we lack data or collecting data is difficult/costly.

Table 1: Deep dive into the top 5 demographic subgroups in the Covid dataset with the largest gains, across 10 seeds, for $|D_{\text{train}}| = 20$. GPT-4 improves performance on the smallest groups.

Subgroup	n_{samples} in D_{train} (min - max)	Avg. Acc. Gain v. D_{train} GPT-4	TVAE
Age_40	0-6	**6.38** +- 2.09	-3.37 +- 2.86
Liver	0-1	**3.85** +- 3.37	-13.1 +- 3.38
Renal	0-3	**4.52** +- 2.01	-18.0 +- 3.22
Amarela	0-1	**8.71** +- 1.40	-2.03 +- 2.88
Parda	3-11	**5.07** +- 1.50	-6.57 +- 1.61

► **Importance of contextual information in the prompt.** A natural question is: *how important is the prompt to elicit the prior knowledge of the LLM?* We explore two variants: (1) *Prompt w/ context*: provides contextual information including background about the dataset, feature names and descriptions (our approach) and (2) *Prompt w/ no context*: only provides the numerical in-context examples (ablation). Fig. 4 qualitatively shows that not including contextual knowledge in the prompt gives lower coverage of D_{oracle} with less extrapolation beyond D_{train}. We quantify this in Table 2 using Precision (Quality) and Recall (Diversity) metrics (Sajjadi et al., 2018), as well as Utility (Downstream performance). GPT-4 *with contextual information* has superior precision and recall in the low data setting. Furthermore, we show that *the lack* of contextual information in the prompt significantly harms the precision (quality) of the data even compared to TVAE. This highlights that LLMs need guidance, as we are only able to get the extrapolation and performance benefits by including contextual information, further motivating our design choices in the prompt. We conduct the same experiment with the Compas dataset in Appendix C.3.

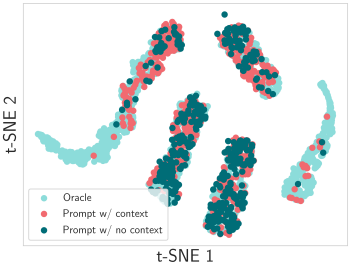

Figure 4: Contextual information in the prompt is important for extrapolation.

2.2. Data curation with learning dynamics

When prompted with Φ (which contains the in-context samples of D_{train}), the LLM generates samples from a distri-

Table 2: Including contextual information in the prompt improves precision (P), recall (R), and utility (U) in low-sample settings (results shown for the Covid dataset).

n_{samples} in D_{train}	GPT-4 w/ context			GPT-4 no context			TVAE		
	P	R	U	P	R	U	P	R	U
20	$\mathbf{0.41}_{(\mathbf{0.04})}$	$\mathbf{0.87}_{(\mathbf{0.03})}$	$\mathbf{0.74}_{(\mathbf{0.01})}$	$0.13_{(0.0)}$	$0.82_{(0.01)}$	$0.66_{(0.01)}$	$0.33_{(0.07)}$	$0.50_{(0.03)}$	$0.59_{(0.02)}$
40	$\mathbf{0.40}_{(\mathbf{0.01})}$	$\mathbf{0.91}_{(\mathbf{0.01})}$	$\mathbf{0.76}_{(\mathbf{0.0})}$	$0.11_{(0.0)}$	$0.89_{(0.0)}$	$0.69_{(0.0)}$	$0.27_{(0.01)}$	$0.68_{(0.01)}$	$0.62_{(0.03)}$
100	$\mathbf{0.42}_{(\mathbf{0.01})}$	$0.86_{(0.02)}$	$\mathbf{0.75}_{(\mathbf{0.01})}$	$0.11_{(0.01)}$	$\mathbf{0.90}_{(\mathbf{0.01})}$	$0.74_{(0.01)}$	$0.39_{(0.02)}$	$0.67_{(0.03)}$	$0.64_{(0.06)}$
200	$0.44_{(0.02)}$	$0.85_{(0.02)}$	$\mathbf{0.75}_{(\mathbf{0.0})}$	$0.08_{(0.01)}$	$\mathbf{0.90}_{(\mathbf{0.0})}$	$0.60_{(0.01)}$	$\mathbf{0.47}_{(\mathbf{0.0})}$	$0.73_{(0.01)}$	$0.65_{(0.02)}$

bution $p_\Phi(X, Y)$ that approximates $p_R(X, Y)$, implicitly exploiting its large-scale pretraining and few-shot capabilities. LLMs are of course not perfect and could generate noisy samples, hence this distribution may be inaccurate [1]. To make this distribution more relevant to the downstream task, we include a data curation mechanism. Specifically, we focus on the noisy feature-label relationship $p_\Phi(Y|X)$, for which we expect $p_\Phi(Y|X) \neq p_R(Y|X)$ given the small size of D_{train}. This motivates us to curate D_{syn} and discard likely mislabeled samples.

We anchor our approach with ideas from learning theory that show that the behavior of individual samples during model training (called *learning dynamics*) contains signal about the nature of the samples themselves (Arpit et al., 2017; Arora et al., 2019; Li et al., 2020). Some samples are easily and confidently predicted over different model checkpoints, whereas other samples might be challenging (e.g. due to mislabeling) and hence might be incorrectly predicted for the given label. Consequently, we operationalize *learning dynamics* as the basis of our proposed curation mechanism. Specifically, we analyze samples in D_{syn} by studying their learning dynamics computed with a classifier trained on D_{train}. We then categorize and filter samples in D_{syn}, and produce a curated dataset $D_{\text{curated}} \subset D_{\text{syn}}$.

Learning dynamics. We now formalize how we compute learning dynamics for individual samples. Assume that a classifier f is trained in an iterative scheme (e.g. neural networks or XGBoost trained over iterations) on D_{train}, which makes it possible to analyze the learning dynamics of samples in D_{syn} over these iterations. The classifier f should be at least as flexible as the model that the practitioner intends to use for the downstream task. f is trained from scratch on D_{train} and goes through $e \in [E]$ different checkpoints leading to the set $\mathcal{F} = \{f_1, f_2, \ldots, f_E\}$, such that f_e is the classifier at the e-th checkpoint. Let $[f_e(x)]_y$ denote the predicted probability for class y and sample x. Our goal is to assess the learning dynamics of samples in D_{syn} over these E training checkpoints, while we train f on D_{train}. For this, we define H, a random variable following a uniform distribution $\mathcal{U}_\mathcal{F}$ over the set of checkpoints $\mathcal{F}$. Specifically, given $H = h$ and a sample (x, y), we define the correctness in the prediction of H as a binary random variable $\hat{Y}_\mathcal{F}(x, y)$ with the following conditional: $P(\hat{Y}_\mathcal{F}(x, y) = 1|H = h) = [h(x)]_y$ and $P(\hat{Y}_\mathcal{F}(x, y) = 0|H = h) = 1 - P(\hat{Y}_\mathcal{F}(x, y) = 1|H = h)$.

[1] We could finetune the model on the scarce D_{train} we have, but is likely to still lead to overfitting due to the extreme data scarcity and LLM parameter size.

Curation metrics. Equipped with a probabilistic interpretation of the predictions of a model, we now define two characterization metrics that we use for curation: (i) average confidence and (ii) aleatoric (data) uncertainty, inspired by (Kwon et al., 2020; Seedat et al., 2022a).

Definition 2.1 (Average confidence)**.** For any set of checkpoints $\mathcal{F} = \{f_1, ..., f_E\}$, the average confidence for a sample (x, y) is defined as the following marginal:

$$\begin{aligned}\bar{P}_\mathcal{F}(x, y) &:= P(\hat{Y}_\mathcal{F}(x, y) = 1) \\ &= \mathbb{E}_{H\sim\mathcal{U}_\mathcal{F}}[P(\hat{Y}_\mathcal{F}(x, y) = 1|H)] \\ &= \frac{1}{E}\sum_{e=1}^{E}[f_e(x)]_y\end{aligned}$$

Definition 2.2 (Aleatoric uncertainty)**.** For any set of checkpoints $\mathcal{F} = \{f_1, ..., f_E\}$, the aleatoric uncertainty for a sample (x, y) is defined as:

$$\begin{aligned}v_{al,\mathcal{F}}(x, y) &:= \mathbb{E}_{H\sim\mathcal{U}_\mathcal{F}}[Var(\hat{Y}_\mathcal{F}(x, y)|H)] \\ &= \frac{1}{E}\sum_{e=1}^{E}[f_e(x)]_y(1 - [f_e(x)]_y)\end{aligned}$$

Intuitively, for binary classification ($k = 2$), the aleatoric uncertainty for a sample x is maximized when $[f_e(x)]_y = \frac{1}{2}$ for all checkpoints f_e, akin to random guessing. Recall aleatoric uncertainty captures the inherent data uncertainty, hence is a principled way to capture issues such as mislabeling. This contrasts epistemic uncertainty, which is model-dependent and can be reduced simply by increasing model parameterization (Hüllermeier & Waegeman, 2021).

Having defined sample-wise confidence and aleatoric uncertainty, we categorize samples in D_{syn} as *Selected* or *Discarded*: for a sample (x, y), a set of training checkpoints $\mathcal{F}$, and two thresholds τ_{conf} and τ_{al}, we define the category $c(x, y, \mathcal{F})$ as *Discarded* if $\bar{P}_\mathcal{F}(x, y) < \tau_{\text{conf}}$ and $v_{al,\mathcal{F}}(x, y) < \tau_{\text{al}}$, and *Selected* otherwise.

Hence, a *Discarded* sample is one for which we have a very low confidence in predicting its associated label whereas we also have low inherent data uncertainty. Finally, given a function f associated with the set of checkpoints $\mathcal{F}$, we define the curated set $D_{\text{curated}} = \{(x, y)|(x, y) \in D_{\text{syn}}, c(x, y, \mathcal{F}) = \textit{Selected}\}$. We also define $D_{\text{discarded}} = D_{\text{syn}} \setminus D_{\text{curated}}$.

To summarize, the objective of the curation step is that training on the curated synthetic data leads to a better clas-

sifier $f_{D_{\text{curated}}}$ for the downstream task, compared to training on the uncurated synthetic data, i.e. $M(f_{D_{\text{curated}}}) > M(f_{\mathcal{D}_{\text{syn}}})$, where M is a performance measure (for example accuracy). In Sec. 3, we empirically show how performance on this curated dataset is superior both for LLM generated data, as well as other classes of generative models.

Dissecting the role of curation. We now empirically demonstrate the role of curation in correcting the noisy feature-label relationship present in D_{syn}, highlighting two insights:
(i) curation discards samples which are atypical in their label with respect to their neighbors in D_{syn}
(ii) discarded samples can be considered "mislabeled", and we quantify their atypicality using a large held-out dataset D_{oracle}.

► **Discarded samples conflict on the label with their neighbors in D_{syn}.** We audit every synthetic sample (x, y) generated by GPT-4 (across 7 datasets) and compute the proportion of its k nearest neighbors in D_{syn} which share the same label y. The agreement with the neighbors assesses the typicality of a sample's y given x, where naturally lower agreement is linked to mislabeling, which we aim to detect via curation. Taking k = 10, we obtain an average agreement of a_{curated} = **0.74** for D_{curated}, compared to $a_{\text{discarded}}$ = **0.58** for $D_{\text{discarded}}$. This shows that the samples removed by our curation mechanism are those which, despite having similar features x, do not agree with the labels of their surrounding neighbors. This corroborates ideas in (Ashmore et al., 2021) of how proximity violations are useful to guide remedial action to improve models. Not removing these mislabeled samples injects noise into the downstream classifier, thus reducing performance.

► **Assessing discarded samples with D_{oracle}.** Ideally, the samples we select should better align with the true feature-label distribution. Since we don't have access to this distribution explicitly, we compute a proxy for $\eta(x) = \arg\max_y p(Y = y|X = x)$, which we call $\hat{\eta}$. It is obtained by training a classifier on a held-out dataset D_{oracle}—the same size as D_{test} and an order of magnitude larger than D_{train}. For each synthetic method, we then report the accuracy of $\hat{\eta}$ on both the curated D_{curated} and discarded $D_{\text{discarded}}$ datasets —see Fig. 5.

We highlight two key observations. First, the curated datasets, for all the generative models, exhibit a higher agreement with the proxy $\hat{\eta}$ than the discarded datasets. This aligns with the desideratum of only keeping samples that exhibit the correct feature-label relationships. This provides a rationale for why curation helps improve discriminative performance, as samples in D_{curated} are much more likely to have the correct feature-label relationship.

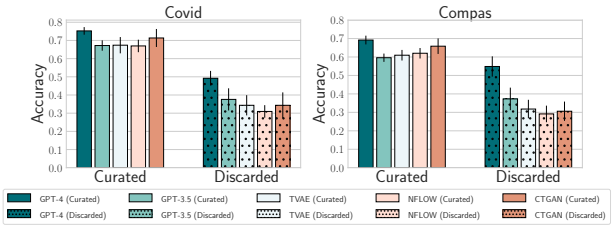

Figure 5: $\hat{\eta}$ aligns more with D_{curated} than $D_{\text{discarded}}$ for each generative model: the curation step keeps high quality samples tailored to the downstream task.

Second, GPT-4 has a higher agreement with $\hat{\eta}$ on $D_{\text{discarded}}$, compared to other generators. This illustrates that GPT-4's prior knowledge enables it to better capture the distribution $p(Y|X = x)$. Note that generative baselines (e.g. TVAE) model the joint $p(X, Y)$, *without any context* of which is the set of features and which is the label. In contrast, we can define in the LLM prompt which column is the target Y, allowing the LLM to better capture the feature-label relationships. This complements the findings from Fig. 2, which showed that GPT-4 extrapolates to unseen regions of the feature manifold, captured by the support of $p(X)$.

3. Curated LLMs for Better Data Augmentation

We now perform an end-to-end quantitative evaluation of `CLLM` [2] across multiple real-world datasets, for **downstream utility**, demonstrating the value of allying the generative capabilities of LLMs with our curation mechanism.

Sec. 3.1 compares the downstream performance of models when trained on uncurated vs curated data for a variety of augmentation approaches. Having evaluated `CLLM` on a range of datasets, we also demonstrate how we can leverage information extracted during curation to characterize datasets via a **hardness proxy**. Sec. 3.2 illustrates how our characterization of samples during the curation step can help to flag synthesized datasets (e.g via the LLM) which, if used for training, will result in poor downstream performance.

Experimental setup. We compare `CLLM` (with GPT-4 (OpenAI, 2023) and GPT-3.5 (Brown et al., 2020)) against a variety of baselines for tabular data generation and augmentation: CTGAN (Xu et al., 2019), TVAE (Xu et al., 2019), Normalizing Flows (Papamakarios et al., 2021), TabDDPM (Kotelnikov et al., 2022), SMOTE (Chawla et al., 2002) and GReaT (Borisov et al., 2023), which fine-tunes an LLM. We evaluate performance on 7 real-world datasets with different feature counts and representative of the diverse domains

[2] Code: https://github.com/seedatnabeel/CLLM or https://github.com/vanderschaarlab/CLLM

Table 3: AUC averaged over 4 downstream models on D_{test}. Curation improves performance for all methods across all sample sizes n, as indicated by ↑. CLLM w/ GPT-4 (Cur.) provides the strongest performance for both private/proprietary datasets and public datasets

	Real data		CLLM (OURS)				Baselines											
			GPT-4		GPT-3.5		CTGAN		TabDDPM		GReaT		NFLOW		SMOTE		TVAE	
Dataset	D_{oracle}	D_{train}	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.
covid (n=20)	74.41	68.50	73.78	**73.87** ↑	69.85	71.41 ↑	59.00	63.67 ↑	66.84	66.85 ↑	57.38	66.46 ↑	62.87	68.56 ↑	66.95	66.82	61.69	66.11 ↑
cutract (n=20)	72.23	70.12	71.15	**72.50** ↑	69.97	71.54 ↑	64.01	67.98 ↑	66.05	66.59 ↑	52.38	67.02 ↑	64.44	70.42 ↑	68.41	69.24 ↑	68.94	70.22 ↑
maggic (n=20)	67.41	57.13	60.70	**61.48** ↑	57.54	58.69 ↑	52.75	54.51 ↑	54.59	55.39 ↑	50.29	55.64 ↑	54.72	57.38 ↑	55.84	56.15 ↑	54.08	56.19 ↑
seer (n=20)	87.92	80.67	84.53	**84.82** ↑	83.34	83.71 ↑	74.34	78.73 ↑	80.59	80.60 ↑	47.57	74.43 ↑	76.06	79.98 ↑	79.23	80.02 ↑	74.53	78.73 ↑
compas (n=20)	67.51	63.11	**68.01**	67.91	62.07	64.43 ↑	55.67	62.56 ↑	57.67	60.87 ↑	53.33	63.59 ↑	59.49	64.62 ↑	61.06	61.59 ↑	58.30	62.58 ↑
adult (n=20)	84.17	77.45	50.39	71.48 ↑	49.23	72.37 ↑	72.23	76.86 ↑	74.35	75.04 ↑	67.00	77.25 ↑	67.46	76.48 ↑	73.75	73.67	73.20	76.90 ↑
drug (n=20)	77.81	70.84	75.08	**75.29** ↑	71.68	72.14 ↑	68.31	72.65 ↑	68.12	69.68 ↑	58.78	68.89 ↑	62.13	67.75 ↑	70.16	70.16	66.60	69.18 ↑
covid (n=40)	74.41	70.77	73.40	**73.95** ↑	70.42	71.93 ↑	63.63	68.46 ↑	70.50	70.44	56.50	68.68 ↑	66.41	70.48 ↑	68.66	68.44	61.03	67.35 ↑
cutract (n=40)	72.23	69.18	69.87	**71.72** ↑	68.47	69.56 ↑	63.01	67.87 ↑	65.63	67.27 ↑	54.39	68.44 ↑	61.40	67.98 ↑	67.86	67.95 ↑	59.79	66.62 ↑
maggic (n=40)	67.41	58.26	59.29	**60.77** ↑	57.50	59.15 ↑	55.00	56.78 ↑	55.24	56.94 ↑	48.81	56.64 ↑	54.68	58.58 ↑	57.40	57.44 ↑	55.04	57.33 ↑
seer (n=40)	87.92	82.93	84.29	**84.93** ↑	83.46	84.44 ↑	80.05	83.67 ↑	82.59	81.37	54.93	81.11 ↑	79.88	84.36 ↑	80.79	82.21 ↑	78.69	83.62 ↑
compas (n=40)	67.51	62.34	67.57	**67.85** ↑	61.34	62.84 ↑	56.29	61.02 ↑	58.85	60.11 ↑	58.88	64.37 ↑	58.61	63.54 ↑	60.83	60.95 ↑	55.94	61.04 ↑
adult (n=40)	84.17	79.44	48.31	73.82 ↑	49.21	74.27 ↑	71.82	79.11 ↑	71.51	77.99 ↑	66.77	78.81 ↑	71.13	79.71 ↑	77.90	78.84 ↑	72.58	**80.02** ↑
drug (n=40)	77.81	71.86	74.30	**75.79** ↑	71.33	72.76 ↑	69.46	72.74 ↑	71.08	73.07 ↑	64.89	73.64 ↑	62.51	70.97 ↑	69.23	69.78 ↑	65.22	70.30 ↑
covid (n=100)	74.41	71.57	73.77	**74.71** ↑	70.71	72.76 ↑	69.05	72.13 ↑	71.60	73.22 ↑	63.52	72.04 ↑	64.25	72.64 ↑	70.08	70.78 ↑	69.05	71.96 ↑
cutract (n=100)	72.23	70.96	70.20	**72.51** ↑	69.97	71.94 ↑	67.94	72.42 ↑	70.53	71.98 ↑	55.72	69.14 ↑	67.59	72.42 ↑	68.79	69.68 ↑	66.89	71.52 ↑
maggic (n=100)	67.41	59.65	58.98	**61.32** ↑	55.71	58.90 ↑	57.20	59.34 ↑	57.26	58.28 ↑	49.54	57.91 ↑	56.36	60.11 ↑	58.89	58.99 ↑	56.17	58.86 ↑
seer (n=100)	87.92	83.95	84.45	**85.37** ↑	83.92	85.08 ↑	81.60	85.14 ↑	83.04	84.83 ↑	70.32	83.83 ↑	81.16	85.03 ↑	81.82	82.49 ↑	78.88	84.50 ↑
compas (n=100)	67.51	62.56	68.02	**68.19** ↑	60.10	62.47 ↑	60.01	63.73 ↑	58.32	61.34 ↑	59.97	64.19 ↑	60.02	64.04 ↑	61.44	61.73 ↑	59.97	62.82 ↑
adult (n=100)	84.17	81.24	46.09	74.57 ↑	47.56	73.97 ↑	74.29	80.45 ↑	75.93	78.22 ↑	77.09	**81.66** ↑	70.70	81.04 ↑	80.56	81.10 ↑	74.04	80.23 ↑
drug (n=100)	77.81	73.58	76.24	**76.74** ↑	69.46	71.05 ↑	68.19	73.28 ↑	72.43	73.79 ↑	67.26	75.28 ↑	62.67	73.12 ↑	70.90	71.53 ↑	68.22	73.59 ↑
covid (n=200)	74.41	72.33	73.40	**74.62** ↑	70.70	73.12 ↑	71.07	73.89 ↑	72.47	74.44 ↑	65.55	73.07 ↑	65.04	72.90 ↑	71.68	71.87 ↑	67.89	72.38 ↑
cutract (n=200)	72.23	71.75	71.39	73.01 ↑	70.28	72.39 ↑	69.28	72.41 ↑	71.83	**74.03** ↑	66.66	72.49 ↑	68.77	73.16 ↑	70.23	70.80 ↑	66.61	71.87 ↑
maggic (n=200)	67.41	61.39	58.92	**61.41** ↑	57.33	60.16 ↑	58.48	61.33 ↑	56.26	57.20 ↑	50.74	59.60 ↑	55.95	60.75 ↑	60.73	60.78 ↑	57.18	60.23 ↑
seer (n=200)	87.92	84.63	84.39	85.56 ↑	83.48	84.80 ↑	82.04	85.34 ↑	84.39	**86.57** ↑	82.15	86.03 ↑	77.73	85.19 ↑	83.38	84.15 ↑	79.71	85.26 ↑
compas (n=200)	67.51	63.27	67.02	**68.15** ↑	60.48	63.39 ↑	60.58	64.32 ↑	60.60	63.52 ↑	61.11	65.08 ↑	56.58	63.60 ↑	61.99	62.80 ↑	60.15	63.99 ↑
adult (n=200)	84.17	82.12	40.96	75.84 ↑	49.89	76.11 ↑	78.18	82.32 ↑	81.66	83.17 ↑	80.06	**83.32** ↑	74.31	82.64 ↑	82.26	82.39 ↑	75.21	82.02 ↑
drug (n=200)	77.81	76.10	75.58	76.06 ↑	70.66	72.81 ↑	71.31	75.98 ↑	69.61	71.79 ↑	72.35	**77.41** ↑	65.25	75.26 ↑	74.38	74.78 ↑	68.39	74.33 ↑

where CLLM can have impact. For each dataset, we vary the number of samples available in D_{train}, repeating each experiment for 10 seeds.

While we do not know the exact makeup of the pretraining data for LLMs like GPT-4, there is the possibility that open-source data might be included. This poses the risk of memorization as the primary source of performance gain. To disentangle the role of memorization, we select 4 real-world medical datasets (Maggic (Pocock et al., 2013), Covid (Baqui et al., 2020), SEER (Duggan et al., 2016), CUTRACT (PCUK, 2019)) that require an authorization process to access, hence are unlikely to form part of the LLMs training corpus. We use common open-source datasets (Adult and Drug from the UCI repository (Asuncion & Newman, 2007) and Compas (Angwin et al., 2016)) that are highly reflective of data scarce domains. Further experimental details can be found in Appendix B.

3.1. Overall performance: downstream utility

We assess overall performance based on *Utility* of the augmented data, which we evaluate in terms of AUC on the real D_{test}, when using four different types of downstream models (see Appendix B). This setup mirrors the widely adopted Train-on-synthetic-Test-on-real (TSTR) (Esteban et al., 2017). Additionally, we compare the performance to training on the small D_{train}, as well as training on the large held-out D_{oracle}, the latter serving as an upper bound.

GPT-4 + Curation has best overall performance. Table 3 shows the performance of the proposed CLLM (GPT- 4 and GPT-3.5) vs baselines — both *with* and *without* our curation mechanism. We find that the GPT-4 + Curation variant of CLLM outperforms baselines in almost all settings (20/28). Interestingly, its performance is close to or even exceeds the performance of D_{oracle}. Table 4 further shows that GPT-4 + Curation ranks first on average vs all generative methods.

Sample size sensitivity. We now investigate the performance gains of CLLM as we vary the number of samples n in D_{train}, in Table 3 and Table 4. Performance improvements and high ranking across datasets for CLLM (GPT-4+Curation) are especially noticeable in the low-data regime (i.e. $n < 100$). In this regime, the limited size of D_{train} severely constrains the other baseline methods. In contrast, as illustrated in Sec. 2.1, CLLM can leverage GPT-4's prior knowledge to extrapolate beyond the small D_{train}, thereby improving downstream performance. As expected, the performance gap between CLLM and other methods decreases as the size of D_{train} grows (e.g. $n = 200$), where sufficient training data helps other generators achieve good performance. We further decouple the prior knowledge of the LLM and the number of in-context samples in Appendix C.1, showing the importance of the in-context samples to guide the LLM's generation.

Table 4: Average rank of approaches across the different datasets and seeds. CLLM w/ GPT-4 ranks first across all n and curation improves all the generative models.

Method	n=20	n=40	n=100	n=200
CLLM w/ GPT-4	**2.71 ± 1.44**	**2.14 ± 1.06**	**2.29 ± 1.19**	**3.29 ± 1.38**
GPT-4	3.86 ± 1.73	4.29 ± 1.83	6.00 ± 1.77	7.57 ± 1.65
CLLM w/ GPT-3.5	4.14 ± 0.94	4.14 ± 0.71	6.86 ± 1.24	7.57 ± 0.70
NFLOW (curated)	6.00 ± 1.21	4.71 ± 0.80	4.00 ± 0.57	4.71 ± 0.63
GPT-3.5	6.71 ± 1.52	7.29 ± 1.26	11.57 ± 0.94	12.57 ± 0.57
TVAE (curated)	7.14 ± 1.17	7.86 ± 1.30	6.43 ± 0.40	6.71 ± 0.52
SMOTE (curated)	7.71 ± 0.33	8.14 ± 0.91	7.71 ± 1.19	7.43 ± 1.07
SMOTE	7.86 ± 0.55	9.57 ± 0.80	9.57 ± 1.09	9.00 ± 1.03
TabDDPM (curated)	8.29 ± 0.98	8.00 ± 0.93	6.00 ± 0.95	5.14 ± 1.68
CTGAN (curated)	8.29 ± 1.42	7.14 ± 0.91	4.14 ± 0.62	3.71 ± 0.39
GReaT (curated)	8.57 ± 1.50	6.57 ± 1.21	6.29 ± 1.38	3.57 ± 0.92
TabDDPM	10.14 ± 1.19	9.86 ± 1.15	10.00 ± 1.03	10.29 ± 1.02
TVAE	12.14 ± 0.89	14.00 ± 0.70	13.71 ± 0.39	14.43 ± 0.40
NFLOW	12.86 ± 0.47	14.14 ± 0.37	14.00 ± 0.45	15.29 ± 0.33
CTGAN	13.86 ± 0.68	13.14 ± 0.47	12.86 ± 0.37	12.00 ± 0.53
GReaT	15.71 ± 0.26	15.00 ± 0.53	14.57 ± 1.03	12.71 ± 0.96

Curation generally helps all generative models. Our curation mechanism consistently benefits all generative models for the different n. It ensures that only high-quality samples are retained, which is crucial for good data augmentation and downstream performance and has been overlooked in previous works. This explains why the combination of the best generative model and curation (CLLM) gives the best results and highest rankings in the low-data regime (e.g. $n = 20$). In addition to GPT-4 and GPT-3.5, we show the versatility of our proposed curation mechanism to provide benefit with other open-source LLM backbones, including Mistral-7b (Jiang et al., 2023), LLAMA-2 (Touvron et al., 2023) and Mixtral (Jiang et al., 2024) (cf. Appendix C.2).

Performance benefits maintained for private and public datasets. One may hypothesize that the strong LLM (e.g. GPT-4) performance is explained by datasets being part of the LLMs' training corpus, hence possibly being memorized. We show in Table 3 that it is unlikely, as we retain strong performance for both open-source datasets, as well as private medical datasets which require authorization processes for access and are unlikely to be part of the LLM pretraining dataset.

Remark on ICL versus fine-tuning. Our results in Table 3 and Table 4 indicate that ICL is better than fine-tuning (GReaT baseline) in the low-data regime. This highlights the difficulty of fine-tuning in this regime, where it is easy to overfit to D_{train}. As we increase the number of samples, this baseline, coupled with curation, improves to the level of CLLM (GPT-4).

3.2. Hardness: a proxy signal to flag poor quality synthetic datasets

Having a systematic way to assess datasets generated by LLMs like GPT-4 is important because their black-box nature provides little control on their generation quality. This contrasts conventional generators for which training loss is an exploitable signal. Hence, we ask: could we have a signal to identify a potential problematic dataset generated by an LLM without an exhaustive manual review? For example, GPT-4 produced low-quality synthetic data for the Adult dataset (across the different sample sizes) resulting in poor downstream performance. While curation improves it, downstream performance is still suboptimal.

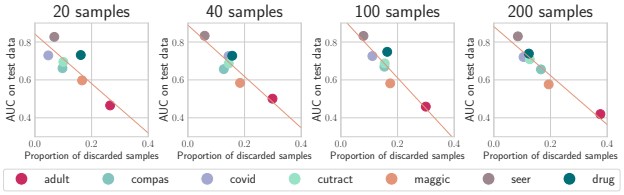

Figure 6: The proportion of discarded samples D_{syn} is a proxy for test performance. This negative linear relationship where each point is a synthetic dataset generated by GPT-4 (e.g. Adult, Covid, Compas) allows us to flag datasets that will lead to unreliable downstream performance.

Addressing this question is important, since datasets are rarely created by the ML model builder in real-world ML workflows, but rather by specialist data teams or data owners (Gebru et al., 2021; Sambasivan et al., 2021; Goncalves et al., 2020). Thus, having a signal to preemptively flag a potentially suboptimal generated dataset spares investment in both storing the subpar data and/or training a model likely to underperform on real data.

To address this, we posit that D_{syn} should intuitively be considered imperfect if curation discards many of its samples, since the number of discarded samples measures the quality of samples with respect to the small but gold-standard D_{train}.

Hence, we investigate the relationship between test performance (AUC) and the proportion of samples discarded by curation. Fig. 6, where each point is a synthetic dataset generated by GPT-4 (e.g. Adult, Compas), shows a strong negative linear relationship between these two quantities. This holds across the different n with slopes fairly stable around -1.4. This relationship corroborates the poor quality of the dataset generated by GPT-4 on the Adult dataset, providing a useful proxy that D_{syn} is unlikely to lead to good downstream performance.

4. Discussion

We introduce CLLM, an approach for data augmentation in the low-data setting. CLLM exploits the prior knowledge of LLMs along with our curation mechanism for improved downstream performance.

As empirically shown, CLLM outperforms traditional generative models—most noticeably on underrepresented subgroups, for which data augmentation is of utmost importance. CLLM is grounded in the ICL capability of LLMs. Further improvements may be achieved through different tuning and prompting of the LLM, as shown in different domains (Meng et al., 2023; Liu et al., 2023). Improving LLM tuning and prompting is beyond the scope of our work, but we regard this as a promising avenue for future work.

While the key contribution of this work is the curation of LLM outputs, overall downstream performance gains are still fundamentally tied to the LLM backbone. Specifically, a practical consideration is that using less parameterized LLMs leads to poorer uncurated data. That said, our curation mechanism naturally addresses this aspect and improves downstream performance (see Appendix C.2).

Finally, while CLLM addresses the data logjam issue, increasing access to ML across regions, domains and societies is also about more than just technology. We believe broader engagement and discussion with various stakeholders is crucial to responsibly expand ML access, thereby realizing the benefits of ML in an equitable way.

Impact statement

Data scarcity and computational limitations are deterrents for developing ML. These challenges should inspire cutting-edge ML research (De-Arteaga et al., 2018). We believe CLLM takes a step in this direction toward improving the use of ML in low-data settings, across *society* (e.g. underrepresented subgroups (Suresh & Guttag, 2021)), *domains* (e.g. healthcare (Alami et al., 2020; Owoyemi et al., 2020)) and *regions* (e.g. LMICs).

However, with that in mind, LLMs may make errors and may reflect or exacerbate societal biases that are present in their data (Li et al., 2023). Though the curation in CLLM improves synthetic data quality, it does not directly aim to remove biases. The quality and fairness of generated data should always be evaluated. We believe broader engagement and discussion with various stakeholders is required before methods like CLLM should be applied to real-world sensitive settings like healthcare and finance, as well as more research into LLM bias and potential mitigation strategies. We provide a more detailed discussion in Appendix C.12.

Additionally, in this work, we evaluate CLLM using multiple real-world datasets. The private datasets are *de-identified* and used in accordance with the guidance of the respective data providers. We follow recommendations to use the Azure OpenAI service when using GPT-4 and GPT-3.5 models, where via the agreement we ensure the medical data is not sent for human review or stored, hence respecting the guidelines given by the dataset providers.

Acknowledgments

The authors are grateful to Fergus Imrie, Andrew Rashbass and the anonymous ICML reviewers for their useful comments and feedback. Nabeel Seedat is supported by the Cystic Fibrosis Trust, Nicolas Huynh by Illumina, and Boris van Breugel by the Office of Naval Research UK. This work was supported by Microsoft's Accelerate Foundation Models Academic Research initiative.

References

Ade-Ibijola, A. and Okonkwo, C. Artificial intelligence in africa: Emerging challenges. In *Responsible AI in Africa: Challenges and Opportunities*, pp. 101–117. Springer International Publishing Cham, 2023.

Alami, H., Rivard, L., Lehoux, P., Hoffman, S. J., Cadeddu, S. B. M., Savoldelli, M., Samri, M. A., Ag Ahmed, M. A., Fleet, R., and Fortin, J.-P. Artificial intelligence in health care: laying the foundation for responsible, sustainable, and inclusive innovation in low-and middle-income countries. *Globalization and Health*, 16:1–6, 2020.

Angwin, J., Larson, J., Kirchner, L., and Mattu, S. Machine bias. *ProPublica: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing*, May 2016.

Arora, S., Du, S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pp. 322–332. PMLR, 2019.

Arpit, D., Jastrzkebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pp. 233–242. PMLR, 2017.

Ashmore, R., Calinescu, R., and Paterson, C. Assuring the machine learning lifecycle: Desiderata, methods, and challenges. *ACM Computing Surveys (CSUR)*, 54(5): 1–39, 2021.

Asiedu, M. N., Dieng, A., Oppong, A., Nagawa, M., Koyejo, S., and Heller, K. Globalizing fairness attributes in machine learning: A case study on health in africa. *arXiv preprint arXiv:2304.02190*, 2023.

Asuncion, A. and Newman, D. UCI machine learning repository, 2007.

Baqui, P., Bica, I., Marra, V., Ercole, A., and van Der Schaar, M. Ethnic and regional variations in hospital mortality from covid-19 in brazil: a cross-sectional observational study. *The Lancet Global Health*, 8(8):e1018–e1026, 2020.

Berk, R., Heidari, H., Jabbari, S., Joseph, M., Kearns, M., Morgenstern, J., Neel, S., and Roth, A. A convex framework for fair regression. *arXiv preprint arXiv:1706.02409*, 2017.

Biswas, A., Nasim, M., Imran, A., Sejuty, A. T., Fairooz, F., Puppala, S., and Talukder, S. Generative adversarial networks for data augmentation. *arXiv preprint arXiv:2306.02019*, 2023.

Borisov, V., Sessler, K., Leemann, T., Pawelczyk, M., and Kasneci, G. Language models are realistic tabular data generators. In *The Eleventh International Conference on Learning Representations*, 2023.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Calmon, F., Wei, D., Vinzamuri, B., Natesan Ramamurthy, K., and Varshney, K. R. Optimized pre-processing for discrimination prevention. *Advances in neural information processing systems*, 30, 2017.

Chapelle, O., Schölkopf, B., and Zien, A. (eds.). *Semi-Supervised Learning*. The MIT Press, 2006. ISBN 9780262033589. URL http://dblp.uni-trier.de/db/books/collections/CSZ2006.html.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Ciecierski-Holmes, T., Singh, R., Axt, M., Brenner, S., and Barteit, S. Artificial intelligence for strengthening healthcare systems in low-and middle-income countries: a systematic scoping review. *npj Digital Medicine*, 5(1): 162, 2022.

De-Arteaga, M., Herlands, W., Neill, D. B., and Dubrawski, A. Machine learning for the developing world. *ACM Transactions on Management Information Systems (TMIS)*, 9(2):1–14, 2018.

Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., and Sui, Z. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Duggan, M. A., Anderson, W. F., Altekruse, S., Penberthy, L., and Sherman, M. E. The surveillance, epidemiology and end results (SEER) program and pathology: towards strengthening the critical relationship. *The American Journal of Surgical Pathology*, 40(12):e94, 2016.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226, 2012.

Esteban, C., Hyland, S. L., and Rätsch, G. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*, 2017.

Farahani, A., Voghoei, S., Rasheed, K., and Arabnia, H. R. A brief review of domain adaptation. *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020*, pp. 877–894, 2021.

Fehrman, E., Muhammad, A. K., Mirkes, E. M., Egan, V., and Gorban, A. N. The five factor model of personality and evaluation of drug consumption risk. In *Data science: innovative developments in data analysis and clustering*, pp. 231–242. Springer, 2017.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 259–268, 2015.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Gallegos, I. O., Rossi, R. A., Barrow, J., Tanjim, M. M., Kim, S., Dernoncourt, F., Yu, T., Zhang, R., and Ahmed, N. K. Bias and fairness in large language models: A survey. *arXiv preprint arXiv:2309.00770*, 2023.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Iii, H. D., and Crawford, K. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

Ghosheh, G. O., Thwaites, C. L., and Zhu, T. Synthesizing electronic health records for predictive models in low-middle-income countries (lmics). *Biomedicines*, 11(6): 1749, 2023.

Goncalves, A., Ray, P., Soper, B., Stevens, J., Coyle, L., and Sales, A. P. Generation and evaluation of synthetic patient data. *BMC medical research methodology*, 20(1): 1–40, 2020.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35:507–520, 2022.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Hegselmann, S., Buendia, A., Lang, H., Agrawal, M., Jiang, X., and Sontag, D. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*, pp. 5549–5581. PMLR, 2023.

Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. Tabpfn: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2022.

Hospedales, T. M., Antoniou, A., Micaelli, P., and Storkey, A. J. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Hüllermeier, E. and Waegeman, W. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110:457–506, 2021.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., Casas, D. d. l., Hanna, E. B., Bressand, F., et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Jin, Q. and Ucar, T. Benchmarking tabular representation models in transfer learning settings. In *NeurIPS 2023 Second Table Representation Learning Workshop*, 2023.

Kim, J., Lee, C., and Park, N. Stasy: Score-based tabular data synthesis. In *The Eleventh International Conference on Learning Representations*, 2022.

Kirsch, A., Van Amersfoort, J., and Gal, Y. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32, 2019.

Kotelnikov, A., Baranchuk, D., Rubachev, I., and Babenko, A. Tabddpm: Modelling tabular data with diffusion models. *arXiv preprint arXiv:2209.15421*, 2022.

Kwon, Y., Won, J.-H., Kim, B. J., and Paik, M. C. Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics & Data Analysis*, 142: 106816, 2020.

Lemaître, G., Nogueira, F., and Aridas, C. K. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. URL http://jmlr.org/papers/v18/16-365.html.

Levin, R., Cherepanova, V., Schwarzschild, A., Bansal, A., Bruss, C. B., Goldstein, T., Wilson, A. G., and Goldblum, M. Transfer learning with deep tabular models. *arXiv preprint arXiv:2206.15306*, 2022.

Li, H., Moon, J. T., Purkayastha, S., Celi, L. A., Trivedi, H., and Gichoya, J. W. Ethics of large language models in medicine and medical research. *The Lancet Digital Health*, 5(6):e333–e335, 2023.

Li, Y., Ma, T., and Zhang, H. R. Learning over-parametrized two-layer neural networks beyond ntk. In *Conference on learning theory*, pp. 2613–2682. PMLR, 2020.

Liang, W., Tadesse, G. A., Ho, D., Fei-Fei, L., Zaharia, M., Zhang, C., and Zou, J. Advances, challenges and opportunities in creating data for trustworthy ai. *Nature Machine Intelligence*, 4(8):669–677, 2022.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., and Tang, J. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Machado, P., Fernandes, B., and Novais, P. Benchmarking data augmentation techniques for tabular data. In *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 104–112. Springer, 2022.

Margeloiu, A., Simidjievski, N., Lio, P., and Jamnik, M. Graph-conditioned mlp for high-dimensional tabular biomedical data. *arXiv preprint arXiv:2211.06302*, 2022.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021.

Meng, Y., Michalski, M., Huang, J., Zhang, Y., Abdelzaher, T., and Han, J. Tuning language models as training data generators for augmentation-enhanced few-shot learning. In *International Conference on Machine Learning*, pp. 24457–24477. PMLR, 2023.

Mirchandani, S., Xia, F., Florence, P., Ichter, B., Driess, D., Arenas, M. G., Rao, K., Sadigh, D., and Zeng, A. Large language models as general pattern machines. *arXiv preprint arXiv:2307.04721*, 2023.

Mollura, D. J., Culp, M. P., Pollack, E., Battino, G., Scheel, J. R., Mango, V. L., Elahi, A., Schweitzer, A., and Dako, F. Artificial intelligence in low-and middle-income countries: innovating global health radiology. *Radiology*, 297 (3):513–520, 2020.

Mussmann, S. and Liang, P. On the relationship between data efficiency and error for uncertainty sampling. In *International Conference on Machine Learning*, pp. 3674–3682. PMLR, 2018.

Nguyen, V.-L., Shaker, M. H., and Hüllermeier, E. How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, 111(1):89–122, 2022.

OpenAI. Gpt-4 technical report, 2023.

Owoyemi, A., Owoyemi, J., Osiyemi, A., and Boyd, A. Artificial intelligence for healthcare in africa. *Frontiers in Digital Health*, 2:6, 2020.

Pan, S. J. and Yang, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 2009.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

PCUK, P. C. U. Cutract. *https://prostatecanceruk.org*, 2019.

Penedo, G., Malartic, Q., Hesslow, D., Cojocaru, R., Cappelli, A., Alobeidli, H., Pannier, B., Almazrouei, E., and Launay, J. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*, 2023.

Pocock, S. J., Ariti, C. A., McMurray, J. J., Maggioni, A., Køber, L., Squire, I. B., Swedberg, K., Dobson, J., Poppe, K. K., Whalley, G. A., et al. Predicting survival in heart failure: a risk score based on 39 372 patients from 30 studies. *European Heart Journal*, 34(19):1404–1413, 2013.

Polyzotis, N. and Zaharia, M. What can data-centric ai learn from data and ml engineering? *arXiv preprint arXiv:2112.06439*, 2021.

Qian, Z., Cebere, B.-C., and van der Schaar, M. Synthcity: facilitating innovative use cases of synthetic data in different data modalities, 2023. URL https://arxiv.org/abs/2301.07573.

Ranathunga, S., Lee, E.-S. A., Prifti Skenduli, M., Shekhar, R., Alam, M., and Kaur, R. Neural machine translation for low-resource languages: A survey. *ACM Computing Surveys*, 55(11):1–37, 2023.

Ruiz, C., Ren, H., Huang, K., and Leskovec, J. Enabling tabular deep learning when d $\setminus \gg n$ with an auxiliary knowledge graph. *arXiv preprint arXiv:2306.04766*, 2023.

Sajjadi, M. S., Bachem, O., Lucic, M., Bousquet, O., and Gelly, S. Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31, 2018.

Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., and Aroyo, L. M. "everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2021.

Seedat, N., Crabbé, J., Bica, I., and van der Schaar, M. Data-iq: Characterizing subgroups with heterogeneous outcomes in tabular data. In *Advances in Neural Information Processing Systems*, 2022a.

Seedat, N., Imrie, F., and van der Schaar, M. Dc-check: A data-centric ai checklist to guide the development of reliable machine learning systems. *arXiv preprint arXiv:2211.05764*, 2022b.

Seedat, N., Crabbé, J., Qian, Z., and van der Schaar, M. Triage: Characterizing and auditing training data for improved regression. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a.

Seedat, N., Imrie, F., and van der Schaar, M. Dissecting sample hardness: Fine-grained analysis of hardness characterization methods. In *The Twelfth International Conference on Learning Representations*, 2023b.

Settles, B. Active learning literature survey. Technical report, University of Wisconsin-Madison, 2009.

Shwartz-Ziv, R. and Armon, A. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., et al. Large language models encode clinical knowledge. *Nature*, pp. 1–9, 2023.

Suresh, H. and Guttag, J. A framework for understanding sources of harm throughout the machine learning life cycle. In *Equity and access in algorithms, mechanisms, and optimization*, pp. 1–9. 2021.

Tanaka, F. H. K. D. S. and Aranha, C. Data augmentation using gans. *arXiv preprint arXiv:1904.09135*, 2019.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023.

van Engelen, J. E. and Hoos, H. H. A survey on semi-supervised learning. *Machine Learning*, 109:373–440, 2019.

Vanschoren, J., van Rijn, J. N., Bischl, B., and Torgo, L. Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. doi:10.1145/2641190.2641198. URL http://doi.acm.org/10.1145/2641190.2641198.

Wang, W. and Pai, T.-W. Enhancing small tabular clinical trial dataset through hybrid data augmentation: Combining smote and wcgan-gp. *Data*, 8(9):135, 2023.

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 2020.

Wei, J., Wei, J., Tay, Y., Tran, D., Webson, A., Lu, Y., Chen, X., Liu, H., Huang, D., Zhou, D., et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.

Whiteson, D. HIGGS. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5V312.

Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32, 2019.

Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q. V., Zhou, D., and Chen, X. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*, 2023.

Zha, D., Bhat, Z. P., Lai, K.-H., Yang, F., Jiang, Z., Zhong, S., and Hu, X. Data-centric artificial intelligence: A survey. *arXiv preprint arXiv:2303.10158*, 2023.

Zhang, B. H., Lemoine, B., and Mitchell, M. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 335–340, 2018.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

Appendix: Curated LLM: Synergy of LLMs and Data Curation for tabular augmentation in low-data regimes

Table of Contents

A. Extended Related Work

This paper primarily engages with the work on data augmentation when we have limited data, where our primary goal is synthetic data generation to augment the dataset. Generating synthetic datasets not only helps improve downstream performance, but it is also a flexible solution as it doesn't tie the data consumer to any particular downstream model. That said, beyond the major difference of synthetic data generation, for completeness we contrast our setting of learning from limited data with other seemingly similar settings and highlight their differences.

Contrasting learning w/ limited data vs other settings. The challenge of learning from limited data, while seemingly related to several other learning paradigms, presents distinct differences and unique intricacies that warrant dedicated study.

Transfer learning (Pan & Yang, 2009), *domain adaptation* (Farahani et al., 2021), and *few-shot learning* (Wang et al., 2020) employ additional data resources or rely on specific task-related assumptions to improve learning performance. These methods exploit large labeled data from a source domain, unlabeled data in a target domain, or leverage knowledge from related tasks respectively. For example, (Levin et al., 2022) and (Jin & Ucar, 2023) use models trained on labeled data from a source domain, while (Ruiz et al., 2023) and (Margeloiu et al., 2022) leverage knowledge-graphs. This is in contrast to our setting, considered of learning with limited data, which must function with whatever scarce labeled data it has, without making any assumptions about the availability of additional data or tasks.

Detailed contrast between `CLLM` and transfer learning / meta-learning / few-shot learning.

In addition to the above, we emphasize below three dimensions along which `CLLM` differs from the transfer learning and meta-learning literature (which permit to do few-shot learning). Specifically, we highlight three specific dimensions which explain why transfer learning and meta-learning generally cannot apply to the setting considered in `CLLM` . We provide empirical evidence on why the following dimensions are important (notably the point on the choice of downstream backbone). Specifically, we compare `CLLM` with TabPFN (Hollmann et al., 2022), a few-shot learning method designed for small tabular problems (see Appendix C.9).

1. *Access to external datasets*: our problem setting in `CLLM` assumes access to a single small training set D_{train}, without access to any external/additional datasets. This mirrors the unique characteristics of low-to-middle income countries (LMICs), where data scarcity may be pervasive, hence making it unrealistic to assume that external datasets are available to the practitioner. In contrast, transfer learning and meta-learning make more stringent assumptions on the data requirements. (i) *Transfer learning* (Definition 3 in (Zhuang et al., 2020)): one typically assumes access to at least one additional source dataset D_{source}, usually bigger than D_{train}. This external D_{source} can then be used to pretrain a model, which is adapted using D_{train}. (ii) *Meta-learning* (Hospedales et al., 2021): one assumes access to a set of m tasks, which define a set of source datasets $\{D^{(i)}_{\text{source}} \mid i \in [m]\}$. It is also worth noting that a common assumption in meta-learning is that the source datasets and D_{train} share the same feature space, restricting its applicability. To summarize, both these learning paradigms often rely on external data, an assumption not required in `CLLM` . We acknowledge that `CLLM` can be seen as a form of transfer, in its broad definition, since it uses prior knowledge, with the LLM. However, a key point is that it does not require external/additional datasets.

2. *Flexibility of the backbone model*: since `CLLM` is a data augmentation method for tabular data, it enables the practitioner to use any downstream model backbone, such as neural networks or tree-based methods (XGBoost, Random forest, CatBoost). This flexibility is important, as it allows the practitioner to choose the best-suited model for the task at hand. For example, one can use any tree-based method with `CLLM` , which is appealing given that tree-based methods are often preferred over neural networks for tabular data (Grinsztajn et al., 2022), (Shwartz-Ziv & Armon, 2022). In contrast, most approaches to transfer learning and meta-learning are not flexible as they traditionally require neural networks as backbone models. This stems from their methodology, where fine-tuning a pretrained model on D_{train} is a prevailing approach (Finn et al., 2017).

3. *Ease of use*: `CLLM` is distinct in that it is easy to use, without the need for costly or complex operations, such as fine-tuning large pretrained models or using them at inference time, which may be impossible in LMICs, due to the cost associated to these operations.

Active learning (Settles, 2009) and *semi-supervised learning* (van Engelen & Hoos, 2019; Chapelle et al., 2006) also operate under the premise of having access to plentiful unlabeled data and the capacity to interactively query labels. However, in our setting, considered learning with limited data does not inherently assume such capabilities, focusing instead on limited

labeled data only. Furthermore, active learning primarily focuses on the iterative process of selecting data samples that, when labeled, are expected to most significantly improve the model's performance. This selection is typically based on criteria such as uncertainty sampling which focuses on **epistemic uncertainty** (Mussmann & Liang, 2018; Houlsby et al., 2011; Kirsch et al., 2019; Nguyen et al., 2022). The primary objective is to minimize labeling effort while maximizing the model's learning efficiency. Additionally, active learning would aims to label instances based on epistemic uncertainty where the model struggles to make accurate predictions, yet the samples themselves are correct. In contrast, `CLLM` leverage training dynamics based on **aleatoric uncertainty** and confidence and is designed to discard samples that might jeopardize the downstream accuracy. These samples can be considered to have inherent issues or are erroneous, such as being "mislabeled". To summarize, in active learning, epistemic uncertainty is used to identify data points that, if labeled, would yield the most significant insights for model training. In our approach, they serve to identify and exclude/filter data points that could potentially deteriorate the model's performance.

Self-supervised learning (Liu et al., 2021) leverages large amounts of unlabeled data to learn useful representations for downstream tasks. However, in our setting, considered learning with limited data does not inherently assume such access to vast amounts of unlabeled data.

Data-centric AI. Ensuring high data quality is a critical but often overlooked problem in ML, where the focus is optimizing models (Sambasivan et al., 2021). Even when it is considered, the process of assessing datasets is adhoc or artisanal (Seedat et al., 2022b). However, the recent push of data-centric AI (Liang et al., 2022; Polyzotis & Zaharia, 2021; Zha et al., 2023; Seedat et al., 2023b) aims to develop systematic tools to curate existing datasets. Our work contributes to this nascent body of work (Seedat et al., 2023a) – presenting `CLLM`, which, to the best of our knowledge, is the first systematic data-centric framework looking at how we can tailor synthetic datasets (rather than real datasets) to downstream task use with data curation.

Why Data Augmentation? Data augmentation is a flexible approach to address the low-data regime. An alternative might be to resort to a pretrained black-box model for classification, which could be for example via in-context learning for classification (Dong et al., 2022). However, such a solution is inadequate for several reasons, many of which would prevent real-world utility (e.g. in LMICs):

► *Not economical over the long term:* While using an LLM like GPT for classification may seem attractive due to its few-shot capabilities, it is likely not economically viable in real-world settings, especially in LMICs. The reason is classifying each sample will incur a cost to call the LLM, hence scales linearly with the number of test samples. Over time, the cumulative cost of these calls will surpass the once-off fixed cost associated with generating data. With data augmentation, once the dataset is augmented, there are no additional deployment time costs associated with the LLM. Indeed, the downstream models e.g. a random forest or XGBoost have negligible inference costs.

► *Control, interpretability and auditability:* Relying on a large, pre-trained LLM as a black-box classifier raises several concerns. (1) we have no control over our downstream classifier and its architecture, (2) lack of interpretability and auditability of the LLM when issuing predictions. In contrast, training a downstream model on augmented data maintains the ability to understand and explain how the model is making decisions (e.g. feature importance). This is especially crucial in contexts where accountability, transparency, and validation of machine learning processes are paramount.

► *Independence and self-sufficiency:* Relying on third-party services for continuous classification means being dependent on their availability, pricing models, and potential changes in the LLM version. By augmenting data and training a downstream classifier on the augmented dataset, we ensure that there is no external dependencies such as increasing costs or reduced performance with LLM version updates.

► *Hardware and financial constraints:* Even if we opt for an open-source LLM (e.g. Falcon (Penedo et al., 2023) or LLaMA-2 (Touvron et al., 2023)), deploying and running it locally demands significant computational resources. Typically, these models require GPUs with high amounts of VRAM for optimal performance (e.g. needing around 40 GB hencing requiring an A100 GPU for Falcon-40b and LLaMA-2 65B). Such high-end GPUs are expensive, and are likely to be inaccessible in a LMIC setting. Furthermore, renting hardware by the hour can quickly become prohibitively expensive. Data augmentation, on the other hand, can often be performed on modest hardware, and once the augmented dataset is created, many classifiers can be trained without the need for high-end GPUs, making the entire process more financially accessible.

In conclusion, while large language models offer vast knowledge, for low-data settings in low-income countries, data augmentation provides a more cost-effective, controllable, and interpretable solution for building robust classifiers.

B. Experimental Details

We provide details on our datasets used, as well as, other experimental specifics including: generation, curation, downstream model, prompt template.

B.1. Datasets

We summarize the different datasets we use in this paper in Table 5. The datasets vary in number of samples, number of features and domain.

Table 5: Summary of the datasets used. * Denotes private/proprietary datasets.

Name	n samples	n features	Domain
Adult Income (Asuncion & Newman, 2007)	30k	12	Finance
Compas (Angwin et al., 2016)	5k	13	Criminal justice
*Covid-19 (Baqui et al., 2020)	7k	29	Healthcare/Medicine
*CUTRACT Prostate (PCUK, 2019)	2k	12	Healthcare/Medicine
Drug (Fehrman et al., 2017)	2k	27	Healthcare/Medicine
*MAGGIC (Pocock et al., 2013)	41k	29	Healthcare/Medicine
*SEER Prostate (Duggan et al., 2016)	20k	12	Healthcare/Medicine

The private datasets are de-identified and used in accordance with the guidance of the respective data providers. We follow recommendations to use the Azure OpenAI service when using GPT-4 and GPT-3.5 models, where via the agreement we ensure the medical data is not sent for human review or stored, hence respecting the guidelines given by the dataset providers.

We detail the dataset splits used in Sec. 3.1. For each dataset and number of samples $n \in \{20, 40, 100, 200\}$, we sample a training set D_{train} such that $|D_{\text{train}}| = n$, and each target class has the same number of samples. We then split the remaining samples into two non-overlapping datasets, D_{oracle} and D_{test}, which have the same cardinality. This procedure is repeated $n_{\text{seed}} = 10$ times, thus leading to different training and test sets. Note that the different generative models use the same D_{train} and D_{test} for a given seed.

Motivation for the choice of datasets.

1. **Open-source:** Adult, Drug and Compas are widely used open-source datasets used in the tabular data literature. Adult and Drug are both UCI datasets that have been used in many papers, while Compas is part of OpenML (Vanschoren et al., 2013). Our reason for selecting them is that, despite them being open-source, they are highly reflective of domains in which we might be unable to collect many samples — hence in reality would often be in a low-data regime.
2. **Private datasets:** We wanted to disentangle the possible role of memorization in the strong performance of the LLM. To ensure the datasets are not in the LLMs training corpus, we selected 4 private medical datasets that need an authorization process to access. Hence, these datasets would not be part of the LLMs training corpus given their proprietary nature and hence would be unseen to the LLM. While the private and unseen aspect was the main motivation, we also wish to highlight that these are real-world medical datasets. Consequently, this allows us to test a highly realistic problem setting.

B.2. Data generation.

GPT-4 and GPT-3.5 We access GPT-4 (OpenAI, 2023) and GPT-3.5-Turbo (Brown et al., 2020) through the API. We use a temperature of 0.9.

GReaT. GReaT (Borisov et al., 2023) is a generative model which fine-tunes an LLM based on a training set. We use the implementation provided by authors.

Generative model based approaches. For the other baselines used in 3.1, we use the library SynthCity (Qian et al., 2023), using the defaults. We detail each next.

- TVAE: this is a conditional Variational Auto Encoder (VAE) for tabular data and is based on Xu et al. (2019)
- CTGAN: A conditional generative adversarial network which can handle tabular data and is based on Xu et al. (2019)
- NFLOW: Normalizing Flows are generative models which produce tractable distributions where both sampling and density evaluation can be efficient and exact.
- TabDDPM: A diffusion model that can be universally applied to any tabular dataset, handles any type of feature and is based on Kotelnikov et al. (2022)

Traditional Data Augmentation. We use SMOTE (Chawla et al., 2002) which augments data by considering nearest neighbors and performing linear interpolations. We use the implementation provided by (Lemaître et al., 2017), and set the number of neighbors k to 5.

B.3. Data curation

Learning dynamics computation We train an XGBoost with 100 estimators on D_{train}. We then compute predictive confidence and aleatoric uncertainty for the samples in D_{syn}. The motivation for the choice of an XGBoost backbone is that we cannot expect good performance by choosing "any" curation model, but rather we require a curation model with enough capacity and generalization properties — where boosting methods like XGBoost used in our work have shown to achieve best performance on tabular data. This leads to our guideline for the curation step: the model used for curation should be **at least as flexible** as the model that the practitioner intends to use for the downstream task.

Learning dynamics thresholds Recall that `CLLM` has two thresholds τ_{conf} and τ_{al} on the predictive confidence and aleatoric uncertainty respectively, as defined in 2.2. We set $\tau_{\text{conf}} = 0.2$, in order to select high confidence samples. We adopt an adaptive threshold for τ_{al} based on the dataset, such that $\tau_{\text{al}} = 0.75 \cdot (\max(v_{al}(D_{\text{syn}})) - \min(v_{al}(D_{\text{syn}})))$. Note that by definition $v_{al}(D_{\text{syn}})$ is bounded between 0 and 0.25.

Example of learning dynamics We include examples of learning dynamics computed for 20 samples in Fig. 7.

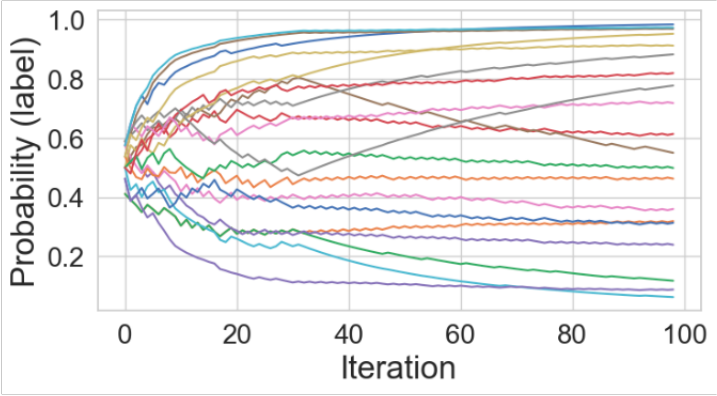

Figure 7: Learning dynamics computed for 20 samples

B.4. Downstream task

We compute downstream performance in Sec. 3.1 using four different downstream models: XGBoost, Random Forest, Decision tree, and Logistic Regression.

B.5. Prompt example

We include the template of the prompts used throughout the paper. We show how we include (1) in-context examples (demonstrations), (2) contextual information including dataset background and feature information and (3) the instruction.

```
System role: 'You are a tabular synthetic data generation model.'

You are a synthetic data generator.
Your goal is to produce data which mirrors \
the given examples in causal structure and feature and label distributions \
but also produce as diverse samples as possible.

I will give you real examples first.

Context: Leverage your medical knowledge about covid and Brazil to generate 1000
realistic but diverse samples.

example data: {data}

The output should be a markdown code snippet formatted in the following schema:

"Sex_male": string  // feature column
"Age": string  // feature column
"Age_40": string  // feature column
"Age_40_50": string  // feature column
"Age_50_60": string  // feature column
"Age_60_70": string  // feature column
"Age_70": string  // feature column
"Fever": string  // feature column
"Cough": string  // feature column
"Sore_throat": string  // feature column
"Shortness_of_breath": string  // feature column
"Respiratory_discomfort": string  // feature column
"SPO2": string  // feature column
"Dihareea": string  // feature column
"Vomitting": string  // feature column
"Cardiovascular": string  // feature column
"Asthma": string  // feature column
"Diabetis": string  // feature column
"Pulmonary": string  // feature column
"Immunosuppresion": string  // feature column
"Obesity": string  // feature column
"Liver": string  // feature column
"Neurologic": string  // feature column
"Renal": string  // feature column
"Branca": string  // feature column
"Preta": string  // feature column
"Amarela": string  // feature column
"Parda": string  // feature column
"Indigena": string  // feature column
"is_dead": string  // label if patient dead or not, is_dead

DO NOT COPY THE EXAMPLES but generate realistic but new and diverse samples which have
 the correct label conditioned on the features.
```

Listing 1: Template of the prompt

C. Additional Results

C.1. Decoupling prior knowledge and data model

Two components can be attributed to the good performances of CLLM: the background knowledge of the LLM, and its capacity to build a strong data model. In this subsection, we provide insights to understand the effect of the LLM's background knowledge (e.g. prior). We considered the Covid dataset (private medical dataset, to avoid memorization issues) and generated data with GPT-4 (same as Section 2.1). We ablate the prompt used in our work (detailed in Appendix B.5), and solely provide one in-context example in the prompt, in order to give the LLM the minimal amount of information about the desired structure of the dataset. This forces the LLM to rely on its own prior (background knowledge), and removes the effect of in-context examples which could be used to build a data model. We report the results for the prior and CLLM in Table 6.

Table 6: Downstream accuracy when varying the number of in-context samples in the prompt to generate the augmented datasets.

In-context samples	Downstream accuracy
$n = 1$ (Prior)	70.20 ± 1.60
$n = 20$	73.87 ± 0.50
$n = 40$	73.95 ± 0.67
$n = 100$	74.71 ± 0.34
D_{oracle}	74.6 ± 0.15

From these results, we conclude that the LLM prior permits to obtain good downstream performance, but is outperformed by $\mathcal{D}_{\text{oracle}}$ by a margin of 4.4%. Hence, we cannot solely rely on the prior. Furthermore, downstream performance increases as the number of in-context samples increases. This shows it is important to include in-context samples if we wish to obtain downstream performance close to $\mathcal{D}_{\text{oracle}}$, as the LLM can build a good data model. This implies that while the LLM uses background knowledge of similar datasets, it still needs in-context samples to refine its prior by creating a good data model.

We then quantify and visualize the strength of the prior, by studying how much the LLMs output distribution adapts to the in-context samples provided. We evaluate data generated by the prior of the LLM ($n = 1$), and for $n = 20, 40, 100$ on the Covid dataset. In particular, we observe in Figure 8 that there is a region in the oracle data which is not captured by the LLM's prior output (the left part of the leftmost blob, circled in blue in Figure 8). However, as the number of in-context real examples increases in the prompt of the LLM, we observe that this steers the LLM to generate data which covers this region. This region is associated to the subgroup of people older than 87 years old, and having many severe comorbidities (e.g. Diabetes, Cardiovascular diseases) and many respiratory symptoms. This subgroup, in the Oracle dataset, represents less than 3.5% of the data, and is completely ignored by the GPT-4 prior. In particular, the prior defaults to more typical patients in the range 70-80 years old. On the contrary, as n increases, the LLM is guided by the in-context samples and generates samples from this subgroup, which are "rarer" or different from the general population.

This shows that the LLM captures the distinct features of this particular region, and is not overwhelmed by the prior. Instead, the data in the form of in-context samples adapts it, and aligns the augmented dataset with the ground-truth distribution.

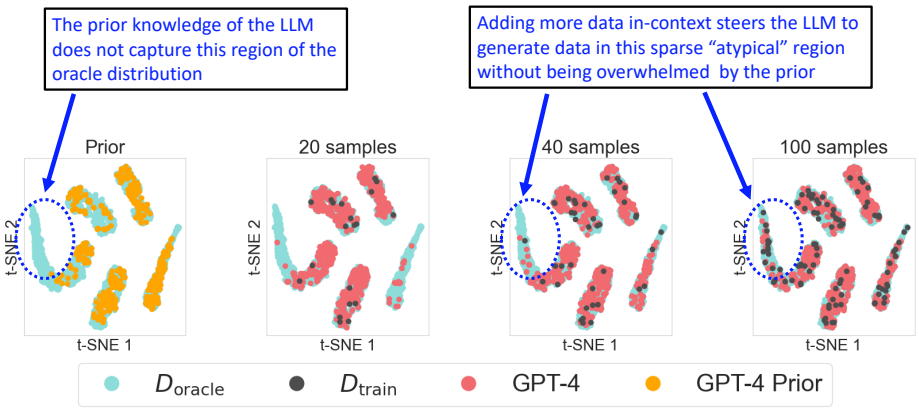

Figure 8: The data generated by the LLM captures the distinct features in "atypical" regions of the Oracle manifold, as in-context samples are added to the prompt. This shows that it is flexible enough to adapt its prior knowledge to the nuances of the data. The group encircled in blue represents patients who are > 88 years old, representing around 3.5% of the Oracle. This illustrates the added in-context samples can successfully guide the LLM to generate these rare samples.

C.2. Curation mechanism also improves open-source LLM generated data

In this subsection, we demonstrate the wide applicability of the `CLLM` framework. Different factors may affect the choice of the LLM backbone, such as operational costs, and the desired parameterization of the LLM (for quality of generation). In addition to GPT-4 and GPT-3.5, we use the open source models Mistral-7b (Jiang et al., 2023) and LLaMa-13b (4-bit quantized) (Touvron et al., 2023), LLaMa-70b (Touvron et al., 2023) and Mixtral-8x7b (Jiang et al., 2024) to generate augmented datasets [3]. We then compute the downstream performance when training a model on both the uncurated and curated data. As can be seen in Table 7, downstream performance with uncurated is lower for these open-source models compared to GPT-4 — which is expected given their significantly smaller size (i.e. parameter count). However, the curation mechanism, which is the key contribution of `CLLM`, almost always improves downstream performance for all LLM backbones investigated. Overall, this demonstrates the versatility and wide applicability of `CLLM` for tabular data augmentation in low-data regimes.

Practical tip: The size of the LLM plays a role: the larger models (with more parameters) outperform those with fewer parameters (e.g. GPT-4 vs Mixtral-8x7b vs Mistral-7b). Hence, while curation helps improve all LLM-generated data, ideally the best LLM possible should be used.

[3] For LLaMa-70b and Mixtral-8x7b, we only run open-source datasets due to data sharing restrictions with the endpoints for those two models.

Table 7: AUC averaged over 4 downstream models on D_{test} for GPT-4, GPT-3.5, Mistral-7b, LLAMA-13b, LLAMA-70b and Mixtral-8x7b (Curated and Uncurated)

Dataset	Open AI				Open source							
	GPT-4		GPT-3.5		Mistral-7b		LLAMA-13b		LLAMA-70b		Mixtral	
	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.
covid (n=20)	73.78	**73.87** ↑	69.85	**71.41** ↑	71.47	**72.80** ↑	63.25	**67.24** ↑	N.A.	N.A.	N.A.	N.A.
cutract (n=20)	71.15	**72.50** ↑	69.97	**71.54** ↑	69.60	**71.34** ↑	67.84	66.71	N.A.	N.A.	N.A.	N.A.
maggic (n=20)	60.70	**61.48** ↑	57.54	**58.69** ↑	53.64	52.06	53.30	**53.96** ↑	N.A.	N.A.	N.A.	N.A.
seer (n=20)	84.53	**84.82** ↑	83.34	**83.71** ↑	83.60	**85.18** ↑	80.94	**82.84** ↑	N.A.	N.A.	N.A.	N.A.
compas (n=20)	68.01	67.91	62.07	**64.43** ↑	56.26	**59.95** ↑	60.08	**60.34** ↑	56.10	**60.68** ↑	54.34	**61.66** ↑
adult (n=20)	50.39	**71.48** ↑	49.23	**72.37** ↑	47.68	**65.84** ↑	48.82	**66.00** ↑	51.96	**68.40** ↑	57.78	**76.84** ↑
drug (n=20)	75.08	**75.29** ↑	71.68	**72.14** ↑	74.14	**75.21** ↑	67.96	67.87	66.77	**69.12** ↑	72.45	71.63
covid (n=40)	73.40	**73.95** ↑	70.42	**71.93** ↑	69.61	**71.47** ↑	61.32	**65.57** ↑	N.A.	N.A.	N.A.	N.A.
cutract (n=40)	69.87	**71.72** ↑	68.47	**69.56** ↑	68.74	**72.36** ↑	64.34	**67.46** ↑	N.A.	N.A.	N.A.	N.A.
maggic (n=40)	59.29	**60.77** ↑	57.50	**59.15** ↑	52.43	**53.79** ↑	52.61	**53.45** ↑	N.A.	N.A.	N.A.	N.A.
seer (n=40)	84.29	**84.93** ↑	83.46	**84.44** ↑	83.88	**85.21** ↑	78.82	**82.65** ↑	N.A.	N.A.	N.A.	N.A.
compas (n=40)	67.57	**67.85** ↑	61.34	**62.84** ↑	57.07	**60.87** ↑	59.52	**61.46** ↑	57.48	**61.20**↑	58.90	**63.32**↑
adult (n=40)	48.31	**73.82** ↑	49.21	**74.27** ↑	48.80	**74.13** ↑	54.34	**69.31** ↑	64.44	**74.83**↑	56.59	**78.30** ↑
drug (n=40)	74.30	**75.79** ↑	71.33	**72.76** ↑	73.12	**74.12** ↑	69.84	**72.50** ↑	64.14	**68.14** ↑	74.34	**76.07** ↑
covid (n=100)	73.77	**74.71** ↑	70.71	**72.76** ↑	71.02	**73.37** ↑	63.76	**70.68** ↑	N.A.	N.A.	N.A.	N.A.
cutract (n=100)	70.20	**72.51** ↑	69.97	**71.94** ↑	68.92	**71.17** ↑	64.81	**69.85** ↑	N.A.	N.A.	N.A.	N.A.
maggic (n=100)	58.98	**61.32** ↑	55.71	**58.90** ↑	52.53	**53.36** ↑	54.27	53.65	N.A.	N.A.	N.A.	N.A.
seer (n=100)	84.45	**85.37** ↑	83.92	**85.08** ↑	82.23	**84.36** ↑	81.00	**81.99** ↑	N.A.	N.A.	N.A.	N.A.
compas (n=100)	68.02	**68.19** ↑	60.10	**62.47** ↑	53.74	**61.28** ↑	57.90	**61.79** ↑	60.96	**63.36** ↑	59.07	**63.13** ↑
adult (n=100)	46.09	**74.57** ↑	47.56	**73.97** ↑	40.51	**71.08** ↑	48.85	**72.91** ↑	54.45	**76.67** ↑	54.27	**77.61** ↑
drug (n=100)	76.24	**76.74** ↑	69.46	**71.05** ↑	74.02	**76.55** ↑	66.86	**75.31** ↑	71.47	**75.00** ↑	73.22	**75.83** ↑
covid (n=200)	73.40	**74.62** ↑	70.70	**73.12** ↑	70.81	**73.26** ↑	62.53	**70.67** ↑	N.A.	N.A.	N.A.	N.A.
cutract (n=200)	71.39	**73.01** ↑	70.28	**72.39** ↑	67.69	**70.29** ↑	65.77	**69.11** ↑	N.A.	N.A.	N.A.	N.A.
maggic (n=200)	58.92	**61.41** ↑	57.33	**60.16** ↑	52.78	52.40	52.56	**52.90** ↑	N.A.	N.A.	N.A.	N.A.
seer (n=200)	84.39	**85.56** ↑	83.48	**84.80** ↑	83.15	**84.18** ↑	80.88	**82.74** ↑	N.A.	N.A.	N.A.	N.A.
compas (n=200)	67.02	**68.15** ↑	60.48	**63.39** ↑	53.96	**59.22** ↑	57.56	**59.97** ↑	61.82	**63.53** ↑	60.99	**64.72** ↑
adult (n=200)	40.96	**75.84** ↑	49.89	**76.11** ↑	44.13	**74.23** ↑	48.25	**78.35** ↑	54.42	**76.23** ↑	42.42	**76.06** ↑
drug (n=200)	75.58	**76.06** ↑	70.66	**72.81** ↑	71.89	**76.54** ↑	70.88	**75.46** ↑	69.12	**74.22** ↑	73.43	**76.31** ↑

C.3. Ablation for contextual information on Compas

We conduct a similar experiment as in Table 2, and use the dataset Compas. We report the results in Table 8.

Table 8: Including contextual information in the prompt improves precision (P), recall (R), and utility (U) in low-sample settings (results shown for Compas).

n_{samples} in D_{train}	GPT-4 w/ context			GPT-4 no context			TVAE		
	P	R	U	P	R	U	P	R	U
20	$\mathbf{0.69_{(0.02)}}$	$\mathbf{0.88_{(0.02)}}$	$\mathbf{0.69_{(0.02)}}$	$0.27_{(0.03)}$	$\mathbf{0.89_{(0.03)}}$	$0.60_{(0.03)}$	$0.43_{(0.02)}$	$0.43_{(0.05)}$	$0.55_{(0.04)}$
40	$\mathbf{0.70_{(0.0)}}$	$\mathbf{0.92_{(0.01)}}$	$\mathbf{0.65_{(0.03)}}$	$0.31_{(0.06)}$	$0.84_{(0.03)}$	$0.57_{(0.01)}$	$0.54_{(0.02)}$	$0.80_{(0.02)}$	$0.50_{(0.04)}$
100	$\mathbf{0.69_{(0.02)}}$	$\mathbf{0.89_{(0.02)}}$	$\mathbf{0.69_{(0.01)}}$	$0.34_{(0.1)}$	$0.85_{(0.05)}$	$0.62_{(0.01)}$	$0.60_{(0.03)}$	$0.86_{(0.02)}$	$0.59_{(0.03)}$
200	$\mathbf{0.70_{(0.01)}}$	$\mathbf{0.89_{(0.02)}}$	$\mathbf{0.69_{(0.01)}}$	$0.31_{(0.05)}$	$0.87_{(0.03)}$	$0.58_{(0.05)}$	$0.65_{(0.02)}$	$0.88_{(0.01)}$	$0.63_{(0.01)}$

These results highlight the importance of incorporating contextual information in the prompt, as it enables to exploit the prior knowledge of the LLM.

C.4. Comparison to random noise baseline

We compare CLLM to a random noise baseline, where we augment the dataset with random additive Gaussian noise. In order to capture the correlations between the different features, we fit a Kernel Density Estimator with a Gaussian kernel and bandwidth given by Scott's rule. We then sample 1000 points to create an augmented dataset D_{syn}. We report the performance gap between CLLM and this baseline (with and without curation) for the Covid and Compas datasets in Figure 9. We observe that the random noise baseline does not match the performance of CLLM (i.e. has a performance gap), although the baseline naturally improves as the dataset D_{train} grows in size.

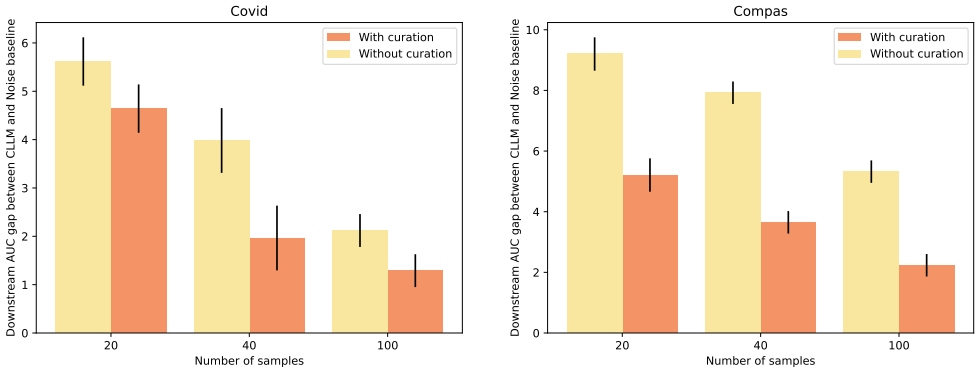

Figure 9: The random noise baseline does not match the performance of CLLM

C.5. Detailed results for Section 3.1

We report additional results for Sec. 3.1, showing the AUC for each downstream model (XGBoost, Random forest, Logistic regression, Decision tree). As we can see, the conclusion that curation helps improve downstream performance holds for each of these various downstream models, as is indicated by the green arrows in Tables 9, 10, 11, 12.

Table 9: AUC for the RF model on D_{test} where curation improves performance for all methods across all sample sizes n, as indicated by ↑.

	Real data		CLLM (OURS)				Baselines											
			GPT-4		GPT-3.5		CTGAN		TabDDPM		GReaT		NFLOW		SMOTE		TVAE	
Dataset	D_{oracle}	D_{train}	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.
covid (n=20)	76.11	72.52	**75.67**	75.59	72.37	73.32 ↑	61.66	65.24 ↑	70.22	70.13	57.78	68.24 ↑	65.54	70.72 ↑	71.40	71.42 ↑	63.90	67.69 ↑
cutract (n=20)	74.16	73.34	73.45	**74.24** ↑	72.22	73.51 ↑	66.24	70.45 ↑	69.73	70.49 ↑	52.61	68.11 ↑	66.63	72.39 ↑	71.42	72.34 ↑	71.18	71.89 ↑
maggic (n=20)	71.28	59.18	62.92	**63.97** ↑	60.10	61.25 ↑	53.21	55.14 ↑	55.61	56.50 ↑	50.92	57.23 ↑	55.91	59.07 ↑	58.20	58.55 ↑	55.43	57.71 ↑
seer (n=20)	90.09	85.22	86.30	**86.82** ↑	85.86	85.73	78.11	80.52 ↑	82.61	82.74 ↑	47.57	75.70 ↑	77.35	81.45 ↑	82.09	83.00 ↑	77.63	81.16 ↑
compas (n=20)	66.79	64.47	**68.40**	67.85	61.93	64.29 ↑	56.87	63.21 ↑	58.56	61.26 ↑	52.23	63.30 ↑	59.60	64.79 ↑	61.05	62.18 ↑	59.31	63.37 ↑
adult (n=20)	85.83	83.31	51.35	73.45 ↑	49.17	74.27 ↑	75.63	78.99 ↑	77.27	77.64 ↑	69.35	78.79 ↑	68.32	78.27 ↑	77.75	78.34 ↑	76.13	79.02 ↑
drug (n=20)	83.12	77.07	**79.23**	78.90	77.11	76.95	72.81	77.24 ↑	72.00	74.03 ↑	63.34	73.29 ↑	65.47	70.19 ↑	75.48	75.36	72.22	74.32 ↑
covid (n=40)	76.11	75.21	75.63	**75.70** ↑	72.27	73.59 ↑	66.64	70.75 ↑	75.56	75.39	56.91	70.88 ↑	70.47	73.15 ↑	73.59	73.63 ↑	64.27	69.71 ↑
cutract (n=40)	74.16	72.31	71.67	**73.20** ↑	69.92	71.01 ↑	64.67	69.21 ↑	69.16	69.67 ↑	53.41	68.93 ↑	61.23	69.21 ↑	70.82	71.21 ↑	59.99	67.56 ↑
maggic (n=40)	71.28	60.91	61.80	**63.11** ↑	59.38	61.50 ↑	56.49	58.17 ↑	56.50	58.21 ↑	48.43	57.84 ↑	55.47	59.94 ↑	59.82	60.11 ↑	56.76	58.92 ↑
seer (n=40)	90.09	86.96	86.01	86.29 ↑	86.73	**87.09** ↑	83.46	86.62 ↑	85.86	85.19	54.71	83.22 ↑	83.23	86.81 ↑	83.76	84.82 ↑	80.34	85.82 ↑
compas (n=40)	66.79	62.73	68.06	**68.13** ↑	61.17	62.27 ↑	56.05	60.92 ↑	59.76	61.03 ↑	57.82	64.25 ↑	58.89	63.79 ↑	61.20	61.25 ↑	55.89	60.54 ↑
adult (n=40)	85.83	83.61	50.26	75.55 ↑	46.68	76.63 ↑	75.13	81.56 ↑	76.10	80.40 ↑	68.04	81.08 ↑	73.29	81.46 ↑	80.35	81.10 ↑	75.81	82.11 ↑
drug (n=40)	83.12	78.81	78.35	**79.54** ↑	77.33	77.94 ↑	74.68	76.64 ↑	74.50	77.18 ↑	70.25	77.77 ↑	65.86	74.55 ↑	74.84	75.26 ↑	71.29	75.98 ↑
covid (n=100)	76.11	75.78	75.86	**76.33** ↑	73.02	74.40 ↑	72.00	74.74 ↑	74.64	75.74 ↑	65.76	74.69 ↑	67.05	75.47 ↑	74.00	74.09 ↑	72.11	74.44 ↑
cutract (n=100)	74.16	73.93	72.92	74.73 ↑	72.51	73.93 ↑	70.18	74.26 ↑	71.71	73.69 ↑	55.48	70.79 ↑	69.79	**74.75** ↑	71.18	72.47 ↑	68.47	73.78 ↑
maggic (n=100)	71.28	63.06	60.99	**63.36** ↑	57.97	60.86 ↑	58.98	60.76 ↑	58.88	60.13 ↑	49.66	59.57 ↑	57.82	62.24 ↑	61.55	61.90 ↑	57.46	60.55 ↑
seer (n=100)	90.00	87.53	86.33	87.31 ↑	86.40	87.06 ↑	84.59	87.27 ↑	85.89	87.00 ↑	70.32	85.97 ↑	84.30	87.52 ↑	85.12	85.82 ↑	81.92	86.41 ↑
compas (n=100)	66.79	63.18	68.44	**68.67** ↑	59.41	62.00 ↑	60.28	64.34 ↑	60.17	62.63 ↑	59.32	63.98 ↑	60.10	65.19 ↑	61.34	61.37 ↑	59.58	63.37 ↑
adult (n=100)	85.83	84.38	46.37	76.29 ↑	47.30	75.74 ↑	77.45	82.63 ↑	80.88	81.69 ↑	79.23	83.33 ↑	73.52	83.38 ↑	82.54	83.05 ↑	76.60	82.28 ↑
drug (n=100)	83.12	80.31	79.29	79.75 ↑	74.55	76.46 ↑	74.54	78.01 ↑	78.34	79.88 ↑	73.19	79.72 ↑	67.01	76.46 ↑	76.92	77.57 ↑	74.53	78.82 ↑
covid (n=200)	76.11	76.08	75.23	75.84 ↑	72.62	74.99 ↑	74.04	76.25 ↑	75.28	**76.82** ↑	67.08	75.62 ↑	67.73	75.67 ↑	74.82	75.27 ↑	70.64	75.03 ↑
cutract (n=200)	74.16	74.25	73.48	75.28 ↑	72.01	74.37 ↑	71.99	74.93 ↑	74.33	**76.19** ↑	68.05	74.68 ↑	71.47	75.67 ↑	72.44	72.85 ↑	68.55	74.43 ↑
maggic (n=200)	71.28	64.77	61.56	63.78 ↑	59.76	62.53 ↑	60.85	63.57 ↑	58.23	58.85 ↑	50.84	61.49 ↑	57.52	63.00 ↑	63.72	64.01 ↑	58.92	62.08 ↑
seer (n=200)	90.09	88.13	86.96	87.72 ↑	85.28	86.70 ↑	85.05	87.83 ↑	87.14	**88.80** ↑	84.86	88.14 ↑	80.80	87.46 ↑	86.57	87.11 ↑	82.30	87.59 ↑
compas (n=200)	66.79	63.61	67.68	**68.80** ↑	60.67	63.43 ↑	61.20	65.11 ↑	60.88	63.87 ↑	60.10	64.91 ↑	56.94	64.76 ↑	61.57	62.76 ↑	59.95	64.14 ↑
adult (n=200)	85.83	85.16	40.18	77.61 ↑	46.74	78.71 ↑	80.92	84.26 ↑	84.50	**85.44** ↑	82.73	85.05 ↑	77.48	84.72 ↑	84.26	84.33 ↑	78.12	83.94 ↑
drug (n=200)	83.12	81.47	78.70	78.98 ↑	75.87	77.59 ↑	77.19	80.03 ↑	73.25	75.75 ↑	78.50	81.13 ↑	67.83	78.81 ↑	79.50	80.16 ↑	74.81	79.45 ↑

Table 10: AUC for the XGB model on D_{test} where curation improves performance for all methods across all sample sizes n, as indicated by ↑.

	Real data		CLLM (OURS)				Baselines											
			GPT-4		GPT-3.5		CTGAN		TabDDPM		GReaT		NFLOW		SMOTE		TVAE	
Dataset	D_{oracle}	D_{train}	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.
covid (n=20)	77.59	69.70	**75.59**	75.49	71.27	72.25 ↑	61.40	65.96 ↑	69.39	68.21	55.99	67.06 ↑	62.28	69.67 ↑	68.49	68.43	62.49	67.16 ↑
cutract (n=20)	73.14	71.04	73.26	**74.30** ↑	71.62	73.04 ↑	65.13	69.67 ↑	68.46	68.87 ↑	51.99	68.31 ↑	62.93	70.87 ↑	68.28	70.72 ↑	69.96	71.98 ↑
maggic (n=20)	71.38	57.98	61.54	**62.81** ↑	58.76	60.01 ↑	53.62	55.44 ↑	55.97	57.06 ↑	50.14	56.96 ↑	54.91	58.22 ↑	56.79	57.15 ↑	55.41	57.72 ↑
seer (n=20)	91.80	80.41	**87.00**	86.56	85.67	85.42	76.84	80.00 ↑	81.79	80.21	45.06	75.39 ↑	76.25	81.45 ↑	78.57	79.78 ↑	75.69	81.01 ↑
compas (n=20)	69.17	65.49	**69.71**	68.79	62.55	64.68 ↑	56.34	63.92 ↑	58.19	61.42 ↑	51.73	63.51 ↑	59.01	64.84 ↑	62.35	63.46 ↑	59.18	63.62 ↑
adult (n=20)	88.64	79.21	52.12	74.28 ↑	47.70	72.91 ↑	71.70	78.26 ↑	76.50	77.22 ↑	67.79	78.67 ↑	66.79	77.77 ↑	77.11	77.97 ↑	75.46	**79.35** ↑
drug (n=20)	81.06	69.26	**78.17**	77.28	75.23	75.71 ↑	70.60	74.31 ↑	70.29	71.25 ↑	57.18	69.67 ↑	62.96	69.12 ↑	70.84	72.30 ↑	69.78	72.45 ↑
covid (n=40)	77.59	71.57	75.23	**75.36** ↑	71.55	73.37 ↑	66.17	70.85 ↑	72.21	72.68 ↑	56.03	70.80 ↑	67.39	71.82 ↑	70.02	69.96	61.19	69.77 ↑
cutract (n=40)	73.14	69.50	71.26	**73.57** ↑	69.66	71.34 ↑	62.52	68.69 ↑	64.50	68.05 ↑	51.23	68.23 ↑	60.74	68.91 ↑	69.42	69.19	59.34	67.80 ↑
maggic (n=40)	71.38	58.76	60.27	**61.88** ↑	57.87	60.29 ↑	55.87	58.47 ↑	55.38	58.03 ↑	48.50	57.02 ↑	54.53	59.01 ↑	58.12	58.29 ↑	56.19	58.33 ↑
seer (n=40)	91.80	84.51	86.63	**87.40** ↑	86.27	86.80 ↑	82.48	86.55 ↑	81.62	81.58	52.80	83.03 ↑	80.04	85.84 ↑	82.53	83.62 ↑	79.35	85.81 ↑
compas (n=40)	69.17	62.90	**69.24**	68.90	61.74	63.21 ↑	56.63	62.55 ↑	59.07	59.38 ↑	58.01	64.95 ↑	58.44	64.44 ↑	61.81	61.89 ↑	56.60	62.26 ↑
adult (n=40)	88.64	82.16	48.51	76.59 ↑	45.92	76.92 ↑	73.38	82.53 ↑	74.03	79.84 ↑	66.60	80.98 ↑	69.47	81.56 ↑	80.01	80.51 ↑	74.20	**82.53** ↑
drug (n=40)	81.06	72.72	77.03	**78.85** ↑	76.37	76.48 ↑	70.99	74.48 ↑	74.65	76.53 ↑	64.52	75.09 ↑	62.58	73.14 ↑	71.80	72.15 ↑	67.61	74.04 ↑
covid (n=100)	77.59	73.16	75.55	**76.56** ↑	72.69	74.36 ↑	70.54	73.99 ↑	72.99	74.50 ↑	63.40	73.98 ↑	63.50	73.79 ↑	72.64	73.35 ↑	70.45	73.76 ↑
cutract (n=100)	73.14	70.49	72.29	74.01 ↑	71.51	73.48 ↑	68.64	73.63 ↑	71.32	**74.02** ↑	54.74	69.69 ↑	66.97	72.98 ↑	67.71	69.42 ↑	66.44	72.32 ↑
maggic (n=100)	71.38	60.55	60.04	**62.57** ↑	56.98	60.49 ↑	58.21	60.42 ↑	58.11	59.42 ↑	49.71	58.77 ↑	55.36	60.58 ↑	60.02	60.15 ↑	56.47	60.28 ↑
seer (n=100)	91.80	85.52	86.81	**87.64** ↑	86.39	87.21 ↑	83.74	87.29 ↑	84.66	86.85 ↑	68.20	85.49 ↑	82.37	86.80 ↑	83.18	83.87 ↑	80.45	86.49 ↑
compas (n=100)	69.17	62.71	**69.37**	69.37	59.83	62.80 ↑	60.56	64.90 ↑	58.20	62.00 ↑	58.51	64.60 ↑	59.76	64.68 ↑	61.33	62.03 ↑	60.37	63.42 ↑
adult (n=100)	88.64	82.78	44.60	76.75 ↑	46.34	76.83 ↑	74.97	82.89 ↑	77.38	81.42 ↑	78.84	**83.96** ↑	68.31	82.87 ↑	82.58	83.24 ↑	76.15	82.47 ↑
drug (n=100)	81.06	75.49	79.12	**79.81** ↑	72.52	74.78 ↑	68.89	75.91 ↑	76.88	78.33 ↑	66.38	77.02 ↑	62.27	75.13 ↑	74.40	74.49 ↑	71.02	76.85 ↑
covid (n=200)	77.59	73.06	75.12	**76.64** ↑	71.74	74.66 ↑	72.57	75.50 ↑	73.56	75.88 ↑	66.02	74.51 ↑	63.29	74.62 ↑	73.34	73.89 ↑	68.14	74.21 ↑
cutract (n=200)	73.14	71.68	73.43	75.14 ↑	71.76	74.31 ↑	69.35	73.80 ↑	72.76	**75.33** ↑	65.83	73.43 ↑	68.05	74.27 ↑	70.06	71.35 ↑	66.65	73.34 ↑
maggic (n=200)	71.38	62.40	59.46	**63.15** ↑	58.05	61.68 ↑	59.04	62.58 ↑	57.63	59.23 ↑	50.63	60.09 ↑	55.55	61.36 ↑	61.73	61.88 ↑	57.09	61.15 ↑
seer (n=200)	91.80	85.87	86.75	88.08 ↑	85.74	86.97 ↑	83.97	87.64 ↑	86.21	**88.46** ↑	83.21	87.65 ↑	76.91	87.16 ↑	84.70	85.59 ↑	81.45	87.46 ↑
compas (n=200)	69.17	63.14	68.80	**69.54** ↑	61.62	64.59 ↑	60.65	65.67 ↑	60.97	65.12 ↑	60.66	65.67 ↑	56.49	64.34 ↑	61.68	62.93 ↑	60.32	64.18 ↑
adult (n=200)	88.64	84.12	40.38	77.39 ↑	47.40	79.40 ↑	80.00	84.61 ↑	83.73	85.73 ↑	81.92	**85.76** ↑	74.90	85.13 ↑	84.78	84.45	75.54	84.39 ↑
drug (n=200)	81.06	78.41	78.25	79.09 ↑	73.31	76.76 ↑	72.82	79.00 ↑	72.62	74.89 ↑	74.30	**79.75** ↑	63.91	77.60 ↑	77.64	77.92 ↑	70.16	77.89 ↑

Table 11: AUC for the DT model on D_{test} where curation improve performance for all methods across all sample sizes n, as indicated by ↑.

	Real data		CLLM (OURS)				Baselines											
			GPT-4		GPT-3.5		CTGAN		TabDDPM		GReaT		NFLOW		SMOTE		TVAE	
Dataset	D_{oracle}	D_{train}	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.
covid (n=20)	63.46	61.92	67.52	**67.67** ↑	61.07	64.89 ↑	52.94	56.62 ↑	58.95	59.19 ↑	51.83	58.73 ↑	54.37	61.53 ↑	59.56	59.01	56.46	59.49 ↑
cutract (n=20)	62.49	64.62	62.40	**66.03** ↑	62.10	65.15 ↑	55.67	61.73 ↑	57.91	58.97 ↑	51.89	62.76 ↑	57.12	64.85 ↑	62.85	62.38	60.25	62.34 ↑
maggic (n=20)	56.53	54.80	54.67	**55.03** ↑	52.54	53.54 ↑	50.90	52.08 ↑	52.39	52.70 ↑	49.70	53.08 ↑	51.52	53.23 ↑	52.55	52.44	50.91	51.52 ↑
seer (n=20)	78.86	71.09	76.17	**77.17** ↑	73.92	75.58 ↑	64.18	70.69 ↑	74.78	74.72	46.60	68.79 ↑	65.77	72.25 ↑	70.54	71.19 ↑	64.47	70.11 ↑
compas (n=20)	61.05	58.73	62.07	**63.91** ↑	57.17	60.85 ↑	53.83	58.83 ↑	51.77	56.11 ↑	52.39	60.96 ↑	54.64	60.27 ↑	58.16	57.57	53.52	58.97 ↑
adult (n=20)	73.84	68.03	52.80	67.10 ↑	49.15	66.52 ↑	61.57	68.93 ↑	70.46	71.19 ↑	62.04	**71.87** ↑	59.07	69.16 ↑	65.52	63.33	64.40	70.41 ↑
drug (n=20)	65.66	63.49	64.40	**66.87** ↑	62.69	64.21 ↑	56.76	64.18 ↑	61.04	62.48 ↑	52.22	62.62 ↑	53.50	61.30 ↑	62.34	60.69	55.00	58.89 ↑
covid (n=40)	63.46	65.10	66.55	**67.85** ↑	62.54	64.64 ↑	57.16	63.41 ↑	63.15	60.93	51.66	61.81 ↑	57.04	63.13 ↑	62.19	60.49	54.51	60.32 ↑
cutract (n=40)	62.49	63.01	62.49	**65.29** ↑	61.61	63.01 ↑	56.40	62.63 ↑	58.18	58.97 ↑	51.41	62.50 ↑	55.64	62.24 ↑	60.93	61.29 ↑	51.94	60.24 ↑
maggic (n=40)	56.53	53.74	53.20	**54.71** ↑	52.33	52.97 ↑	51.85	53.85 ↑	52.48	53.45 ↑	49.61	52.87 ↑	51.24	54.45 ↑	53.69	53.37	50.60	52.66 ↑
seer (n=40)	78.86	72.40	76.02	77.13 ↑	75.18	**78.26** ↑	68.31	73.89 ↑	77.94	73.24	51.45	72.32 ↑	67.78	75.95 ↑	70.32	72.81 ↑	68.79	74.48 ↑
compas (n=40)	61.05	58.23	62.00	**63.43** ↑	56.73	59.19 ↑	52.23	56.84 ↑	56.67	57.55 ↑	54.52	60.09 ↑	54.47	58.92 ↑	56.59	56.47	52.62	57.20 ↑
adult (n=40)	74.84	69.75	49.47	68.90 ↑	47.85	66.79 ↑	60.55	70.14 ↑	64.06	70.56 ↑	58.77	71.63 ↑	60.26	71.50 ↑	69.43	70.58 ↑	59.76	**72.41** ↑
drug (n=40)	65.66	64.18	62.86	**65.90** ↑	60.09	63.43 ↑	60.84	65.40 ↑	63.05	64.90 ↑	54.80	65.25 ↑	54.68	62.40 ↑	61.21	61.98 ↑	53.77	60.23 ↑
covid (n=100)	63.46	63.17	67.14	**68.60** ↑	62.40	66.02 ↑	61.34	64.63 ↑	63.33	66.42 ↑	55.86	64.53 ↑	55.44	64.91 ↑	61.96	63.40 ↑	59.75	64.47 ↑
cutract (n=100)	62.49	62.69	60.96	65.60 ↑	62.12	65.95 ↑	59.06	**66.20** ↑	63.38	65.72 ↑	52.62	61.73 ↑	58.11	65.08 ↑	60.15	61.34 ↑	58.82	63.76 ↑
maggic (n=100)	56.53	54.83	53.58	**55.70** ↑	50.72	53.98 ↑	52.29	55.06 ↑	53.93	54.61 ↑	49.71	53.34 ↑	51.82	54.53 ↑	54.11	54.34 ↑	51.31	53.33 ↑
seer (n=100)	78.86	73.83	76.50	**77.84** ↑	74.81	77.80 ↑	69.61	76.91 ↑	74.07	77.21 ↑	59.62	75.40 ↑	69.51	76.54 ↑	71.04	72.63 ↑	66.35	76.44 ↑
compas (n=100)	61.05	56.61	63.04	**63.39** ↑	55.57	57.40 ↑	55.07	58.95 ↑	53.38	55.69 ↑	54.61	59.31 ↑	54.38	57.92 ↑	56.69	57.01 ↑	54.34	56.72 ↑
adult (n=100)	73.84	72.19	46.68	66.58 ↑	47.94	67.37 ↑	62.42	71.78 ↑	68.06	70.77 ↑	66.29	**74.36** ↑	58.28	72.14 ↑	71.51	72.68 ↑	62.11	72.18 ↑
drug (n=100)	65.66	64.20	66.18	**67.41** ↑	60.77	61.21 ↑	57.67	63.41 ↑	63.26	64.55 ↑	54.89	65.72 ↑	52.06	63.38 ↑	61.21	62.30 ↑	55.40	63.32 ↑
covid (n=200)	63.46	63.45	67.19	**68.74** ↑	63.25	65.73 ↑	61.58	66.27 ↑	63.55	66.98 ↑	57.20	65.54 ↑	55.08	63.77 ↑	63.60	63.40	60.01	64.27 ↑
cutract (n=200)	62.49	63.53	63.36	65.33 ↑	63.01	65.47 ↑	60.51	65.01 ↑	62.55	**66.94** ↑	57.40	65.14 ↑	59.13	65.20 ↑	61.66	62.40 ↑	58.29	64.85 ↑
maggic (n=200)	56.53	55.06	52.76	54.41 ↑	52.41	54.34 ↑	52.47	**55.21** ↑	52.40	53.39 ↑	50.43	53.66 ↑	51.58	54.82 ↑	54.64	54.36	51.78	54.22 ↑
seer (n=200)	78.86	74.51	75.20	77.26 ↑	75.33	77.41 ↑	70.96	77.08 ↑	74.51	**79.20** ↑	71.44	78.66 ↑	65.84	76.60 ↑	73.11	74.77 ↑	66.37	76.36 ↑
compas (n=200)	61.05	57.20	61.09	**62.89** ↑	56.15	59.02 ↑	54.20	57.78 ↑	55.72	58.68 ↑	55.83	59.81 ↑	52.13	57.25 ↑	56.55	57.34 ↑	54.07	58.66 ↑
adult (n=200)	73.84	72.26	44.41	67.99 ↑	48.46	67.24 ↑	66.56	74.12 ↑	71.61	74.89 ↑	69.91	**76.52** ↑	60.70	74.05 ↑	73.11	74.08 ↑	63.10	73.82 ↑
drug (n=200)	65.66	67.04	66.29	66.57 ↑	61.88	64.34 ↑	59.10	66.06 ↑	63.59	63.40	59.11	**69.04** ↑	53.78	64.55 ↑	64.56	65.35 ↑	56.82	63.02 ↑

Table 12: AUC for the LR model on D_{test} where curation improves performance for all methods across all sample sizes n, as indicated by ↑.

	Real data		CLLM (OURS)				Baselines											
			GPT-4		GPT-3.5		CTGAN		TabDDPM		GReaT		NFLOW		SMOTE		TVAE	
Dataset	D_{oracle}	D_{train}	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.
covid (n=20)	80.47	69.85	76.35	**76.73** ↑	74.69	75.18 ↑	59.99	66.87 ↑	68.78	69.84 ↑	63.94	71.81 ↑	69.28	72.32 ↑	68.36	68.44 ↑	63.92	70.12 ↑
cutract (n=20)	79.12	71.47	**75.50**	75.41	73.93	74.47 ↑	69.01	70.06 ↑	68.11	68.02	53.04	68.90 ↑	71.09	73.55 ↑	71.09	71.53 ↑	74.37	74.69 ↑
maggic (n=20)	70.46	56.55	63.67	**64.12** ↑	58.74	59.98 ↑	53.27	55.37 ↑	54.40	55.28 ↑	50.39	55.27 ↑	56.52	59.00 ↑	55.83	56.45 ↑	54.59	57.82 ↑
seer (n=20)	90.91	85.96	88.63	**88.74** ↑	87.90	88.11 ↑	78.22	83.69 ↑	83.17	84.74 ↑	51.07	77.84 ↑	84.86	84.75	85.72	86.11 ↑	80.33	82.63 ↑
compas (n=20)	73.02	63.74	**71.88**	71.10	66.62	67.89 ↑	55.62	64.29 ↑	62.15	64.70 ↑	56.97	66.59 ↑	64.71	68.59 ↑	62.68	63.16 ↑	61.19	64.36 ↑
adult (n=20)	88.37	79.24	45.28	71.07 ↑	50.89	75.77 ↑	80.03	**81.27** ↑	73.19	74.10 ↑	68.84	79.66 ↑	75.65	80.74 ↑	74.61	75.04 ↑	76.82	78.84 ↑
drug (n=20)	81.41	73.56	**78.54**	78.10	71.69	71.68	73.07	74.88 ↑	69.14	70.97 ↑	62.38	69.99 ↑	66.57	70.37 ↑	71.99	72.28 ↑	69.39	71.08 ↑
covid (n=40)	80.47	71.20	76.17	**76.91** ↑	75.32	76.13 ↑	64.56	68.82 ↑	71.08	72.77 ↑	61.40	71.21 ↑	70.74	73.84 ↑	68.85	69.65 ↑	64.13	69.58 ↑
cutract (n=40)	79.12	71.90	74.05	**74.82** ↑	72.69	72.89 ↑	68.43	70.97 ↑	70.67	72.39 ↑	61.50	74.10 ↑	67.98	71.55 ↑	70.26	70.12	67.87	70.88 ↑
maggic (n=40)	70.46	59.65	61.89	**63.39** ↑	60.41	61.84 ↑	55.81	56.64 ↑	56.58	58.08 ↑	48.68	58.82 ↑	57.50	60.90 ↑	57.98	58.00 ↑	56.61	59.42 ↑
seer (n=40)	90.91	87.86	88.50	**88.89** ↑	85.66	85.62	85.95	87.62 ↑	84.92	85.45 ↑	60.75	85.84 ↑	88.46	88.85 ↑	86.56	87.58 ↑	86.30	88.36 ↑
compas (n=40)	73.02	65.50	**70.96**	70.93	65.70	66.72 ↑	60.24	63.78 ↑	59.91	62.49 ↑	65.17	68.19 ↑	62.63	67.00 ↑	63.70	64.18 ↑	58.63	64.17 ↑
adult (n=40)	88.37	82.23	44.99	74.25 ↑	56.38	76.76 ↑	78.24	82.22 ↑	71.83	81.15 ↑	73.65	81.54 ↑	81.49	**84.34** ↑	81.80	83.16 ↑	80.54	83.02 ↑
drug (n=40)	81.41	71.74	**78.98**	78.88	71.54	73.21 ↑	71.34	74.44 ↑	72.12	73.68 ↑	70.00	76.45 ↑	66.91	73.78 ↑	69.09	69.75 ↑	68.21	70.94 ↑
covid (n=100)	80.47	74.19	76.53	**77.34** ↑	74.74	76.27 ↑	72.30	75.15 ↑	75.45	76.20 ↑	69.05	74.94 ↑	71.01	76.38 ↑	71.71	72.28 ↑	73.90	75.17 ↑
cutract (n=100)	79.12	76.73	74.62	75.70 ↑	73.75	74.40 ↑	73.89	75.61 ↑	75.70	74.49	60.03	74.35 ↑	75.49	**76.86** ↑	76.13	75.47	73.85	76.21 ↑
maggic (n=100)	70.46	60.14	61.31	**63.64** ↑	57.17	60.28 ↑	59.32	61.13 ↑	58.12	58.97 ↑	49.10	59.95 ↑	60.43	63.06 ↑	59.87	59.59	59.43	61.28 ↑
seer (n=100)	90.91	88.92	88.16	88.69 ↑	88.09	88.25 ↑	88.45	89.11 ↑	87.53	88.25 ↑	83.13	88.44 ↑	88.46	**89.27** ↑	87.95	87.65	86.81	88.67 ↑
compas (n=100)	73.02	67.75	71.21	**71.34** ↑	65.61	67.67 ↑	64.14	66.72 ↑	61.55	65.03 ↑	67.43	68.88 ↑	65.84	68.39 ↑	66.39	66.53 ↑	65.61	67.77 ↑
adult (n=100)	88.37	85.59	46.71	78.68 ↑	48.64	75.94 ↑	82.30	84.48 ↑	77.41	78.98 ↑	83.99	84.99 ↑	82.67	**85.77** ↑	85.63	85.43	81.32	83.97 ↑
drug (n=100)	81.41	74.34	**80.35**	80.01	70.00	71.76 ↑	71.66	75.80 ↑	71.26	72.40 ↑	74.57	78.65 ↑	69.34	77.50 ↑	71.04	71.75 ↑	71.91	75.38 ↑
covid (n=200)	80.47	76.73	76.06	77.25 ↑	75.18	77.09 ↑	76.11	77.54 ↑	77.51	**78.09** ↑	71.92	76.62 ↑	74.06	77.53 ↑	74.94	74.94	72.75	76.00 ↑
cutract (n=200)	79.12	77.53	75.28	76.30 ↑	74.36	75.38 ↑	75.25	75.88 ↑	**77.66**	77.66	75.37	76.69 ↑	76.45	77.50 ↑	76.77	76.59	72.94	74.86 ↑
maggic (n=200)	70.46	63.32	61.89	**64.30** ↑	59.10	62.10 ↑	61.57	63.98 ↑	56.77	57.35 ↑	51.05	63.17 ↑	59.17	63.84 ↑	62.86	62.86	60.94	63.47 ↑
seer (n=200)	90.91	90.01	88.65	89.20 ↑	87.57	88.14 ↑	88.20	88.80 ↑	89.69	89.81 ↑	89.10	89.68 ↑	87.35	89.54 ↑	89.13	89.14 ↑	88.74	89.65 ↑
compas (n=200)	73.02	69.14	70.53	**71.36** ↑	63.49	66.50 ↑	66.26	68.73 ↑	64.83	66.40 ↑	67.85	69.93 ↑	60.78	68.07 ↑	68.18	68.18	66.26	68.99 ↑
adult (n=200)	88.37	86.94	38.89	80.35 ↑	56.97	79.07 ↑	85.25	86.29 ↑	86.81	86.63	85.70	85.94 ↑	84.18	86.65 ↑	86.88	86.70	84.09	85.91 ↑
drug (n=200)	81.41	77.47	79.06	79.61 ↑	71.56	72.54 ↑	76.15	78.84 ↑	69.00	73.14 ↑	77.51	79.72 ↑	75.47	**80.07** ↑	75.82	75.69	71.76	76.95 ↑

C.6. Full results for performance evaluation

We report full results with standard deviation for the results from the main paper. The performance is AUC averaged over XGBoost, Random forest, Logistic regression, Decision tree.

Table 13: AUC averaged over 4 downstream models on D_{test} where curation improves performance for all methods across all sample sizes n, as indicated by ↑. CLLM w/ GPT-4 (Curated) dataset provides the strongest performance for both private/proprietary datasets and public datasets

	Real data		CLLM (OURS)				Baselines											
			GPT-4		GPT-3.5		CTGAN		TabDDPM		GReaT		NFLOW		SMOTE		TVAE	
Dataset	D_{oracle}	D_{train}	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.	Uncur.	Cur.
covid (n=20)	74.41$_{(0.11)}$	68.50$_{(1.57)}$	73.78$_{(0.31)}$	**73.87**$_{(0.50)}$	69.85$_{(0.75)}$	71.41$_{(0.92)}$	59.00$_{(2.25)}$	63.67$_{(2.51)}$	66.84$_{(1.66)}$	66.85$_{(1.56)}$	57.38$_{(1.47)}$	66.46$_{(0.80)}$	62.87$_{(0.98)}$	68.56$_{(1.07)}$	66.95$_{(1.66)}$	66.82$_{(1.89)}$	61.69$_{(2.72)}$	66.11$_{(2.79)}$
cutract (n=20)	72.23$_{(0.65)}$	70.12$_{(1.16)}$	71.15$_{(0.46)}$	**72.50**$_{(0.76)}$	69.97$_{(1.13)}$	71.54$_{(1.38)}$	64.01$_{(2.26)}$	67.98$_{(1.63)}$	66.05$_{(1.14)}$	66.59$_{(1.49)}$	52.38$_{(1.78)}$	67.02$_{(0.97)}$	64.44$_{(1.05)}$	70.42$_{(1.25)}$	68.41$_{(1.34)}$	69.24$_{(1.25)}$	68.94$_{(1.38)}$	70.22$_{(1.12)}$
maggic (n=20)	67.41$_{(0.06)}$	57.13$_{(0.85)}$	60.70$_{(0.38)}$	**61.48**$_{(0.49)}$	57.54$_{(0.83)}$	58.69$_{(0.76)}$	52.75$_{(1.34)}$	54.51$_{(1.40)}$	54.59$_{(1.02)}$	55.39$_{(1.07)}$	50.29$_{(0.52)}$	55.64$_{(0.43)}$	54.72$_{(1.38)}$	57.38$_{(1.06)}$	55.84$_{(1.21)}$	56.15$_{(1.20)}$	54.08$_{(1.05)}$	56.19$_{(0.76)}$
seer (n=20)	87.92$_{(0.08)}$	80.67$_{(1.67)}$	84.53$_{(0.33)}$	**84.82**$_{(0.47)}$	83.34$_{(0.95)}$	83.71$_{(0.52)}$	74.34$_{(4.11)}$	78.73$_{(3.13)}$	80.59$_{(1.32)}$	80.60$_{(1.32)}$	47.57$_{(2.51)}$	74.43$_{(3.36)}$	76.06$_{(1.68)}$	79.98$_{(1.65)}$	79.23$_{(2.17)}$	80.02$_{(2.08)}$	74.53$_{(1.37)}$	78.73$_{(1.24)}$
compas (n=20)	67.51$_{(0.03)}$	63.11$_{(0.96)}$	**68.01**$_{(0.44)}$	67.91$_{(0.55)}$	62.07$_{(1.51)}$	64.43$_{(1.38)}$	55.67$_{(1.60)}$	62.56$_{(1.61)}$	57.67$_{(1.90)}$	60.87$_{(1.33)}$	53.33$_{(1.85)}$	63.59$_{(1.30)}$	59.49$_{(1.72)}$	64.62$_{(1.14)}$	61.06$_{(1.92)}$	61.59$_{(2.15)}$	58.30$_{(1.77)}$	62.58$_{(1.87)}$
adult (n=20)	84.17$_{(0.10)}$	77.45$_{(1.25)}$	50.39$_{(3.99)}$	71.48$_{(2.34)}$	49.23$_{(2.57)}$	72.37$_{(2.26)}$	72.23$_{(1.26)}$	76.86$_{(1.25)}$	74.35$_{(1.48)}$	75.04$_{(1.82)}$	67.00$_{(4.72)}$	77.25$_{(1.49)}$	67.46$_{(3.76)}$	76.48$_{(1.77)}$	73.75$_{(1.64)}$	73.67$_{(1.46)}$	73.20$_{(1.51)}$	76.90$_{(1.64)}$
drug (n=20)	77.81$_{(0.55)}$	70.84$_{(2.25)}$	75.08$_{(1.17)}$	**75.29**$_{(1.11)}$	71.68$_{(2.25)}$	72.14$_{(2.64)}$	68.31$_{(2.81)}$	72.65$_{(2.00)}$	68.12$_{(2.38)}$	69.68$_{(2.41)}$	58.78$_{(4.26)}$	68.89$_{(3.57)}$	62.13$_{(4.94)}$	67.75$_{(3.72)}$	70.16$_{(1.87)}$	70.16$_{(1.75)}$	66.60$_{(2.95)}$	69.18$_{(2.59)}$
covid (n=40)	74.41$_{(0.11)}$	70.77$_{(0.96)}$	73.40$_{(0.61)}$	**73.95**$_{(0.67)}$	70.42$_{(0.92)}$	71.93$_{(0.60)}$	63.63$_{(1.15)}$	68.46$_{(0.93)}$	70.50$_{(1.49)}$	70.44$_{(1.37)}$	56.50$_{(0.83)}$	68.68$_{(1.39)}$	66.41$_{(2.51)}$	70.48$_{(1.48)}$	68.66$_{(1.48)}$	68.44$_{(1.28)}$	61.03$_{(0.96)}$	67.35$_{(0.60)}$
cutract (n=40)	72.23$_{(0.65)}$	69.18$_{(0.65)}$	69.87$_{(0.62)}$	**71.72**$_{(0.46)}$	68.47$_{(0.59)}$	69.56$_{(0.50)}$	63.01$_{(2.33)}$	67.87$_{(1.36)}$	65.63$_{(2.71)}$	67.27$_{(2.38)}$	54.39$_{(1.59)}$	68.44$_{(0.41)}$	61.40$_{(1.95)}$	67.98$_{(1.06)}$	67.86$_{(0.72)}$	67.95$_{(0.59)}$	59.79$_{(1.85)}$	66.62$_{(1.06)}$
maggic (n=40)	67.50$_{(0.04)}$	58.26$_{(0.55)}$	59.29$_{(0.50)}$	**60.77**$_{(0.36)}$	57.50$_{(0.69)}$	59.15$_{(0.48)}$	55.00$_{(1.27)}$	56.78$_{(1.17)}$	55.24$_{(0.63)}$	56.94$_{(0.50)}$	48.81$_{(0.73)}$	56.64$_{(0.63)}$	54.68$_{(0.53)}$	58.58$_{(0.54)}$	57.40$_{(0.57)}$	57.44$_{(0.67)}$	55.04$_{(1.14)}$	57.33$_{(1.06)}$
seer (n=40)	87.92$_{(0.08)}$	82.93$_{(0.55)}$	84.29$_{(0.39)}$	**84.93**$_{(0.46)}$	83.46$_{(1.20)}$	84.44$_{(0.69)}$	80.05$_{(0.93)}$	83.67$_{(0.51)}$	82.59$_{(1.48)}$	81.37$_{(1.11)}$	54.93$_{(2.06)}$	81.11$_{(1.24)}$	79.88$_{(0.53)}$	84.36$_{(0.62)}$	80.79$_{(0.67)}$	82.21$_{(0.67)}$	78.69$_{(2.32)}$	83.62$_{(1.06)}$
compas (n=40)	67.51$_{(0.03)}$	62.34$_{(0.79)}$	67.57$_{(0.40)}$	**67.85**$_{(0.37)}$	61.34$_{(1.67)}$	62.84$_{(1.21)}$	56.29$_{(1.96)}$	61.02$_{(1.69)}$	58.85$_{(1.28)}$	60.11$_{(1.22)}$	58.88$_{(1.04)}$	64.37$_{(0.91)}$	58.61$_{(1.35)}$	63.54$_{(1.11)}$	60.83$_{(1.19)}$	60.95$_{(1.22)}$	55.94$_{(1.50)}$	61.04$_{(1.56)}$
adult (n=40)	84.17$_{(0.10)}$	79.44$_{(1.03)}$	48.31$_{(3.58)}$	73.82$_{(2.19)}$	49.21$_{(1.78)}$	74.27$_{(1.49)}$	71.82$_{(1.22)}$	79.11$_{(0.71)}$	71.51$_{(1.85)}$	77.99$_{(0.65)}$	66.77$_{(3.32)}$	78.81$_{(1.36)}$	71.13$_{(2.27)}$	79.71$_{(1.02)}$	77.90$_{(1.07)}$	78.84$_{(1.19)}$	72.58$_{(1.34)}$	**80.02**$_{(0.76)}$
drug (n=40)	77.81$_{(0.55)}$	71.86$_{(1.07)}$	74.30$_{(0.59)}$	**75.79**$_{(0.39)}$	71.33$_{(0.88)}$	72.76$_{(0.97)}$	69.46$_{(2.23)}$	72.74$_{(1.70)}$	71.08$_{(1.72)}$	73.07$_{(1.03)}$	64.89$_{(1.39)}$	73.64$_{(0.87)}$	62.51$_{(3.03)}$	70.97$_{(1.91)}$	69.23$_{(1.68)}$	69.78$_{(1.46)}$	65.22$_{(1.37)}$	70.30$_{(1.04)}$
covid (n=100)	74.41$_{(0.11)}$	71.57$_{(0.48)}$	73.77$_{(0.27)}$	**74.71**$_{(0.34)}$	70.71$_{(0.46)}$	72.76$_{(0.44)}$	69.05$_{(0.96)}$	72.13$_{(0.66)}$	71.60$_{(0.59)}$	73.22$_{(0.46)}$	63.52$_{(1.29)}$	72.04$_{(0.57)}$	64.25$_{(1.51)}$	72.64$_{(0.64)}$	70.08$_{(0.67)}$	70.78$_{(0.59)}$	69.05$_{(0.48)}$	71.96$_{(0.49)}$
cutract (n=100)	72.23$_{(0.65)}$	70.96$_{(0.68)}$	70.20$_{(0.45)}$	**72.51**$_{(0.55)}$	69.97$_{(0.97)}$	71.94$_{(0.89)}$	67.94$_{(1.01)}$	72.42$_{(0.66)}$	70.53$_{(1.51)}$	71.98$_{(1.39)}$	55.72$_{(2.04)}$	69.14$_{(0.93)}$	67.59$_{(0.71)}$	72.42$_{(0.54)}$	68.79$_{(0.83)}$	69.68$_{(0.76)}$	66.89$_{(1.03)}$	71.52$_{(0.70)}$
maggic (n=100)	67.50$_{(0.04)}$	59.65$_{(0.50)}$	58.98$_{(0.23)}$	**61.32**$_{(0.42)}$	55.71$_{(0.83)}$	58.90$_{(0.72)}$	57.20$_{(0.91)}$	59.34$_{(0.64)}$	57.26$_{(0.50)}$	58.28$_{(0.46)}$	49.54$_{(0.71)}$	57.91$_{(0.74)}$	56.36$_{(0.54)}$	60.11$_{(0.54)}$	58.89$_{(0.51)}$	58.99$_{(0.42)}$	56.17$_{(0.68)}$	58.86$_{(0.69)}$
seer (n=100)	87.92$_{(0.08)}$	83.95$_{(0.32)}$	84.45$_{(0.38)}$	**85.37**$_{(0.47)}$	83.92$_{(0.81)}$	85.08$_{(0.32)}$	81.60$_{(0.73)}$	85.14$_{(0.36)}$	83.04$_{(0.78)}$	84.83$_{(0.49)}$	70.32$_{(2.52)}$	83.83$_{(0.39)}$	81.16$_{(1.03)}$	85.03$_{(0.39)}$	81.82$_{(0.45)}$	82.49$_{(0.46)}$	78.88$_{(0.71)}$	84.50$_{(0.44)}$
compas (n=100)	67.51$_{(0.03)}$	62.56$_{(0.72)}$	68.02$_{(0.29)}$	**68.19**$_{(0.37)}$	60.10$_{(1.60)}$	62.47$_{(1.11)}$	60.01$_{(1.16)}$	63.73$_{(0.99)}$	58.32$_{(0.91)}$	61.34$_{(0.94)}$	59.97$_{(0.82)}$	64.19$_{(0.83)}$	60.02$_{(0.63)}$	64.04$_{(0.60)}$	61.44$_{(0.84)}$	61.73$_{(0.81)}$	59.97$_{(1.00)}$	62.82$_{(1.05)}$
adult (n=100)	84.17$_{(0.10)}$	81.24$_{(0.48)}$	46.09$_{(1.86)}$	74.57$_{(1.74)}$	47.56$_{(3.43)}$	73.97$_{(1.57)}$	74.29$_{(1.23)}$	80.45$_{(0.81)}$	75.93$_{(1.45)}$	78.22$_{(1.22)}$	77.09$_{(0.74)}$	**81.66**$_{(0.53)}$	70.70$_{(1.02)}$	81.04$_{(0.37)}$	80.56$_{(0.53)}$	81.10$_{(0.44)}$	74.04$_{(1.44)}$	80.23$_{(0.98)}$
drug (n=100)	77.81$_{(0.55)}$	73.58$_{(0.72)}$	76.24$_{(0.64)}$	**76.74**$_{(0.51)}$	69.46$_{(2.43)}$	71.05$_{(2.30)}$	68.19$_{(2.06)}$	73.28$_{(1.41)}$	72.43$_{(0.96)}$	73.79$_{(0.65)}$	67.26$_{(1.22)}$	75.28$_{(0.46)}$	62.67$_{(2.23)}$	73.12$_{(0.97)}$	70.90$_{(1.08)}$	71.53$_{(1.13)}$	68.22$_{(1.60)}$	73.59$_{(0.97)}$
covid (n=200)	74.41$_{(0.11)}$	72.33$_{(0.52)}$	73.40$_{(0.25)}$	**74.62**$_{(0.13)}$	70.70$_{(0.73)}$	73.12$_{(0.48)}$	71.07$_{(0.35)}$	73.89$_{(0.40)}$	72.47$_{(0.50)}$	74.44$_{(0.44)}$	65.55$_{(0.67)}$	73.07$_{(0.38)}$	65.04$_{(0.69)}$	72.90$_{(0.37)}$	71.68$_{(0.49)}$	71.87$_{(0.49)}$	67.89$_{(0.56)}$	72.38$_{(0.42)}$
cutract (n=200)	72.23$_{(0.65)}$	71.75$_{(0.71)}$	71.39$_{(0.76)}$	73.01$_{(0.85)}$	70.28$_{(0.67)}$	72.39$_{(0.84)}$	69.28$_{(0.55)}$	72.41$_{(0.65)}$	71.83$_{(0.65)}$	**74.03**$_{(0.69)}$	66.66$_{(0.94)}$	72.49$_{(0.71)}$	68.77$_{(0.71)}$	73.16$_{(0.68)}$	70.23$_{(0.83)}$	70.80$_{(0.79)}$	66.61$_{(0.76)}$	71.87$_{(0.71)}$
maggic (n=200)	67.50$_{(0.04)}$	61.39$_{(0.44)}$	58.92$_{(0.47)}$	**61.41**$_{(0.39)}$	57.33$_{(0.51)}$	60.16$_{(0.41)}$	58.48$_{(0.59)}$	61.33$_{(0.57)}$	56.26$_{(0.82)}$	57.20$_{(0.91)}$	50.74$_{(0.74)}$	59.60$_{(0.55)}$	55.95$_{(0.72)}$	60.75$_{(0.39)}$	60.73$_{(0.44)}$	60.78$_{(0.49)}$	57.18$_{(0.56)}$	60.23$_{(0.44)}$
seer (n=200)	87.92$_{(0.08)}$	84.63$_{(0.35)}$	84.39$_{(0.19)}$	85.56$_{(0.30)}$	83.48$_{(0.41)}$	84.80$_{(0.34)}$	82.04$_{(0.70)}$	85.34$_{(0.36)}$	84.39$_{(0.44)}$	**86.57**$_{(0.27)}$	82.15$_{(0.43)}$	86.03$_{(0.23)}$	77.73$_{(1.65)}$	85.19$_{(0.22)}$	83.38$_{(0.32)}$	84.15$_{(0.42)}$	79.71$_{(0.58)}$	85.26$_{(0.25)}$
compas (n=200)	67.51$_{(0.03)}$	63.27$_{(0.60)}$	67.02$_{(0.53)}$	**68.15**$_{(0.40)}$	60.48$_{(1.86)}$	63.39$_{(1.28)}$	60.58$_{(0.83)}$	64.32$_{(0.78)}$	60.60$_{(0.90)}$	63.52$_{(1.04)}$	61.11$_{(0.63)}$	65.08$_{(0.53)}$	56.58$_{(1.03)}$	63.60$_{(0.83)}$	61.99$_{(0.64)}$	62.80$_{(0.61)}$	60.15$_{(1.14)}$	63.99$_{(0.71)}$
adult (n=200)	84.17$_{(0.10)}$	82.12$_{(0.41)}$	40.96$_{(2.47)}$	75.84$_{(1.43)}$	49.89$_{(3.22)}$	76.11$_{(1.43)}$	78.18$_{(0.26)}$	82.32$_{(0.29)}$	81.66$_{(0.19)}$	83.17$_{(0.20)}$	80.06$_{(0.53)}$	**83.32**$_{(0.35)}$	74.31$_{(0.81)}$	82.64$_{(0.33)}$	82.26$_{(0.38)}$	82.39$_{(0.33)}$	75.21$_{(0.69)}$	82.02$_{(0.25)}$
drug (n=200)	77.81$_{(0.55)}$	76.10$_{(0.45)}$	75.58$_{(0.55)}$	76.06$_{(0.42)}$	70.66$_{(1.56)}$	72.81$_{(1.18)}$	71.31$_{(1.17)}$	75.98$_{(0.76)}$	69.61$_{(2.76)}$	71.79$_{(1.99)}$	72.35$_{(0.61)}$	**77.41**$_{(0.56)}$	65.25$_{(2.47)}$	75.26$_{(0.77)}$	74.38$_{(0.50)}$	74.78$_{(0.54)}$	68.39$_{(1.26)}$	74.33$_{(0.61)}$

C.7. CLLM with context in low-resource languages

We investigate how the language of tabular column names affect the performance of CLLM. To do so, for a given dataset, we translate the feature names from English to 2 low-resource languages (Ranathunga et al., 2023), i.e. to (i) *Swahili* and (ii) *Hausa*. Changing the language of the features alters the contextual information provided to the LLM to generate synthetic data.

We report the results in Table 14. As expected, we observe for both Swahili and Hausa a performance drop compared to using English feature names, thereby mirroring the results in Section 2.1 on the importance of contextual information in the prompt. However, we note that CLLM for both Swahili and Hausa remains highly competitive compared to other baselines, notably thanks to the curation mechanism.

Table 14: Test AUC in settings where the names of features have been translated to Swahili and Hausa, with and without curation

Dataset	Uncurated *English*	Curated *English*	Uncurated *Swahili*	Curated *Swahili*	Uncurated *Hausa*	Curated *Hausa*
covid ($n = 20$)	73.78	73.87	71.32	73.06	68.75	70.01
seer ($n = 20$)	84.53	84.82	83.92	85.47	81.57	82.73
compas ($n = 20$)	68.01	67.91	64.42	61.61	62.85	63.06
covid $n = 40$	73.40	73.95	71.53	73.20	70.15	72.13
seer $n = 40$	84.29	84.93	79.71	84.25	81.47	84.66
compas $n = 40$	67.57	67.85	64.92	65.78	62.31	63.20

Takeaways:

1. CLLM can provide valuable augmentation and performance gains even with feature names in low-resource languages.
2. For optimal performance, we recommend using CLLM with English feature names or translating the feature names to English before applying CLLM.

C.8. Comparison of curation mechanism vs Data-IQ

We compare CLLM's curation mechanism with the approach taken by Data-IQ (Seedat et al., 2022a).

They differ along four dimensions:

1. **Problem setting** (data augmentation vs data understanding): CLLM focuses on the problem of data augmentation in low-data regimes by curating synthetic samples. In contrast, Data-IQ aims to understand and characterize subgroups within a given real dataset.
2. **Conceptually**: CLLM curates a large synthetic dataset D_{syn} with respect to a very small gold standard real dataset D_{train}, whereas Data-IQ aims to characterize subgroups of learnable samples in a single large real dataset D (e.g. to find the hard samples in D).
3. **Technically**: CLLM trains the curation model only on the small but gold standard D_{train}. The learning dynamics of the synthetic samples in D_{syn} are then assessed with respect to this curation model. In contrast, Data-IQ computes learning dynamics for real samples in D using a curator model trained on the same D it assesses.
4. **Empirical performance**: In CLLM , the curation aims at discarding synthetic samples which contradict the learning signal obtained from the real data. If we were to perform the curation like in Data-IQ, this implies that we would instead have to merge D_{train} and D_{syn}, and train a curator model on $D_{\text{merged}} = D_{\text{train}} \cup D_{\text{syn}}$ to assess D_{syn}. Intuitively, with such an approach, the signal from the small D_{train} would be overshadowed by D_{syn}, as the latter is a lot bigger in size (1000 samples), thus making the curation irrelevant.

We now show empirical evidence regarding the last point, by comparing CLLM with a baseline where we train the curator on $D_{\text{merged}} = D_{\text{train}} \cup D_{\text{syn}}$. The results are in Table 15. We observe that:

1. CLLM's curation outperforms Data-IQ curation on downstream performance across all datasets and n.
2. Data-IQ curation does not always improve upon the uncurated baseline, which aligns with our intuition that D_{syn} overshadows D_{train} because of its size.

Table 15: Comparing the CLLM curation mechanism to Data-IQ curation (Seedat et al., 2022a). We report the AUC averaged over 4 downstream models on $\mathcal{D}_{\text{test}}$.

Dataset	Uncurated	Curated (CLLM) - Ours	Curated (Data-IQ)
covid (n=20)	73.78	73.87	73.45
cutract (n=20)	71.15	72.50	70.84
maggic (n=20)	60.70	61.48	60.68
seer (n=20)	84.53	84.82	83.97
compas (n=20)	68.01	67.91	67.45
adult (n=20)	50.39	71.48	43.59
drug (n=20)	75.08	75.29	74.23
covid (n=40)	73.40	73.95	73.36
cutract (n=40)	69.87	71.72	69.57
maggic (n=40)	59.29	60.77	59.65
seer (n=40)	84.29	84.93	84.47
compas (n=40)	67.57	67.85	67.07
adult (n=40)	48.31	73.82	48.92
drug (n=40)	74.30	75.79	74.45
covid (n=100)	73.77	74.71	73.77
cutract (n=100)	70.20	72.51	70.29
maggic (n=100)	58.98	61.32	58.66
seer (n=100)	84.45	85.37	84.33
compas (n=100)	68.02	68.19	67.82
adult (n=100)	46.09	74.57	45.13
drug (n=100)	76.24	76.74	76.11
covid (n=200)	73.40	74.62	73.27
cutract (n=200)	71.39	73.01	71.43
maggic (n=200)	58.92	61.41	58.71
seer (n=200)	84.39	85.56	84.49
compas (n=200)	67.02	68.15	67.26
adult (n=200)	40.96	75.84	41.01
drug (n=200)	75.58	76.06	75.39

C.9. Comparing CLLM vs TabPFN

We outlined in Appendix A various dimensions on which CLLM and transfer learning / meta-learning / few-shot learning differ. To provide further empirical evidence on why the above dimensions are important (notably the point on the choice of downstream backbone), we compare CLLM with TabPFN (Hollmann et al., 2022), a few-shot learning method designed for small tabular problems. We chose TabPFN because it meets our criteria where it does not require access to external datasets, since the model is pretrained on an extensive set of synthetic tabular datasets and can perform few-shot learning with its transformer backbone.

We use the pretrained model released by the authors at https://github.com/automl/TabPFN and show the results in Table 16.

Takeaways: We see CLLM outperforms TabPFN on $6/7$ datasets, for the different n. The performance gains are especially noticeable in the ultra low-sample regime ($n = 20$).

Finally, we acknowledge that CLLM is not a one-size-fits-all approach. When the assumptions underpinning transfer learning or meta-learning hold (e.g. availability of external datasets), combining ideas from these learning paradigms with the augmentation methodology of CLLM could constitute an interesting research direction, but this falls beyond the scope of our current work.

Table 16: Comparison of CLLM vs TabPFN. We report the AUC averaged over 4 downstream models on $\mathcal{D}_{\text{test}}$.

Dataset	Uncurated	Curated (CLLM)	TabPFN
covid (n=20)	73.78	73.87	66.31
cutract (n=20)	71.15	72.50	63.86
maggic (n=20)	60.70	61.48	55.49
seer (n=20)	84.53	84.82	75.30
compas (n=20)	68.01	67.91	57.04
adult (n=20)	50.39	71.48	71.70
drug (n=20)	75.08	75.29	69.18
covid (n=40)	73.40	73.95	67.26
cutract (n=40)	69.87	71.72	65.93
maggic (n=40)	59.29	60.77	57.75
seer (n=40)	84.29	84.93	79.48
compas (n=40)	67.57	67.85	57.72
adult (n=40)	48.31	73.82	75.36
drug (n=40)	74.30	75.79	71.32
covid (n=100)	73.77	74.71	69.50
cutract (n=100)	70.20	72.51	70.01
maggic (n=100)	58.98	61.32	59.07
seer (n=100)	84.45	85.37	80.96
compas (n=100)	68.02	68.19	63.42
adult (n=100)	46.09	74.57	77.36
drug (n=100)	76.24	76.74	72.68
covid (n=200)	73.40	74.62	70.82
cutract (n=200)	71.39	73.01	71.59
maggic (n=200)	58.92	61.41	60.67
seer (n=200)	84.39	85.56	82.05
compas (n=200)	67.02	68.15	65.51
adult (n=200)	40.96	75.84	79.18
drug (n=200)	75.58	76.06	74.29

C.10. CLLM in specialized domains.

We conduct an experiment with the additional dataset Higgs (Whiteson, 2014). We chose this dataset because it comes from a specialized domain (physics), where the dataset consists in kinematic properties of particles measured by the particle detectors of an accelerator, which is likely under-represented in the LLMs training corpus.

We note that the contextual information for this dataset is quite specific, as can be seen from the names of the features, which include for example "m_bb", "m_jj", "m_jjj". This particular contextual information, which is not as semantically meaningful as other datasets, makes it interesting to compare the performance of CLLM with the traditional baselines. We show the results in Table 17. As we can see, the performance of CLLM is good for the smaller n. As n increases, the downstream task benefits more from the use of the other baselines. We highlight that in this case, our curation mechanism still benefits both CLLM and the baselines.

Table 17: CLLM performance on the Higgs dataset

Dataset	CLLM		CTGAN		TVAE		NFLOW	
	Uncur	Cur	Uncur	Cur	Uncur	Cur	Uncur	Cur
higgs (n=20)	65.82	70.25	62.73	67.31	59.14	69.89	56.45	69.63
higgs (n=40)	65.38	71.01	63.07	70.41	59.29	67.61	60.27	71.54
higgs (n=100)	68.62	73.42	68.46	75.59	68.90	76.05	59.73	75.86
higgs (n=200)	66.39	75.18	74.27	79.75	73.65	79.32	60.99	79.42

C.11. Behavior beyond the low-data regime.

We examine the behavior of baselines beyond $n = 200$ and the low-data regime. We provide a plot in Figure 10 for three datasets examining the behavior of CTGAN, TVAE and NFLOW for increasing n. We fix the CLLM method at $n = 200$ (as we do not see too much of an additional performance increase, due to the LLM context window which limits the number of in-context samples we can provide).

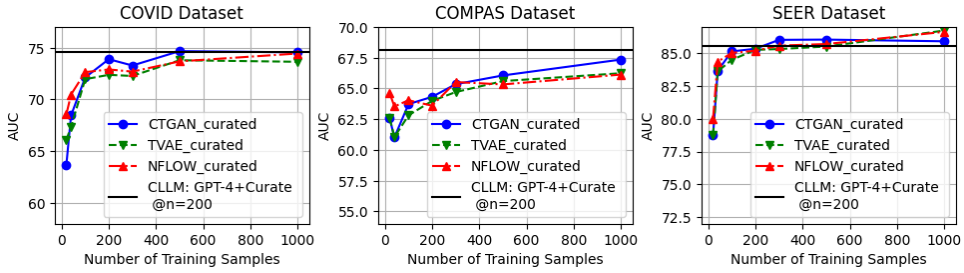

Figure 10: Behavior of baselines at high n, i.e. beyond the low data regime.

Takeaways:

1. Other baselines initially improve with more real samples, however their performance gains tend to plateau out at around $n \in \{200, 400\}$ samples. This suggests we only get minimal gains as increasing numbers of samples are added to the baselines.

2. CLLM either outperforms the baselines even for very high n (Compas, Covid) or remains competitive (SEER), even when the number of samples used by CTGAN, TVAE and NFLOW is twice or thrice bigger.

C.12. Addressing potential biases from the LLM

There are many challenges which arise from the use of LLMs. An example is potential biases from the LLM which might stem from their training data and might affect our synthetic data generated by the LLM.

We first describe some considerations which show how a practicioner using CLLM can address this issue.

1. **Choice of LLM:** Different LLMs may exhibit different levels of bias, due to their respective pretraining and alignment with human feedback. With CLLM , the practitioner has a lot of flexibility in the choice of the LLM. This is an important aspect when it comes to bias. The practitioner may want to minimize the risk of having bias in the augmented data by selecting one particular LLM for their task based on prior knowledge about different LLMs and their respective potential bias (Gallegos et al., 2023).

2. **Fairness analysis on the augmented data:** Practitioners can directly assess potential bias in the data generated by CLLM : (i) by doing a standard fairness analysis and computing different fairness metrics (Dwork et al., 2012; Hardt et al., 2016; Mehrabi et al., 2021) on the generated data, as can be done more generally when building a model with any given dataset. (ii) If some bias is detected, the practitioner can use any off-the-shelf debiasing method (Calmon et al., 2017; Feldman et al., 2015), or use a training objective to make the downstream model fair (Berk et al., 2017; Zhang et al., 2018). Note that this approach is possible because CLLM is a data augmentation method, which gives a lot of flexibility for the practitioner to adjust the data according to their needs.

3. **CLLM can reduce representation bias:** While these aforementioned considerations involve choices external to CLLM , we also want to emphasize that our CLLM approach has the potential to already address bias indirectly. As shown in Figure 3, underrepresented groups in the population benefit the most from CLLM . This suggests that we could use CLLM to remove representation bias, by leveraging CLLM to augment $\mathcal{D}_{\text{train}}$ with synthetic data from underrepresented groups. Table 1 and Table 2 show that data generated with CLLM better aligns with the ground-truth distribution, compared to using a widely used conventional generative approach (TVAE).

4. **Curation aligns the feature/label relationships:** The curation step of CLLM can already help indirectly address issues of bias in feature-label relationships if training data $\mathcal{D}_{\text{train}}$ is unbiased, even if it is not its primary purpose. The curation mechanism discards samples in $\mathcal{D}_{\text{syn}}$ which do not obey the same feature / label relationship as in $\mathcal{D}_{\text{train}}$. The results shown in Figure 5 demonstrate that the curation step aligns the feature/label relationship between the curated set and the ground-truth distribution. Thus, assuming that $\mathcal{D}_{\text{train}}$ is not biased and has the correct $Y|X$, discriminatory correlations stemming from the LLM bias could be detected in the form of feature/label relationships which differ from those in the gold standard $\mathcal{D}_{\text{train}}$.

We now provide additional empirical evidence into how curation can help with bias, in a synthetic setup.

Synthetic setup: We consider a synthetic setup where X and Y are random variables such that $X \in \mathbb{R}^2$ denotes the features, and $Y \in \{0, 1\}$ is the label. Furthermore, we let X_1 (the first coordinate of X) be a sensitive attribute. We then create a biased distribution such that a downstream predictor $\hat{Y}$ trained on this biased distribution will violate the fairness criterion of equalized odds (Hardt et al., 2016).

Formally, we consider a mixture of Gaussians to define X, i.e. $X = ZA + (1 - Z)B$ where $Z \sim Ber(1/2)$, $A \sim \mathcal{N}([-1.5, 0], I_2)$ and $B \sim \mathcal{N}([1.5, 0], I_2)$. Furthermore, we let $Y = Z$.

In order to introduce some bias in the data, we consider the variable Y' such that $Y' = 1 - Y$ if $|X_1| > 2.5$, else $Y' = Y$.

Let $\hat{Y}$ denote a downstream predictor, supposedly trained on samples drawn from the distribution of (X, Y') (instead of the ground-truth distribution of (X, Y)). Intuitively, for $y \in \{0, 1\}$, because of the definition of Y', we can expect $P(\hat{Y} = y \mid |X_1| > 2.5, Y = y)$ to be much smaller than $P(\hat{Y} = y \mid |X_1| \leq 2.5, Y = y)$, hence strongly violating equality of odds. To quantify that, we will compute the absolute equality of odds differences defined as

$$\Delta_{Y=1} = |P(\hat{Y} = 1 \mid |X_1| \leq 2.5, Y = 1) - P(\hat{Y} = 1 \mid |X_1| > 2.5, Y = 1)|$$

and

$$\Delta_{Y=0} = |P(\hat{Y} = 0 \mid |X_1| \leq 2.5, Y = 0) - P(\hat{Y} = 0 \mid |X_1| > 2.5, Y = 0)|$$

We investigate if curation can help address the bias. We generate a training dataset $\mathcal{D}_{\text{train}}$ of size $n = 20$, by sampling independent samples from the distribution of (X, Y). We generate an augmented dataset $\mathcal{D}_{\text{syn}}$ of size 1000, by sampling from the biased distribution of (X, Y'). We then curate $\mathcal{D}_{\text{syn}}$ using the curation mechanism of CLLM, and obtain $\mathcal{D}_{\text{curated}} \subset \mathcal{D}_{\text{syn}}$. Finally, we train three XGBoost models on each of the three datasets $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{syn}}$, and $\mathcal{D}_{\text{curated}}$.

Results. We evaluate the performance of these downstream models on a held-out $\mathcal{D}_{\text{test}}$.

We then report the average test accuracy, along with $\Delta_{Y=0}$ and $\Delta_{Y=1}$ for 10 different seeds in Table 18, which demonstrate that the curation mechanism helps ensure that the bias present in the augmented dataset $\mathcal{D}_{\text{syn}}$ does not propagate to the downstream model, as can be seen with the low values of $\Delta_{Y=0}$ and $\Delta_{Y=1}$ for the models trained on $\mathcal{D}_{\text{curated}}$.

Table 18: Curation can help address bias.

Dataset used for downstream training	**Test accuracy (%)** (↑)	$\Delta_{Y=1}$ (↓)	$\Delta_{Y=0}$ (↓)
$\mathcal{D}_{\text{syn}}$ (Biased)	76.7	0.89	0.90
$\mathcal{D}_{\text{train}}$ (Unbiased)	90.0	0.17	**0.07**
$\mathcal{D}_{\text{curated}}$	**91.3**	**0.13**	0.08

C.13. Choice of thresholds

In CLLM we have thresholds which are used as part of the curation mechanism. The intuition is that our choice of thresholds should discard hard samples, that is, samples for which the confidence is low while the aleatoric uncertainty is also low. Given this intuition, we set an adaptive threshold on the aleatoric uncertainty, with $\tau_{al} = 0.75 \cdot (\max(v_{al}(\mathcal{D}_{\text{syn}}) - \min(v_{al}(\mathcal{D}_{\text{syn}}))$, where $v_{al}(\mathcal{D}_{\text{syn}})$ denotes the set of aleatoric uncertainties for the samples in $\mathcal{D}_{\text{syn}}$. Hence, this threshold is adaptive and depends on the dataset at hand, which reduces the number of degrees of freedom to 1, namely the choice of the confidence threshold. For the latter, we set $\tau_{conf} = 0.2$. While we do not claim that this value is optimal for all datasets, we find in practice that it is robust for datasets across different domains.

We explore other combinations of thresholds to confirm our intuition. In addition to the choice of thresholds used in our main experiments, we consider two alternatives:

- Aggressive filtering: we set $\tau_{conf} = 0.95$ and $\tau_{al} = 0.2$.
- Permissive filtering: we set $\tau_{conf} = 0.5$ and $\tau_{al} = 0.05$

We then evaluate the test AUC for $n = 20$, with an XGBoost as the downstream model, and GPT-3.5 as the LLM backbone. The results are shown in Figure 11.

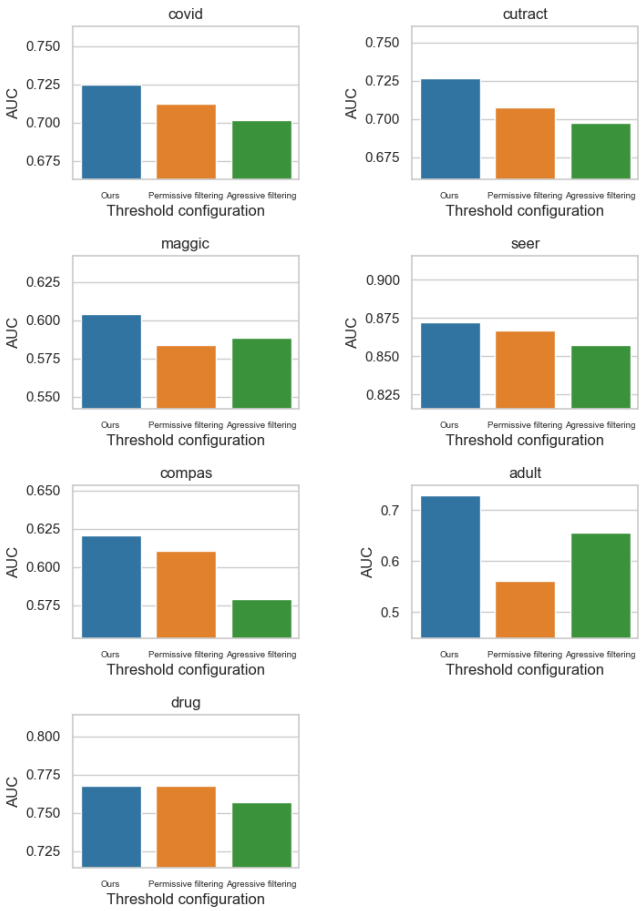

Figure 11: Assessment of the effects of different thresholds in CLLM

As we can see, our configuration strikes a good balance between the aggressive filtering and permissive filtering baselines, across the 7 datasets.

In an ideal scenario, access to an external validation set could help determine optimal thresholds. However, given our focus on the low-sample regime ($n \leq 100$), we prioritized a versatile configuration that performs consistently across datasets and sample sizes.