Appendix: Curated LLM: Synergy of LLMs and Data Curation for tabular augmentation in ultra 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.

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) aims to develop systematic tools to curate existing datasets. Our work contributes to this nascent body of work (Seedat et al., 2023) – 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 ultra 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

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 an ultra-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.

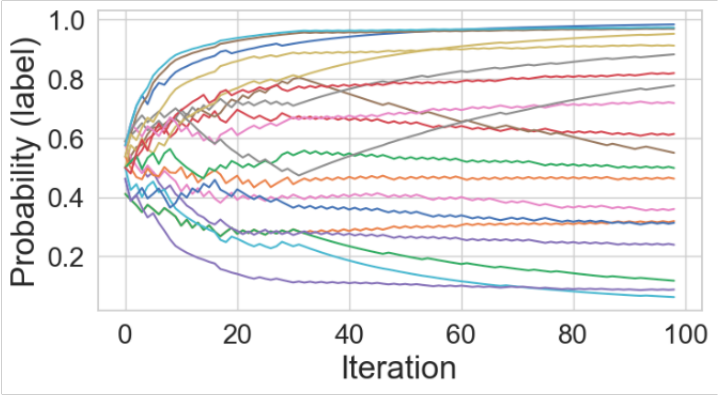

Figure 7: Learning dynamics computed for 20 samples

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 DETAILED RESULTS FOR SECTION 3.1

We report additional results for Sec. 3.1, showing the AUC for each downstream model used (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 6, 7, 8, 9.

Table 6: 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)	76.71	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)	74.96	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)	71.32	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)	90.06	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)	66.66	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)	86.04	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)	82.76	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)	76.21	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)	74.59	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)	89.94	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)	66.38	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)	85.99	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)	83.18	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)	76.31	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)	75.04	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)	71.07	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)	89.99	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)	66.22	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)	86.01	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)	82.14	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 7: 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)	77.93	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)	73.83	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)	71.45	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)	91.79	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)	69.26	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)	88.88	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)	80.63	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)	77.70	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)	73.04	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)	71.42	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)	91.81	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)	68.88	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)	88.71	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)	81.44	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)	77.56	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)	73.68	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)	71.41	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)	91.83	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)	69.04	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)	88.86	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)	80.13	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 8: 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)	64.94	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)	62.32	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)	56.85	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)	78.84	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)	61.12	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)	74.13	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)	66.19	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)	63.57	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)	62.88	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)	56.67	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)	78.48	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)	61.17	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)	74.22	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)	65.98	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)	64.28	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)	62.77	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)	56.72	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)	78.58	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)	60.82	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)	73.61	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)	65.85	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 9: 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)	80.50	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)	79.14	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)	70.36	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)	90.93	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)	72.35	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)	88.65	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)	81.26	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)	80.59	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)	78.93	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)	70.48	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)	90.93	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)	72.30	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)	88.42	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)	81.42	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)	80.62	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)	78.58	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)	70.28	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)	90.94	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)	72.47	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)	88.50	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)	81.33	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.2 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 10: 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)	75.02$_{(0.22)}$	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)	72.57$_{(0.48)}$	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$_{(1.85)}$	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)	87.90$_{(0.09)}$	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)	67.35$_{(0.16)}$	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)	84.43$_{(0.06)}$	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)	77.71$_{(0.32)}$	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)	74.52$_{(0.16)}$	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)	72.36$_{(0.47)}$	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)	67.46$_{(0.07)}$	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)	87.79$_{(0.07)}$	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)	67.18$_{(0.30)}$	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)	84.34$_{(0.07)}$	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)	78.00$_{(0.19)}$	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)	74.69$_{(0.19)}$	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)	72.52$_{(0.49)}$	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)	67.37$_{(0.06)}$	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)	87.84$_{(0.08)}$	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)	67.14$_{(0.20)}$	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)	84.25$_{(0.04)}$	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)	77.36$_{(0.51)}$	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**$_{(0.56)}$	65.29$_{(2.47)}$	75.26$_{(0.77)}$	74.38$_{(0.50)}$	74.78$_{(0.54)}$	68.39$_{(1.26)}$	74.33$_{(0.61)}$

C.3 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 lack of examples 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 in Table 11, along with the results for CLLM. From these results, we conclude the following:

1. The LLM prior permits to obtain good downstream performance, but it is outperformed by $\mathcal{D}_{\text{oracle}}$ by a margin of 4.4%. Hence, we cannot solely rely on the prior.
2. Downstream performance increases as the number of in-context samples increases. This shows it is indeed important to include the in-context examples 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 does use background knowledge of similar datasets, it still requires in-context samples to refine its prior by creating a good data model.

Table 11: 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

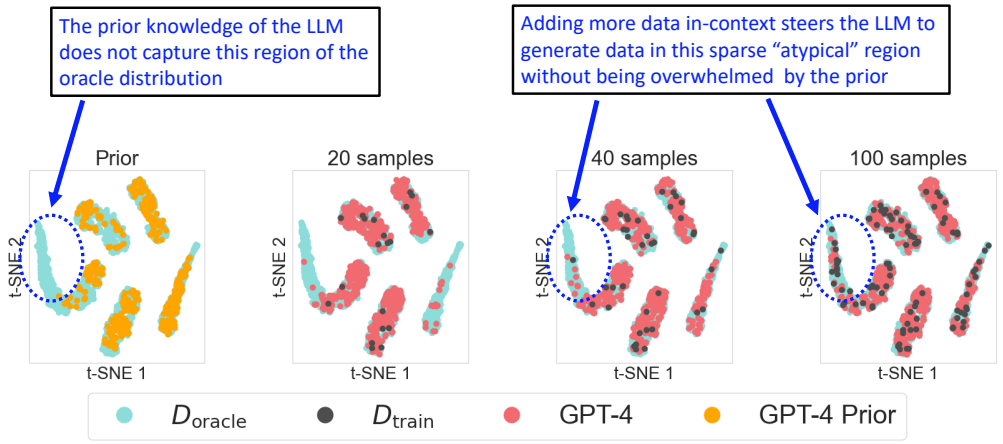

Figure 8: The data generated by the LLM captures the distinct features in "atypical" regions of the Oracle manifold, as in-context samples are added to the prompt. This shows that it is flexible enough to adapt its prior knowledge to the nuances of the data. The group encircled in blue represents patients 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.

Next, we 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 demonstrates that the LLM captures the distinct features of this particular region, and hence is not overwhelmed by the prior, but instead the data in the form of in-context samples adapts it, hence aligning the augmented dataset with the ground-truth distribution.

C.4 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 12.

Table 12: 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	$\mathbf{0.69}_{(\mathbf{0.02})}$	$\mathbf{0.88}_{(\mathbf{0.02})}$	$\mathbf{0.69}_{(\mathbf{0.02})}$	$0.27_{(0.03)}$	$\mathbf{0.89}_{(\mathbf{0.03})}$	$0.60_{(0.03)}$	$0.43_{(0.02)}$	$0.43_{(0.05)}$	$0.55_{(0.04)}$
40	$\mathbf{0.70}_{(\mathbf{0.0})}$	$\mathbf{0.92}_{(\mathbf{0.01})}$	$\mathbf{0.65}_{(\mathbf{0.03})}$	$0.31_{(0.06)}$	$0.84_{(0.03)}$	$0.57_{(0.01)}$	$0.54_{(0.02)}$	$0.80_{(0.02)}$	$0.50_{(0.04)}$
100	$\mathbf{0.69}_{(\mathbf{0.02})}$	$\mathbf{0.89}_{(\mathbf{0.02})}$	$\mathbf{0.69}_{(\mathbf{0.01})}$	$0.34_{(0.1)}$	$0.85_{(0.05)}$	$0.62_{(0.01)}$	$0.60_{(0.03)}$	$0.86_{(0.02)}$	$0.59_{(0.03)}$
200	$\mathbf{0.70}_{(\mathbf{0.01})}$	$\mathbf{0.89}_{(\mathbf{0.02})}$	$\mathbf{0.69}_{(\mathbf{0.01})}$	$0.31_{(0.05)}$	$0.87_{(0.03)}$	$0.58_{(0.05)}$	$0.65_{(0.02)}$	$0.88_{(0.01)}$	$0.63_{(0.01)}$

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

C.5 COMPARISON TO RANDOM NOISE BASELINE

We now compare to a random noise baseline. Specifically, 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

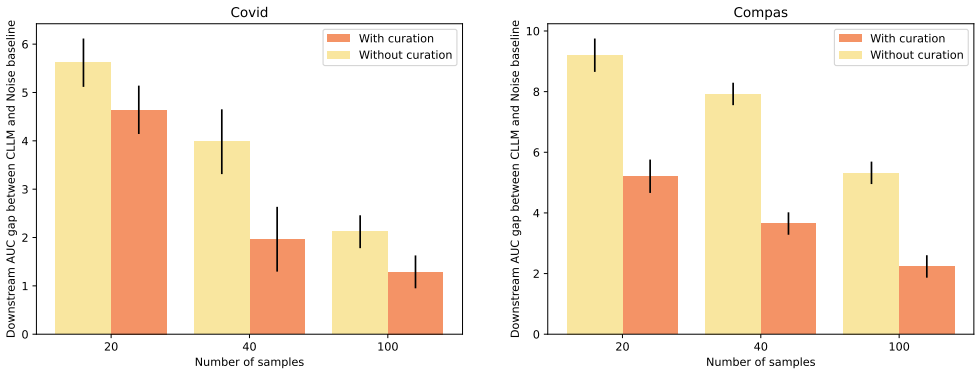

Figure 9: The random noise baseline does not match the performance of CLLM

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.