DIPBox: A Multi-Scale Testing Framework for Tracking Dataset Regeneration

Tian Dong
Shanghai Jiao Tong University
Shanghai, China
tian.dong@sjtu.edu.cn

Yan Meng*
Shanghai Jiao Tong University
Shanghai, China
yan_meng@sjtu.edu.cn

Shaofeng Li
Southeast University
Nanjing, China
shaofengli@seu.edu.cn

Guoxing Chen
Shanghai Jiao Tong University
Shanghai, China
guoxingchen@sjtu.edu.cn

Yuling Chen
Guizhou University
Guiyang, China
ylchen3@gzu.edu.cn

Zhen Liu
Shanghai Jiao Tong University
Shanghai, China
liuzhen@sjtu.edu.cn

Haojin Zhu
Shanghai Jiao Tong University
Shanghai, China
zhu-hj@sjtu.edu.cn

Hao Chen
The University of Hong Kong
Hong Kong, China
chenho@hku.hk

Abstract

Training datasets have tremendous proprietary value and are vulnerable to unauthorized copying. Existing defenses mainly focus on tracking individual data points, but pay little attention to the threat of dataset regeneration. Through a measurement study of public tumor datasets, we identify substantial real-world partial-dataset replication, raising concerns about potential license noncompliance. To counter the challenge of tracking previously unknown adversarial regeneration, our key insight is that utility-preserving regeneration tends to preserve measurable signals across multiple feature scales. We categorize these dataset features into sample-, set-, and distribution-level features and design four similarity metrics to accurately identify regeneration. Based on these metrics, we develop DIPBox, which to our knowledge is the first testing framework that tracks regeneration suspects via multi-scale similarity testing across a spectrum of defender access settings, from limited to full information. We further provide a learning-theoretic analysis that justifies these multi-scale metrics and characterizes constraints on utility-preserving evasive regeneration. Extensive experiments on 16 vision and text base datasets, 320 regenerated datasets, and 590 derived models validate that DIPBox outperforms previous solutions while characterizing its robustness and limits under three adaptive attacks.

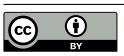
CCS Concepts

• **Security and privacy → Digital rights management**.

Keywords

Dataset regeneration, unauthorized usage, similarity testing

*Yan Meng is the corresponding author.



CCS '26, The Hague, Netherlands

ACM ISBN 979-8-4007-2871-6/2026/11
https://doi.org/10.1145/3830454.3832619

ACM Reference Format:
Tian Dong, Yan Meng, Shaofeng Li, Guoxing Chen, Yuling Chen, Zhen Liu, Haojin Zhu, and Hao Chen. 2026. DIPBox: A Multi-Scale Testing Framework for Tracking Dataset Regeneration. In *Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security (CCS '26), November 15–19, 2026, The Hague, Netherlands.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3830454.3832619

1 Introduction

Training datasets are *copyrightable* assets because of the high cost of curation and the increasing scarcity of high-quality data sources [19], but they are routinely replicated and used without the owner's consent, leading to potential copyright or license violations [57]. For example, more than 8,284 open-source datasets on Hugging Face use Creative Commons Non-Commercial licenses, among which 1,621 also prohibit derivatives, but practical compliance mechanisms are lacking [41]. Even worse, the absence of an explicit license for over 70% of datasets on GitHub and Hugging Face complicates legal accountability [41].

Existing dataset tracking approaches can be categorized into *proactive* and *passive* defenses. Proactive defenses watermark datasets prior to release via backdoors [16, 36, 37] or feature perturbations [29, 49]. Passive defenses, which apply to in-the-wild datasets, instead exploit model memorization to infer whether a model was trained on a target dataset in whole or in part [14, 17, 28, 38, 44]. A key limitation of prior art is its reliance on memorization of individual data points to track a dataset in its entirety, together with the assumption that adversaries directly reuse or at most mildly modify the pirated data. This leaves room for real-world adversaries to evade detection via *regeneration attacks* (see Figure 1), including aggressive per-datum modification, set regrouping [41], or generative reproduction (e.g., using Stable Diffusion) that yields no identical samples while preserving equivalent model-training utility [61].

To validate this threat, we first conduct a measurement study of publicly available tumor datasets and identify substantial duplication that suggests potential license noncompliance. For example, a dataset licensed to restrict derivatives and commercial use appears

Table 1: Comparison with prior work on dataset regeneration tracking. In the suspect-dataset access column, ●denotes no dataset access and ◗denotes access to a subset. In the suspect-model access column, ●denotes black-box query access and ○denotes white-box access. Commas indicate support for multiple settings.

Methods	Type	Non-Invasiveness	Access to Suspect Dataset $\mathcal{X}_{\mathcal{A}}$	Access to Suspect Model $f_{\mathcal{A}}$	Regeneration Attacks ❶ Post-processing	❷ Reorganization	❸ Polishing
Radioactive data [49], DVBW [37]	Watermarking	✗	●	○,●	✗	✗	✗
Data-use Audit [29]	Watermarking	✗	●	●	✓	✓	✗
Dataset Inference (DI) [43, 44]	Inference	✓	●	○,●	✗	✗	✗
EMA [28], MeFA [38], RAI^2 [14]	Inference	✓	●	●	✓	✓	✗
ORL-AUDITOR [17]	Inference	✓	●	●	✗	✓	✗
DIPBox (Ours)	Similarity Testing	✓	◗,●	○,●	✓	✓	✓

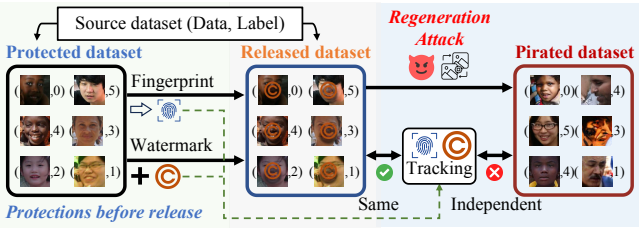


Figure 1: The adversary regenerates a derived dataset to evade detection and gain economic benefit.

to have been repackaged into four other open-source datasets, with an exact-duplicate overlap rate ranging from 49% to 100%. These measurements confirm the existence of regenerated datasets in practice and motivate our research question: *How can we robustly track dataset regeneration for datasets released on public platforms?*

Similar to other digital artifacts, the core challenge lies in handling varied and unforeseen regeneration strategies under the constraints of limited access (e.g., a pirated trial subset) or no access (e.g., only a trained model) to the full pirated dataset. To tackle this challenge, we observe that, regardless of efforts to obfuscate regeneration, an adversary typically seeks to preserve dataset quality so that equally accurate models can be trained—an objective that resembles software infringement. Consequently, inspired by software copyright protections that compare multiple components [1, 12, 59, 65–67] (e.g., code, functionality, interfaces), we adopt a multi-scale similarity testing approach that delivers robustness and extensibility to new tracking metrics.

This paper introduces DIPBox, which to our knowledge is the first framework designed for third parties (e.g., responsible sharing platforms) to track dataset regeneration via multi-scale testing against the source. We identify three core feature levels targeted by regeneration attacks: sample-level, set-level (shallow features), and distribution-level (deep features). To *quantify* feature discrepancy, in addition to existing metrics such as *output distance* [14, 17] and *sample distance* that primarily capture shallow features, we introduce two new metrics—*gradient distance* and *distribution distance*—to generalize testing across scales. We further design metric-adaptive test-case selection and a principled decision procedure, covering a broader spectrum of regeneration attacks under wider access settings compared with prior work (see Table 1) and provide a learning-theoretic analysis (see Section 4.5) that (i) justifies why utility-preserving regeneration retains detectable signals, and (ii) formalizes a utility-divergence trade-off for evasive regeneration.

We implement DIPBox as an automated toolbox to support future measurement and defense research. To evaluate, we build a regeneration benchmark covering 320 regenerated datasets derived from 16 widely used base datasets and 590 trained models spanning 12 classic and foundational model architectures. The results show that DIPBox provides accurate, evidence-based identification of regenerated datasets, remains model-agnostic, and is resilient to diverse adversarial training conditions and regeneration attacks. For instance, DIPBox can track all distribution-level regeneration variants of FairFace containing 128×128 facial images. Additionally, DIPBox surpasses state-of-the-art methods [14, 29] with at least a 25% higher TPR, due to its holistic, multi-scale characterization of dataset features. Finally, we show that in our evaluated adaptive attacks, evading DIPBox requires substantial utility loss or additional computation and memory cost.

Contribution. In summary, our contributions are:

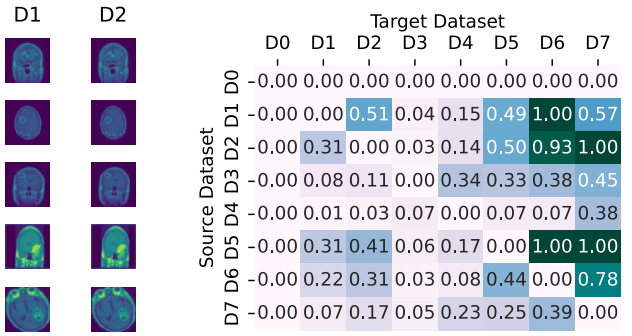
- We conduct a measurement study of real-world datasets and identify substantial duplication that suggests potential license noncompliance.
- We formalize the problem of tracking dataset regeneration and present attack primitives.
- We propose DIPBox, which to our knowledge is the first extensible framework for identifying dataset regeneration based on multi-scale testing metrics with a learning-theoretic analysis.
- We conduct extensive evaluations and implement DIPBox as an automated toolbox to support future research.

2 Motivating Measurement

We conduct a measurement of real-world datasets and identify potentially regenerated datasets. We manually screened over 74 datasets with keywords "brain" and "tumor" on Hugging Face to ensure they target the same domain and task. We filter for datasets with license "CC-BY-NC-ND-4.0", which restricts commercial use and derivative works. Then, we select datasets with less restrictive or unspecified licenses using the following criteria: (i) the dataset has been downloaded at least 10 times; (ii) the dataset is not described as derived from existing datasets; and (iii) the dataset is downloadable and usable. In the end, we gathered 8 datasets (see Table 2), annotated from "D0" to "D7", where "D0" and "D1" have licenses forbidding commercial use and derivatives, while the rest are

Table 2: Statistics of tumor datasets in the measurement.

Dataset	Size	Classes	License	Repository ID
D0	275	10	CC-BY-NC-ND-4.0	TrainingDataPro/brain-mri-dataset
D1	447	4	CC-BY-NC-ND-4.0	haydenbanz/TumorVisionDatasets
D2	800	4	Apache-2.0	thubZ9/MRI_Classification-Tumor
D3	506	/	/	miladfa7/Brain-MRI-Images-for-Brain-Tumor-Detection
D4	7342	3	/	youngp5/BrainMRI
D5	1429	2	/	tanzuhuggingface/brainmri
D6	3264	4	MIT	sartajbhuvaji/Brain-Tumor-Classification
D7	7023	4	/	Simezu/brain-tumour-MRI-scan



(a) Exact duplicate examples. **(b) Proportion of common samples in the source dataset among the measured datasets.**

Figure 2: Measurement on open-source brain tumor datasets.

more permissive or unspecified. We unify the data size to 256×256 and convert all images to grayscale.

We observe exact duplicate samples among the collected datasets. Figure 2a presents examples of common samples (L_2 distance is 0) between two datasets D1 and D2. Furthermore, for all datasets, we identify candidate duplicates by thresholding L_2 distance at 0.1 and manually verify the resulting pairs. In Figure 2b, each grid shows the proportion of duplicate samples in the source dataset. There are three notable findings:

(1) D0 has no overlapping samples with the measured datasets, which is also verified through manual checking. Considering that D0 is a trial subset for attracting users to purchase the complete dataset, it may originate from a different data source than the commonly used tumor data.
(2) Despite the more restrictive license, D1 dataset might be composed of the existing samples (e.g., subset of D6), because D1 was uploaded later than the other datasets. This might raise compliance and attribution concerns.
(3) Dataset regeneration is more prevalent among datasets with more permissive or unspecified licenses. This not only confuses users about data sources but also causes potential disputes in the future.

These findings suggest that real-world dataset regeneration can arise as sample-level or set-level reuse (Attack ❶ and ❷ defined in Section 3) and motivate us to investigate whether existing defenses remain robust under such regeneration primitives. For example, since D0 is a trial subset used to attract users, an adversary may purchase, regenerate and resell the full dataset.

3 Problem Statement

In this section, we formalize the problem of tracking regenerated datasets and present the adversary's goals and capacities.

3.1 Formulation & Threat Model

We focus on small-scale *domain datasets*, typically used for training specialized classification models [72] and widely used across the AI sector [23]. Constructing high-quality domain datasets is resource-intensive, with data source quality being crucial for ensuring model performance with real-world test data [41, 45].

We consider three parties: the victim $\mathcal{V}$, the adversary $\mathcal{A}$, and the defender (a.k.a., auditor). $\mathcal{V}$ owns a proprietary dataset $\mathcal{X}_\mathcal{V}$. The victim also trains a model $f_\mathcal{V}$ on $\mathcal{X}_\mathcal{V}$. The value of $\mathcal{X}_\mathcal{V}$ lies in the fact that its distribution $\mathcal{P}_\mathcal{V}$ closely matches the real-world distribution, leading to models that achieve higher accuracy on live test data $\mathcal{P}_t$ after deployment, i.e., $\mathcal{X}_\mathcal{V} \sim \mathcal{P}_\mathcal{V} \approx \mathcal{P}_t$. For the adversary $\mathcal{A}$, directly collecting in-distribution data is infeasible due to exclusive data sources [5] or high cost. We assume $\mathcal{A}$ has access only to a distribution $\mathcal{P}_\mathcal{A}^{(0)}$ not identically distributed as $\mathcal{P}_t$ [73], thus the model $f_{\mathcal{P}_\mathcal{A}^{(0)}}$ trained on $\mathcal{P}_\mathcal{A}^{(0)}$ achieves at least $\varepsilon^u_{\mathcal{V},\mathcal{A}}$ lower test accuracy:

$$\mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{P}_t}[v(f_{\mathcal{P}_\mathcal{A}^{(0)}}(\mathbf{x}),y)] < \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{P}_t}[v(f_\mathcal{V}(\mathbf{x}),y)] - \varepsilon^u_{\mathcal{V},\mathcal{A}}, \quad (1)$$

where v is the utility function and $\mathbb{E}$ is the expectation.

We denote the adversary's auxiliary dataset drawn from $\mathcal{P}_\mathcal{A}^{(0)}$ as $\mathcal{X}_\mathcal{A}^{(0)} \sim \mathcal{P}_\mathcal{A}^{(0)}$.

We assume the adversary obtains $\mathcal{X}_\mathcal{V}$ through leakage [15] or other unauthorized means, and then crafts and distributes $\mathcal{X}_\mathcal{A}$ on platforms for profit. The adversary also knows the existing protections [29, 36, 37], thus cannot directly reuse $\mathcal{X}_\mathcal{V}$; instead, they regenerate their dataset $\mathcal{X}_\mathcal{A}$ (of distribution $\mathcal{P}_\mathcal{A}$) based on $\mathcal{X}_\mathcal{V}$ and train their model $f_\mathcal{A}$. The adversary can perform hyperparameter tuning to maximize the accuracy of $f_\mathcal{A}$. Formally, we model the regeneration and tracking as an adversarial game:

(1) Initially, the adversary has $\mathcal{X}_\mathcal{A}^{(0)}$ and $\mathcal{X}_\mathcal{V}$.
(2) The adversary regenerates $\mathcal{X}_\mathcal{A} \leftarrow \mathcal{A}(\mathcal{X}_\mathcal{V}, \mathcal{X}_\mathcal{A}^{(0)})$.
(3) The defender calculates $d_{\mathcal{V},\mathcal{A}} \leftarrow \mathrm{DIST}(\mathcal{X}_\mathcal{V}, \mathcal{X}_\mathcal{A})$.
(4) The adversary wins if $d_{\mathcal{V},\mathcal{A}} > \tau_{\mathcal{V},\mathcal{A}}$ and achieves:

$$\left| \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{P}_t}[v(f_\mathcal{A}(\mathbf{x}),y)] - \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{P}_t}[v(f_\mathcal{V}(\mathbf{x}),y)] \right| < \varepsilon^u_{\mathcal{V},\mathcal{A}}, \quad (2)$$

where v is the dataset utility function, $\varepsilon^u_{\mathcal{V},\mathcal{A}}$ is the utility drop limit, $\tau_{\mathcal{V},\mathcal{A}}$ is the predefined threshold by $\mathcal{V}$.

In this formulation, $\mathrm{DIST}(\cdot,\cdot)$ is a generic distance-based auditing procedure; DIPBox instantiates it using multiple distances and a joint decision rule (see Section 4). The adversary aims to win the game by:

- *Preserving Utility*: $f_\mathcal{A}$ has comparable accuracy on $\mathcal{P}_t$.
- *Evading Defense*: $\mathcal{X}_\mathcal{A}$ cannot be judged as a copy of $\mathcal{X}_\mathcal{V}$.

3.2 Attack Primitives

To satisfy Eq. 2, an adversary must perturb the dataset while ensuring its utility. Motivated by software infringement analysis, which leverages both shallow (e.g., layout) and inherent (e.g., source code)

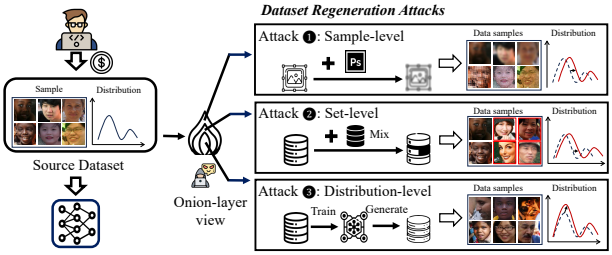


Figure 3: Overview of the regeneration attack primitives on our identified dataset features. The dataset has an onion-layered structure, where the surface-level feature (e.g., datum) is the skin and the inner feature (i.e., distribution) represents the onion core. The adversary can construct a pirated dataset by regeneration attacks at three levels.

features, we propose a top-down dataset feature hierarchy. As shown in Figure 3, we distinguish shallow features at the sample and set levels from deep features at the distribution level, and categorize attack primitives accordingly into sample-, set-, and distribution-level primitives.

- Attack ❶: Post-processing Attack. The adversary post-processes (e.g., via Gaussian blurring) all samples in $\mathcal{X}_{\mathcal{V}}$ to prevent the auditor from identifying copied samples.
- Attack ❷: Reorganization Attack. The adversary changes the dataset organization by combining a subset of $\mathcal{X}_{\mathcal{V}}$ with curated data following $\mathcal{P}_{\mathcal{A}}^{(0)}$ to build $\mathcal{X}_{\mathcal{A}}$.
- Attack ❸: Polishing Attack. The adversary uses a generator (i.e., generative models trained on $\mathcal{X}_{\mathcal{V}}$ or public foundation models) to learn $\mathcal{P}_{\mathcal{V}}$. The generator produces $\mathcal{X}_{\mathcal{A}}$ from the same distribution as $\mathcal{X}_{\mathcal{V}}$ which has no identical sample but can train models of comparably high test accuracy [52, 61].

We find that dataset regenerations can generally be decomposed into a combination of three attack primitives. However, combining them (e.g., combining ❸ and ❶) often compromises a dataset's utility (see Section 5). Therefore, instead of studying these complex combinations, we focus on individual primitives and vary their intensity to evaluate our tracking framework's robustness. Recognizing that new primitives may appear in the future, we design our framework to be *extensible* (see Section 4).

4 System Design of DIPBox

In this section, we elaborate DIPBox's design goals, defender assumptions, and workflow overview.

4.1 Overview

4.1.1 Design Goals. DIPBox is tailored for tracking datasets from exclusive or scarce sources because of their proprietary value and the low likelihood of a coincidental match, which reflects a broader industry trend. DIPBox can serve as a technical tool for auditors to support downstream legal assessment. Moreover, DIPBox can automatically screen uploaded datasets for potential regeneration attempts, analogous to automated plagiarism detection on preprint platforms. To sum up, the design goals include:

Algorithm 1: DIPBox auditing workflow

Input: Victim's model $f_{\mathcal{V}}$ and dataset $\mathcal{X}_{\mathcal{V}}$, suspect model $f_{\mathcal{A}}$ and subset $\mathcal{X}_{\mathcal{A}}^{s}$, audit set sizes N_{audit}^{M} and N_{audit}^{D}, random seed $seed$, access mode $\mathcal{M}_{\mathcal{A}}$, and judge requirement $R_{\mathcal{J}}$
Output: Judgment $\mathcal{J}$
// Audit set generation (Section 4.2)
1 $\mathcal{X}_{\mathcal{J}}^{M}, \mathcal{X}_{\mathcal{J}}^{D} \leftarrow$ GenerateAuditSet $(f_{\mathcal{V}}, \mathcal{X}_{\mathcal{V}}, N_{audit}^{M}, N_{audit}^{D}, seed)$
// Similarity metrics (Section 4.3)
2 $\mathbf{s}_{\mathcal{J}} \leftarrow$ SimilarityChecking $(\mathcal{X}_{\mathcal{J}}^{M}, \mathcal{X}_{\mathcal{J}}^{D}, f_{\mathcal{V}}, (f_{\mathcal{A}}, \mathcal{X}_{\mathcal{A}}^{s}), \mathcal{M}_{\mathcal{A}})$
// Judge derived dataset (Section 4.4)
3 $\mathcal{J} \leftarrow$ Judging $(\mathbf{s}_{\mathcal{J}}, R_{\mathcal{J}})$
4 **return** $\mathcal{J}$

- **Non-intrusiveness**: The tracking should not damage the source dataset *utility*.
- **Robustness**: The regeneration threat should be accurately identified as long as the model-training utility is preserved.
- **Efficiency**: The tracking algorithm is of low complexity and only requires a small proportion of audited datasets.
- **Extensibility**: The framework should easily incorporate newer tracking metrics and should work under a variety of access levels to $f_{\mathcal{A}}$ and $\mathcal{X}_{\mathcal{A}}$.

4.1.2 Defender Assumptions. To support varying access settings, we consider the weakest defense assumption (i.e., black-box access to $f_{\mathcal{A}}$ and no access to $\mathcal{X}_{\mathcal{A}}$) and provide more accurate tracking as the defender capacity improves (i.e., white-box access to $f_{\mathcal{A}}$ and a subset of $\mathcal{X}_{\mathcal{A}}$). We clarify the defender capacity as follows:

(1) *Access to the suspect model* $f_{\mathcal{A}}$. We consider two possibilities: 1) black-box access. The defender can only query $f_{\mathcal{A}}$ to obtain confidence scores. 2) white-box access. The defender (e.g., platform) can acquire $f_{\mathcal{A}}$ weights.
(2) *Access to the suspect dataset* $\mathcal{X}_{\mathcal{A}}$. Similarly, the defender has 1) no access to $\mathcal{X}_{\mathcal{A}}$'s samples, and 2) white-box access to the suspect sub-dataset $\mathcal{X}_{\mathcal{A}}^{s}$, which is assumed to be uniformly sampled from $\mathcal{X}_{\mathcal{A}}$.

Note that white-box access to a subset $\mathcal{X}_{\mathcal{A}}^{s}$ is realistic. For instance, current open-source or commercial platforms often provide a free preview subset (e.g., D0 in Table 2). Moreover, platforms or other parties responsible for copyright and license compliance [64], can, subject to sharer authorization or regulatory mandates, leverage privacy-preserving computing techniques (e.g., Trusted Execution Environments (TEEs) [39]) to obtain partial or complete access to the dataset.

4.1.3 Workflow of DIPBox. As Figure 4 shows, DIPBox consists of three modules and four similarity metrics spanning sample-, set-, and distribution-level features. Algorithm 1 depicts the procedure of DIPBox. The *Audit Set Creation* module generates two audit sets $\mathcal{X}_{\mathcal{J}}^{M}$ and $\mathcal{X}_{\mathcal{J}}^{D}$ that are used by model-access and dataset-access metrics, respectively (Line 1). To ensure efficiency, the audit sets should be small-sized and represent the source dataset. To ensure extensibility, the *Similarity Calculation* module contains multiple metrics (Line 2) that depend on the defender's access to $(f_{\mathcal{A}}, \mathcal{X}_{\mathcal{A}})$, denoted by access mode $\mathcal{M}_{\mathcal{A}}$. For each metric supported under $\mathcal{M}_{\mathcal{A}}$, the defender computes a metric value (a distance; smaller

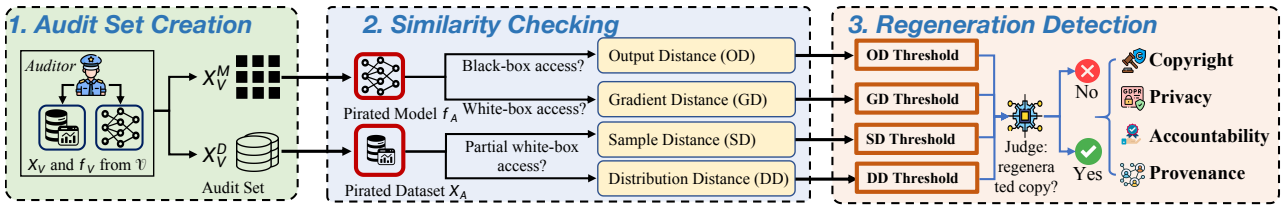
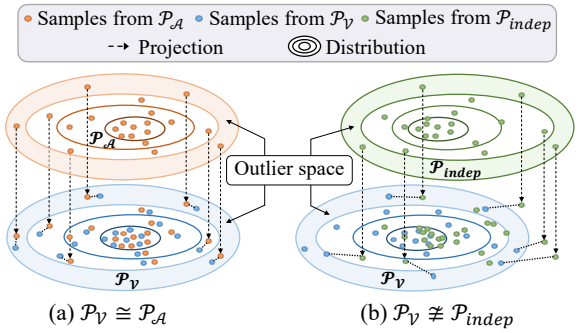


Figure 4: The workflow of DIPBox. With the created audit set, the auditor measures the similarity scores based on their access to the target model/dataset. The final judgment is based on thresholding and applies to multiple downstream investigations.



(a) $\mathcal{P}_{\mathcal{V}} \cong \mathcal{P}_{\mathcal{A}}$ (b) $\mathcal{P}_{\mathcal{V}} \ncong \mathcal{P}_{indep}$

Figure 5: Intuition of our audit set generation. The distribution of a regenerated dataset is close to the victim's distribution ($\mathcal{P}_{\mathcal{V}} \cong \mathcal{P}_{\mathcal{A}}$). On the other hand, for an independent dataset, the underlying distribution $\mathcal{P}_{indep}$ has larger distance to the victim's distribution ($\mathcal{P}_{\mathcal{V}} \ncong \mathcal{P}_{indep}$). We select the outlier points (i.e., points on the shaded area) for further testing as they characterize the distribution contour.

indicates higher similarity) and then thresholds it in the next module. Finally, the *Regeneration Judgment* module accomplishes the judgment based on the comprehensive evaluation of requirements $R_{\mathcal{J}}$, which will be elaborated in Section 4.4. The decision is accompanied by per-metric scores, which help interpret which feature scale drives the decision.

4.2 Audit Set Creation

The defender first generates an audit set from $\mathcal{X}_{\mathcal{V}}$. The audit set should be small to reduce computing cost and to well characterize the source dataset $\mathcal{X}_{\mathcal{V}}$ so that the derived datasets can be more accurately detected and independent datasets are less likely to be mistaken as a copy. We consider two audit sets $\mathcal{X}_{\mathcal{J}}^D$ and $\mathcal{X}_{\mathcal{J}}^M$ used to calculate similarities for dataset metrics and model metrics, respectively. The audit-set sizes are denoted by N_{audit}^D and N_{audit}^M for $\mathcal{X}_{\mathcal{J}}^D$ and $\mathcal{X}_{\mathcal{J}}^M$, respectively. In Algorithm 1, we implement these steps as $\texttt{GenerateAuditSet}(f_{\mathcal{V}}, \mathcal{X}_{\mathcal{V}}, N_{audit}^M, N_{audit}^D, seed)$ that outputs $(\mathcal{X}_{\mathcal{J}}^M, \mathcal{X}_{\mathcal{J}}^D)$. When the defender has access to the suspect sub-dataset $\mathcal{X}_{\mathcal{A}}^s$, we uniformly sample from the victim dataset, i.e., $\mathcal{X}_{\mathcal{J}}^D = \texttt{Uniform}(\mathcal{X}_{\mathcal{V}}, N_{audit}^D, seed)$, to better represent the distribution of $\mathcal{X}_{\mathcal{V}}$, where $\texttt{Uniform}$ denotes a uniform sampler seeded with *seed*.

Considering the case with black-box or white-box access to the suspect model $f_{\mathcal{A}}$, our insight is leveraging outliers to enlarge the difference between regenerated and independent datasets. As illustrated in the Figure 5 (a), such outliers depict the dataset shape in the distribution space. Hence, the regeneration effect on models can be amplified on the outliers, leading to more accurate detection.

For an independent dataset which follows a different $\mathcal{P}_{indep} \ncong \mathcal{P}_{\mathcal{V}}$, as shown in Figure 5 (b), the peripheral points can better characterize the distribution shift than the in-distribution points which are basically easy-to-fit samples. Since the outliers are generally hard to fit, we sample N_{audit}^M high-loss points as our audit set:

$$\mathcal{X}_{\mathcal{J}}^M = \texttt{Sample}(\{x | x \in \mathcal{X}_{\mathcal{V}} \wedge l(f_{\mathcal{V}}(x)) \geq \tau_{audit}\}, N_{audit}^M), \quad (3)$$

where l is the loss function, τ_{audit} is the threshold of outlier range.

Table 3: Summary of similarity metrics. ●denotes black-box access, ❍denotes white-box access, ◗denotes partial white-box access, and "/" indicates that the corresponding access is not involved.

Similarity Metric	Defense Target	Defender Access $\mathcal{X}_{\mathcal{A}}$	Defender Access $f_{\mathcal{A}}$
Output Distance (OD)	Sample & set features	/	●
Gradient Distance (GD)		/	❍
Sample Distance (SD)		◗	/
Distribution Distance (DD)	Distribution features	◗	/

4.3 Similarity Calculation

In this section, we present the four similarity metrics under different access modes. As illustrated in Figure 4, the metrics are divided into two categories: metrics requiring model access and metrics requiring dataset access. Table 3 presents the defender access and target for each metric. If the defender does not have the required access for a metric, then DIPBox skips this metric and proceeds with the rest. Note that metrics designed for black-box model access can also be tested with white-box model access, and all metrics can be evaluated when the auditor has white-box access to the suspect model and partial white-box access to the suspect dataset. The final judgment (Section 4.4) is based on the holistic evaluation of metrics.

4.3.1 Metrics with Model Access. Inspired by previous works [17, 44], we found model memorization on training samples can reliably assess shallow dataset features. Based on the access to the suspect model $f_{\mathcal{A}}$, we propose two *model-agnostic* metrics, Output Distance (OD), adapting from prior work [14, 17] and our novel metric Gradient Distance (GD), respectively.

Output Distance. The OD measures the distance of outputs on the audit set $\mathcal{X}_{\mathcal{J}}^M$ between the victim's model $f_{\mathcal{V}}$ and the adversary's

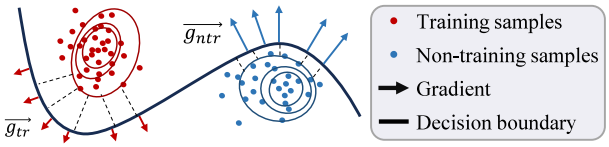


Figure 6: Intuition of GD: Training samples tend to induce smaller per-example gradients than non-training samples under converged training.

model $f_{\mathcal{A}}$. Since the audit set typically comprises hard-to-learn samples, a lower value indicates a higher likelihood that the adversarial model $f_{\mathcal{A}}$ has also been trained on these audit samples. We use l_p norm to measure the distance between outputs:

$$OD(f_{\mathcal{V}}, f_{\mathcal{A}}, \mathcal{X}^M_{\mathcal{J}}) = \frac{1}{\left|\mathcal{X}^M_{\mathcal{J}}\right|} \sum_{\mathbf{x}\in\mathcal{X}^M_{\mathcal{J}}} \left\|f_{\mathcal{V}}(\mathbf{x}) - f_{\mathcal{A}}(\mathbf{x})\right\|_p, \tag{4}$$

where $\|\cdot\|_p$ means the l_p norm. There are also other similar metrics such as Jensen-Shannon distance [20] in place of l_p norm, but they are shown to have equivalent performance as OD [10], thus we consider only l_p and implement other metrics in future work.

Gradient Distance. We propose GD for more detailed analysis with white-box access to $f_{\mathcal{A}}$. Empirically, for well-trained models, training data often exhibit lower loss, which correlates with smaller per-example gradients [44] under common losses (Figure 6). We propose to compute the *layer-wise normalized gradients*, $LNG(f, \mathbf{x})$, as a practical proxy signal, to quantify the self-influence of point $\mathbf{x}$ on model f:

$$\mathrm{LNG}(f, x) \triangleq \frac{1}{N_f} \sum_{i=1}^{N_f} \frac{g_i(f, x)}{\left\|\mathbf{w}_{f_i}\right\|_p + \varepsilon_{\text{num}}}, \tag{5}$$

where N_f is the number of layers of f, $\mathbf{w}_{f_i}$ is the weight matrix of i-th layer and where $g_i(f, x) \triangleq \left\|\nabla_{\mathbf{w}_{f_i}} \ell(f(x), y)\right\|_p$ and $\nabla_{\mathbf{w}_{f_i}} \ell(f(x), y)$ denotes the per-example gradient of the loss with respect to the parameters of the i-th layer, evaluated at input $\mathbf{x}$. Here, y denotes the ground-truth label associated with $\mathbf{x}$. The lower normalized gradient reflects the smaller impact of data $\mathbf{x}$ to the weight, and the layer-wise average represents the impact of $\mathbf{x}$ to the whole model. Our novel metric GD is computed based on distance between LNG:

$$GD(f_{\mathcal{V}}, f_{\mathcal{A}}, \mathcal{X}^M_{\mathcal{J}}) = \frac{1}{\left|\mathcal{X}^M_{\mathcal{J}}\right|} \sum_{\mathbf{x}\in\mathcal{X}^M_{\mathcal{J}}} |LNG(f_{\mathcal{V}}, \mathbf{x}) - LNG(f_{\mathcal{A}}, \mathbf{x})|. \tag{6}$$

4.3.2 Metrics with Dataset Access. With the suspect sub-dataset $\mathcal{X}^s_{\mathcal{A}}$, the auditor can obtain more detailed results.

Sample Distance. As demonstrated in Section 2, SD estimates the fraction of audit samples in $\mathcal{X}^D_{\mathcal{J}}$ whose identical or similar copy also appears in $\mathcal{X}^s_{\mathcal{A}}$, providing an approximation of actual proportion of duplicates. As the adversary preserves the dataset quality, regenerated samples should have small distance to the original ones. We use the closed r-ball $\mathcal{B}_d(\mathbf{x}, r) = \{\mathbf{x}' | d(\mathbf{x}', \mathbf{x}) \le r\}$ for each $\mathbf{x} \in \mathcal{X}^D_{\mathcal{J}}$ to delineate the border of similar copies. Formally:

Algorithm 2: `Judging`$(\mathbf{s}_{\mathcal{J}}, R_{\mathcal{J}})$

Input: Metric values $\mathbf{s}_{\mathcal{J}}$ (smaller indicates higher similarity), and judge requirement $R_{\mathcal{J}}$
Output: Judgment $\mathcal{J}$

```
if ∃λ ∈ {OD, GD} tested such that s_J[λ] ≤ R_J[λ] then
    if SD is tested then
        return Attack ❶ / ❷ with s_J[SD].
    else
        return Attack ❶ / ❷.
else
    if DD is tested and s_J[DD] ≤ R_J[DD] then
        return Attack ❸.
return Independent.
```

$$SD(\mathcal{X}^D_{\mathcal{J}}, \mathcal{X}^s_{\mathcal{A}}, r) = \left|\{\mathbf{x} | \mathbf{x} \in \mathcal{X}^D_{\mathcal{J}} \wedge \mathcal{B}_d(\mathbf{x}, r) \cap \mathcal{X}^s_{\mathcal{A}} \neq \emptyset\}\right| / \left|\mathcal{X}^D_{\mathcal{J}}\right|. \tag{7}$$

Distribution Distance. It is hard to directly measure distribution-level similarity, especially with only a small suspect subset. To simultaneously satisfy the design goals, inspired by the notion of Maximum Mean Discrepancy (MMD) [8, 22], we empirically estimate the distribution gap between $\mathcal{X}^D_{\mathcal{J}}$ and $\mathcal{X}^s_{\mathcal{A}}$ with:

$$DD(\mathcal{X}^D_{\mathcal{J}}, \mathcal{X}^s_{\mathcal{A}}) = \mathbb{E}_{\theta\sim\mathcal{P}_\theta} \left\| \frac{1}{\left|\mathcal{X}^D_{\mathcal{J}}\right|} \sum_{i=1}^{\left|\mathcal{X}^D_{\mathcal{J}}\right|} \psi_\theta(\mathbf{x}_i) - \frac{1}{\left|\mathcal{X}^s_{\mathcal{A}}\right|} \sum_{j=1}^{\left|\mathcal{X}^s_{\mathcal{A}}\right|} \psi_\theta(\mathbf{x}'_j) \right\|_p, \tag{8}$$

where ψ_θ is the feature extractor, $\mathbf{x}_i \in \mathcal{X}^D_{\mathcal{J}}$ and $\mathbf{x}'_j \in \mathcal{X}^s_{\mathcal{A}}$.

4.4 Regeneration Judgment

The judgment is based on the pre-computed threshold contained in the judge requirement $R_{\mathcal{J}}$ and the thresholding pattern among available metrics. Inspired by one-sided hypothesis testing [10], we calibrate each metric-specific threshold τ_λ using surrogate negative datasets $\{\mathcal{X}^i_{neg}\}_i$ (independently constructed datasets). For $\lambda \in \{OD, GD\}$, we train negative models $\{f^i_{neg}\}_i$ on $\{\mathcal{X}^i_{neg}\}_i$ and compute $s^i_{neg} = \lambda(f_{\mathcal{V}}, f^i_{neg}, \mathcal{X}^M_{\mathcal{J}})$. For $\lambda = DD$, we compute $s^i_{neg} = DD(\mathcal{X}^D_{\mathcal{J}}, (\mathcal{X}^i_{neg})^s)$, where $(\mathcal{X}^i_{neg})^s$ is a uniformly sampled audited subset of $\mathcal{X}^i_{neg}$. We then set $\tau_\lambda = \alpha_\lambda \cdot \min_i s^i_{neg}$, where $0 < \alpha_\lambda \le 1$ controls the sensitivity (larger α_λ is more permissive). Since smaller distances indicate stronger similarity to $\mathcal{X}_{\mathcal{V}}$, using the closest negative score makes the threshold conservative for a small negative set. The thresholding pattern consists of metrics whose values are lower than the corresponding thresholds (see Algorithm 2), serving as supporting evidence for the auditor to assess the regeneration attack implemented by the adversary. When surrogate negative datasets are unavailable, the defender can alternatively set $\tau_\lambda = \alpha_\lambda \cdot \max_i s^i_{pos}$ based on potential pirated datasets $\{\mathcal{X}^i_{pos}\}_i$, where s^i_{pos} is computed analogously. In this case, the threshold can also be determined via empirical strategies, e.g., simulating dataset modifications with surrogate datasets and selecting τ_λ to balance false positives and false negatives. The computational cost mainly comes from surrogate model training, leading to a complexity of $O(N_f + N_g)$, where N_f is

the number of surrogate models and N_g is the number of surrogate generators for simulating attacks.

4.5 Theoretical Analysis

To justify DIPBox, we analyze (i) the principles underlying our similarity metrics and (ii) the utility–evasion trade-off faced by a regeneration adversary.

4.5.1 Justification of Multi-Scale Similarity Metrics. Our metrics capture distinct, fundamental properties of the learning pipeline.

OD is motivated by *prediction stability*: under the utility constraint in Eq. (2), a successful regeneration should preserve generalization behavior, suggesting limited sensitivity to small data perturbations, leading to close outputs on the audit set. GD is motivated by *influence-function* analysis: the self-influence of a sample is controlled by a quadratic form involving its gradient and the inverse Hessian; GD serves as an efficient proxy for distinguishing seen-like samples from unseen-like ones. DD estimates an MMD-inspired RKHS integral probability metric. Under a characteristic kernel, the corresponding mean embedding is identifiable; in our implementation, we approximate this idea using finite random neural feature maps [3, 21, 51, 71], yielding a practical distribution-level similarity test.

4.5.2 Detectability via Domain Adaptation. The adversary faces a trade-off between evading detection and preserving utility (Eq. (2)). We model this trade-off using domain adaptation theory. Let $\mathcal{P}_{\mathcal{A}}$ denote the adversary's (regenerated) training distribution and $\mathcal{P}_{\mathcal{V}}$ denote the victim's distribution on which utility is evaluated. A standard domain-adaptation bound [6] suggests that performing well on $\mathcal{P}_{\mathcal{V}}$ depends on both (i) low risk on $\mathcal{P}_{\mathcal{A}}$ and (ii) small distributional divergence between $\mathcal{P}_{\mathcal{A}}$ and $\mathcal{P}_{\mathcal{V}}$.

Theorem 4.1 (Utility–Evasion Trade-off)**.** Let $\mathcal{H}$ be a hypothesis class and consider the 0–1 loss. For any $h \in \mathcal{H}$,

$$R_{\mathcal{P}_{\mathcal{V}}}(h) \;\le\; R_{\mathcal{P}_{\mathcal{A}}}(h) \;+\; \tfrac{1}{2}\, d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{P}_{\mathcal{A}}, \mathcal{P}_{\mathcal{V}}) \;+\; \lambda, \tag{9}$$

where $\lambda = \min_{h\in\mathcal{H}} \left(R_{\mathcal{P}_{\mathcal{A}}}(h) + R_{\mathcal{P}_{\mathcal{V}}}(h)\right)$ is the ideal joint error and $d_{\mathcal{H}\Delta\mathcal{H}}(\cdot,\cdot)$ is the $\mathcal{H}\Delta\mathcal{H}$-divergence.

Theorem 4.1 highlights a generic trade-off between preserving utility on $\mathcal{P}_{\mathcal{V}}$ and increasing cross-domain discrepancy between $\mathcal{P}_{\mathcal{A}}$ and $\mathcal{P}_{\mathcal{V}}$. Our DD metric instantiates a practical, distribution-level MMD-inspired discrepancy that is well-suited for detecting generative polishing, while the bound uses $d_{\mathcal{H}\Delta\mathcal{H}}$ to capture domain shift. We treat these two notions as complementary rather than equivalent; empirically, increasing distribution-level deviation to evade DD tends to make maintaining high utility on $\mathcal{P}_{\mathcal{V}}$ harder.

5 Evaluation

In this section, we first present the experimental settings in Section 5.1. Then, in Sections 5.2 to 5.4, we evaluate the effectiveness of DIPBox under different defender access assumptions against Attack ❶, Attack ❷, and Attack ❸, respectively. We report OD, GD, and DD values to represent the defender's black-box model access, white-box model access, and white-box access to a partial suspect dataset, respectively. Next, we compare the True Positive Rate (TPR) of DIPBox with previous baselines in Section 5.5. Finally, we study adaptive attacks in Section 5.6.

5.1 Datasets & Setup

We use widely adopted AI training datasets (e.g., CIFAR-10) for our evaluation. For each task, the victim holds a training dataset as $\mathcal{X}_{\mathcal{V}}$ and trains models achieving high accuracy on corresponding test data (simulated as real-world data of distribution $\mathcal{P}_t$). The adversary only has access to independent (negative) datasets for the same task, whose distributions are similar but not identical to $\mathcal{X}_{\mathcal{V}}$. For text data, we use BERT `[CLS]` embeddings so that inputs are represented as vectors, analogous to image features.

Regenerated Datasets. We craft 320 regenerated datasets in total. For the Attack ❶, we first test the 6 post-processing attacks as listed in Table 4, and fix the post-processing severity to the low level. We then increase the severity to test the robustness in Section 5.2 and extend to 20 different post-processing attack variants in Table 8. For the set-level attack (Attack ❷), we consider the adversary randomly mixing a subset of the victim dataset $\mathcal{X}_{\mathcal{V}}$ with one independent dataset to form $\mathcal{X}_{\mathcal{A}}$ with overlap ratio $p_{\mathcal{V}} \in \{0.2, 0.4, 0.6, 0.8\}$ where $p_{\mathcal{V}} = |\mathcal{X}_{\mathcal{V}} \cap \mathcal{X}_{\mathcal{A}}| / |\mathcal{X}_{\mathcal{V}}|$. For the distribution-level attack (Attack ❸), we consider both generative adversarial networks (GANs) and diffusion models in our experiments as they represent common generation strategies. We use the conditional GAN for MNIST and one of the state-of-the-art diffusion models EDM [30] for CIFAR-10. For FairFace, we train the GAN model StyleGAN3 [32]. For text data, we use a ChatGPT-enhanced T5 paraphraser [58].

Negative Datasets. Prior work [14, 28, 29, 44] evaluates negative datasets from the same distribution as the victim dataset, which contradicts our assumption that adversaries lack access to in-distribution data. To the best of our knowledge, no public benchmark exists for dataset regeneration. Therefore, we construct a benchmark using research-permissive open-source datasets. We select datasets that address the same task but differ from the victim dataset to introduce distribution discrepancy. Since few public datasets provide exact class matches, we survey 26 datasets from prior work, manually harmonize labels, and select the three most similar datasets as negatives (see Table 4). For CIFAR-10, due to limited negative dataset choices, we additionally use ImageNet as negative datasets, which shares a common source with CIFAR-10.

Audit Set. In terms of the audit set $\mathcal{X}^M_{\mathcal{J}}$ (Section 4.2), we set $\tau_{audit} = 0.2$ and $N^M_{audit} = 1000$, and sample $\mathcal{X}^M_{\mathcal{J}}$ uniformly at random from the high-loss set induced by τ_{audit}. To compute SD and DD, we set $N^D_{audit} = 0.01 \cdot |\mathcal{X}_{\mathcal{V}}|$ and sample $\mathcal{X}^D_{\mathcal{J}}$ uniformly from $\mathcal{X}_{\mathcal{V}}$, and we set $|\mathcal{X}^s_{\mathcal{A}}| = 0.01 \cdot |\mathcal{X}_{\mathcal{A}}|$ by uniformly sampling a fixed subset from $\mathcal{X}_{\mathcal{A}}$; both subsets remain unchanged throughout the experiments. We approximate the expectation in DD (Eq. 8) by averaging over 1000 random draws of ψ_θ. Each draw corresponds to sampling the parameters θ of a two-layer MLP from $\mathcal{N}(0,1)$ as a universal feature map ψ_θ [3, 21, 51, 71].

Models. In total, we trained 590 classification and generative models. We choose one widely studied architecture for the victim model $f_{\mathcal{V}}$ for each task. In addition to the same architecture as the victim, the adversary chooses more advanced architectures as listed in Table 4 to make up for the dataset utility degradation. We further evaluate DIPBox's robustness against different training strategies by altering training epochs, architecture, and attack severity.

Utility Metrics. We quantify the adversary's gain by the Utility Retention (UR), reported as a percentage: $UR = 100 \times v(f_{\mathcal{A}})/v(f_{\mathcal{V}})$,

Table 4: Details of benchmark datasets and experimental setup. For each task, we adopt one widely used training dataset as the victim's dataset $\mathcal{X}_\mathcal{V}$, and select three most similar public datasets used in the prior literature as the independent (negative) ones.

Task	Victim's Dataset	Domain	Input Size	# of Classes	Victim's Architecture	Additional Architecture for $f_\mathcal{A}$	Post-processing in Attack ❶	Generators for Attack ❷	Independent Datasets
Handwritten Digit Classification	MNIST	Image	32×32	10	LeNet-5	3-layer MLP	Glass blur, Impulse noise, JPEG compression, Pixelate, Spatter, Speckle noise	Conditional DCGAN	DIDA [35] (Neg-1), ARDIS [34] (Neg-2), SVHN [46] (Neg-3)
Object Classification	CIFAR-10	Image	32×32	10	ResNet-18 [26]	MobileNet-v2 [50], VGG-13 [53]		EDM [30]	STL-10 [11] (Neg-1), ImageNet-Set1 (Neg-2), ImageNet-Set2 (Neg-3)
Facial Attribute Classification	FairFace	Image	128×128	20	ResNet-101	EfficientNet-v2-s [54], RegNet-y-8gf [47]		StyleGAN3 [32]	LFWA+ [40] (Neg-1), PubFig [33] (Neg-2), UTKFace [70] (Neg-3)
News Classification	AG-News	Text	768	4	Tiny-BERT [55]	Mini-BERT, Small-BERT [7]	Word Addition, Word Deletion, Synonym Swap	ChatGPT-enhanced T5 paraphraser [58]	Yahoo-News [63] (Neg-1), CNN-News [56] (Neg-2), Guardian News (Guard-News) [25] (Neg-3)

Table 5: Similarity measurement with our proposed metrics on regenerated datasets produced by the sample-level attack and on the independent datasets for vision tasks.

Dataset Type		MNIST					CIFAR-10					FairFace				
		UR (%)	OD	GD	SD	DD	UR (%)	OD	GD	SD	DD	UR (%)	OD	GD	SD	DD
Suspect Dataset (Positive)	Glass blur	73.77	0.155	0.338	0.0	7.988	90.17	0.235	0.438	0.254	1.336	84.13	0.032	0.158	0.946	2.485
	Impulse noise	98.60	0.066	0.148	0.0	2.889	92.36	0.197	0.421	0.0	1.427	82.67	0.075	0.374	0.0	3.200
	JPEG compression	98.34	0.041	0.122	1.0	1.453	90.78	0.209	0.383	0.988	1.262	85.25	0.044	0.228	1.0	2.348
	Pixelate	98.28	0.068	0.171	0.002	3.241	89.96	0.296	0.691	0.988	1.269	87.46	0.027	0.117	1.0	2.393
	Spatter	96.97	0.070	0.155	0.008	3.479	92.58	0.240	0.497	0.129	2.693	82.22	0.056	0.284	0.006	4.272
	Speckle noise	98.73	0.033	0.106	0.526	2.374	91.08	0.259	0.508	0.597	1.321	76.27	0.061	0.305	0.068	2.768
Independent Dataset (Negative)	Neg-1	13.48	1.254	4.073	0.0	35.927	32.19	1.053	5.951	0.0	4.586	15.23	1.122	6.256	0.000	9.441
	Neg-2	18.10	1.309	4.693	0.0	33.181	77.16	0.467	4.944	0.009	1.652	22.05	1.098	6.018	0.000	13.258
	Neg-3	63.78	0.339	0.901	0.0	26.737	82.74	0.451	4.860	0.008	1.870	58.60	0.995	6.598	0.001	13.589

Table 6: Similarity measurement with our metrics in the news classification task.

Modification Type		UR (%)	OD	GD	SD	DD
Suspect Dataset (Positive)	Insertion	99.98	0.106	2.967	0.002	4.789
	Deletion	99.31	0.137	5.123	0.084	5.952
	Synonym	99.48	0.160	3.493	0.002	7.104
Independent Dataset (Negative)	Neg-1	47.95	1.070	15.786	0.0	11.294
	Neg-2	60.83	1.069	19.709	0.0	13.913
	Neg-3	80.78	0.600	15.157	0.0	13.165

where v is the test accuracy. Take CIFAR-10 as an example, $v(f_\mathcal{V})$ represents the test accuracy on 10,000 test samples of model $f_\mathcal{V}$ trained on 50,000 training samples. A UR closer to 100% indicates greater utility retention.

5.2 Defending Sample-level Attack

We first evaluate the effectiveness of DIPBox for the sample-level attack (Attack ❶) with the default training setting. Then, we investigate the robustness against various training settings of $f_\mathcal{A}$ and the post-processing severity. We mainly report OD, GD and DD results for the defender with access to the black-box suspect model, white-box suspect model and partial suspect dataset.

Effectiveness. In Table 5, we present the metric values on datasets regenerated by Attack ❶ and on the negative datasets for the vision data. Table 6 shows corresponding results for the text data, where the text embeddings produced by $f_\mathcal{V}$ are compared. We also provide the UR to assess utility loss during regeneration. Under low attack severity, DIPBox consistently identifies all tested sample-level regenerations as shallow-feature regeneration following Algorithm 2. Specifically, their OD, GD, and DD values are lower than those of the negative datasets and the corresponding thresholds, enabling detection under all applicable access modes.

Unequal Effect of Post-processing Methods. We note that the same post-processing approach can produce unequal impact across different datasets or models. For example, MNIST data are more sensitive to the "Glass blur" than the other two vision tasks (CIFAR-10 and FairFace) because of larger accuracy degradation. In addition, facial attribute data are more sensitive across post-processing methods.

Utility Gap of Negative Datasets. The negative datasets have lower UR than datasets regenerated under Attack ❶ because of the inherent data distribution discrepancy. For example, the negative datasets of the facial attribute classification have less balanced data, which hinders the model from learning facial features equally among races and genders. This also implies that, in case of exclusive data source, a dataset of almost equal model-training utility to the proprietary dataset is more suspicious for unauthorized copying under the exclusive-source assumption.

Results on Texts. We have similar observations for text data (Table 6): the models trained on datasets that undergo random insertion, synonym swap, and deletion for 30% words can achieve near-perfect UR on test data. We note that the URs of derived datasets are much higher and close to 100%. This signifies that the text embedding models are not fully influenced by individual words but also the semantic meaning of input texts.

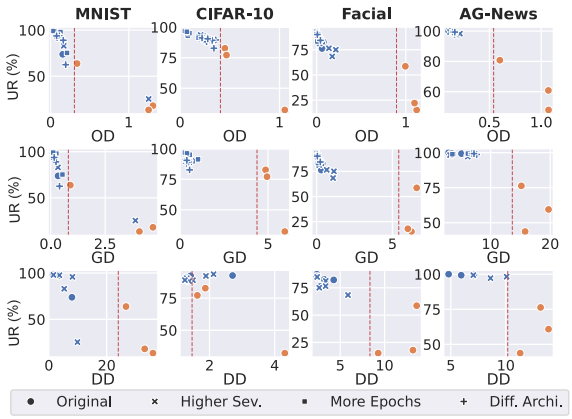


Figure 7: Robustness evaluation for defending the Attack ❶. We show the trade-off between UR and distances under various attack settings, where each point represents a model. The vertical dashed line is the threshold.

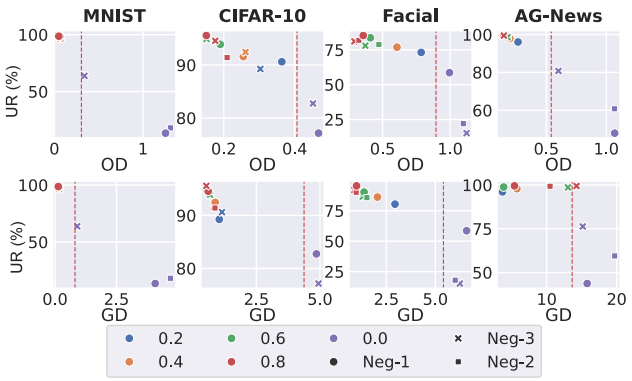


Figure 8: Trade-off between the measured distances (OD and GD) and the UR for intersection ratios 0.2, 0.4, 0.6, 0.8 and the negative ones (i.e., ratio 0.0), where each point represents a model. The vertical dashed line is the threshold.

Evaluation of SD. It is worth noting that SD metric generates unstable values for the sample-level regeneration attack. Ideally, nearly all samples are copied from the original dataset with additional modification, thus SD should be close to 1. However, the value of SD varies from 0 to 1. For instance, the SD is significantly lower than other post-processing methods for CIFAR-10 post-processed by "Impulse noise" and is nearly 0 for the facial dataset post-processed by "Spatter". The reason is that we choose a conservative radius for SD and a single choice of radius cannot fully characterize closeness between samples. Higher radius should be better for capturing the post-processed samples but can cause more false positives. Another potential reason is that we adopt classic L_p norm to compare sample closeness, which is computationally efficient but does not capture the visual similarity. We will investigate the more robust metrics (e.g., LPIPS [69]) in the future work.

Robustness. Figure 7 also validates the robustness of our framework when the adversary adopts different settings to train their model $f_{\mathcal{A}}$, and post-processing methods of medium and high severity. For OD and GD, we investigate the robustness against the adversary's different training settings. As expected, more training epochs (More Epochs) or more advanced model architecture (Diff. Archi.) result in better generalization over the modified samples and thus higher UR; however, these settings do not enable evasion and the regenerated datasets remain detectable under our previously determined thresholds. Besides, we also investigate when the adversary increases the modification severity (Higher Sev.): amplifying the post-processing to medium and high level.

Trade-off between UR and Similarity. From the results of vision (Table 5) and text data (Table 6), we notice the trade-off between UR and the value of OD, GD and DD: dissimilar datasets result in lower UR and higher metric values, and datasets regenerated by the sample-level attack have higher UR and lower metric values. Note that the trade-off is not linear: some negative datasets can achieve simultaneously higher UR and lower testing metric value than another negative dataset. However, they are still separable from the regenerated datasets.

In Figure 7, we visualize the trade-off under more complicated training and post-processing settings. The red vertical line represents the decision threshold determined previously. We observe that there are clear intervals between positive and negative datasets for OD and GD, which ensures the accurate judgment by Algorithm 2. As we consider the sample-level attack, the inherent data distribution can be shifted from the original distribution, leading to a DD value higher than the decision threshold. Thus, under the tested attack settings, the adversary cannot simultaneously retain high utility and evade DIPBox.

Case Study: CIFAR-10 and ImageNet. In Figure 7, we observe two negative datasets for CIFAR-10 are closer to the positive datasets than other tasks. After manual checking, they are sampled from ImageNet. The reason is both CIFAR-10 and ImageNet are sampled from the same source (i.e., searching on Internet using hierarchical structure of WordNet during close periods). Therefore, they are more similar than other tasks.

Takeaway 1. DIPBox can effectively detect datasets regenerated under Attack ❶ on both vision and text data and is robust against various sample-level attack settings.

5.3 Defending Set-level Attack

In this section, we first study the effectiveness against set-level regeneration under overlap ratios $p_{\mathcal{V}} \in \{0.2, 0.4, 0.6, 0.8\}$, then show robustness under different training strategies for $f_{\mathcal{A}}$.

Effectiveness. We investigate the model-level metrics (OD and GD) in Figure 8 under defender's black-box and white-box access to the suspect model, and the dataset-level metrics (SD and DD) in Figure 9 when the defender can access partial suspect dataset.

Evaluation of OD and GD. Figure 8 presents the UR-metric trade-off for OD and GD on our tested severity. We show the test intersection ratios in different colors and use the point style (e.g., circle or square) to represent the mixed independent datasets. As a higher similarity means more mixed instances in the suspect dataset, the

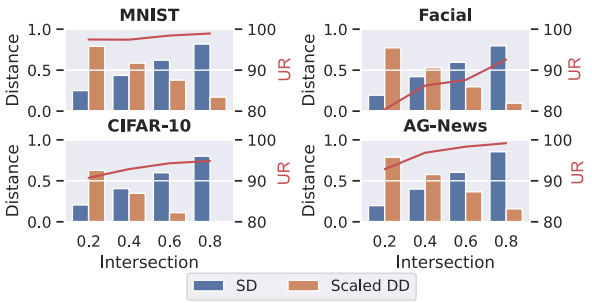


Figure 9: Trade-off between dataset metrics (SD and DD) and the UR for the intersection ratios 0.2, 0.4, 0.6, 0.8 and the negative one 0.0. The red curve is the UR (right axis).

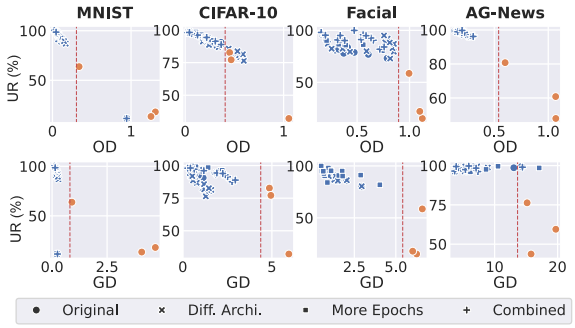


Figure 10: Robustness evaluation of defending Attack ❷. The trade-off between the UR and our metrics under different attack severity and adversary model training settings. The vertical dashed line represents the threshold.

OD and GD have smaller value as the intersection ratio increases due to sample-level closeness. Note that we do not aim to estimate the exact similarity but only to judge whether the suspect dataset contains a significant proportion of victim samples. Additionally, we note that the GDs measured on models finetuned on AG-News are close to or higher than the threshold. We provide the in-depth analysis along with the robustness. However, we observe that in general the trade-off between UR and our metrics (OD and GD) holds, and that the adversary cannot craft a mixed dataset under Attack ❷ without sacrificing the utility.

Evaluation of SD and DD. We plot the results in Figure 9. These two metrics are evaluated with access to the audit set and suspect sub-dataset, enabling a more direct dataset-level comparison. Here, SD estimates sample-level overlap based on the audit set, while DD captures distribution-level distance, which decreases as the ground-truth overlap ratio increases.

Robustness. Figure 10 evaluates the robustness under different training settings for the adversary's model. DIPBox can correctly detect datasets regenerated under Attack ❷ setting because the GD metric can accurately distinguish the positive and negative datasets on the vision task. Note that there are tested positive models with OD values higher than the threshold for CIFAR-10, because ImageNet, from which the two negative datasets are sampled with

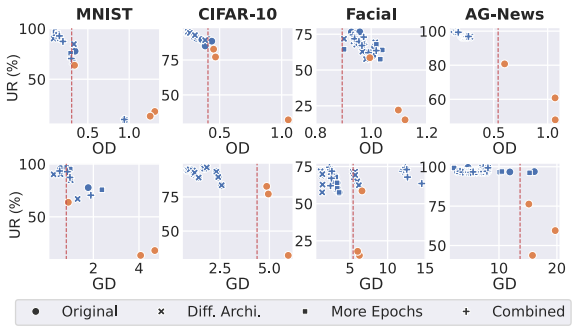


Figure 11: The trade-off between the UR and the model metrics (OD and GD) under different attack severity for Attack ❸ and adversary model training settings. The blue and orange points are positive and negative datasets.

manual annotation alignment, is similar to CIFAR-10 and often selected as the source for CIFAR-10 alternatives (e.g., CINIC [13]). On the other hand, the STL-10 is more different from CIFAR-10 (e.g., in terms of the image tone) and is accurately detected.

For the text classification, the GD metric is not reliable. The reason is rooted in the architecture difference: the models are based on the pretrained LMs of more parameters than the conventional CNNs thus have more complex decision boundary and can be sensitive to the non-training samples. We examined the evaded cases and found the cases of GD values higher than threshold are those with a small intersection ratio 0.2 and with more finetuning epochs. Nevertheless, the OD metric guarantees the correct judgment.

Takeaway 2: For the Attack ❷, OD and GD are robust against various training settings adopted by the adversary, and SD and DD can correctly quantify the regeneration ratio in the adversary's dataset.

5.4 Defending Distribution-level Attack

In this section, we investigate the effectiveness and robustness against the distribution-level dataset regeneration attack. For vision tasks, we train generative models (see Table 4) and select checkpoints to obtain generated content of varying quality. For the default generator, we train the EDM model on CIFAR-10 (size 32×32) to an FID score of 2.341 and train StyleGAN3 on FairFace (size 128×128) to an FID score of 12.151, which is comparable to the reported performance [30, 32].

Limitation of OD and GD. We provide the evaluation results of model-level metrics in Figure 11 which are measurable when the defender has black-box or white-box model access and no access to partial suspect dataset. As the adversary dataset is re-sampled from the same distribution as the source dataset, our model metrics can fail under different training settings. We observe that multiple positive derivatives (blue points) can have both high UR and higher values of OD or GD than the negative ones at the same time. Take the facial dataset as an example, a remarkable proportion of positive models achieves ~75% UR (the same level as Attack ❶ in Figure 7 and Attack ❷ in Figure 8) and has both model metrics higher than the threshold. This is expected, because there are no

Table 7: Robustness against Attack ❸ with different generator types and generation sizes of the advanced generator.

Adversary Setting	MNIST		CIFAR-10		FairFace		AG-News	
	UR (%)	DD	UR (%)	DD	UR (%)	DD	UR (%)	DD
Default Generator	95.59	1.43	92.87	1.36	66.09	2.52	96.75	8.23
Advanced Generator	97.44	1.18	95.63	1.45	93.50	2.53	97.40	8.22
Double size	97.69	0.94	97.29	1.13	91.93	2.06	97.63	8.00
Half size	96.89	1.18	93.39	1.47	81.36	2.68	96.34	8.21
Threshold	/	24.03	/	1.49	/	8.37	/	10.17

exact overlapping samples and the training-data distribution shift between the positive models and the source model is minimized by the generative model. Therefore, it is necessary for the dataset metric to identify the distribution-level regenerated dataset.

Evaluation of SD and DD. Now we investigate whether the SD and DD, measurable under defender's access to partial suspect dataset, can track the distribution-level regeneration. As the dataset is regenerated, there are very few samples that resemble the original data samples, and thus the SD values are close to 0 for all the four benchmarks we selected. As for DD, we evaluate the robustness by varying the attack settings and summarize the results in Table 7. In particular, we evaluate two different generative models for each benchmark. For the vision task, we use GANs and diffusion models of appropriate scales: we use the StyleGAN2-Ada [31] for CIFAR-10 and the Denoising Diffusion Probabilistic Models (DDPM) [27] for MNIST and FairFace. For the text, we use both back-translation and Pegasus-based paraphraser [48, 68] as advanced generator. We also test if the adversary generates dataset of double or half size of the source dataset.

In Table 7, we denote the generators that produce training data enabling model to achieve higher UR by "Advanced" and the other one as the "Default", and we report the generator checkpoint that achieves the highest UR. Regardless of the generator type or the size of generated dataset, our DD can effectively and *robustly* evaluate the closeness between the distributions of the regenerated dataset and the source dataset. Next, we perform an in-depth analysis of the factors involved in distribution-level regeneration.

Generator Capacity. The adversary may adopt a more advanced generator to produce more in-distribution samples. The dataset produced by the advanced generator has higher quality but not necessarily lower DD value, because our DD assesses the closeness between the distribution of two datasets, instead of the dataset utility closeness. We manually inspect the difference between generators and found that the better quality comes from finer details (e.g., clearer facial contours) in the generation, which can ameliorate the model convergence to the target distribution. Nonetheless, the DD remains robust in identifying the regenerated dataset.

Amount of Generated Samples. We note that having fewer generated samples for training is a limiting factor in developing a highly accurate suspect model. Consequently, the Attack ❸ may be preferred by the adversary because the adversary can further improve the model performance by increasing the training data size. However, this comes at the cost of additional computation for generator training, finetuning, and especially inference. For example, if the adversary's dataset size doubles, the training time also roughly doubles, not counting the cost of dataset generation. Therefore, we consider doubled data size in this paper and leave larger sizes for future work. Among the tested cases (i.e., double-size), our dataset metrics achieve robust tracking.

Table 8: Comparison with existing methods. Each entry is TP/Total for the corresponding attack, and "Overall TPR" is computed over all attacks.

Method	CIFAR-10				FairFace			
	A ❶	A ❷	A ❸	Overall TPR (%)	A ❶	A ❷	A ❸	Overall TPR (%)
Radioactive [49]	1 / 20	1 / 9	0 / 13	4.76	0 / 20	1 / 8	0 / 7	2.86
DI [44]	1 / 20	2 / 9	0 / 13	7.14	0 / 20	1 / 8	0 / 7	2.86
EMA [28]	3 / 20	2 / 9	0 / 13	11.90	0 / 20	1 / 8	0 / 7	2.86
UBW [36]	4 / 20	3 / 9	0 / 13	16.67	0 / 20	1 / 8	0 / 7	2.86
RAI^2 [14]	6 / 20	5 / 9	0 / 13	26.19	18 / 20	7 / 8	0 / 7	71.43
Data-use [29]	15 / 20	7 / 9	0 / 13	52.38	18 / 20	8 / 8	0 / 7	74.29
Ours	**20 / 20**	**9 / 9**	**13 / 13**	**100.0**	**20 / 20**	**8 / 8**	**7 / 7**	**100.0**

Takeaway 3. DIPBox can perform accurate dataset tracking with DD under stronger distribution-level attacks.

5.5 Comparison with Baselines

Setup. We compare with state-of-the-art methods including dataset watermarking UBW [36] and data-use audit [29], inference-based auditing EMA [28] and RAI^2 [14]. We use CIFAR-10 and FairFace as they are two representative benchmarks used in prior work. Since detection tasks typically prioritize a high TPR to minimize missed detections, we compare TPR, defined as $TPR = TP/(TP + FN)$, which also aligns with previous work [14, 28, 29, 36, 44] that mainly reports the detection rate. For fair comparison, we need to unify the output format. Note that UBW and EMA output binary auditing results that align with DIPBox's output format, while RAI^2 estimates the proportion of copied samples in the form of a dataset similarity score ranging from 0 to 1. We map the output to 1 if the dataset similarity is higher than 0.5, and 0 otherwise. We use 0.5 as a midpoint for binarization; other thresholds trade off misses and false alarms and would require task-specific threshold selection. In terms of testing samples, we add up to 20 sample-level modifications for Attack ❶ to expand attack settings, select all possible overlap settings for Attack ❷, and consider datasets generated from various intermediate checkpoints of the generator for Attack ❸. In total, we obtain 42 and 35 tested regenerated datasets for CIFAR-10 and FairFace, respectively.

Comparison Results. In Table 8, we observe that DIPBox systematically outperforms prior work across different threats and benchmarks. The improvement comes from more robust detection against Attack ❶ and Attack ❷ and the coverage of Attack ❸. For instance, DIPBox can robustly detect sample-level attacks while previous watermarking (e.g., [49]) can be bypassed by certain sample-level perturbations (e.g., Glass blur) because of watermark loss caused by compression. Notably, on FairFace, RAI^2 [14] and Data-use [29] have comparable performance with our DIPBox in detecting the sample-level and set-level attacks but fail to detect the distribution-level attack. This signifies the necessity of a holistic approach to track dataset regeneration.

Takeaway 4. DIPBox outperforms previous approaches in terms of the detection rate through its multi-scale testing.

Table 9: Evaluation of DP-based adaptive attacks on MNIST. Red / green indicates values above/below the threshold, respectively.

Attacks	$\varepsilon = 1$				$\varepsilon = 10$			
	UR (%)	OD	GD	DD	UR (%)	OD	GD	DD
DP-SGD [2]	76.15	0.248	4.725	/	91.14	0.109	1.637	/
DP-MERF [24]	65.76	1.070	4.864	2.577	71.78	1.213	5.224	2.582
PrivSet [9]	73.36	1.323	9.482	12.67	92.33	1.362	8.402	11.73

5.6 Robustness to Adaptive Attacks

The adversary can attack our framework from the model level and the dataset level with knowledge of DIPBox's procedure, including the audit set selection and the metric design. The main goal is to weaken the signals captured by DIPBox. To achieve this goal, we consider three types of adaptive attacks: 1) a universal adaptive attack based on differential privacy (DP) that reduces the influence of the original training data on the suspect model or dataset, 2) a metric-specific adaptive attack that optimizes against a given metric to raise the metric value above the threshold known by the adversary, and 3) an audit set-targeted adaptive attack where the adversary knows the audit set design and adjusts the suspect dataset before training or preview release. We do not additionally combine these adaptive attacks with every base attack from Attack ❶ to Attack ❸ because such combinations introduce additional utility loss. We mainly use MNIST and FairFace as they represent datasets of simple and complex features.

Evaluation of DP-based Adaptive Attack. For the DP-based adaptive attack, the adversary can exploit DP-SGD [2] for model-level metrics OD and GD. For our SD and DD metrics, the adversary generates data with DP for the distribution-level attack in order to increase the difference between regenerated and protected datasets. We consider MNIST here because existing DP generation techniques mainly focus on simple benchmarks such as MNIST. For DP-generated data, we use DP-MERF [24] to train a generator that can produce data of better quality (i.e., enabling training models of higher test accuracy) than conventional DP GANs. In addition, the adversary can directly distill the target dataset into a much smaller dataset using dataset distillation techniques [71]. We adopt PrivSet [9] as the most recent dataset condensation solution with DP constraints. Table 9 shows the results of DP-based attacks against our framework DIPBox.

Model-level DP-based Attack. As expected, DP applied during training on MNIST has a negative impact on utility (i.e., test accuracy). The small budget $\varepsilon = 1$ can reduce the UR to $\sim$ 70%, while the large budget $\varepsilon = 10$ only results in a slight UR drop; it still achieves UR over 90%, compared to UR$\approx$ 98% under Attack ❶. According to Algorithm 2, our framework DIPBox still flags these cases because at least one decisive model metric (OD or GD) remains below the determined threshold. This suggests that the DP-based attack against OD/GD does not enable evasion without incurring a noticeable utility loss in our setting.

Dataset-level DP-based Attack. We evaluate the DP-based attack against our remaining dataset metrics. As the bottom two rows of Table 9 show, the regenerated datasets are not identified as the sample-level attack or the set-level attack according to our framework DIPBox because the OD and GD values are higher than the threshold. In addition, similar to the evaluation in Section 5.4, the SD values are almost 0 because there are few approximate sample pairs between the original and regenerated datasets. Hence, we rely on DD to distinguish whether the regenerated dataset has a nearly identical distribution. As Table 9 shows, the DD values are below the threshold, demonstrating that the dataset-level attack does not evade DIPBox. Compared to the MNIST results of Table 7, the DD values of dataset-distillation-generated datasets (i.e., PrivSet) are higher than those previously evaluated with naïve generators, further confirming that a smaller regeneration dataset size can increase DD at the cost of degraded UR.

Metric-specific Adaptive Attacks. With knowledge of the metric design and the target threshold (i.e., estimated from the design of DIPBox), the adversary can add a metric-maximization term weighted by w to the optimization objective to increase the metric value for their trained model or generated dataset. When the defender can only compute OD and GD metrics, the adversary minimizes the model-training loss minus $w \cdot d(f_{\mathcal{V}}, f_{\mathcal{A}}, \mathcal{X}_{\mathcal{J}}^{\mathcal{A}})$, where $d \in \{OD, GD\}$ and $\mathcal{X}_{\mathcal{J}}^{(M,\mathcal{A})}$ is an adversary-generated audit set following the same procedure as $\mathcal{X}_{\mathcal{J}}^{M}$. As the adversary does not know the actual $\mathcal{X}_{\mathcal{J}}^{M}$ due to random sampling, the $\mathcal{X}_{\mathcal{J}}^{(M,\mathcal{A})}$ is sampled in each step. The adversary can adopt better hyperparameters than the default setting, so we applied a grid search on optimizers, training epochs from 10 to 200, learning rates $\{0.001, 0.01, 0.1\}$, and finally selected training of 200 epochs with learning rate 0.1 using Adam. For the DD-specific attack (i.e., improving Attack ❸), we mainly focus on MNIST because of lower training cost. We use conditional GAN on MNIST for experiments, and add DD as an optimization term with different penalty weights w. We apply similar hyperparameter optimization on the generator training.

Figure 12 (a) and Figure 12 (c) show the OD-specific adaptive attacks on MNIST and FairFace, respectively. We can see that small w is already effective to increase the auditing metrics and large penalty weight w can significantly degrade the model utility. The sweet spot between the utility and the auditing metric value differs between MNIST and FairFace. For instance, on MNIST the adversary begins to bypass OD-only auditing around $w = 0.5$, but the UR has already dropped below 80% and decreases further as w grows. Meanwhile, on FairFace, $w = 0.5$ retains about 60% UR but remains near the OD threshold, while larger weights cross the threshold at substantially lower UR because the facial data are of larger scale and make the model more sensitive to the metric penalty.

For the GD-specific adaptive attacks on MNIST and FairFace in Figure 12 (b) and Figure 12 (d), we observe a similar trend but the model's performance is more sensitive to the GD penalty, as UR drops faster than under the OD penalty. This is expected because the GD penalty can have a larger impact on the model weights during training through gradients of larger norms. To make GD higher than our predefined threshold, the UR is around 70% and less than 60% for MNIST and FairFace respectively, suggesting that GD is harder to attack adaptively. Meanwhile, it increases the adversary's computational cost because GD depends on per-example parameter gradients, and optimizing it builds a second-order computation graph which slows the training and occupies more memory.

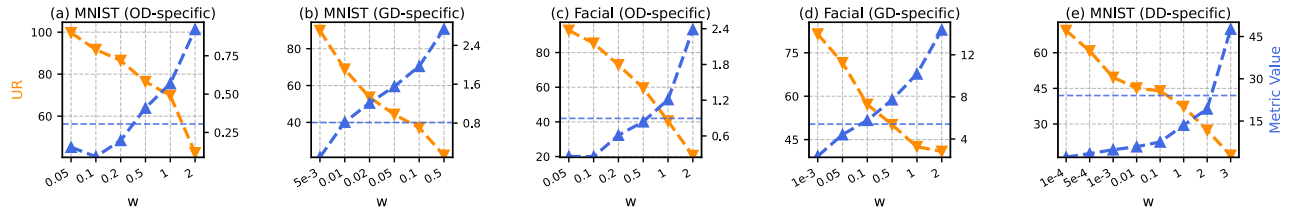


Figure 12: Evaluation of metric-specific adaptive attacks.

Table 10: Evaluation of audit set-targeted adaptive attacks on FairFace.

Target	Strategy	UR	Judgment
OD/GD	10% high-loss sample removal	86.86%	Attack ❶/❷
	20% high-loss sample removal	85.71%	Attack ❶/❷
	30% high-loss sample removal	84.24%	Independent
DD	10% Low-loss sampled preview	-	Attack ❸
	10% High-loss sampled preview	-	Independent
	Single-class preview	-	Independent

For the DD-specific adaptive attack shown in Figure 12 (e), we found that the UR values are generally lower than those in previous OD-specific and GD-specific adaptive attacks on MNIST. Specifically, in this case, the generative model is more sensitive to the penalty weight and often fails to converge stably to a low loss. Therefore, the final generator produces low-quality images, which makes the data unlearnable for the subsequent classification models. The results demonstrate a trade-off between the adaptive attack strength (i.e., w) and the final model or dataset utility: increasing the metric-specific penalty enough to bypass detection comes with utility loss.

Adaptive Attacks for Audit Set. We consider two adaptive strategies with knowledge of the design of $\mathcal{X}_{\mathcal{J}}^{M}$ and $\mathcal{X}_{\mathcal{J}}^{D}$, targeting the model-level metrics OD/GD and dataset-level metrics SD/DD, respectively. The adversary cannot control $\mathcal{X}_{\mathcal{J}}^{M}$ but can remove potential audit-set samples by filtering out samples with loss higher than τ_{audit}. Since τ_{audit} is a hyperparameter set by the defender and should be adjusted based on the adversary's own model, one direct method for the adversary is to remove high-loss samples. Then, the adversary directly trains $f_{\mathcal{A}}$ on the remaining data. The top half of Table 10 shows that removing 30% of the samples can lead to an independent judgment by Algorithm 2 when the audit is limited to model-level metrics (OD/GD). Further exploration on the removed data shows that our previous τ_{audit} corresponds to the top 24%, and removing less than 20% high-loss samples is equivalent to Attack ❷, which is detectable as shown in Figure 8. Although removing 30% can evade model-level tracking with OD/GD, without knowledge of τ_{audit}, the adversary cannot reliably guess the correct removal ratio to win the game in Section 3.1 and still risks utility loss or being tracked.

In addition, we consider an adversary that regenerates with Attack ❸ and controls $\mathcal{X}_{\mathcal{J}}^{D}$ (e.g., an adversary-controlled preview subset on the platform), which is sampled from regenerator-produced data of 10% low-loss or high-loss samples assessed by a surrogate model directly trained on the obtained $\mathcal{X}_{\mathcal{A}}$, or from a random single class. In the bottom half of Table 10, we find that DD can still track the low-loss case but fails for single-class and high-loss cases because of significant distribution shift. Although the adversarial selection of $\mathcal{X}_{\mathcal{J}}^{D}$ does not degrade $\mathcal{X}_{\mathcal{A}}$, the high-loss samples can contain low-quality generated data, and a single class does not fully represent $\mathcal{X}_{\mathcal{A}}$ well enough to attract users.

Takeaway 5. The evaluated adaptive attacks against DIPBox can evade detection by increasing metric values or manipulating the audit set, but they incur dataset utility degradation, poorer preview representativeness, or memory and computational cost.

6 Related Work

In this section, we review existing copyright protections and auditing strategies for models and datasets.

Models. The owner can watermark the proprietary model before release and testify the watermark to claim ownership, but this type of watermarking can be vulnerable to various adaptive attacks [42]. Similarly, white-box watermarking, which manipulates the weights with watermarking order in a functionality-preserving way, is also shown unreliable [62] under adversarial settings. Another direction is leveraging the inherent model patterns for ownership verification. For example, PublicCheck [60] realizes public verification of a deployed model at runtime through fingerprint samples. RAI^2 [14] verifies model identity using random projection without the need for fingerprints or watermarks. DeepJudge [10] implements a testing framework of six metrics for model similarity testing. Our work also informs model ownership disputes by auditing dataset usage, which directly affects model performance.

Datasets. There is also a line of work attempting to track dataset usage [18]. Dataset watermarking [37, 49] refers to marking datasets with backdoor-alike samples. Yet, the robustness is questionable under white-box access and low-sample settings [4]. On the other hand, non-invasive auditing can better preserve the dataset utility. Dataset Inference [44] tests whether the model is trained on the source dataset and is recently extended to the LLM datasets [43]. EMA [28] infers the dataset identity through enhanced membership inference to the suspect model. RAI^2 [14] uses a look-up table to estimate dataset similarity via black-box querying of the suspect model. ORL-AUDITOR [17] audits trajectory usage to protect the copyright of offline reinforcement learning datasets. Nevertheless, prior methods largely provide binary evidence of dataset use from shallow signals (e.g., memorization) and typically do not aim to attribute the underlying regeneration mechanism, especially

for distribution-level polishing. Our framework covers more advanced regeneration attacks and provides holistic auditing with coarse-grained attribution between shallow-feature regeneration (Attack ❶/❷) and distribution-level polishing (Attack ❸). In addition to the sample-, user-, and dataset-level categories surveyed in prior work [18], we further study distribution-level dataset tracking.

7 Discussion & Conclusion

In this work, we propose DIPBox for multi-scale tracking of regenerated datasets and validate its effectiveness on vision and text classification tasks.

Limitation & Future Work. DIPBox currently assumes white-box model access and partial dataset access to enable full-scale testing with all metrics. Although realistic for platform-based audits, this limits applicability in offline scenarios. In the black-box regime, distribution-level regeneration remains challenging to track because model outputs may provide insufficient signals without near-duplicate samples. As future work, we plan to explore representation-based techniques for black-box tracking.

Scalability. Although DIPBox currently targets small datasets, it can be extended to larger datasets. For large classification datasets, a divide-and-conquer strategy can aggregate local similarity measurements into global judgments. For generative datasets, OD and GD can be adapted by comparing latent representations, as model representations may also memorize training data.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No. U24A20241, 62325207, 62302298, 62332013). Hao Chen was partially supported by the JC STEM Lab Initiative.

References

[1] 10Duke. 2023. Software Copyright – Basics Explained. https://www.10duke.com/resources/glossary/software-copyright.

[2] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016*. ACM, 308–318.

[3] Ehsan Amid, Rohan Anil, Wojciech Kotlowski, and Manfred K. Warmuth. 2022. Learning from Randomly Initialized Neural Network Features. *CoRR* abs/2202.06438 (2022).

[4] Buse Gul Atli Tekgul and N. Asokan. 2022. On the Effectiveness of Dataset Watermarking. In *Proceedings of the 2022 ACM on International Workshop on Security and Privacy Analytics (IWSPA '22)*. Association for Computing Machinery, 93–99.

[5] Ms Aayushi Bansal, Dr Rewa Sharma, and Dr Mamta Kathuria. 2022. A systematic review on data scarcity problem in deep learning: solution and applications. *ACM Computing Surveys (Csur)* 54, 10s (2022), 1–29.

[6] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* 79, 1 (2010), 151–175.

[7] Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in NLI: Ways (Not) To Go Beyond Simple Heuristics. arXiv:2110.01518 [cs.CL]

[8] Mikolaj Binkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying MMD GANs. In *6th International Conference on Learning Representations, ICLR 2018*.

[9] Dingfan Chen, Raouf Kerkouche, and Mario Fritz. 2022. Private Set Generation with Discriminative Information. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[10] Jialuo Chen, Jingyi Wang, Tinglan Peng, Youcheng Sun, Peng Cheng, Shouling Ji, Xingjun Ma, Bo Li, and Dawn Song. 2022. Copy, Right? A Testing Framework for Copyright Protection of Deep Learning Models. In *43rd IEEE Symposium on Security and Privacy, SP 2022*. IEEE, 824–841.

[11] Adam Coates, Andrew Ng, and Honglak Lee. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 215–223.

[12] Former Contributor. 2015. Five Reasons To Copyright Register Your Software Now. https://www.forbes.com/sites/oliverherzfeld/2015/10/26/five-reasons-to-copyright-your-software-now.

[13] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. 2018. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505* (2018).

[14] Tian Dong, Shaofeng Li, Guoxing Chen, Minhui Xue, Haojin Zhu, and Zhen Liu. 2023. RAI2: Responsible Identity Audit Governing the Artificial Intelligence. In *30th Annual Network and Distributed System Security Symposium, NDSS 2023*. The Internet Society.

[15] Tian Dong, Yan Meng, Shaofeng Li, Guoxing Chen, Zhen Liu, and Haojin Zhu. 2025. Depth Gives a False Sense of Privacy: LLM Internal States Inversion. In *34th USENIX Security Symposium (USENIX Security 25)*. USENIX Association, Seattle, WA, 1629–1648.

[16] Tian Dong, Minhui Xue, Guoxing Chen, Rayne Holland, Yan Meng, Shaofeng Li, Zhen Liu, and Haojin Zhu. 2025. The Philosopher's Stone: Trojaning Plugins of Large Language Models. In *Network and Distributed System Security Symposium, NDSS 2025*. The Internet Society.

[17] Linkang Du, Min Chen, Mingyang Sun, Shouling Ji, Peng Cheng, Jiming Chen, and Zhikun Zhang. 2024. ORL-AUDITOR: Dataset Auditing in Offline Deep Reinforcement Learning. In *31st Annual Network and Distributed System Security Symposium, NDSS 2024*. The Internet Society.

[18] Linkang Du, Xuanru Zhou, Min Chen, Chusong Zhang, Zhou Su, Peng Cheng, Jiming Chen, and Zhikun Zhang. 2025. SoK: Dataset Copyright Auditing in Machine Learning Systems. In *IEEE Symposium on Security and Privacy, SP 2025*. IEEE.

[19] The Economist. 2024. AI firms will soon exhaust most of the internet's data. https://www.economist.com/schools-brief/2024/07/23/ai-firms-will-soon-exhaust-most-of-the-internets-data.

[20] Bent Fuglede and Flemming Topsøe. 2004. Jensen-Shannon divergence and Hilbert space embedding. In *Proceedings of the 2004 IEEE International Symposium on Information Theory, ISIT 2004*. IEEE, 31.

[21] Raja Giryes, Guillermo Sapiro, and Alexander M. Bronstein. 2016. Deep Neural Networks with Random Gaussian Weights: A Universal Classification Strategy? *IEEE Trans. Signal Process.* 64, 13 (2016), 3444–3457.

[22] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. 2012. A Kernel Two-Sample Test. *J. Mach. Learn. Res.* 13 (2012), 723–773.

[23] Kathrin Grosse, Lukas Bieringer, Tarek R. Besold, and Alexandre M. Alahi. 2024. Towards More Practical Threat Models in Artificial Intelligence Security. In *Proc. of USENIX Security*.

[24] Frederik Harder, Kamil Adamczewski, and Mijung Park. 2021. DP-MERF: Differentially Private Mean Embeddings with RandomFeatures for Practical Privacy-preserving Data Generation. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021 (Proceedings of Machine Learning Research, Vol. 130)*. PMLR, 1819–1827.

[25] Sameed Hayat. 2026. Guardian News Dataset. https://www.kaggle.com/datasets/sameedhayat/guardian-news-dataset. Accessed: 2026-06-18.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. of IEEE CVPR*.

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.

[28] Yangsibo Huang, Chun-Yin Huang, Xiaoxiao Li, and Kai Li. 2022. A Dataset Auditing Method for Collaboratively Trained Machine Learning Models. *IEEE Transactions on Medical Imaging* (2022).

[29] Zonghao Huang, Neil Zhenqiang Gong, and Michael K. Reiter. 2024. A General Framework for Data-Use Auditing of ML Models. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security, 2024*. ACM.

[30] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. 2022. Elucidating the Design Space of Diffusion-Based Generative Models. In *NeurIPS*.

[31] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2020. Training Generative Adversarial Networks with Limited Data. In *Proc. NeurIPS*.

[32] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-Free Generative Adversarial Networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*. 852–863.

[33] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. 2009. Attribute and simile classifiers for face verification. In *IEEE 12th International Conference on Computer Vision, ICCV 2009*. IEEE Computer Society, 365–372.

[34] Huseyin Kusetogullari, Amir Yavariabdi, Abbas Cheddad, Håkan Grahn, and Johan Hall. 2020. ARDIS: a Swedish historical handwritten digit dataset. *Neural Comput. Appl.* 32, 21 (2020), 16505–16518.

[35] Huseyin Kusetogullari, Amir Yavariabdi, Johan Hall, and Niklas Lavesson. 2021. DIGITNET: A Deep Handwritten Digit Detection and Recognition Method Using

a New Historical Handwritten Digit Dataset. *Big Data Res.* 23 (2021), 100182.
[36] Yiming Li, Yang Bai, Yong Jiang, Yong Yang, Shu-Tao Xia, and Bo Li. 2022. Untargeted Backdoor Watermark: Towards Harmless and Stealthy Dataset Copyright Protection. In *Advances in Neural Information Processing Systems (NeurIPS)*.
[37] Yiming Li, Mingyan Zhu, Xue Yang, Yong Jiang, Tao Wei, and Shu-Tao Xia. 2023. Black-box Dataset Ownership Verification via Backdoor Watermarking. *IEEE Transactions on Information Forensics and Security* (2023), 1–1. doi:10.1109/TIFS.2023.3265535
[38] Gaoyang Liu, Tianlong Xu, Xiaoqiang Ma, and Chen Wang. 2022. Your Model Trains on My Data? Protecting Intellectual Property of Training Data via Membership Fingerprint Authentication. *IEEE Trans. Inf. Forensics Secur.* 17 (2022), 1024–1037.
[39] Hui Liu, Hongzhi Luo, Shaofeng Li, Tian Dong, Guoxing Chen, Yan Meng, and Haojin Zhu. 2023. Privacy Computing with Right to Be Forgotten in Trusted Execution Environment. In *IEEE Global Communications Conference (GlobeCom), Kuala Lumpur, Malaysia*.
[40] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.
[41] Shayne Longpre, Robert Mahari, Anthony Chen, Naana Obeng-Marnu, Damien Sileo, William Brannon, Niklas Muennighoff, Nathan Khazam, Jad Kabbara, Kartik Perisetla, et al. 2024. A large-scale audit of dataset licensing and attribution in AI. *Nature Machine Intelligence* 6, 8 (2024), 975–987.
[42] Nils Lukas, Edward Jiang, Xinda Li, and Florian Kerschbaum. 2022. SoK: How Robust is Image Classification Deep Neural Network Watermarking?. In *43rd IEEE Symposium on Security and Privacy, SP 2022*. IEEE, 787–804.
[43] Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. 2024. LLM Dataset Inference: Did you train on my dataset?. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
[44] Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. 2021. Dataset Inference: Ownership Resolution in Machine Learning. In *International Conference on Learning Representations*.
[45] Bertin Martens. 2018. The importance of data access regimes for artificial intelligence and machine learning. *JRC Digital Economy Working Paper 2018-09* (2018).
[46] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. 2011. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning* (2011).
[47] Ilija Radosavovic, Raj Prateek Kosaraju, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2020. Designing Network Design Spaces. In *Proc. of IEEE/CVF CVPR*.
[48] Arpit Rajauria. 2020. pegasus paraphrase. https://huggingface.co/tuner007/pegasus_paraphrase.
[49] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. 2020. Radioactive data: tracing through training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020 (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 8326–8335.
[50] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*. Computer Vision Foundation / IEEE Computer Society, 4510–4520.
[51] Andrew M. Saxe, Pang Wei Koh, Zhenghao Chen, Maneesh Bhand, Bipin Suresh, and Andrew Y. Ng. 2011. On Random Weights and Unsupervised Feature Learning. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*. Omnipress, 1089–1096.
[52] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024. AI models collapse when trained on recursively generated data. *Nature* 631, 8022 (2024), 755–759.
[53] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[54] Mingxing Tan and Quoc V. Le. 2021. EfficientNetV2: Smaller Models and Faster Training. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021 (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 10096–10106.
[55] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *arXiv preprint arXiv:1908.08962* (2019).
[56] Hadas Unger, CHERGUELAINE Ayoub, and BOUBEKRI Faycal. 2023. CNN News Articles from 2011 to 2022. https://huggingface.co/datasets/AyoubChLin/CNN_News_Articles_2011-2022.
[57] James Vincent. 2023. Meta's powerful AI language model has leaked online. https://www.theverge.com/2023/3/8/23629362/meta-ai-language-modelllama-leak-online-misuse.
[58] Maxim Kuznetsov Vladimir Vorobev. 2023. A paraphrasing model based on ChatGPT paraphrases. https://huggingface.co/humarin/chatgpt_paraphraser_on_T5_base.
[59] Haibo Wang, Zhuolin Xu, Huaien Zhang, Nikolaos Tsantalis, and Shin Hwei Tan. 2026. Towards Understanding Refactoring Engine Bugs. *ACM Trans. Softw. Eng. Methodol.* 35, 5, Article 138 (April 2026), 55 pages. doi:10.1145/3747289
[60] Shuo Wang, Sharif Abuadbba, Sidharth Agarwal, Kristen Moore, Ruoxi Sun, Minhui Xue, Surya Nepal, Seyit Camtepe, and Salil Kanhere. 2022. PublicCheck: Public Integrity Verification for Services of Run-time Deep Models. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 1239–1256.
[61] Yifei Wang, Jizhe Zhang, and Yisen Wang. 2024. Do Generated Data Always Help Contrastive Learning?. In *The Twelfth International Conference on Learning Representations*.
[62] Yifan Yan, Xudong Pan, Mi Zhang, and Min Yang. 2023. Rethinking White-Box Watermarks on Deep Learning Models under Neural Structural Obfuscation. In *32nd USENIX Security Symposium, USENIX Security 2023*. USENIX Association, 2347–2364.
[63] Ze Yang, Can Xu, Wei Wu, and Zhoujun Li. 2019. Read, Attend and Comment: A Deep Architecture for Automatic News Comment Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*. Association for Computational Linguistics, 5076–5088.
[64] Yuxia Zhan, Yan Meng, Yichang Xiong Lu Zhou, Xiaokuan Zhang, Lichuan Ma, Guoxing Chen, Qingqi Pei, and Haojin Zhu. 2024. VPVet: Vetting Privacy Policies of Virtual Reality Apps. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security, 2024*. ACM.
[65] Huaien Zhang, Yu Pei, Junjie Chen, and Shin Hwei Tan. 2023. Statfier: Automated Testing of Static Analyzers via Semantic-Preserving Program Transformations. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (San Francisco, CA, USA) *(ESEC/FSE 2023)*. Association for Computing Machinery, New York, NY, USA, 237–249. doi:10.1145/3611643.3616272
[66] Huaien Zhang, Yu Pei, Shuyun Liang, and Shin Hwei Tan. 2024. Understanding and Detecting Annotation-Induced Faults of Static Analyzers. *Proc. ACM Softw. Eng.* 1, FSE, Article 33 (July 2024), 23 pages. doi:10.1145/3643759
[67] Huaien Zhang, Yu Pei, Shuyun Liang, Zezhong Xing, and Shin Hwei Tan. 2024. Characterizing and Detecting Program Representation Faults of Static Analysis Frameworks. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis* (Vienna, Austria) *(ISSTA 2024)*. Association for Computing Machinery, New York, NY, USA, 1772–1784. doi:10.1145/3650212.3680398
[68] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020 (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 11328–11339.
[69] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*. Computer Vision Foundation / IEEE Computer Society, 586–595.
[70] Zhifei Zhang, Yang Song, and Hairong Qi. 2017. Age Progression/Regression by Conditional Adversarial Autoencoder. In *Proc. of IEEE CVPR*.
[71] Bo Zhao and Hakan Bilen. 2023. Dataset Condensation with Distribution Matching. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023*. IEEE, 6503–6512.
[72] Haodong Zhao, Wei Du, Fangqi Li, Peixuan Li, and Gongshen Liu. 2023. Fedprompt: Communication-efficient and privacy-preserving prompt tuning in federated learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.
[73] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated Learning with Non-IID Data. *CoRR* abs/1806.00582 (2018).

A Ethical Considerations & Open Science

DIPBox[1] is an auditing support tool for assessing possible dataset regeneration, not a standalone legal verdict. It may benefit dataset owners, auditors, and users by improving accountability, but false or misinterpreted results could harm legitimate developers or platforms, and the attack analysis could aid evasion. Results should therefore be corroborated through human review and independent evidence. We conducted only offline experiments on publicly available datasets and models under their stated terms, obtained institutional IRB approval.

[1] https://zenodo.org/records/20746861