Synthetic Artifact Auditing: Tracing LLM-Generated Synthetic Data Usage in Downstream Applications

Yixin Wu[1] Ziqing Yang[1] Yun Shen[2] Michael Backes[1] Yang Zhang[1]*

[1] *CISPA Helmholtz Center for Information Security* [2] *Netapp*

Abstract

Large language models (LLMs) have facilitated the generation of high-quality, cost-effective synthetic data for developing downstream models and conducting statistical analyses in various domains. However, the increased reliance on synthetic data may pose potential negative impacts. Numerous studies have demonstrated that LLM-generated synthetic data can perpetuate and even amplify societal biases and stereotypes, and produce erroneous outputs known as "hallucinations" that deviate from factual knowledge. In this paper, we aim to audit artifacts, such as classifiers, generators, or statistical plots, to identify those trained on or derived from synthetic data and raise user awareness, thereby reducing unexpected consequences and risks in downstream applications. To this end, we take the first step to introduce synthetic artifact auditing to assess whether a given artifact is derived from LLM-generated synthetic data. We then propose an auditing framework with three methods including metric-based auditing, tuning-based auditing, and classification-based auditing. These methods operate without requiring the artifact owner to disclose proprietary training details. We evaluate our auditing framework on three text classification tasks, two text summarization tasks, and two data visualization tasks across three training scenarios. Our evaluation demonstrates the effectiveness of all proposed auditing methods across all these tasks. For instance, black-box metric-based auditing can achieve an average accuracy of 0.868 ± 0.071 for auditing classifiers and 0.880 ± 0.052 for auditing generators using only 200 random queries across three scenarios. We hope our research will enhance model transparency and regulatory compliance, ensuring the ethical and responsible use of synthetic data.[1]

1 Introduction

Large language models (LLMs) are revolutionizing data acquisition for natural language processing (NLP) tasks. Collecting high-quality data has always been a challenge due to its labor-intensive and time-consuming nature. With the advent of the LLM era, the explosive growth in training scales has rapidly increased the corresponding demand for data, further escalating these challenges. LLMs, known for their ability to generate high-quality data cost-effectively, have spurred a surge in leveraging them for synthetic data generation [27, 35, 99].

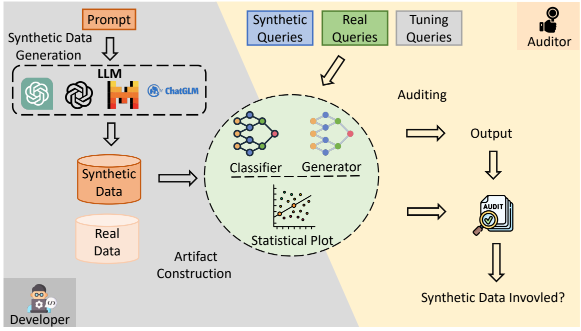


Figure 1: Overview of the synthetic artifact auditing. The auditing targets are: classifiers (Section 4), generators (Section 5), and statistical plots (Section 6).

This approach allows for tailoring training data, especially for low-resource NLP tasks like medicine and healthcare [71, 81, 82]. It can also strategically enhance training data by rebalancing under-represented classes [25, 39], and mitigate the privacy risks associated with privacy-concerned data sharing and analysis [15, 39, 41]. Building on these benefits, synthetic data has rapidly gained widespread adoption across academia and industry. It facilitates developing NLP models and conducting statistical analyses across various domains, from healthcare [16, 71, 81, 86] and law [54] to education [65] and scientific discovery [36, 98] and marketing [58]. Frameworks such as Microsoft's AgentInstruct [63] and Hazy's synthetic data generator (acquired by SAS) [7] are already being actively used in real-world settings. In practice, Microsoft post-trains Mistral-7b using synthetic data, achieving notable performance gains on multiple benchmarks [63], while

*Yang Zhang is the corresponding author.

[1] Our code is available at https://github.com/TrustAIRLab/synthetic_artifact_auditing.

OpenAI employs synthetic data generated by o1-preview to fine-tune Canvas [3].

The increasing reliance on LLMs for generating synthetic data, however, raises significant concerns regarding potential adverse impacts [26]. Numerous studies have demonstrated that LLMs inherently perpetuate or even amplify societal biases and stereotypes related to race, sex, and culture in the uncurated training data, neglecting perspectives from other regions of the world [29, 47, 91]. LLMs can further produce erroneous outputs known as "hallucinations," generating fictional text misaligned with factual knowledge [44,59,74]. With the persistent engagement of LLM-generated synthetic data in downstream training and analysis processes, the dissemination of exaggerated biases and inaccurate information could significantly undermine the reliability of decision-making processes and erode user trust. Although various studies aim to detect and mitigate bias and hallucination, existing approaches predominantly depend on the self-correction mechanisms of models [56] and external classifiers [90, 92]. Both strategies exhibit limitations in their correction and detection capabilities, making them insufficient to fully eliminate bias and hallucination. Additionally, concerns have also arisen over the potential unauthorized use of LLM-generated data, with reports [1] indicating instances where competitors leverage such data to develop competing products in violation of usage terms [10, 12]. Worse yet, given the rapid development of LLMs, it is highly conceivable that unforeseen issues may emerge in the future.

These issues highlight the necessity for a deeper investigation to determine whether LLM-generated synthetic data was involved in the construction processes of a given artifact. In response, this paper aims to audit artifacts to label those trained on or derived from synthetic data and raise users' awareness, thereby reducing unexpected consequences and risks in downstream applications. We first introduce the concept of *synthetic artifact auditing* and frame it as a binary classification task. Artifact owners typically only reveal the trained models and the analysis results to their users, while keeping the details of training data confidential. Considering that, we propose an auditing framework with three methods: metric-based auditing, tuning-based auditing, and classification-based auditing. These methods obviate the need for disclosure of proprietary training specifics. We currently focus on three types of artifacts that commonly appear in real-world applications: classifiers, generators, and statistical plots (Figure 1).

We evaluate our auditing framework on three text classification tasks, two text summarization tasks, and two data visualization tasks with four LLMs across three scenarios. In general, it can achieve good auditing performance across all tasks and scenarios. With black-box access and limited resources, it achieves an average accuracy of 0.868 ± 0.071 for auditing classifiers and 0.880 ± 0.052 for auditing generators. Meanwhile, it can also achieve 0.966 ± 0.003 average accuracy for auditing statistical plots. We attribute the high performance to the fact that these downstream synthetic artifacts can learn unique patterns from synthetic data and capture the relationships between them, the target labels, and the reference texts. In this manner, for example, synthetic classifiers trained on synthetic data exhibit more confidence than classifiers trained on real data in making predictions for synthetic data, thereby aiding in distinguishing between the two.

Contributions. We summarize our contributions as follows:

- We introduce the concept of synthetic artifact auditing. Given an artifact, it determines whether it is trained on or derived from LLM-generated synthetic data.
- We propose an auditing framework with three methods that require no disclosure of proprietary training specifics: metric-based auditing, tuning-based auditing, and classification-based auditing. This framework is extendable, currently supporting auditing for classifiers, generators, and statistical plots.
- We evaluate our auditing framework on three text classification tasks, two text summarization tasks, and two data visualization tasks across three training scenarios. The evaluation demonstrates the effectiveness of all proposed auditing methods across all these tasks.

Impact. Our work has a real-world impact, particularly in promoting the responsible use of synthetic data. Regulatory and governmental bodies are increasingly prioritizing data governance and transparency in the development of AI systems. For instance, the UK's ICO requires documentation of synthetic data creation and its properties [14]. Similarly, California recently passed Law AB 2013 [9], mandating the disclosure of training datasets, including the use of synthetic data [2]. Our framework provides a practical means for third parties to audit artifacts without requiring the disclosure of proprietary training details by artifact owners. This supports compliance with data governance and transparency requirements, enhances alignment with regulatory and legal standards, and facilitates responsible and accountable AI practices. We will open-source our code to facilitate further research.

2 Synthetic Data Generation

Synthetic data generation [19] provides a variable solution to scenarios where real data is limited due to high costs [72], privacy constraints [40], and biased distributions [46]. The fundamental objective of synthetic data generation is to produce data that is both plausible and representative of the underlying distribution observed in real data. The evolution of synthetic data generation models has been closely tied to advancements in machine learning research. Early approaches predominantly employed statistical methods such as (hidden) Markov chains [83], and n-grams [22], which excel at capturing token co-occurrences but struggle to capture nuanced semantic meanings, resulting in lower-quality generated data. With the advent of Deep Learning [33], synthetic data generation models adopt deep sequential models, such as RNN [17] and (Variational) Autoencoder [43], and more recent GANs [94]. These models, trained on larger datasets, can better comprehend token meanings and subsequently generate more realistic data. The introduction of Transformer architecture [85] has further revolutionized these efforts by utilizing attention mechanisms to model token relationships. Recent advancements in large language models (LLMs) lead to a surge in synthetic data generation for NLP tasks, such as time series [97], text [51], and code [53]).

3 Problem Statement

Auditing Scenario. We consider an auditing scenario in which auditors aim to determine whether given artifacts, such as classifiers, generators, and statistical plots, are trained on or derived from LLM-generated synthetic data. An overview of this auditing scenario is shown in Figure 1. Developers prompt LLMs to create synthetic datasets tailored to specific NLP tasks and then use the synthetic data or a combination of real and synthetic data to train models and conduct statistical analyses, such as data visualization. The auditor can be third-party regulatory agencies, downstream users, or LLM service providers investigating whether certain artifacts are derived from synthetic data.

Auditor's Capability. We consider capabilities as follows:

- *Access to the target artifact.* Auditors possess either black-box or white-box access to the auditing target. Black-box access allows querying the model via an API with input data and receiving outputs. In contrast, white-box access grants more knowledge including model architecture and parameters. These capabilities align with previous studies [42, 79, 80, 88, 89]. For artifacts like statistical plots, auditors have direct access to the targets.

- *Access to a reference real dataset.* Auditors leverage their understanding of the target artifact, including its functions and input data, to independently gather a reference real dataset and perform the same task. We primarily assume that the reference real dataset and the target real dataset originate from a similar distribution. In Section 8, we demonstrate our auditing methods still work well when they come from different distributions.

- *Access to a synthetic dataset.* Auditors can instruct LLMs to generate a synthetic dataset tailored to the functions of auditing targets. We primarily assume that the reference synthetic dataset and the target synthetic dataset are generated from the same source LLM. In Section 8, we demonstrate that our auditing methods still work well when they can come from different source LLMs.

We stress that the auditor has no direct access to the training datasets used to develop the target artifact. They also do not have access to certain training hyperparameters (e.g., epochs) considered proprietary to the artifact owners. Furthermore, the reference real data collected by auditors remains entirely independent from the target artifact. Our experimental setup reflects the above settings (see Section 4.3 for details).

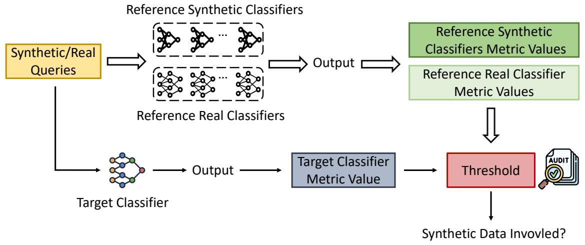

Reference Synthetic Classifiers
Synthetic/Real Queries
Output
Reference Synthetic Classifiers Metric Values
Reference Real Classifier Metric Values
Reference Real Classifiers
Target Classifier
Output
Target Classifier Metric Value
Threshold
AUDIT
Synthetic Data Invovled?

Figure 2: Overview of the metric-based auditing.

Synthetic Artifact Auditing. Formally, we formulate our auditing as a binary clarification problem. That is, given a target artifact $\mathcal{A}_{target}$ and external knowledge $\mathcal{K}$ of an auditor, synthetic artifact auditing can be defined as follows:

$$\mathcal{A}_{target}, \mathcal{K} \rightarrow \{\mathbf{0}, \mathbf{1}\}, \tag{1}$$

where **1** denotes that synthetic data was involved in the target classifiers' and generators' training procedure or used to generate target statistical plots, and **0** indicates otherwise.

Note. Building models from scratch for NLP tasks has been uncommon in recent years. Instead, developers typically fine-tune pre-trained language models (PLMs) for specific tasks. Therefore, we consider that our artifacts are fine-tuned from PLMs, such as DistilBERT [76] and BART [49]. Moreover, we do not consider the scenario where auditing targets are LLMs for the following reasons. First, in real-world applications, efficiency, cost-effectiveness, and customizability are essential. Smaller and less complex models are not only easier to train and deploy but also enable faster inference. For example, using an expensive LLM for sentiment analysis may not be financially sustainable. Therefore, the mainstream approach remains to fine-tune much smaller PLMs tailored to specific downstream tasks [35, 39, 45, 64, 65]. Second, recent studies [54, 81] show that these fine-tuned PLMs achieve overall even better performance in domain-specific tasks.

4 Classifier Auditing

4.1 Metric-Based Auditing

Intuition. We first consider the case where the auditors only have black-box access to the target classifier C_{target} and query it with input texts to obtain outputs. Recent studies [38, 55] show that LLM-generated synthetic data has unique lexical, structural, and semantic features that distinguish it from real data. Classifiers trained with such data may likely learn to recognize and leverage these distinctive features to predict labels effectively. Therefore, we hypothesize that classifiers trained with more synthetic data, referred to as *synthetic classifiers*, tend to be more confident when predicting labels for synthetic input texts and less confident with real input texts compared to *real classifiers*. Conversely, classifiers trained predominantly on real data may exhibit lower confidence when predicting labels for synthetic inputs and higher confidence with real input texts, as these real classifiers have not "seen" these distinctive synthetic features during training.

Methodology. Building upon the hypothesis of these behavior disparities, we develop a metrics-based auditing approach. It involves designing a query set $Q = \{(x_i, y_i)\}_{i=1}^{n}$, where x_i represents the input text and y_i denotes the target label, and evaluating the classifier's outputs for this query set via a performance metric to conduct auditing. We present the overview of our metric-based auditing method in Figure 2. The auditor first queries the target classifier with Q, either Q_{syn} consisting of synthetic data or Q_{real} consisting of real data, and obtains outputs. The auditor then computes the values of the performance metric for all data in the query set and compares the average values of those metrics with a certain threshold to determine whether the target classifier C_{target} is a *synthetic classifier* or a *real classifier*. More formally, we define the metric-based auditing using Q_{syn} and Q_{real} as follows:

$$\mathcal{I}_{conf}(C_{target}, Q_{syn}) = \mathbb{1}\{\frac{1}{n}\sum_{i=1}^{n} C_{target}(x_i)_{y_i} > \tau\}, \tag{2}$$

$$\mathcal{I}_{conf}(C_{target}, Q_{real}) = \mathbb{1}\{\frac{1}{n}\sum_{i=1}^{n} C_{target}(x_i)_{y_i} < \tau\}. \tag{3}$$

We leverage the average confidence score of the query set as an example of the performance metric. We can also select other performance metrics, such as the average of entropy values and accuracy. The corresponding definitions are shown in Appendix A. The auditor empirically determines the threshold τ based on their reference classifiers. Specifically, the auditor (1) trains synthetic reference classifiers ($\Delta(C_{ref}^{syn})$ $=\{C_{ref,1}^{syn}, C_{ref,2}^{syn}, \ldots, C_{ref,k}^{syn}\}$) using a mix of synthetic and real data, and real reference classifiers ($\Delta(C_{ref}^{real})$ $=\{C_{ref,1}^{real}, C_{ref,2}^{real}, \ldots, C_{ref,k}^{real}\}$) exclusively using real data; (2) then leverages the query set (Q_{syn}/Q_{real}) to obtain the reference classifiers' outputs and computes performance metric values; (3) establishes an empirical threshold value that achieves the highest accuracy in distinguishing between C_{ref}^{syn} and C_{ref}^{real}. The auditor inevitably needs to obtain training data for these reference classifiers. Since the auditor understands the exact downstream task that the target classifier performs, in turn, for C_{ref}^{real}, the auditor can independently source corresponding training data (e.g., from the Web). For C_{ref}^{syn}, the auditor can prompt LLMs with a task description to generate corresponding synthetic data. Note that the proportion of synthetic data in the training dataset of different C_{ref}^{syn} can vary. This allows the C_{ref}^{syn} to be trained solely on synthetic data or on a mix of synthetic and real data with random proportions.

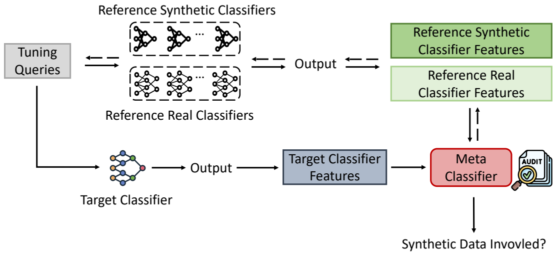

Reference Synthetic Classifiers
Tuning Queries
Output
Reference Synthetic Classifier Features
Reference Real Classifier Features
Reference Real Classifiers
Output
Target Classifier Features
Meta Classifier
AUDIT
Target Classifier
Synthetic Data Invovled?

Figure 3: Overview of the tuning-based auditing.

4.2 Tuning-Based Auditing

Intuition. When the auditor has white-box access to the audit target, they gain full visibility into the model's architecture and parameters. A common approach in such scenarios, as suggested by prior research [30], may utilize a flattened vector of all parameters as input and train a binary classifier to conduct the audit. However, our empirical analysis demonstrates that this approach yields results close to random guessing. Besides the knowledge of the model's architecture and parameters, white-box access further exempts the auditing target from being confined to processing generic text inputs, i.e., discrete tokens, enabling it to accept continuous embeddings as input. This flexibility allows us to iteratively refine an input query set from a continuous optimization space, thereby yielding more precise outputs to effectively identify behavior discrepancies between synthetic and real targets.

Methodology. We present the overview of tuning-based auditing in Figure 3. Similar to metric-based auditing, the auditor first trains a small set of synthetic reference classifiers $\Delta(C_{ref}^{syn})$ and real reference classifiers $\Delta(C_{ref}^{real})$. Given these reference classifiers, the auditor leverages a simple gradient-based approach where they directly optimize the query set Q_ϕ parameterized by ϕ and a *meta-classifier* $\mathcal{M}_{\omega_1}$ parametrized by ω_1 via backpropagation. The meta-classifier uses the output probabilities (posteriors) from a given classifier to predict its assigned label. More formally, the auditor aims to maximize the likelihood of the correct label $y \in \mathcal{Y}$, i.e., indicating whether it is a synthetic or real classifier, for the corresponding reference classifiers as follows:

$$\max_{\phi;\omega_1} P_{\omega_1;\theta;\phi}(\mathcal{Y}|\mathcal{M}_{\omega_1}(C_{ref,\theta}(Q_\phi))), \tag{4}$$

where the parameters ϕ of the query set and the parameters ω_1 of the meta-classifier are learned via back-propagation and the parameters θ of the reference classifiers are frozen. At inference time, the auditor queries the target classifier with learned Q_ϕ, and then feeds the outputs into the trained meta-classifier $\mathcal{M}_{\omega_1}$ to make the predictions. The learned Q_ϕ is a format of embedding vectors, the auditor thereby leverages white-box access to the target classifier to feed the embeddings into it.

4.3 Target and Reference Classifier Setup

We mainly consider three text classification tasks: sentiment analysis on the IMDB dataset [8] ($\mathcal{T}_{C_1}$); topic classification on the AG's news (abbreviated as AG) dataset [96] ($\mathcal{T}_{C_2}$); spam detection on the Enron-Spam dataset [60] ($\mathcal{T}_{C_3}$). We provide task details in Appendix B.1. Our overall setup for target and reference classifiers (see Figure 4) contains three primary steps. We provide a brief overview of their main objectives below, with further details elaborated in subsequent sections.

- **Data Splitting.** We prepare the real dataset and the auxiliary dataset. We ensure that both datasets remain mutually exclusive to maintain the integrity and objectivity of the evaluation process.
- **Synthetic Data Generation.** We utilize four representative large language models (LLMs) as sources, alongside two prompting strategies, to generate synthetic data. This guarantees diversity and enhances the quality of the synthetic data produced.
- **Training Scenarios.** We mainly establish three distinct training data composition scenarios for the synthetic classifier. It is designed to simulate the training process of the auditing target, forming the core setup of our evaluation.

4.3.1 Data Split

As illustrated in Figure 4, we first partition the whole dataset evenly into two disjoint subsets: the target dataset $\mathcal{D}_{target}$ and reference dataset $\mathcal{D}_{ref}$. $\mathcal{D}_{target}$ is further divided evenly into two disjoint splits as the target real dataset $\mathcal{D}^{real}_{target}$ and the target auxiliary dataset $\mathcal{D}^{aux}_{target}$. We reserve a fixed 1,000 samples in $\mathcal{D}_{ref}$ as the testing set $\mathcal{D}_{test}$, which is exclusively used to assess classifier performance. The remaining samples in $\mathcal{D}_{ref}$ are further evenly divided into two disjoint subsets $\mathcal{D}^{real}_{ref}$ and $\mathcal{D}^{aux}_{ref}$. We later randomly sample instances from $\mathcal{D}_{test}$ to construct Q_{real} and use them as reference samples for constructing Q_{syn} in a paraphrasing prompt strategy. All text classification tasks follow the above process for data split, where the specific details for each task are shown in Appendix B.2.

4.3.2 Synthetic Data Generation

Sources. Four representative LLMs are employed as the synthetic data generation sources, including `gpt-3.5-turbo-1106` (GPT-3.5 Turbo) [5], `gpt-4-0613` (GPT-4) [6], `Mistral-7B-Instruct-v0.2` (Mistral) [11], and `chatglm3-6b` (ChatGLM3) [4] as the synthetic data generation sources.

Strategies. As illustrated in Figure 4, we leverage $\mathcal{D}^{aux}_{target}$, $\mathcal{D}^{aux}_{ref}$, and a random subset of $\mathcal{D}_{test}$ to generate the target synthetic dataset $\mathcal{D}^{syn}_{target}$, reference synthetic dataset $\mathcal{D}^{syn}_{ref}$, and synthetic query set Q_{syn}. All four LLMs are utilized in this process. Specifically, with $\mathcal{D}^{aux}_{target}$, we construct four synthetic datasets $\mathcal{D}^{syn}_{target}$ containing synthetic data from GPT-3.5, GPT-4, Mistral, and ChatGLM3, respectively. To accommodate different task requirements, we employ two representative prompting strategies to generate synthetic data [51, 55, 64].

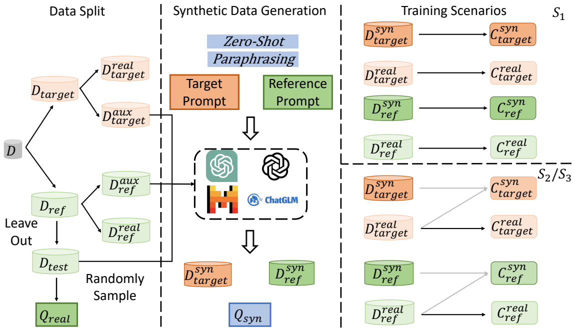


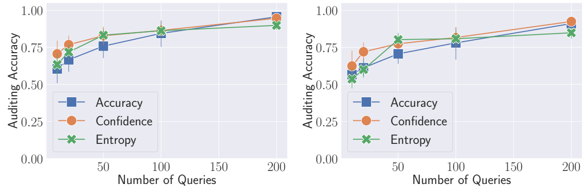
Figure 4: Overview of target/reference classifier setup.

- *Zero-shot*. We provide LLMs with labels (and additional information) and instruct them to generate content from scratch.
- *Paraphrasing*. We provide the label along with an entire input text as a reference to the LLMs and instruct them to generate new content based on the reference input.

For $\mathcal{T}_{C_1}$, we adopt the zero-shot strategy by providing movie names and outlines to ensure the quality of the generated review. The goal is to achieve comparable performance between synthetic and real classifiers. For $\mathcal{T}_{C_2}$ and $\mathcal{T}_{C_3}$, the zero-shot strategy cannot yield high-quality news articles and email messages. We thus opt for the paraphrasing strategy. Note that we do not assume that the auditor knows the prompts used to construct the target synthetic dataset $\mathcal{D}^{syn}_{target}$. We therefore use different prompts to construct Q_{syn} and $\mathcal{D}^{syn}_{ref}$ from those used for $\mathcal{D}^{syn}_{target}$ across all tasks. We experiment with prompts for synthetic data generation to ensure that synthetic artifacts achieve stable performance, ensuring the reliability of our evaluation. Further details on synthetic data generation are available in Appendix B.3.

4.3.3 Training Scenarios

Training Data for Synthetic Classifier. The use of synthetic data in real-world applications falls into two primary categories: (1) augmenting training data in low-resource scenarios [24, 39, 48, 64], and (2) generating synthetic datasets from scratch to support model training [31, 45, 51, 81]. Stemming from these applications, we consider the following three scenarios in our experimental setup:



(a) $\mathcal{S}_1$

(b) $\mathcal{S}_2$

Figure 5: Metric-based auditing performance for target classifiers fine-tuned on pre-trained DistilBERT with varying query budgets of Q_{syn} $\{10, 20, 50, 100, 200\}$ for $\mathcal{T}_{C_3}$ in (a) $\mathcal{S}_1$ and (b) $\mathcal{S}_2$. The source LLM is GPT-3.5.

- $\mathcal{S}_1$: Training exclusively on synthetic data (100%) generated from a single LLM.
- $\mathcal{S}_2$: Training on a combination of real and synthetic data from a single LLM. In evaluation, we vary the proportion of synthetic data from 10% to 100% in increments of 10%.
- $\mathcal{S}_3$: Training on a mix of real and synthetic data from multiple LLMs (all four LLMs). The proportions of synthetic data and the sources of synthetic data from different LLMs are randomized.

Training Data for Real Classifier. Reference/target real classifiers are exclusively trained on real data. C^{real}_{target} is trained solely on $\mathcal{D}^{real}_{target}$, and C^{real}_{ref} is trained solely on $\mathcal{D}^{real}_{ref}$. We consider all three scenarios for training synthetic classifiers in each task and include more details of the training scenarios in Appendix B.4.

Target/Reference Classifiers. We utilize DistilBERT [76] as the base classifier. A linear classification layer is tuned on top of these pre-trained models to predict target labels for various tasks. Our fine-tuning process employs cross-entropy loss and the Adam optimizer with a learning rate set to 2e-5. We fine-tune the classifiers of $\mathcal{T}_{C_1}$ for 5 epochs and those of $\mathcal{T}_{C_2}$ and $\mathcal{T}_{C_3}$ for 3 epochs. We develop 50 target synthetic classifiers based on each LLM in both $\mathcal{S}_1$ and $\mathcal{S}_2$. These target classifiers are used to evaluate the auditing performance. Overall, we train a total of 50 target real classifiers, 200 target synthetic classifiers in $\mathcal{S}_1$, 200 target synthetic classifiers in $\mathcal{S}_2$, and 50 target synthetic classifiers in $\mathcal{S}_3$ for each task. We also train the same number of reference classifiers to determine the threshold values and to train the meta-classifier. Both target and reference classifier sets are balanced in terms of class distribution.

Target Classifier Performance. The primary metric used to assess classifier performance (i.e., utility) is accuracy on the testing dataset. We ensure that synthetic classifiers achieve performance comparable to real classifiers.

Table 1: Average metric-based auditing performance for target classifiers fine-tuned on pre-trained DistilBERT, enabled by three different metrics and different query budgets, across three tasks in $\mathcal{S}_1$ and $\mathcal{S}_2$.

$\|Q\|$	Query Type	Accuracy	Confidence	Entropy
10	Q_{syn}	0.617 ± 0.116	$\mathbf{0.777 \pm 0.139}$	0.714 ± 0.126
	Q_{real}	0.624 ± 0.120	0.751 ± 0.142	$\mathbf{0.766 \pm 0.094}$
200	Q_{syn}	0.765 ± 0.072	$\mathbf{0.870 \pm 0.070}$	0.782 ± 0.075
	Q_{real}	0.721 ± 0.098	0.735 ± 0.120	$\mathbf{0.796 \pm 0.065}$

4.4 Auditing Setup

Auditing Model. The metric-based auditing calculates a performance metric value for each classifier and uses a reference classifier set to establish a threshold value for auditing targets. We consider accuracy (Accuracy), average confidence scores (Confidence), and average entropy values (Entropy) on the query set (Q_{syn}/Q_{real}) as the performance metrics to enable metric-based auditing. For tuning-based auditing, a meta-classifier is trained to conduct the auditing, which is a 3-layer MLP model with 32 neurons in the hidden layer. It directly takes the outputs (posteriors) from the classifiers as input. We use the cross-entropy loss and optimize it with the Adam optimizer with a learning rate of 1e-3. The meta-classifier is trained on the reference classifier set for 50 epochs.

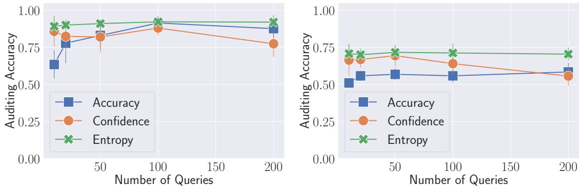
Auditing Evaluation Protocols. We ensure balanced class distributions in both the target and reference classifiers. Consequently, auditing accuracy on the target classifiers is employed as the primary metric. The target classifiers include 50 target real classifiers and 50 target synthetic classifiers for each scenario. In $\mathcal{S}_1$, the target synthetic classifiers have a 100% synthetic proportion. In $\mathcal{S}_2$, each of the ten different synthetic proportions corresponds to five classifiers. In $\mathcal{S}_3$, the target synthetic classifiers have a random synthetic proportion. Each experiment is run five times with different seeds and evaluated on all 100 target classifiers. The final average score is reported alongside its corresponding error bar.

4.5 Preliminary Investigation

In this section, we investigate the appropriate query budget, performance metric, and the number of reference classifiers to enable metric-based auditing with Q_{syn} (Section 4.5.1) and Q_{real} (Section 4.5.2) and tuning-based auditing (Section 4.5.3).

4.5.1 Metric-Based Auditing with Q_{syn}

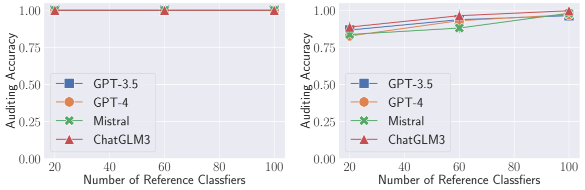
We conduct this investigation in $\mathcal{S}_1$ and $\mathcal{S}_2$. We initially leverage 10 reference real and 10 reference synthetic classifiers for each scenario. In $\mathcal{S}_1$, the synthetic classifiers have a 100% synthetic proportion. In $\mathcal{S}_2$, each synthetic classifier has one of ten different synthetic proportions. We show the metric-based auditing performance with varying query budgets in Figure 5. The varying query budgets are described as different numbers



(a) $\mathcal{S}_1$

(b) $\mathcal{S}_2$

Figure 6: Metric-based auditing performance for target classifiers fine-tuned on DistilBERT with varying query budgets of Q_{real} $\{10, 20, 50, 100, 200\}$ for $\mathcal{T}_{C_3}$. The source is GPT-3.5.



(a) $\mathcal{S}_1$

(b) $\mathcal{S}_2$

Figure 7: Tuning-based auditing performance for target classifiers fine-tuned on DistilBERT with varying numbers of reference classifiers $\{20, 60, 100\}$ for $\mathcal{T}_{C_3}$ in (a) $\mathcal{S}_1$ and (b) $\mathcal{S}_2$.

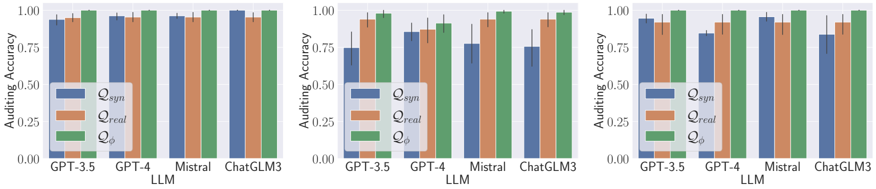
of queries, and we randomly select synthetic queries each time. We find that in both scenarios, more synthetic queries help determine a better threshold, resulting in improved auditing performance. This is reflected not only in higher accuracy but also in a smaller standard deviation. For example, the metric-based auditing using Confidence achieves only 0.624 ± 0.164 with 10 synthetic queries in $\mathcal{S}_2$, but it increases to 0.924 ± 0.046, a significant margin of increase of 0.3, with 200 queries. Meanwhile, we observe that conducting metric-based auditing with different metrics also leads to varying auditing performance. To determine which metric yields better auditing performance, we average all results conducted with 200 synthetic queries across three tasks in $\mathcal{S}_1$ and $\mathcal{S}_2$. As reported in Table 1, the average confidence scores (Confidence) enable the best auditing performance with higher accuracy and lower standard deviation when using Q_{syn}. We speculate that this is because synthetic data is generated based on their labels as conditions, resulting in features that represent the target class and a clear decision boundary between input texts of different classes. In Figure 15, it is also demonstrated that synthetic data has clearer boundaries compared to real data, and we defer more discussions in Section 6.3. Consequently, both synthetic and real classifiers tend to make the right predictions for synthetic data, and thus it is difficult to audit based on the accuracy of Q_{syn}. Furthermore, since synthetic classifiers have involved more synthetic data featuring similar characteristics during training, they display higher confidence, thereby enabling them to be distinguished from real classifiers. Meanwhile, in Appendix C, we demonstrate that 20 reference classifiers are sufficient to launch a successful metric-based auditing with Q_{syn}, and the benefit of more reference classifiers is minimal. Similar conclusions can be drawn from other source LLMs and tasks.

4.5.2 Metric-Based Auditing with Q_{real}

We follow the same evaluation setup and show the auditing performance with varying query budgets in Figure 6. The varying query budgets are described as different numbers of queries, and we randomly select real queries each time. We find that as the query budget for Q_{real} increases, the auditing performance is relatively stable. For example, metric-based auditing using Entropy achieves only 0.892 ± 0.126 with 10 real queries in S1, and it increases to 0.920 ± 0.091, a small margin of only 0.028, with 200 queries. It might be because real data, carefully handcrafted by humans, has more stable text quality compared to synthetic data, enabling a considerable auditing performance even with $|Q_{real}| = 10$ in some cases. Meanwhile, we can observe the varying auditing performance using different metrics. As illustrated in Table 1, we also average all auditing results conducted with 200 real queries. Metric-based auditing using Entropy achieves the best auditing performance, i.e., 0.766 ± 0.094 on average with 10 queries and 0.796 ± 0.065 on average with 200 queries. In addition, we observe that, with Confidence as the metric, real queries not only underperform compared to using Entropy, but their performance also deteriorates with the growth of the query budget. We attribute it to the less distinct decision boundary among real input from different classes (Figure 15). This results in both synthetic and real classifiers displaying less decisive confidence, thereby making predictions more difficult. Consequently, it is challenging to establish a threshold for synthetic artifact auditing based on Confidence which is a metric focused on a single class and leverages Accuracy. Alternatively, real queries can rely on Entropy, which measures the uncertainty across the entire probability distribution of all classes. This approach can lead to superior auditing performance. Similar to using Q_{syn}, 20 reference classifiers are also sufficient to launch auditing. Similar conclusions can be drawn on other LLMs and tasks.

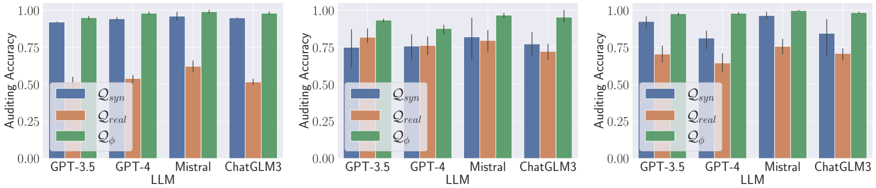
4.5.3 Tuning-Based Auditing

With white-box access, we adopt a simple gradient-based approach to learn a query set Q_ϕ, i.e., embedding vectors, to feed into the target classifier. In this section, we determine the appropriate number of tuned queries and reference classifiers to enable tuning-based auditing. We present the auditing performance with varying numbers of reference classifiers in Figure 7. The size of the tuned query set is five, i.e., $|Q_\phi| = 5$, since we demonstrate that this is sufficient to launch the tuning-based auditing. We maintain the numbers of reference real and



(a) $\mathcal{T}_{C_1}$ (b) $\mathcal{T}_{C_2}$ (c) $\mathcal{T}_{C_3}$

Figure 8: Auditing performance for target classifiers fine-tuned on the pre-trained DistilBERT model using metric-based auditing with Q_{real} and Q_{syn} and tuning-based auditing with Q_{ϕ} across three tasks and four LLMs in $\mathcal{S}_1$.



(a) $\mathcal{T}_{C_1}$ (b) $\mathcal{T}_{C_2}$ (c) $\mathcal{T}_{C_3}$

Figure 9: Auditing performance for target classifiers fine-tuned on the pre-trained DistilBERT model using metric-based auditing with Q_{real} and Q_{syn} and tuning-based auditing with Q_{ϕ} across three tasks and four LLMs in $\mathcal{S}_2$.

synthetic classifiers the same, and the synthetic proportions of synthetic classifiers are the same as those in Section 4.5.1. We observe that tuning-based auditing achieves strong performance. For example, in $\mathcal{S}_1$, the tuning-based auditing shows superior auditing performance, i.e., 1.000 ± 0.000, even with only 20 reference classifiers. Meanwhile, in $\mathcal{S}_2$, we observe that more reference classifiers (e.g., 100 reference classifiers), meaning a larger training dataset lead to better tuning-based auditing performance. Similar conclusions can be drawn on other LLMs and tasks.

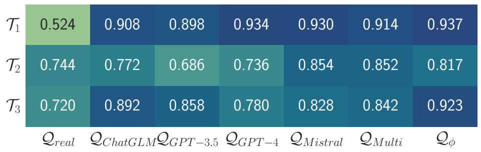
4.6 Main Evaluation

In this section, we evaluate three tasks and three scenarios. We set up the auditing method based on the results in previous sections. For metric-based auditing (Q_{syn} and Q_{real}), we set the query budget to 200. The performance metrics used are Confidence for Q_{syn} and Entropy for Q_{real}. The reference classifiers include 10 reference real classifiers and 10 reference synthetic classifiers for each scenario. In $\mathcal{S}_1$, the reference synthetic classifiers have a 100% synthetic proportion. In $\mathcal{S}_2$, we select a reference synthetic classifier for each of the ten different proportions. In $\mathcal{S}_3$, each reference synthetic classifier has a random proportion. For tuning-based auditing (Q_{ϕ}), we learn five tuned queries, and the number of reference classifiers is set to 100, including 50 real and 50 synthetic classifiers for each scenario. The synthetic proportions in $\mathcal{S}_1$ and $\mathcal{S}_3$ are the same as those for the metric-based auditing. In $\mathcal{S}_2$, we select five reference synthetic classifiers for each of the ten different proportions.

Cost. We mainly consider the cost of training reference classifiers. Training a reference classifier for $\mathcal{T}_{C_1}$, $\mathcal{T}_{C_2}$, and $\mathcal{T}_{C_3}$ costs 88.28 seconds, 65.16 seconds, and 69.45 seconds, respectively, across three training scenarios on average. The corresponding costs of metric-based auditing (20 reference classifiers) for conducting training on a Google GCP A100 are $1.82, $1.34, and $1.44, respectively, while those of tuning-based auditing (100 reference classifiers) are $9.10, $6.70, and $7.20, respectively.

Auditing Performance in $\mathcal{S}_1$. As shown in Figure 8, we observe that, in general, all proposed methods are effective across three tasks and four LLMs in $\mathcal{S}_1$. Especially for tuning-based auditing, it can achieve higher accuracy and a smaller standard deviation compared to metric-based auditing. For example, it achieves an average accuracy of 0.989 ± 0.009 in $\mathcal{S}_1$. Metric-auditing using Q_{real}, with an accuracy of 0.932 ± 0.069, follows behind yet outperforms Q_{syn}, which achieves an accuracy of 0.882 ± 0.077.

Auditing Performance in $\mathcal{S}_2$. As illustrated in Figure 9, we then report the auditing performance with the same evaluation setting in $\mathcal{S}_2$. Tuning-based auditing still achieves the best performance, with an average accuracy of 0.962 ± 0.017. In contrast, metric-based auditing with Q_{real} shows a substantial decline in performance, while Q_{syn} demonstrates greater resilience, with its auditing accuracy in $\mathcal{S}_2$ nearly matching that of $\mathcal{S}_1$. Specifically, Q_{real} achieves an average accuracy of 0.675 ± 0.060 in $\mathcal{S}_2$, reflecting a notable decrease of 0.257 compared to its performance in $\mathcal{S}_1$. In contrast, Q_{syn} achieves an average accuracy of 0.868 ± 0.077 in $\mathcal{S}_2$, only a marginal decrease of 0.014 from $\mathcal{S}_1$. We speculate that this divergence in performance may be attributed to the inclusion of real data in the training dataset for synthetic classifiers in $\mathcal{S}_2$. In $\mathcal{S}_1$,



	$\mathcal{Q}_{real}$	$\mathcal{Q}_{ChatGLM}$	$\mathcal{Q}_{GPT-3.5}$	$\mathcal{Q}_{GPT-4}$	$\mathcal{Q}_{Mistral}$	$\mathcal{Q}_{Multi}$	$\mathcal{Q}_{\phi}$
$\mathcal{T}_1$	0.524	0.908	0.898	0.934	0.930	0.914	0.937
$\mathcal{T}_2$	0.744	0.772	0.686	0.736	0.854	0.852	0.817
$\mathcal{T}_3$	0.720	0.892	0.858	0.780	0.828	0.842	0.923

Figure 10: Auditing performance for target classifiers fine-tuned on pre-trained DistilBERT across three tasks in $\mathcal{S}_3$.

the synthetic classifier (100% synthetic proportion) and the real classifier (0% synthetic proportion) exhibit clear behavior disparities with both $\mathcal{Q}_{syn}$ and $\mathcal{Q}_{real}$, thereby enabling good performance when using both types of queries. Continuing with our earlier speculation, the decision boundary between different classes in the synthetic data is clearer than in the real data, facilitating easier feature-label correlation during classifier training. Consequently, in $\mathcal{S}_2$, even a synthetic classifier with a 10% synthetic proportion can make predictions on $\mathcal{Q}_{syn}$ with higher confidence compared to a real classifier, resulting in behavior disparities. Nevertheless, a synthetic classifier with 10% synthetic proportion, i.e., 90% real data, and a real classifier (100% real data) may have similar confidence levels when making predictions on $\mathcal{Q}_{real}$, resulting in negligible behavior disparities.

Auditing Performance in $\mathcal{S}_3$. Finally, we report the auditing performance in $\mathcal{S}_3$. Synthetic classifiers are trained on a combination of real and synthetic data from multiple sources (all four LLMs in our evaluation). As such, we consider synthetic data from different source LLMs as the query to enable our metric-based auditing. Specifically, we consider constructing the query set using $\mathcal{Q}_{real}$, $\mathcal{Q}_{\phi}$, and synthetic data solely from ChatGLM ($\mathcal{Q}_{ChatGLM}$), GPT-3.5 ($\mathcal{Q}_{GPT\text{-}3.5}$), GPT-4 ($\mathcal{Q}_{GPT\text{-}4}$), Mistral ($\mathcal{Q}_{Mistral}$), and a mix of random data from all LLMs ($\mathcal{Q}_{Multi}$). As shown in Figure 10, tuning-based auditing outperforms metric-based auditing in most cases, achieving an average accuracy of 0.892 ± 0.023 on three tasks. Metric-based auditing using $\mathcal{Q}_{Mistral}$ and $\mathcal{Q}_{Multi}$ follow closely behind. The performance of $\mathcal{Q}_{real}$ remains unsatisfactory, with an accuracy of only 0.663 ± 0.061. We leverage $\mathcal{Q}_{Multi}$ as the default setting for auditing synthetic classifiers with the synthetic query set, as it can achieve decent performance in various settings.

Takeaways. In general, tuning-based auditing achieves the best performance, with an average accuracy of 0.944 ± 0.018, in all three scenarios. However, it requires white-box access to target classifiers, and the auditor needs extra training resources to train more reference classifiers and develop a meta-classifier. If the auditor only has black-box access to target classifiers or limited training resources, we recommend utilizing metric-based auditing with $\mathcal{Q}_{syn}$. This non-NN-based auditing method requires fewer reference classifiers and achieves decent auditing performance, with an average accuracy of 0.868 ± 0.071 using 200 synthetic queries and 20 reference classifiers.

5 Generator Auditing

5.1 Metric-Based Auditing

Intuition. The black-box auditing approach for generators shares similarities with auditing classifiers as discussed in Section 4.1. In this study, we focus on text summarization generators. The input text x_i used for training these models is human-crafted, i.e., real, while the corresponding output summary y_i can be synthetic. We hypothesize that generators trained using real summaries, referred to as the *real generator*, may outperform the *synthetic generator* when generating summaries for real input text. Based on this hypothesis, we utilize the real input text as the query and its corresponding real summary as the reference text (ground truth) to form the query set $\mathcal{Q}_{real}$. We employ standard performance metrics for text summarization tasks to enable metric-based auditing.

Methodology. The metric-based auditing process for the generator is similar to the process for the classifier, as illustrated in Figure 2. Here, an auditor interacts with the target generator by submitting a query set and receiving the generated summary. Subsequently, they calculate the performance metric using this generated summary alongside a reference text. A predefined threshold is utilized to classify the target generator either as a *synthetic generator* or a *real generator*. Formally, we define the metric-auditing for the target generator $\mathcal{G}_{target}$ using $\mathcal{Q}_{real}$ as follows:

$$\mathcal{I}(\mathcal{G}_{target}, \mathcal{Q}_{real}) = \mathbb{1}\{f(\mathcal{G}_{target}(x_i), y_i) < \tau, \forall (x_i, y_i) \in \mathcal{Q}_{real}\}, \tag{5}$$

where $f(\cdot)$ denotes the performance metric. The auditor empirically determines the threshold through comparisons with reference generators. The processes of collecting training data, training the reference generators $\{\mathcal{G}^{syn}_{ref,1}, \mathcal{G}^{syn}_{ref,2}, \ldots, \mathcal{G}^{syn}_{ref,k}\}$ and $\{\mathcal{G}^{real}_{ref,1}, \mathcal{G}^{real}_{ref,2}, \ldots, \mathcal{G}^{real}_{ref,k}\}$, and selecting the final threshold are identical to those used when auditing classifiers in Section 4.1.

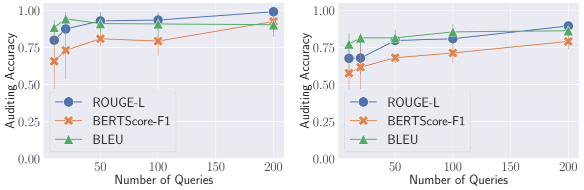
Note. Here, we do not consider tuning-based auditing. The reasons are two-fold: (1) it is difficult to assign a reference text to a tuned query; (2) for a reference-free metric, such as perplexity [73, 78], the meta classifier struggles to converge during training, as the generators only output placeholder tokens for this nonsensical query in the initial stage of training.

5.2 Evaluation Setup

We conduct two text summarization tasks on representative datasets: CNN/DM [77] ($\mathcal{T}_{\mathcal{G}_1}$) and XSum [66] ($\mathcal{T}_{\mathcal{G}_2}$). We provide the specific details for these two tasks in Appendix B.2.

5.2.1 Target and Reference Generator Setup

We primarily follow the setup in Figure 4, including the same data split, paraphrasing-based synthetic data generation, and identical training scenarios. We provide specific details in Appendix B. We use the widely adopted pre-trained model



(a) $\mathcal{S}_1$

(b) $\mathcal{S}_2$

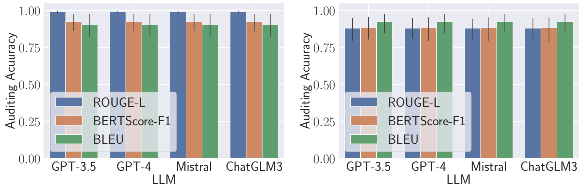
Figure 11: Metric-based auditing performance for target generators fine-tuned on pre-trained BART with varying query budgets of Q_{real} $\{10, 20, 50, 100, 200\}$ for $\mathcal{T}_{\mathcal{G}_1}$ in (a) $\mathcal{S}_1$ and (b) $\mathcal{S}_2$. The source LLM of synthetic data is GPT-3.5.

BART [49] as the backbone for *target/reference generators* in both tasks. We employ cross-entropy as the loss function and use Adam as the optimizer, with a learning rate of 2e-5. The number of beams is set to 4. We fine-tune the generators for 3 epochs. Overall, we train a total of 50 target real generators, 200 target synthetic generators in $\mathcal{S}_1$ (50 per LLM), 200 target synthetic generators in $\mathcal{S}_2$, and 50 target synthetic generators in $\mathcal{S}_3$ for each task. We also train the same number of reference generators to determine the threshold values. We ensure that target synthetic generators achieve performance comparable to target real generators.

5.2.2 Auditing Setup

Auditing Model. We only consider metric-based auditing for generators. Since it is a non-NN-based method, we calculate a performance metric value for each generator and use the reference generator set to determine a threshold value for this performance metric for auditing. We select three widely used performance metrics BERTScore [95], ROUGE [52], and BLEU [70] to enable the metric-based auditing. They all evaluate the quality of synthetic text relative to reference text but differ in focus. BERTScore measures semantic similarity using contextual embeddings from models like BERT. BLEU emphasizes precision, reflecting how much relevant information matches the reference, while ROUGE focuses on recall, indicating how much information from the reference is captured in the synthetic text.

Summarization/Auditing Evaluation Metrics. For summarization, we utilize the three standard text summarization evaluation metrics mentioned above. A higher value of the metric indicates better performance. For auditing, we balance the class distribution in target generators and reference generators, so we consider auditing accuracy on target generators as the main metric. The target generators include 50 real generators and 50 synthetic generators for each scenario, and the synthetic proportions for each scenario are the same as those in Section 4.3. Each experiment is run five times with different seeds and evaluated on all 100 target generators. We present the average score with the error bar.



(a) $\mathcal{T}_{\mathcal{G}_1}$

(b) $\mathcal{T}_{\mathcal{G}_2}$

Figure 12: Auditing performance for target generators fine-tuned on pre-trained BART using metric-based auditing with three metrics across two tasks and four source LLMs in $\mathcal{S}_1$.

5.3 Preliminary Investigation

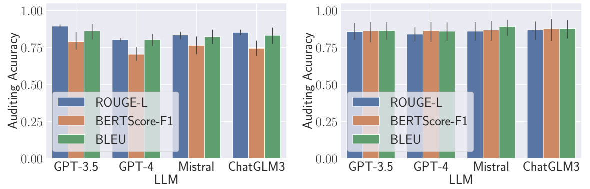
We investigate the appropriate query budget, the number of reference generators, and performance metrics to enable metric-based auditing with Q_{real}. For each scenario, we initially leverage 10 reference real generators and 10 reference synthetic generators, and the synthetic proportions are the same as those in Section 4.5.1. We start by investigating the auditing performance with varying query budgets. As shown in Figure 11, the metric-based auditing with Q_{real} achieves good performance even with 10 random real queries using all three metrics in $\mathcal{S}_1$. For example, using BLEU achieves an accuracy of 0.896 ± 0.082. Meanwhile, we find that in both two scenarios, more real queries result in better auditing performance, i.e., higher accuracy and lower standard deviation. For example, the metric-based auditing using ROUGE-L achieves 0.707 ± 0.231 with 10 queries in $\mathcal{S}_1$, but it increases to 0.924 ± 0.074 with 200 queries, a large margin of 0.217. We demonstrate that 20 reference classifiers are sufficient to find a good threshold to achieve decent auditing performance, and the benefit of more reference generators is minimal in Section C.1. Hence, we default to using all three metrics, 200 random real queries, and 20 reference classifiers to enable metric-based auditing with Q_{real}.

5.4 Main Evaluation

We evaluate two tasks and three scenarios. The query budget is set to 200. We leverage 10 reference real generators and 10 reference synthetic generators. The synthetic proportions for each scenario are the same as those in Section 4.6. We report the auditing performance for two tasks in $\mathcal{S}_1$ in Figure 12.

Costs. Training a reference generator for $\mathcal{T}_{\mathcal{G}_1}$ and $\mathcal{T}_{\mathcal{G}_2}$ costs 613.81 seconds, and 656.18 seconds across three training scenarios on average, respectively. The corresponding total costs of metric-based auditing (20 reference generators) for conducting training on a Google GCP A100 are \$12.80 and \$13.60, respectively.

Results. We observe that all metrics achieve decent auditing performance, indicating the effectiveness of our auditing methods. For example, using ROUGE-L can achieve an average



(a) $\mathcal{T}_{\mathcal{G}_1}$

(b) $\mathcal{T}_{\mathcal{G}_2}$

Figure 13: Auditing performance for target generators fine-tuned on pre-trained BART using metric-based auditing with three metrics across two tasks and four source LLMs in $\mathcal{S}_2$.

Table 2: Auditing performance for target generators using metric-based auditing with three metrics on two tasks in $\mathcal{S}_3$.

Task	ROUGE-L	BERTScore-F1	BLEU
$\mathcal{T}_{\mathcal{G}_1}$	0.780 ± 0.025	0.734 ± 0.065	0.806 ± 0.061
$\mathcal{T}_{\mathcal{G}_2}$	0.818 ± 0.064	0.840 ± 0.076	0.852 ± 0.060

accuracy of 0.990 ± 0.010 in $\mathcal{T}_{\mathcal{G}_1}$, and using BLEU can achieve an average accuracy of 0.902 ± 0.100 in $\mathcal{T}_{\mathcal{G}_2}$. Next, we present the auditing performance in $\mathcal{S}_2$ (Figure 13) and $\mathcal{S}_3$ (Table 2). We observe that our metric-based auditing still achieves good performance, even targeting synthetic generators trained on a mix of synthetic data from multiple sources and real data. Overall, these metrics consistently achieve good auditing performance across two tasks and three scenarios. ROUGE-L achieves the best performance with an average accuracy of 0.880 ± 0.052. BLEU follows closely, achieving 0.877 ± 0.081 on average, outperforms BERTScore-F1.

Takeaways. We demonstrate that the proposed metric-based auditing method using different performance metrics consistently achieves decent performance in all experiment settings.

6 Statistical Plot Auditing

6.1 Classification-Based Auditing

Intuition. Previous work [55] shows that real data and LLM-generated data from the same task represent different patterns in the statistical plots. Intuitively, we consider developing a binary image classifier $\mathcal{M}_{\omega_2}$ to determine whether the input of the given plot contains synthetic data.

Methodology. The classification-based auditing approach for statistical plots is outlined in Figure 14. To construct the training dataset of $\mathcal{M}_{\omega_2}$, the auditor first generates a set of synthetic reference plots $\{\mathcal{P}^{syn}_{ref,1}, \mathcal{P}^{syn}_{ref,2}, \dots, \mathcal{P}^{syn}_{ref,k}\}$ that contain synthetic data as input and real reference plots $\{\mathcal{P}^{real}_{ref,1}, \mathcal{P}^{real}_{ref,2}, \dots, \mathcal{P}^{real}_{ref,k}\}$ that are derived solely from real data. Specifically, we focus on data visualization, i.e., t-distributed Stochastic Neighbor Embedding (t-SNE) [84], on the text classification datasets. We directly leverage the synthetic data and real data in Section 4.1 as the input data to generate these reference plots. The composition of synthetic versus real data within each $\mathcal{P}^{syn}_{ref}$ can vary. This allows plots to be generated solely from synthetic data or from a blend of synthetic and real data in random proportions. We train the image classifier $\mathcal{M}_{\omega_2}$ parameterized by ω_2 via optimizing the following loss function:

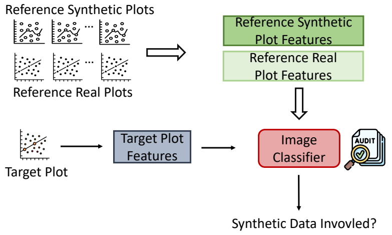


Figure 14: Overview of the classification-based auditing.

$$\mathcal{L} = \sum_{i=1}^{k} \mathbf{loss}(1, \mathcal{M}_{\omega_2}(\mathcal{P}^{syn}_{ref,i})) + \sum_{i=1}^{k} \mathbf{loss}(0, \mathcal{M}_{\omega_2}(\mathcal{P}^{real}_{ref,i})). \quad (6)$$

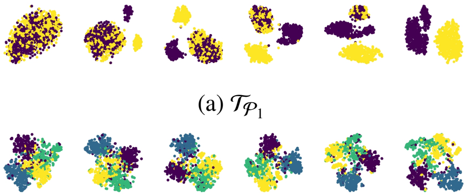
At inference time, the auditor determines a given plot by querying $\mathcal{M}_{\omega_2}$ with it and obtaining the prediction result.

6.2 Evaluation Setup

We conduct two t-distributed Stochastic Neighbor Embedding (t-SNE) visualization [84] tasks. The task $\mathcal{T}_{\mathcal{P}_1}$ employs the IMDB dataset from $\mathcal{T}_{C_1}$ and generates the synthetic data through a zero-shot prompt strategy. The task $\mathcal{T}_{\mathcal{P}_2}$ employs the AG dataset from $\mathcal{T}_{C_2}$ and generates the synthetic data using a paraphrasing strategy.

6.2.1 Target and Reference Plot Setup

We primarily follow the setup in Figure 4, including the same data split, the synthetic data generation settings of $\mathcal{T}_{C_1}$ and $\mathcal{T}_{C_2}$ from Section 4.3 to $\mathcal{T}_{\mathcal{P}_1}$ and $\mathcal{T}_{\mathcal{P}_2}$, and the same training scenario. Standard procedures for generating *target/reference t-SNE plots* are adopted based on established practices [93]. Specifically, we first preprocess all input texts by removing stopwords and punctuation. We then leverage a representative method – Word2Vec [61] – to create word embeddings. Subsequently, we use t-SNE to visualize these embeddings in two-dimensional space. The resulting plots are saved as 300×300 PNG images, showing only the scattered data points without axes, labels, or titles. Different colors are employed to indicate the target labels assigned to each data instance in the text classification task. Overall, we generate a total of 200 target real plots, 800 target synthetic plots in $\mathcal{S}_1$ (200 per LLM), 800 target synthetic plots in $\mathcal{S}_2$, and 200 target synthetic plots in $\mathcal{S}_3$ for each task. We develop the same number of reference plots to train the auditing model.



(a) $\mathcal{T}_{\mathcal{P}_1}$

(b) $\mathcal{T}_{\mathcal{P}_2}$

Figure 15: T-SNE plots using Word2Vec with synthetic proportions of input data are set at intervals of 20%, ranging from 0 to 100% (left to right) in (a) $\mathcal{T}_{\mathcal{P}_1}$ and (b) $\mathcal{T}_{\mathcal{P}_2}$. The synthetic data for $\mathcal{T}_{\mathcal{P}_1}$ and $\mathcal{T}_{\mathcal{P}_2}$ are generated using zero-shot and paraphrasing prompt strategies, respectively. Different colors denote the target labels in the text classification task.

6.2.2 Auditing Setup

Auditing Model. It is essentially an image classifier. We leverage the pre-trained RN18 [37] as the backbone of the image classifier $\mathcal{M}_{\omega_2}$. We fit a linear classifier on top of the pre-trained RN18 to conduct synthetic artifact auditing. We employ cross-entropy as the loss function and Adam as the optimizer with a learning rate of 1e-3. The model is trained for 50 epochs on the reference plots set. Note that we convert t-SNE plots into grayscale to eliminate the possibility that the auditing relies on differences in color schemes.

Auditing Evaluation Protocol. We use test accuracy on all target plots as the key metric to assess auditing performance. The target plots include 200 target real plots and 200 target synthetic plots for each scenario, and the synthetic proportions are the same as those in Section 4.3. Each experiment is run five times with different seeds, and we present the average score along with the error bar.

6.3 Main Evaluation

Figure 15 shows examples of t-SNE plots for $\mathcal{T}_{\mathcal{P}_1}$ and $\mathcal{T}_{\mathcal{P}_2}$. From left to right, the proportion of synthetic data increases from 0 to 100% at intervals of 10%. As shown in Figure 15a, when we leverage a zero-shot prompt strategy to generate synthetic data, there are clear separations between real data and synthetic data in the reduced-dimension space created by t-SNE. However, when we leverage a paraphrasing prompt strategy ($\mathcal{T}_{\mathcal{P}_2}$), the synthetic data are scattered and intertwined with real data, as shown in Figure 15b, making the auditing challenging. This observation is consistent with previous work [20, 55, 62]. We further observe that as the proportion of synthetic data increases, the distinction between data samples from different classes becomes more pronounced, leading to a more distinct decision boundary. We attribute the clearer decision boundary among synthetic data of different classes to its generation based on their labels as conditions, which results in features that highly represent the target class. Meanwhile, this clearer decision boundary distinguishes the real and synthetic data in the reduced-dimension space and thus motivates us to exploit the distinct patterns to conduct the synthetic artifact auditing.

Table 3: Auditing performance for target plots across two tasks and four LLMs in three scenarios.

Scenario	Task	LLMs			
		GPT-3.5	GPT-4	Mistral	ChatGLM3
$\mathcal{S}_1$	$\mathcal{T}_{\mathcal{P}_1}$	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
	$\mathcal{T}_{\mathcal{P}_2}$	1.000 ± 0.000	0.899 ± 0.018	1.000 ± 0.000	1.000 ± 0.000
$\mathcal{S}_2$	$\mathcal{T}_{\mathcal{P}_1}$	1.000 ± 0.000	1.000 ± 0.000	0.999 ± 0.001	0.931 ± 0.004
	$\mathcal{T}_{\mathcal{P}_2}$	0.927 ± 0.006	0.866 ± 0.012	0.956 ± 0.004	0.945 ± 0.003
$\mathcal{S}_3$	$\mathcal{T}_{\mathcal{P}_1}$	0.976 ± 0.002			
	$\mathcal{T}_{\mathcal{P}_2}$	0.882 ± 0.009			

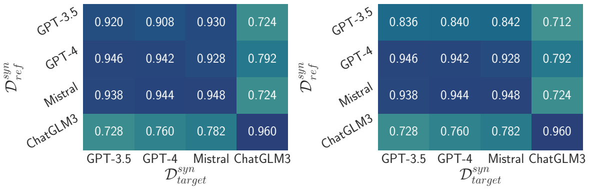
We report the auditing performance for plots using Word2Vec in Table 3. The reference plots include 200 reference real plots and 200 reference synthetic plots for each scenario, and the synthetic proportions are the same as those for target plots in Section 6.2. The classification-based auditing achieves a superior performance in both tasks. It can achieve an average accuracy of 0.990 ± 0.001 on $\mathcal{T}_{\mathcal{P}_1}$ and 0.942 ± 0.006 on $\mathcal{T}_{\mathcal{P}_2}$ across three scenarios. This demonstrates that although synthetic data for this task is generated through a paraphrasing prompt strategy and highly overlaps with real data, we can still successfully conduct auditing by distinguishing the differences in the decision boundaries between data of different classes.

Takeaways. We demonstrate that synthetic data, whether generated through zero-shot or paraphrasing, exhibit clear differences from real data, i.e., the decision boundary of data samples of different classes, and these differences result in distinct patterns on statistical plots. This signal can facilitate the synthetic artifact auditing of t-SNE plots.

7 Related Work

Synthetic Data Detection. This task can be formulated as a classification problem that distinguishes texts generated by language models (i.e., synthetic data) from those authored by humans (i.e., real data) [20, 28, 32, 34, 55, 62, 68, 87]. These methods address the distinctions between synthetic and real data by exploiting their different characteristics. Recent studies [38, 55] show that LLM-generated synthetic data has unique lexical, structural, and semantic features that distinguish it from real data. There are fine-tuning-based methods [20, 23, 55, 87] that analyze texts' latent features and train classifiers to identify synthetic data. The differences between synthetic and real data, along with the success of classifiers in identifying synthetic data, inspire us to propose a hypothesis. Classifiers trained on synthetic data tend to be more confident with synthetic inputs but less confident with real inputs, as they memorize the latent patterns of synthetic data. We then propose a metric-based auditing method grounded in this hypothesis (Section 4.1) and demonstrate its effectiveness through evaluation (Section 4.6).

Membership Inference Attacks (MIAs) [21, 50, 67, 75, 79].



(a) IMDB

(b) Rotten Tomatoes

Figure 16: Auditing performance on $\mathcal{T}_{C_1}$ with different source LLMs in $\mathcal{S}_2$. $\mathcal{D}_{target}^{real}$ is derived from IMDB. $\mathcal{D}_{ref}^{real}$ is (a) disjoint data from IMDB, and (b) data from Rotten Tomatoes.

Although both MIAs and synthetic artifact auditing employ binary classification and leverage classifier outputs (confidence scores, entropy, posteriors), they differ fundamentally in their attack targets and goals. MIAs target a given sample's membership, aiming to detect whether a specific data sample was used during training. In contrast, we target trained classifiers, generators, or plots, aiming to distinguish between artifacts trained on or derived from real versus synthetic data. Additionally, our attack processes diverge: MIAs train shadow models solely to mimic the target model's behavior, while tuning-based auditing trains reference artifacts to optimize queries. MIAs use specific queries to infer their membership, whereas we employ optimized (tuning-based) or random (metric-based) queries to audit target artifacts.

8 Discussion

Data Contamination. The PLMs used in our evaluation, i.e., BART, and DistilBERT, were trained on a corpus consisting of Wikipedia (2,500 million words) and Google's BookCorpus (800 million words) and released in 2019. Their pretraining occurred before the release of ChatGPT in 2022, which facilitated the generation of synthetic data at a massive scale. Given this timeline, we argue that it is improbable that they incorporated synthetic data in the pre-training process.

Practicability. We further explore more practical scenarios, and the experimental settings are consistent with those detailed in Section 4.6. First, we explore the scenarios where $\mathcal{D}_{ref}^{syn}$ and $\mathcal{D}_{target}^{syn}$ are generated from different source LLMs. As shown in Figure 16, our auditing framework maintains comparable performance when the source LLMs differ in most cases. We hypothesize that this outcome arises due to the inherent similarities in the synthetic data produced by different LLMs. Therefore, we recommend that the auditors incorporate multiple source LLMs when constructing $\mathcal{D}_{ref}^{syn}$. Additionally, we observe that even when the distributions of $\mathcal{D}_{ref}^{real}$ and $\mathcal{D}_{target}^{real}$ differ (Rotten Tomatoes [69] for $\mathcal{D}_{ref}^{real}$ and IMDB for $\mathcal{D}_{target}^{real}$), our approach still achieves performance comparable to when the distributions are the same (see more details in Appendix D). We then evaluate the impact of the size of the reference dataset on training each classifier. Specifically, we conduct an experiment where each reference classifier is trained using one-third of the original dataset size. In this setting, the auditing accuracy is 0.870, compared to 0.938 in the original setting with $\mathcal{T}_{C_1}$ and $\mathcal{S}_1$, suggesting that the dataset size could potentially be reduced further. Furthermore, we validate that our methods can distinguish artifacts from different LLMs (GPT-3.5, GPT-4, Mistral, and ChatGLM). For example, in $\mathcal{T}_{C_1}$ and $\mathcal{S}_1$, tuning-based auditing achieves 0.945 accuracy on this four-class classification task. These results indicate that our methods could be potentially used to infer the unauthorized use of LLM-generated data to develop competitive artifacts, which is often prohibited by tech giants [10, 13].

9 Limitations

Design Choices. We take the initial step to introduce synthetic artifact auditing and propose an auditing framework with three methods. For simplicity, we aim to present the intuition behind our methods in straightforward terms and develop auditing processes using simple yet effective design choices, such as using random synthetic queries. Our method, especially metric-based auditing, is flexible, with many different design possibilities yet to be explored. For instance, the methodology may incorporate mixing synthetic and real queries, and leveraging other performance metrics, e.g., METEOR [18] for generators. We plan to explore additional design options and further extend our auditing framework in future research.

Evaluation on LLMs and Additional Tasks. Evaluating LLMs requires training reference LLMs from scratch, but this is impractical due to infrastructure constraints. Our work remains valuable, as small models are still widely used for their efficiency, cost-effectiveness, and flexibility. Furthermore, we empirically demonstrate that synthetic artifacts capture the unique patterns of synthetic data, distinguishing them from real artifacts. As a result, our methods are expected to be feasible for LLMs and generalizable to other tasks.

10 Conclusion

In this paper, we introduce the concept of synthetic artifact auditing. We propose an auditing framework with three methods that require no disclosure of proprietary training specifics: metric-based auditing, tuning-based auditing, and classification-based auditing. This framework is extendable, currently supporting auditing for classifiers, generators, and statistical plots. We evaluate it on three text classification tasks, two text summarization tasks, and two data visualization tasks across three scenarios. The evaluation demonstrates the effectiveness of all proposed auditing methods across all these tasks. We hope our research will promote the ethical and responsible use of synthetic data.

Acknowledgments

We thank all anonymous reviewers for their constructive comments. This work is partially funded by the European Health and Digital Executive Agency (HADEA) within the project “Understanding the individual host response against Hepatitis D Virus to develop a personalized approach for the management of hepatitis D” (DSolve, grant agreement number 101057917) and the BMBF with the project “Repräsentative, synthetische Gesundheitsdaten mit starken Privatsphärengarantien” (PriSyn, 16KISAO29K).

Ethics Considerations

The datasets used in our evaluation are either publicly available or generated by LLMs, so there is a minimal or non-existent presence of personally identifiable information (PII). Hence, there is no risk of user de-anonymization, and our work does not fall under the category of human subjects research according to our Institutional Review Boards (IRB). The proposed framework is to audit digital artifacts, identify those that have been trained on or derived from synthetic data, and enhance user awareness. This effort aims to mitigate unforeseen consequences and risks in downstream applications, such as data hallucinations and inherent biases. Through our research, we aspire to improve model transparency and support regulatory compliance, promoting the ethical and responsible use of synthetic data while building trust among users and stakeholders.

Open Science

We hope our research will enhance model transparency and regulatory compliance, ensuring ethical and responsible use of synthetic data and fostering trust among users and stakeholders. Hence, we open-source our datasets and code to facilitate further research.

References

[1] ByteDance is Secretly Using OpenAI’s Tech to Build a Competitor. https://www.theverge.com/2023/12/15/24003151/bytedance-china-openai-microsoft-competitor-llm.

[2] California Passes New Generative Artificial Intelligence Law Requiring Disclosure of Training Data. https://www.mayerbrown.com/en/insights/publications/2024/09/california-passes-new-generative-artificial-intelligence-law-requiring-disclosure-of-training-data.

[3] Canvas. https://openai.com/index/introducing-canvas/.

[4] ChatGLM3. https://github.com/THUDM/ChatGLM3.

[5] GPT-3.5-Turbo. https://platform.openai.com/docs/models/gpt-3-5-turbo.

[6] GPT-4. https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4.

[7] Hazy Synthetic Data Framework. https://hazy.com/.

[8] IMDB. https://www.imdb.com/.

[9] Law AB2013. https://leginfo.legislature.ca.gov/faces/billNavClient.xhtml?bill_id=202320240AB2013.

[10] LLaMA License. https://ai.meta.com/llama/license/.

[11] Mistral-7B-Instruct-v0.2. https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2.

[12] OpenAI Business License. https://openai.com/policies/business-terms/.

[13] OpenAI License. https://openai.com/policies/row-terms-of-use/.

[14] UK ICO. https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/artificial-intelligence/explaining-decisions-made-with-artificial-intelligence/part-3-what-explaining-ai-means-for-your-organisation/documentation/.

[15] UNECE. https://unece.org/sites/default/files/2023-12/HLGMOS%20LLM%20Paper_Preprint_1.pdf.

[16] Asma Ben Abacha, Wen-wai Yim, Yadan Fan, and Thomas Lin. An Empirical Study of Clinical Note Generation from Doctor-Patient Encounters. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 2283–2294. ACL, 2023.

[17] Razvan Pascanu andTomás Mikolov and Yoshua Bengio. On the difficulty of training Recurrent Neural Networks. In *International Conference on Machine Learning (ICML)*, pages 1310–1318. JMLR, 2013.

[18] Satanjeev Banerjee and Alon Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 65–72. ACL, 2005.

[19] André Bauer, Simon Trapp, Michael Stenger, Robert Leppich, Samuel Kounev, Mark Leznik, Kyle Chard, and Ian T. Foster. Comprehensive Exploration of Synthetic Data Generation: A Survey. *CoRR abs/2401.02524*, 2024.

[20] Mazal Bethany, Brandon Wherry, Emet Bethany, Nishant Vishwamitra, and Peyman Najafirad. Deciphering Textual Authenticity: A Generalized Strategy through the Lens of Large Language Semantics for Detecting Human vs. Machine-Generated Text. *CoRR abs/2401.09407*, 2024.

[21] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership Inference Attacks From First Principles. In *IEEE Symposium on Security and Privacy (S&P)*, pages 1897–1914. IEEE, 2022.

[22] Rui Chen, Gergely Acs, and Claude Castelluccia. Differentially private sequential data publication via variable-length n-grams. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 638–649. ACM, 2012.

[23] Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. GPT-Sentinel: Distinguishing Human and ChatGPT Generated Content. *CoRR abs/2305.07969*, 2023.

[24] Bharath Chintagunta, Namit Katariya, Xavier Amatriain, and Anitha Kannan. Medically Aware GPT-3 as a Data Generator for Medical Dialogue Summarization. In *Proceedings of the Machine Learning for Healthcare Conference (MLHC)*, pages 354–372. PMLR, 2021.

[25] Emiliano De Cristofaro. What Is Synthetic Data? The Good, The Bad, and The Ugly. *CoRR abs/2303.01230*, 2023.

[26] Debarati Das, Karin de Langis, Anna Martin-Boyle, Jaehyung Kim, Minhwa Lee, Zae Myung Kim, Shirley Anugrah Hayati, Risako Owan, Bin Hu, Ritik Parkar, Ryan Koo, Jong Inn Park, Aahan Tyagi, Libby Ferland, Sanjali Roy, Vincent Liu, and Dongyeop Kang. Under the Surface: Tracking the Artifactuality of LLM-Generated Data. *CoRR abs/2401.14698*, 2024.

[27] Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. Data Augmentation using LLMs: Data Perspectives, Learning Paradigms and Challenges. *CoRR abs/2403.02990*, 2024.

[28] Leon Fröhling and Arkaitz Zubiaga. Feature-based detection of automated language models: tackling GPT-2, GPT-3 and Grover. *PeerJ Computer Science*, 2021.

[29] Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md. Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. Bias and Fairness in Large Language Models: A Survey. *CoRR abs/2309.00770*, 2023.

[30] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 619–633. ACM, 2018.

[31] Jiahui Gao, Renjie Pi, Yong Lin, Hang Xu, Jiacheng Ye, Zhiyong Wu, Weizhong Zhang, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. Self-Guided Noise-Free Data Generation for Efficient Zero-Shot Learning. In *International Conference on Learning Representations (ICLR)*, 2023.

[32] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. GLTR: Statistical Detection and Visualization of Generated Text. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 111–116. ACL, 2019.

[33] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.

[34] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. *CoRR abs/2301.07597*, 2023.

[35] Xu Guo and Yiqiang Chen. Generative AI for Synthetic Data Generation: Methods, Challenges and the Future. *CoRR abs/2403.04190*, 2024.

[36] Perttu Hämäläinen, Mikke Tavast, and Anton Kunnari. Evaluating Large Language Models in Generating Synthetic HCI Research Data: a Case Study. In *Annual ACM Conference on Human Factors in Computing Systems (CHI)*. ACM, 2023.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.

[38] Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. MGTBench: Benchmarking Machine-Generated Text Detection. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2024.

[39] Zexue He, Marco Túlio Ribeiro, and Fereshte Khani. Targeted Data Generation: Finding and Fixing Model Weaknesses. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8506–8520. ACL, 2023.

[40] Mikel Hernandez, Gorka Epelde, Ane Alberdi, Rodrigo Cilla, and Debbie Rankin. Synthetic data generation for tabular health records: A systematic review. *Neurocomputing*, 2022.

[41] Markus Hittmeir, Andreas Ekelhart, and Rudolf Mayer. On the Utility of Synthetic Data: An Empirical Evaluation on Machine Learning Tasks. In *Conference on Availability, Reliability and Security (ARES)*. ACM, 2019.

[42] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S. Yu, and Xuyun Zhang. Membership Inference Attacks on Machine Learning: A Survey. *ACM Computing Surveys*, 2021.

[43] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. Toward Controlled Generation of Text. In *International Conference on Machine Learning (ICML)*, pages 1587–1596. PMLR, 2017.

[44] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 2023.

[45] Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. Exploiting Asymmetry for Synthetic Training Data Generation: SynthIE and the Case of Information Extraction. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1555–1574. ACL, 2023.

[46] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter. Analyzing and Reducing the Damage of Dataset Bias to Face Recognition With Synthetic Data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2268. IEEE, 2019.

[47] Hadas Kotek, Rikker Dockum, and David Q. Sun. Gender bias and stereotypes in Large Language Models. In *ACM Collective Intelligence Conference (CI)*. ACM, 2023.

[48] Md. Tahmid Rahman Laskar, Mizanur Rahman, Israt Jahan, Enamul Hoque, and Jimmy Huang. CQSumDP: A ChatGPT-Annotated Resource for Query-Focused Abstractive Summarization Based on Debatepedia. *CoRR abs/2305.06147*, 2023.

[49] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7871–7880. ACL, 2020.

[50] Zheng Li and Yang Zhang. Membership Leakage in Label-Only Exposures. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 880–895. ACM, 2021.

[51] Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. Synthetic Data Generation with Large Language Models for Text Classification: Potential and Limitations. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10443–10461. ACL, 2023.

[52] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 74–81. ACL, 2004.

[53] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2023.

[54] Shuaiqi Liu, Jiannong Cao, Yicong Li, Ruosong Yang, and Zhiyuan Wen. Low-Resource Court Judgment Summarization for Common Law Systems. *CoRR abs/2403.04454*, 2024.

[55] Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. Check Me If You Can: Detecting ChatGPT-Generated Academic Writing using CheckGPT. *CoRR abs/2306.05524*, 2023.

[56] Junyu Luo, Cao Xiao, and Fenglong Ma. Zero-Resource Hallucination Prevention for Large Language Models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3586–3602. Association for Computational Linguistics, 2024.

[57] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning Word Vectors for Sentiment Analysis. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 142–150. ACL, 2011.

[58] Antoine Magron, Anna Dai, Mike Zhang, Syrielle Montariol, and Antoine Bosselut. JOBSKAPE: A Framework for Generating Synthetic Job Postings to Enhance Skill Matching. *CoRR abs/2402.03242*, 2024.

[59] Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. *CoRR abs/2303.08896*, 2023.

[60] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. Spam Filtering with Naive Bayes - Which Naive Bayes? In *Conference on Email and Anti-Spam (CEAS)*. CEAS, 2006.

[61] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *International Conference on Learning Representations (ICLR)*, 2013.

[62] Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature. *CoRR abs/2301.11305*, 2023.

[63] Arindam Mitra, Luciano Del Corro, Guoqing Zheng, Shweti Mahajan, Dany Rouhana, Andrés Codas, Yadong Lu, Weige Chen, Olga Vrousgos, Corby Rosset, Fillipe Silva, Hamed Khanpour, Yash Lara, and Ahmed Awadallah. AgentInstruct: Toward Generative Teaching with Agentic Flows. *CoRR abs/2407.03502*, 2024.

[64] Anders Møller, Arianna Pera, Jacob Aarup Dalsgaard, and Luca Maria Aiello. The Parrot Dilemma: Human-Labeled vs. LLM-augmented Data in Classification Tasks. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 179–192. ACL, 2024.

[65] Ryosuke Nakamoto, Brendan Flanagan, Taisei Yamauchi, Yiling Dai, Kyosuke Takami, and Hiroaki Ogata. Enhancing Automated Scoring of Math Self-Explanation Quality Using LLM-Generated Datasets: A Semi-Supervised Approach. *Computers*, 2023.

[66] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1797–1807. ACL, 2018.

[67] Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine Learning with Membership Privacy using Adversarial Regularization. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 634–646. ACM, 2018.

[68] Hoang-Quoc Nguyen-Son, Ngoc-Dung T. Tieu, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. Identifying computer-generated text using statistical analysis. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1504–1511. IEEE, 2017.

[69] Bo Pang and Lillian Lee. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 115–124. ACL, 2005.

[70] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318. ACL, 2002.

[71] Cheng Peng, Xi Yang, Aokun Chen, Kaleb E. Smith, Nima M. Pournejatian, Anthony B. Costa, Cheryl Martin, Mona G. Flores, Ying Zhang, Tanja Magoc, Gloria P. Lipori, Duane A. Mitchell, Naykky Singh Ospina, Mustafa M. Ahmed, William R. Hogan, Elizabeth A. Shenkman, Yi Guo, Jiang Bian, and Yonghui Wu. A study of generative large language model for medical research and healthcare. *NPJ Digital Medicine*, 2023.

[72] Florentin Poucin, Andrea Kraus, and Martin Simon. Boosting Instance Segmentation with Synthetic Data: A study to overcome the limits of real world data sets. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 945–953. IEEE, 2021.

[73] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 2019.

[74] Vipula Rawte, Amit P. Sheth, and Amitava Das. A Survey of Hallucination in Large Foundation Models. *CoRR abs/2309.05922*, 2023.

[75] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2019.

[76] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR abs/1910.01108*, 2019.

[77] Abigail See, Peter J. Liu, and Christopher D. Manning. Get To The Point: Summarization with Pointer-Generator Networks. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1073–1083. ACL, 2017.

[78] Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. In ChatGPT We Trust? Measuring and Characterizing the Reliability of ChatGPT. *CoRR abs/2304.08979*, 2023.

[79] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *IEEE Symposium on Security and Privacy (S&P)*, pages 3–18. IEEE, 2017.

[80] Liwei Song and Prateek Mittal. Systematic Evaluation of Privacy Risks of Machine Learning Models. In *USENIX Security Symposium (USENIX Security)*. USENIX, 2021.

[81] Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. Does Synthetic Data Generation of LLMs Help Clinical Text Mining? *CoRR abs/2303.04360*, 2023.

[82] Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature Medicine*, 2023.

[83] Aäron van den Oord and Benjamin Schrauwen. Factoring Variations in Natural Images with Deep Gaussian Mixture Models. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3518–3526. NIPS, 2014.

[84] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 2008.

[85] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 5998–6008. NIPS, 2017.

[86] Junda Wang, Zonghai Yao, Avijit Mitra, Samuel Osebe, Zhichao Yang, and Hong Yu. UMASS_BioNLP at MEDIQA-Chat 2023: Can LLMs generate high-quality synthetic note-oriented doctor-patient conversations? In *Clinical Natural Language Processing Workshop (ClinicalNLP)*, pages 460–471. ACL, 2023.

[87] Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohanned Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. M4GT-Bench: Evaluation Benchmark for Black-Box Machine-Generated Text Detection. *CoRR abs/2402.11175*, 2024.

[88] Yixin Wu, Rui Wen, Michael Backes, Pascal Berrang, Mathias Humbert, Yun Shen, and Yang Zhang. Quantifying Privacy Risks of Prompts in Visual Prompt Learning. In *USENIX Security Symposium (USENIX Security)*. USENIX, 2024.

[89] Yixin Wu, Ning Yu, Zheng Li, Michael Backes, and Yang Zhang. Membership Inference Attacks Against Text-to-image Generation Models. *CoRR abs/2210.00968*, 2022.

[90] Zekun Wu, Sahan Bulathwela, and Adriano Soares Koshiyama. Towards Auditing Large Language Models: Improving Text-based Stereotype. *CoRR abs/2311.14126*, 2023.

[91] Sierra Calanda Wyllie, Ilia Shumailov, and Nicolas Papernot. Fairness Feedback Loops: Training on Synthetic Data Amplifies Bias. In *Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 2113–2147. ACM, 2024.

[92] Shukang Yin, Chaoyou Fu, Sirui Zhao, Tong Xu, Hao Wang, Dianbo Sui, Yunhang Shen, Ke Li, Xing Sun, and Enhong Chen. Woodpecker: Hallucination Correction for Multimodal Large Language Models. *CoRR abs/2310.16045*, 2023.

[93] Boyang Zhang, Xinlei He, Yun Shen, Tianhao Wang, and Yang Zhang. A Plot is Worth a Thousand Words: Model Information Stealing Attacks via Scientific Plots. In *USENIX Security Symposium (USENIX Security)*. USENIX, 2023.

[94] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[95] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations (ICLR)*, 2020.

[96] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level Convolutional Networks for Text Classification. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 649–657. NIPS, 2015.

[97] Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K. Gupta, and Jingbo Shang. Large Language Models for Time Series: A Survey. *CoRR abs/2402.01801*, 2024.

[98] Yizhen Zheng, Huan Yee Koh, Jiaxin Ju, Anh T. N. Nguyen, Lauren T. May, Geoffrey I. Webb, and Shirui Pan. Large Language Models for Scientific Synthesis, Inference and Explanation. *CoRR abs/2310.07984*, 2023.

[99] Yue Zhou, Chenlu Guo, Xu Wang, Yi Chang, and Yuan Wu. A Survey on Data Augmentation in Large Model Era. *CoRR abs/2401.15422*, 2024.

A Metric-Based Auditing Using Entropy

Formally, we define the metric-based auditing using Entropy as the performance metric as follows:

$$\mathcal{I}_{entr}(C_{target}, Q_{syn}) = \mathbb{1}\{-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=0}^{c} C_{target}(x_i)_j log(C_{target}(x_i)_j) < \tau\}, \tag{7}$$

where c is the number of the target class.

$$\mathcal{I}_{entr}(C_{target}, Q_{real}) = \mathbb{1}\{-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=0}^{c} C_{target}(x_i)_j log(C_{target}(x_i)_j) > \tau\}, \tag{8}$$

B Details of Evaluation Setup

B.1 Details of Tasks

We consider three text classification tasks, two text summarization tasks, and two data visualization tasks on these five representative datasets. The details of text classification tasks are as follows:

- $\mathcal{T}_{C_1}$: We consider a classifier that classifies the sentiment of movie reviews from the IMDB website [8] into *positive* or *negative*. The IMDB dataset [57] contains 25,000 (review, sentiment label) pairs for training and 25,000 for testing.

- $\mathcal{T}_{C_2}$: We consider a classifier that categories the AG's news dataset (abbreviated as AG) [96] into four topics: *World*, *Sports*, *Business*, and *Sci/Tech*. AG is collected from over 2,000 news sources, containing 120,000 training samples and 7,600 testing samples.

- $\mathcal{T}_{C_3}$: We consider a classifier that identifies emails from the Enron-Spam dataset (abbreviated as ES) [60] as *ham* (legitimate) or *spam*. ES contains around 31,000 training samples and 2,000 testing samples.

The details of text summarization tasks are as follows:

- $\mathcal{T}_{\mathcal{G}_1}$: The CNN/DM dataset contains over 312,000 unique news articles, including 287,113 training instances, 13,368 validation instances, and 11,490 testing instances, as written by journalists at CNN and the Daily Mail. Each sample includes an article with its corresponding highlights written by the article's author.

- $\mathcal{T}_{\mathcal{G}_2}$: The XSum dataset, collected from online articles from the British Broadcasting Corporation (BBC), includes 204,045 training instances, 11,132 validation instances, and 11,334 testing instances. Each instance includes a news article with its corresponding one-sentence summary.

The details of data visualization tasks are as follows:

- $\mathcal{T}_{\mathcal{P}_1}$: This task uses the same dataset, i.e., IMDB, along with its data split and synthetic data generation settings, as $\mathcal{T}_{C_1}$. The synthetic data is generated through a zero-shot prompt method. We provide movie titles, outlines, and target labels to an LLM to generate reviews.

- $\mathcal{T}_{\mathcal{P}_2}$: This task uses the same dataset, i.e., AG, along with its data split and synthetic data generation settings, as $\mathcal{T}_{C_2}$. The synthetic data is generated using a paraphrasing prompt strategy. We provide the original article and the target label to the LLM and instruct it to rewrite a new article.

B.2 Details of Data Split

Below are details of data splitting for each task.

- $\mathcal{T}_{C_1}$ (zero-shot). We use the IMDB training set as the target real dataset $\mathcal{D}^{real}_{target}$. We leave out 1,000 samples in the IMDB testing set as $\mathcal{D}_{test}$ and use the rest as the reference real dataset $\mathcal{D}^{real}_{ref}$. We randomly sample instances from $\mathcal{D}_{test}$ to construct Q_{real}. We also leave out 1,000 samples as Q_{aux} from the retrieved movies as additional information to constructing Q_{syn} in a zero-shot prompt strategy. The rest of the retrieved movies are evenly split into $\mathcal{D}^{aux}_{target}$ and $\mathcal{D}^{aux}_{target}$.

- $\mathcal{T}_{C_2}$, $\mathcal{T}_{C_3}$, $\mathcal{T}_{\mathcal{G}_1}$, and $\mathcal{T}_{\mathcal{G}_2}$ (paraphrasing). We randomly split the AG/Enron-Spam/CNNDM/XSum training set into two evenly disjoint subsets $\mathcal{D}_{ref}$ and $\mathcal{D}_{target}$. $\mathcal{D}_{ref}$ and $\mathcal{D}_{target}$ are further divided evenly into $\mathcal{D}^{real}_{ref}$, $\mathcal{D}^{aux}_{ref}$, $\mathcal{D}^{real}_{target}$, and $\mathcal{D}^{aux}_{target}$. We randomly sample 1000 samples from the AG/Enron-Spam/CNNDM/XSum testing set to serve as $\mathcal{D}_{test}$. We then randomly sample instances from $\mathcal{D}_{test}$ to construct Q_{real} and use them as reference samples Q_{aux} for constructing Q_{syn} in a paraphrasing prompt strategy.

B.3 Details of Synthetic Data Generation

Below are details of synthetic data generation for each task.

- $\mathcal{T}_{C_1}$ (zero-shot). These auxiliary sets $\mathcal{D}^{aux}_{target}$, $\mathcal{D}^{aux}_{ref}$, and Q_{aux} consist of (movie title, outline, sentiment label) pairs. We instruct the LLM to use the reference prompt to generate a positive review and a negative review for each pair in $\mathcal{D}^{aux}_{ref}$ and Q_{aux} and use the target prompt to generate a positive review and a negative review for each pair in $\mathcal{D}^{aux}_{target}$. The final synthetic sets $\mathcal{D}^{syn}_{ref}$, Q_{syn}, and $\mathcal{D}^{syn}_{target}$ consist of (generated review, label) pairs. We set the temperature to 0.5 to balance the diversity and usability.

- $\mathcal{T}_{C_2}$ and $\mathcal{T}_{C_3}$ (paraphrasing). These auxiliary sets $\mathcal{D}^{aux}_{target}$, $\mathcal{D}^{aux}_{ref}$, and Q_{aux} consist of (input text, label) pairs. For each pair, we include each original input with its target label in the prompt and ask LLMs to paraphrase it into a new synthetic sample. We use the target prompt for constructing $\mathcal{D}^{syn}_{target}$ and the reference prompt for constructing $\mathcal{D}^{syn}_{ref}$ and Q_{syn}. The final synthetic sets $\mathcal{D}^{syn}_{ref}$, Q_{syn}, and $\mathcal{D}^{syn}_{target}$ consist of (synthetic input, label) pairs. To ensure that the synthetic examples generated by the paraphrasing strategy exhibit substantial differences from the original examples, we set the temperature to 1.

- $\mathcal{T}_{\mathcal{G}_1}$ and $\mathcal{T}_{\mathcal{G}_2}$ (paraphrasing). These auxiliary sets $\mathcal{D}^{aux}_{target}$, $\mathcal{D}^{aux}_{ref}$, and Q_{aux} consist of (article, summary) pairs. For each original pair, we include both the article and summary in the prompt and ask LLMs to paraphrase the original summary into a new synthetic sample. We use the target prompt for constructing $\mathcal{D}^{syn}_{target}$ and the reference prompt for constructing $\mathcal{D}^{syn}_{ref}$ and Q_{syn}. The final synthetic sets $\mathcal{D}^{syn}_{ref}$, Q_{syn}, and $\mathcal{D}^{syn}_{target}$ consist of (article, synthetic summary) pairs. To ensure that the synthetic examples generated by the paraphrasing strategy exhibit substantial differences from the original examples, we set the temperature to 1.

Note that, before constructing the final synthetic dataset for each task, we perform a filtering process that filters out the refusal outputs.

B.4 Details of Training Classifiers

Training Dataset Size for Each Classifier. We list the size of the training dataset for each classifier as follows:

- $\mathcal{T}_{C_1}$: We randomly sample 1,500 reviews per class (3,000 reviews in total).
- $\mathcal{T}_{C_2}$: We randomly sample 2,000 news articles per class (8,000 articles in total).
- $\mathcal{T}_{C_3}$: We randomly sample 2,000 emails per class (4,000 emails in total).

Target/Reference Classifier Set. We construct the target classifier set, which includes 50 target real classifiers $\Delta(C^{real}_{target})$ and 50 target synthetic classifiers $\Delta(C^{syn}_{target})$, to evaluate the proposed auditing methods. Meanwhile, we construct the reference classifier set, comprising 50 reference real classifiers $\Delta(C^{real}_{ref})$ and 50 reference synthetic classifiers $\Delta(C^{syn}_{ref})$, to train the meta-classifier and empirically obtain the threshold values We leverage the following setup for each task with each LLM:

- *Real classifiers:* We run the training procedure 100 times, each with a different seed, to train $\Delta(C^{real}_{target})$ on $\mathcal{D}^{real}_{target}$ and $\Delta(C^{real}_{ref})$ on $\mathcal{D}^{real}_{ref}$. Each set contains 50 classifiers.
- $\mathcal{S}_1$: We run the training procedure 100 times, each with a different seed, to train $\Delta(C^{syn}_{target})$ on $\mathcal{D}^{syn}_{target}$ and $\Delta(C^{syn}_{ref})$ on $\mathcal{D}^{syn}_{ref}$. Each set contains 50 classifiers.
- $\mathcal{S}_2$: We run the training procedure 100 times, each with a different seed, to train $\Delta(C^{syn}_{target})$ on $\mathcal{D}^{syn}_{target}$ and $\mathcal{D}^{real}_{target}$, and $\Delta(C^{syn}_{ref})$ on $\mathcal{D}^{syn}_{ref}$ and $\mathcal{D}^{real}_{ref}$. We train five classifiers for each of the ten synthetic proportions.
- $\mathcal{S}_3$: We run the training procedure 100 times, each with a different seed, to train $\Delta(C^{syn}_{target})$ on $\mathcal{D}^{syn}_{target}$ and $\mathcal{D}^{real}_{target}$, and $\Delta(C^{syn}_{ref})$ on $\mathcal{D}^{syn}_{ref}$ and $\mathcal{D}^{real}_{ref}$, with random synthetic proportions generated by different seeds. Each set contains 50 classifiers.

Note that we include all four LLMs in $\mathcal{S}_1$ and $\mathcal{S}_2$, so there are a total of 400 synthetic classifiers in $\mathcal{S}_1$, 400 synthetic classifiers in $\mathcal{S}_2$, and 100 synthetic classifiers in $\mathcal{S}_3$ for each task.

B.5 Details of Training Generators

Training Dataset Size for Each Generator. For $\mathcal{T}_{\mathcal{G}_1}$ and $\mathcal{T}_{\mathcal{G}_2}$, we randomly sample 5,000 articles as the training dataset for each generator.

Target/Reference Generator Set. We construct the target generator set, which includes 50 target real generators $\Delta(\mathcal{G}^{real}_{target})$ and 50 target synthetic generators $\Delta(\mathcal{G}^{syn}_{target})$, to evaluate the proposed auditing methods. Meanwhile, we construct the reference generator set, comprising 50 reference real generators $\Delta(\mathcal{G}^{real}_{ref})$ and 50 reference synthetic generators $\Delta(\mathcal{G}^{syn}_{ref})$, to train the meta-classifier and empirically obtain the threshold values. We leverage the following setup for each task with each LLM:

- *Real generators:* We run the training procedure 100 times, each with a different seed, to train $\Delta(\mathcal{G}^{real}_{target})$ on $\mathcal{D}^{real}_{target}$ and $\Delta(\mathcal{G}^{real}_{ref})$ on $\mathcal{D}^{real}_{ref}$. Each set contains 50 generators.
- $\mathcal{S}_1$: We run the training procedure 100 times, each with a different seed, to train $\Delta(\mathcal{G}^{syn}_{target})$ on $\mathcal{D}^{syn}_{target}$ and $\Delta(\mathcal{G}^{syn}_{ref})$ on $\mathcal{D}^{syn}_{ref}$. Each set contains 50 generators.
- $\mathcal{S}_2$: We run the training procedure 100 times, each with a different seed, to train $\Delta(\mathcal{G}^{syn}_{target})$ on $\mathcal{D}^{syn}_{target}$ and $\mathcal{D}^{real}_{target}$, and $\Delta(\mathcal{G}^{syn}_{ref})$ on $\mathcal{D}^{syn}_{ref}$ and $\mathcal{D}^{real}_{ref}$. We train five generators for each of the ten synthetic proportions.
- $\mathcal{S}_3$: We run the training procedure 100 times, each with a different seed, to train $\Delta(\mathcal{G}^{syn}_{target})$ on $\mathcal{D}^{syn}_{target}$ and $\mathcal{D}^{real}_{target}$, and $\Delta(\mathcal{G}^{syn}_{ref})$ on $\mathcal{D}^{syn}_{ref}$ and $\mathcal{D}^{real}_{ref}$, with random synthetic proportions generated by different seeds. Each set contains 50 generators.

Note that we include all four LLMs in $\mathcal{S}_1$ and $\mathcal{S}_2$, so there are a total of 400 synthetic generators in $\mathcal{S}_1$, 400 synthetic generators in $\mathcal{S}_2$, and 100 synthetic generators in $\mathcal{S}_3$ for each task.

B.6 Details of Generating Plots

Input Dataset Size for Each Plot. We randomly sample 1,000 samples from the dataset to generate t-SNE plots, and we follow the same three scenarios in Section 4.3 to control the synthetic proportion in the input data.

Target/Reference Plot Set. We construct the target plot set, which includes 200 target real plots $\Delta(\mathcal{P}^{real}_{target})$ and 200 target synthetic plots $\Delta(\mathcal{P}^{syn}_{target})$, to evaluate the proposed auditing methods. Meanwhile, we construct the reference plot set, comprising 200 reference real plots $\Delta(\mathcal{P}^{real}_{ref})$ and 200 reference synthetic plots $\Delta(\mathcal{P}^{syn}_{ref})$, to train the meta-classifier and empirically obtain the threshold values. We leverage the following setup for each task with each LLM:

- *Real plots:* We run the plotting procedure 400 times, each with a different seed, to plot $\Delta(\mathcal{P}^{real}_{target})$ on $\mathcal{D}^{real}_{target}$ and $\Delta(\mathcal{P}^{real}_{ref})$ on $\mathcal{D}^{real}_{ref}$. Each set contains 200 plots.
- $\mathcal{S}_1$: We run the plotting procedure 400 times, each with a different seed, to plot $\Delta(\mathcal{P}^{syn}_{target})$ on $\mathcal{D}^{syn}_{target}$ and $\Delta(\mathcal{P}^{syn}_{ref})$ on $\mathcal{D}^{syn}_{ref}$. Each set contains 200 plots.
- $\mathcal{S}_2$: We run the plotting procedure 400 times, each with a different seed, to plot $\Delta(\mathcal{P}^{syn}_{target})$ on $\mathcal{D}^{syn}_{target}$ and $\mathcal{D}^{real}_{target}$, and $\Delta(\mathcal{P}^{syn}_{ref})$ on $\mathcal{D}^{syn}_{ref}$ and $\mathcal{D}^{real}_{ref}$. We generate 20 plots for each of the ten synthetic proportions.
- $\mathcal{S}_3$: We run the plotting procedure 400 times, each with a different seed, to plot $\Delta(\mathcal{P}^{syn}_{target})$ on $\mathcal{D}^{syn}_{target}$ and $\mathcal{D}^{real}_{target}$, and $\Delta(\mathcal{P}^{syn}_{ref})$ on $\mathcal{D}^{syn}_{ref}$ and $\mathcal{D}^{real}_{ref}$, with random synthetic proportions generated by different seeds. Each set contains 200 plots.

Overall, for each task, we include all four LLMs in $\mathcal{S}_1$ and $\mathcal{S}_2$, resulting in a total of 400 real plots, 1600 synthetic plots in $\mathcal{S}_1$, 1600 synthetic plots in $\mathcal{S}_2$, and 400 synthetic plots in $\mathcal{S}_3$.

C Impact of Reference Classifiers

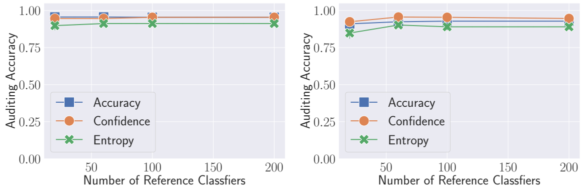
As illustrated in Figure 17 and Figure 18, we demonstrate that 20 reference classifiers are sufficient to find a good threshold to achieve decent auditing performance, and the benefit of more reference classifiers for metric-based auditing with Q_{syn}/Q_{real} is minimal.

C.1 Impact of Reference Generators

As illustrated in Figure 19, we demonstrate that 20 reference classifiers are sufficient to find a good threshold to achieve decent auditing performance, and the benefit of more reference generators for metric-based auditing with Q_{real} is minimal.

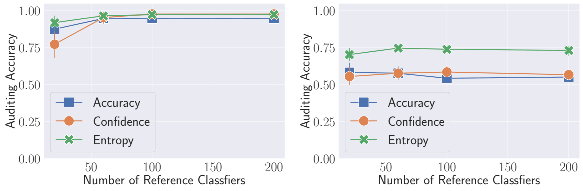
D Training Details of Reference Classifiers On Rotten Tomatoes

The Rotten Tomatoes dataset [69] contains 5,331 positive reviews and 5,331 negative reviews collected from the Rotten Tomatoes movie reviews. Similar to the original $\mathcal{T}_{C_1}$, we randomly sample 1,500 reviews per class using different seeds each time to train a classifier. The reference real classifiers are trained solely on the Rotten Tomatoes dataset. The reference synthetic classifiers are trained on a mix of the Rotten Tomatoes dataset and synthetic data from different source LLMs. The rest of the experimental settings are the same as those for $\mathcal{T}_{C_1}$ in Section 4.3.



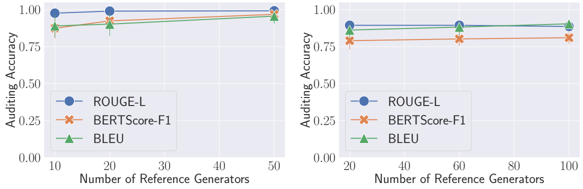
(a) $\mathcal{S}_1$ (b) $\mathcal{S}_2$

Figure 17: Metric-based auditing performance for target classifiers fine-tuned on DistilBERT with varying number of reference classifiers $\{20, 60, 100, 200\}$ in (a) $\mathcal{S}_1$ and (b) $\mathcal{S}_2$ for $\mathcal{T}_{C_3}$. $|Q_{syn}|$ is set to 200. The source LLM is GPT-3.5.



(a) $\mathcal{S}_1$ (b) $\mathcal{S}_2$

Figure 18: Metric-based auditing performance for target classifiers fine-tuned on DistilBERT with varying number of reference classifiers $\{20, 60, 100, 200\}$ in (a) $\mathcal{S}_1$ and (b) $\mathcal{S}_2$ for $\mathcal{T}_{C_3}$. $|Q_{real}|$ is set to 200. The source LLM is GPT-3.5.



(a) $\mathcal{S}_1$ (b) $\mathcal{S}_2$

Figure 19: Metric-based auditing performance for target generators with varying numbers of reference generators for $\mathcal{T}_{\mathcal{G}_1}$. The numbers are $\{10, 20, 50\}$ in (a) $\mathcal{S}_1$ and $\{20, 60, 100\}$ in (b) $\mathcal{S}_2$. The LLM is GPT-3.5.