# ZeroMark: Towards Dataset Ownership Verification without Disclosing Watermarks

**Junfeng Guo**[1,*], **Yiming Li**[2,*], **Ruibo Chen**[1], **Yihan Wu**[1], **Chenxi Liu**[1], **Heng Huang**[1]
[1]Department of Computer Science, Institute of Health Computing
University of Maryland College Park
[2]College of Computing and Data Science, Nanyang Technology University
{gjf2023,ruibo,yihanwu,chenxi,heng}@umd.edu; liyiming.tech@gmail.com

## Abstract

High-quality public datasets significantly prompt the prosperity of deep neural networks (DNNs). Currently, dataset ownership verification (DOV), which consists of dataset watermarking and ownership verification, is the only feasible solution to protect their copyright by preventing unauthorized use. In this paper, we revisit existing DOV methods and find that they all mainly focused on the first stage by designing different types of dataset watermarks and directly exploiting watermarked samples as the verification samples for ownership verification. As such, their success relies on an underlying assumption that verification is a *one-time* and *privacy-preserving* process, which does not necessarily hold in practice. To alleviate this problem, we propose *ZeroMark* to conduct ownership verification without disclosing dataset-specified watermarks. Our method is inspired by our empirical and theoretical findings of the intrinsic property of DNNs trained on the watermarked dataset. Specifically, ZeroMark first generates the closest boundary version of given benign samples and calculates their boundary gradients under the label-only black-box setting. After that, it examines whether the given suspicious method has been trained on the protected dataset by performing a hypothesis test, based on the cosine similarity measured on the boundary gradients and the watermark pattern. Extensive experiments on benchmark datasets verify the effectiveness of our ZeroMark and its resistance to potential adaptive attacks. The codes for reproducing our main experiments are publicly available at GitHub.

## 1 Introduction

Deep neural networks (DNNs) have demonstrated their strong ability in widespread applications, such as face recognition [1, 2, 3]. Currently, there are many (high-quality) public datasets, such as CIFAR [4] and ImageNet [5], that can be easily downloaded and used. Arguably, their availability is one of the key factors in the prosperity of DNNs, as developers can evaluate and improve their models upon them. In particular, these datasets are usually only freely available for non-commercial use since their collection and annotation are time-consuming and even expensive.

To the best of our knowledge, dataset ownership verification (DOV) [6, 7, 8, 9, 10, 11] is currently the only feasible solution for protecting the copyright of public datasets. Specifically, DOV consists of two main stages, including dataset watermarking and ownership verification. In the first stage, dataset owners will introduce some imperceptible watermarked samples to generate the released watermarked version of the original dataset, so that all models trained on it will have specific distinctive prediction behaviors on particular samples ($i.e.$, verification samples) while having normal

[*]The first two authors contributed equally to this work.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).


https://doi.org/10.52202/079017-3829

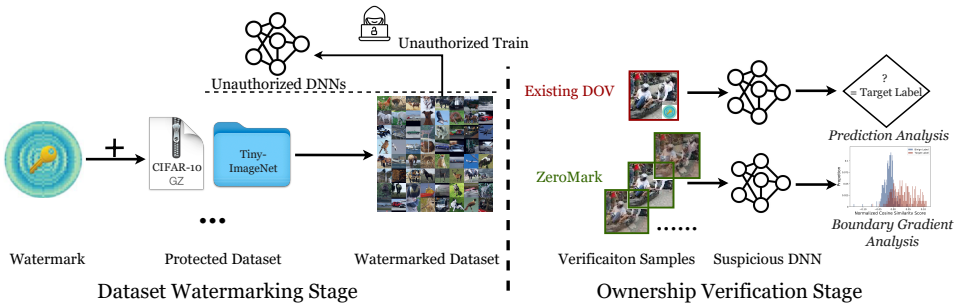



Figure 1: The overview of existing dataset ownership verification (DOV) methods and our ZeroMark. In the verification phase, existing DOV approaches directly exploit watermarked samples for verification purposes. In contrast, ZeroMark queries the suspicious model with boundary samples without disclosing dataset-specified watermarks to safeguard the verification process.

behaviors on standard testing samples. In the second stage, given the API of a suspicious third-party deployed model, the dataset owners will detect whether it is trained on the protected dataset by examining its prediction behaviors on verification samples.

In this paper, we revisit existing DOV methods. We find that they all mainly focused on the first stage by designing different types of dataset watermarks, no matter whether their watermark is backdoor-based [6, 7, 8] or not [9, 11]. All of them directly exploited watermarked samples as the verification samples in their second stage. Accordingly, their success relies on an underlying assumption that verification is a *one-time* and *privacy-preserving* process. Otherwise, as the watermark pattern is leaked during the (first) verification process, the malicious dataset users can easily remove the watermark from the model trained on the stolen dataset or from the verification samples. However, this assumption does not necessarily hold true in practice since adversaries can always update their unauthorized models. As such, an intriguing and critical question arises: *Could we verify dataset ownership without disclosing dataset-specified watermarks to ensure a secure verification process*?

The answer to the aforementioned problem is positive. In this paper, we first delve into the intrinsic property of DNNs trained on the watermarked dataset. We empirically and theoretically demonstrate that the gradient of watermarked models calculated upon the closest samples located at the decision boundary (dubbed 'boundary gradient') has a similar direction to their corresponding watermark patterns, measured by their cosine similarity. Specifically, the distribution of cosine similarity of the watermarked model on the dataset-specified target class has significantly larger values than that of the remaining (benign) classes. In particular, samples located at the decision boundary (dubbed 'boundary samples') of watermarked DNNs contain limited information about the dataset-specified watermarks, measured by several metrics ($e.g.$, mutual information). Motivated by these intriguing findings, we propose to conduct dataset ownership verification with closest boundary samples instead of samples containing dataset-specified watermarks (as shown in Figure 1). We call this method as ZeroMark. Specifically, our ZeroMark has three main steps. In the first step, ZeroMark generates the closest boundary version of given benign samples. The second step calculates the boundary gradients of generated closest boundary samples based on the Monte Carlo method. Both steps are conducted under the label-only black-box verification setting, where dataset owners can only query the suspicious model with verification samples via API and get its predicted labels. In the third step, ZeroMark examines whether the given suspicious method has been trained on the protected dataset via a hypothesis test, based on the distribution of cosine similarity measured between the boundary gradients and the corresponding watermark pattern.

In conclusion, our main contributions are four-fold: **(1)** We revisit existing DOV methods and reveal their underlying assumption regarding the verification phase. It does not necessarily hold, hindering the protection of dataset copyright. **(2)** We empirically and theoretically discover an intrinsic property of watermarked DNNs regarding boundary samples ($i.e.$, those located at the decision boundary). **(3)** We propose a simple yet effective method ($i.e.$, ZeroMark) to verify dataset ownership without disclosing dataset-specified watermarks. **(4)** We conduct experiments on benchmark datasets, verifying the effectiveness of ZeroMark and its resistance to potential adaptive methods.

## 2 Related Work

### 2.1 Data Protection

**Classical Data Protection.** Data protection is a classical and significant research area, which aims to prevent unauthorized data usage or protect private data. Existing classical data protection con-

sists of three main categories, including **(1)** encryption, **(2)** digital watermarking, and **(3)** privacy protection. Specifically, encryption [12, 13, 14] encrypts the sensitive data so that only authorized users with a secret key can decrypt and use it. Digital watermarking [15, 16, 17] embeds an owner-specified pattern to the protected data as the watermark to claim ownership. Privacy protection focuses on preventing the leakage of sensitive information of the data in both empirical [18, 19, 20] and certified manners [21, 22, 23]. Unfortunately, existing classical approaches can not directly protect the copyright of open-source datasets since they either hinder the dataset accessibility ($e.g.$, encryption) or require the information of the training process of models trained on them ($e.g.$, digital watermarking and privacy protection) that will not be disclosed by authorized dataset users.

**Dataset Ownership Verification.** Dataset ownership verification (DOV) aims to verify whether a suspicious model is trained on the protected dataset. To the best of our knowledge, this is currently the only feasible method to protect the copyright of open-source datasets. Specifically, DOV intends to introduce specific prediction behaviors (towards verification samples) in models trained on the protected dataset while preserving their performance on benign testing samples. Dataset owners can verify ownership by examining whether the suspicious model has dataset-specified distinctive behaviors. Previous DOV methods exploit either backdoor attacks [6, 7, 8, 10] or others [9, 11] to watermark the original (unprotected) benign dataset. For example, recently, backdoor-based DOV [6, 8, 7] adopted poisoned-/clean-label backdoor attacks to watermark the protected dataset. Most recently, Guo *et al*. [9] adopted samples from the hardly-generalized domain as watermark samples without introducing any new security vulnerability. However, all existing dataset ownership verification (DOV) approaches [6, 8, 7, 9, 10, 11] mainly focus on designing watermarks with different properties ($e.g.$, harmless and stealthy) and directly exploit the watermarked samples for verification. The security study of their verification stage remains blank and is worth further exploration.

### 2.2 Secure Machine Learning Inference

Currently, there are also a few works to safeguard the inference process of models. In the context of machine learning, secure inference is a two-party cryptographic protocol applied in the inference phase of machine learning models [24, 25, 26]. The server learns nothing about clients' input, while a client learns nothing about the server's machine learning model but can only get the results. Technically, it is implemented by having the server and client involved in a specific protocol and running the encrypted model over the encrypted input through cryptographic techniques such as homomorphic encryption [27] and secret sharing [28]. However, secure inference requires both the client and server to encrypt input data and adapt the machine learning model's operations accordingly through cryptographic mechanisms [27, 28]. As such, it is infeasible to protect the verification process of DOV methods since suspicious third-party models may not support these protocols.

## 3 The Property of Models Trained on the Watermarked Dataset

### 3.1 Preliminaries

**The Main Pipeline of Existing DOV Methods.** Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ denotes the original training dataset. In context of image classification task with $K$-classes, $i.e.$, $\boldsymbol{x}_i \in \mathcal{X} = [0,1]^{C\times W\times H}$ represents the image with $y_i \in \mathcal{Y} = \{1, \cdots, K\}$ as its label. In the first stage of DOV ($i.e.$, dataset watermarking), the dataset owner will embed watermarks to the original dataset to generate its watermarked version ($i.e.$, $\mathcal{D}_w$). Particularly, $\mathcal{D}_w = \mathcal{D}_m \cup \mathcal{D}_b$, where $\mathcal{D}_m$ represents the watermarked version of samples from a small selected subset $\mathcal{D}_s$ of $\mathcal{D}$ ($i.e.$, $\mathcal{D}_s \subset \mathcal{D}$) and $\mathcal{D}_b$ contains remaining benign samples ($i.e.$, $\mathcal{D}_b = \mathcal{D} - \mathcal{D}_s$). The $\mathcal{D}_m$ is generated by the dataset-specified image generator $G_x : \mathcal{X} \to \mathcal{X}$ and the label generator $G_y : \mathcal{Y} \to \mathcal{Y}$, $i.e.$, $\mathcal{D}_m = \{(G_x(\boldsymbol{x}), G_y(y)) | (\boldsymbol{x}, y) \in \mathcal{D}_s\}$. For example, $G_x = (\boldsymbol{1} - \boldsymbol{m}) \odot \boldsymbol{\Delta} + \boldsymbol{m} \odot \boldsymbol{x}$ and $G_y = y_t$ in BadNets-based DOV [29, 6], where $\boldsymbol{m} \in \{0,1\}^{C\times W\times H}$ is the trigger mask, $\boldsymbol{\delta} \in [0,1]^{C\times W\times H}$ is the trigger pattern, $\odot$ denotes the element-wise product, and $y_t$ is the target label. In particular, $\gamma \triangleq \frac{|\mathcal{D}_m|}{|\mathcal{D}_w|}$ is the *watermarking rate*. In the second phase ($i.e.$, ownership verification), for a suspicious model $C : \mathcal{X} \to \mathcal{Y}$ that may be trained on $\mathcal{D}_w$, the dataset owners will investigate whether it conducts unauthorized training by querying it with verification samples under the black-box setting. In general, the verification process of existing DOV is to directly uses watermarked sample $G_x(\boldsymbol{x})$ as verification samples to examine

 https://doi.org/10.52202/079017-3829

whether $C(G_x(\boldsymbol{x})) = G_y(y)$. In contrast, our goal is to perform the verification process without disclosing the watermark samples $G_x(\boldsymbol{x})$ during the inference phase of the suspicious classifier $C$.

**Boundary Samples.** Let the logit margin of model $f : \mathcal{X} \to [0,1]^K$ on the label $y$ is denoted by:

$$\phi_y(\boldsymbol{x}; \boldsymbol{w}) = f_y(\boldsymbol{x}; \boldsymbol{w}) - \max_{y' \neq y} f_{y'}(\boldsymbol{x}; \boldsymbol{w}). \tag{1}$$

It can be observed that $\boldsymbol{x}$ can be classified as $y$ by $f(\cdot; \boldsymbol{w})$ if and only if $\phi_y(\boldsymbol{x}; \boldsymbol{w}) \geq 0$. As such, the set for boundary samples of class $y$ can be denoted by $\mathcal{B}(y; \boldsymbol{w}) = \{x : \phi_y(\boldsymbol{x}; \boldsymbol{w}) = 0\}$.

### 3.2 Approach the Closest Boundary Sample

To obtain the boundary samples, we can easily use a gradient-free method ($i.e.$, geometric search) to move each given sample $\boldsymbol{x}$ forward the decision boundary of $f(\cdot; \boldsymbol{w})$ under the label $y$, as follows:

$$\overline{\boldsymbol{x}} = \alpha \cdot \boldsymbol{x} + (1 - \alpha) \cdot \boldsymbol{x}_y, \ s.t. \ \phi_y(\overline{\boldsymbol{x}}; \boldsymbol{w}) = 0, \tag{2}$$

where $\alpha \in [0, 1]$ is a line search parameter and $\boldsymbol{x}_y$ is a sample classified by the model $f(\cdot; \boldsymbol{w})$ as $y$.

However, the obtained boundary sample of $\boldsymbol{x}$ would be varied according to different $\boldsymbol{x}_y$. As such, we use the *closest boundary sample* of $\boldsymbol{x}$ to study the characteristics of watermarked models.

Following the previous work [30], we define the closest boundary sample of $\boldsymbol{x}$ ($i.e.$, $\overline{\boldsymbol{x}}^*$) as:

$$\overline{\boldsymbol{x}}^* \triangleq \arg\min_{\overline{\boldsymbol{x}}} ||\overline{\boldsymbol{x}} - \boldsymbol{x}||_p \ s.t. \ \phi_y(\overline{\boldsymbol{x}}; \boldsymbol{w}) = 0, \tag{3}$$

where $|| \cdot ||_{1 \leq p \leq \infty}$ is the $\ell_p$ norm.

Specifically, we can exploit the fast adaptive boundary attack (FAB) [31] to calculate the closest boundary sample. In particular, we adapt FAB to implement an iterative algorithm with gradient ascend using $\nabla_{\boldsymbol{x}} \phi_y(\boldsymbol{x}; \boldsymbol{w})$, whose update in $(t+1)$-th iteration is as follows:

$$\overline{\boldsymbol{x}}_{t+1} = \alpha_t \cdot \boldsymbol{x}_0 + (1 - \alpha_t) \cdot \left\{ \overline{\boldsymbol{x}}_t + \beta_t \cdot \frac{\nabla_{\boldsymbol{x}} \phi_y(\overline{\boldsymbol{x}}_t; \boldsymbol{w})}{||\nabla_{\boldsymbol{x}} \phi_y(\overline{\boldsymbol{x}}_t; \boldsymbol{w})||} \right\}, \tag{4}$$

where $\beta_t$ is a positive step size, $\boldsymbol{x}_0$ is an initial point such that $\phi_y(\boldsymbol{x}_0; \boldsymbol{w}) \leq 0$ and $\alpha_t \in [0, 1]$ is chosen to ensure $\overline{\boldsymbol{x}}_{t+1}$ lies in the decision boundary as Eq. (2). In practice, $\boldsymbol{x_0}$ is randomly selected in the validation set whose label is different from $y$.

In general, using the closest boundary samples generated via Eq. (4) is mostly because they are closely related to the intrinsic property of watermarked DNNs, as shown in the next subsection.

### 3.3 The Characteristic of Boundary Gradient of Watermarked DNNs

In this section, we will show that the gradient of closest boundary samples $\nabla_{\boldsymbol{x}} \phi_y(\overline{\boldsymbol{x}}^*; \boldsymbol{w})$ (dubbed 'boundary gradients') of watermarked DNNs is closely related to the watermark patterns. Before we present our technical details, we first define $\cos\angle(\boldsymbol{x}, \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_t))$ as follows:

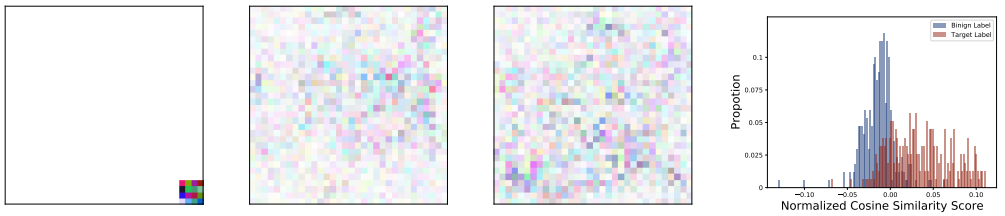
$$\cos\angle(\boldsymbol{x}, \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_t)) \triangleq \frac{< \boldsymbol{x}, \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_t) >}{||\boldsymbol{x}||_2 \cdot ||\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_t)||_2}. \tag{5}$$

Following previous works [32, 9], we use a model $f(\cdot; \boldsymbol{w})$ watermarked through the standard BadNets backdoor attack ($i.e.$, $G_x(\boldsymbol{x}) = (\boldsymbol{1} - \boldsymbol{m}) \odot \boldsymbol{x} + \boldsymbol{m} \odot \boldsymbol{\Delta}$) [29] as a basic example to shed light on the intriguing characteristic of watermarked DNNs.

**Theorem 1** (Property of Boundary Gradient on the Closest Boundary Sample)**.** *Assume that $\phi_y(\overline{\boldsymbol{x}}_t; \boldsymbol{w})$ is twice differentiable with a Lipschitz gradient, if $|\mathcal{D}_m| \to \infty$ and by updating $\overline{\boldsymbol{x}}_t$ in Eq. (4) with step size $\beta_t = ||\overline{\boldsymbol{x}}_t - \boldsymbol{x}_0||_2 \cdot t^{q-1}$, there exists a constant $c \geq 0$ such that*

$$\lim_{|\mathcal{D}_m| \to \infty} 1 - cos\angle(\boldsymbol{m} \odot \boldsymbol{\delta}, \boldsymbol{m} \odot \nabla_{\boldsymbol{x}} \phi_{y_t}(\overline{\boldsymbol{x}}^*, \boldsymbol{w})) \ \leq c \cdot (t^*)^{q-1}, \tag{6}$$

*where $q \in (\frac{1}{2}, 1)$, $y_t$ is the target label ($i.e.$, $y_t = C(G_x(\boldsymbol{x}))$), $\boldsymbol{\delta}$ is the watermark pattern ($i.e.$, $\boldsymbol{\delta} \triangleq G_x(\boldsymbol{x}_0) - \boldsymbol{x}_0$), and $t^*$ is the number of convergence iterations of $\overline{\boldsymbol{x}}^*$'s update.*




(a) Watermark (b) Grad (Benign) (c) Grad (Target) (d) Distribution of $\cos\angle(\cdot)$

Figure 2: **(a)** shows the watermark pattern for BadNets [29] used in our empirical study. **(b)** and **(c)** are examples of boundary gradients calculated under benign and target labels. **(d)** shows the distribution for the cosine similarity calculated over boundary gradients for benign and target labels. More empirical studies on other types of watermarks are included in the appendix.

In general, since the right side of inequality (6) decreases with the increase of $t^*$, Theorem 1 indicates that the cosine similarity between the watermark pattern $\boldsymbol{\delta}$ and the gradient calculated on $\overline{\boldsymbol{x}}_t$ located in the watermark region increases along with its update process. Its proof is in the appendix.

In the following parts, we empirically verify that the distribution of the cosine similarity between watermark patterns and boundary gradients of the closest boundary samples is different ($i.e.$, has larger values) from that on their benign versions to further justify our Theorem 1.

**Settings.** We hereby exploit BadNets-based dataset watermark [29, 6] with ResNet-18 [33] on the CIFAR-10 [4] dataset as an example for the discussion. Specifically, we watermark a sufficient amount of samples to achieve a high watermark success rate ($i.e.$, $\geq 99\%$). The watermark pattern is a $4 \times 4$ square filled with random pixels, as shown in Figure 2(a). We randomly select 400 benign samples and use Eq. (4) to generate their closest boundary version for the target label $y_t$ ($i.e.$, class '0'). To reduce the randomness caused by the selection of watermark patterns, we introduce $\overline{cos}$ as the reference for normalization, as follows:

$$\overline{\cos} \triangleq \frac{1}{N}\sum_{i=1}^{N} \cos\angle(\boldsymbol{m} \odot \boldsymbol{\delta}_i, \boldsymbol{m} \odot \nabla_{\boldsymbol{x}} \phi_{y_t}(\overline{\boldsymbol{x}}^*, \boldsymbol{w})), \tag{7}$$

and we calculate the normalized cosine similarity score as:

$$\widehat{\cos}\angle(\boldsymbol{m} \odot \boldsymbol{\delta}, \boldsymbol{m} \odot \nabla_{\boldsymbol{x}} \phi_{y_t}(\overline{\boldsymbol{x}}^*, \boldsymbol{w})) \triangleq \cos\angle(\boldsymbol{m} \odot \boldsymbol{\delta}, \boldsymbol{m} \odot \nabla_{\boldsymbol{x}} \phi_{y_t}(\overline{\boldsymbol{x}}^*, \boldsymbol{w})) - \overline{\cos}, \tag{8}$$

where $\boldsymbol{\delta}_i$ is $i$-th random watermark pattern that is different from the original one ($i.e.$, $\boldsymbol{\delta}$). We generate the gradient $\nabla_{\boldsymbol{x}} \phi_y(\overline{\boldsymbol{x}}^*, \boldsymbol{w})$ of the watermarked model on the target label (dubbed 'Grad (Target)') and benign labels (dubbed 'Grad (Benign)') others than the target one, as shown in Figure 2(b) and Figure 2(c), respectively. We then calculate their normalized cosine similarity scores with 400 samples. Since the values within the boundary gradient are sparse and not evenly distributed, we follow previous work [34] to select the largest 10 values within the $\boldsymbol{m} \odot \boldsymbol{\delta}$ and $\boldsymbol{m} \odot \nabla_{\boldsymbol{x}} \phi_{y_t}(\overline{\boldsymbol{x}}^*, \boldsymbol{w}))$ to calculate their corresponding cosine similarity score. More settings details are in the appendix.

**Results.** As shown in Figure 2(d), the normalized cosine similarity scores of the target label have significantly larger values compared with those of benign labels. However, their similarity scores still have some overlap (nearly $74\%$ of the target label). It suggests that not all gradients calculated on the closest boundary samples can reflect the watermark pattern $\boldsymbol{\delta}$. It is mostly caused by the deviations introduced by the gradient estimation process under the black-box setting (as in Eq. (11)).

Nevertheless, we can still distinguish between watermarked and benign models based on their similarity distributions by comparing their *maximum* instead of random values. Specifically, suppose we define a threshold $\tau > 0$ as the maximum cosine similarity value for benign labels, and there exists $\mathbb{P}[\widehat{\cos}(\cdot) > \tau] \approx 0.26$ for the watermark model. If we randomly sample $m$ samples to calculate the (closest) boundary gradients and their corresponding normalized cosine similarity, we have:

$$\mathbb{P}(\max\{\widehat{\cos}(\cdot)_1, \widehat{\cos}(\cdot)_2, \cdots, \widehat{\cos}(\cdot)_m\} \leq \tau) = (\mathbb{P}(\widehat{\cos}(\cdot) \leq \tau))^m. \tag{9}$$

There will be at least one sample having $\widehat{\cos}(\cdot) > \tau$ with a probability of $1 - \mathbb{P}[\widehat{\cos}(\cdot) < \tau]^m$. As such, if we sample sufficient samples ($e.g.$, 100), we will have a large chance ($\geq 99\%$) to find at least one boundary gradient larger than $\tau$ to successfully identify the watermark models. These phenomena inspire the design of our ZeroMark method, as proposed in the next section.


https://doi.org/10.52202/079017-3829

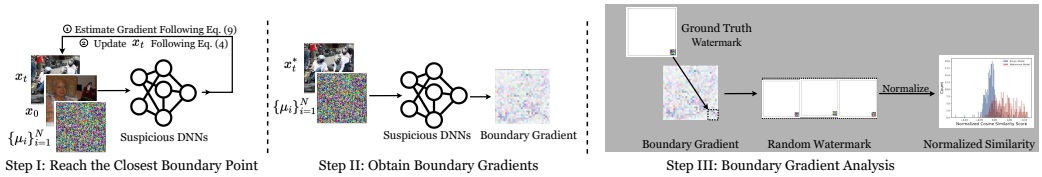



Figure 3: The main pipeline of our ZeroMark, which consists of three main steps. In the first steps, ZeroMark applies Eq. (4) and Eq. (10) to generate the closest boundary samples. In the second step, ZeroMark estimates the boundary gradients for generated boundary samples. In the third step, ZeroMark first leverages the ground truth watermark pattern and random watermark patterns to calculate the normalized cosine similarity following Eq. (7) and Eq. (8). After that, ZeroMark uses the distribution of similarity scores to conduct $t$-test for dataset ownership verification. In particular, in the third stage, patterns ($i.e.$, random and ground truth watermarks) displayed in the gray background are only available to the defender and inaccessible to the suspicious model.

## 4 Methodology

In this section, we describe the threat model and the technical details of our ZeroMark method.

**Threat Model.** Consistent with previous DOV methods [7, 6, 8], we assume that dataset owners will watermark the original dataset to generate its watermarked version. The dataset owner has full knowledge of watermark patterns and the validation samples used for ownership verification. In the verification stage, given a suspicious model, they will examine whether it was trained on the protected dataset under the label-only black-box setting, where they can only query the model with verification samples via model API and get its predicted labels, without accessing its intermediate results ($e.g.$, gradients) and model parameters.

**The Main Pipeline of ZeroMark.** In general, our ZeroMark consists of three main steps, as shown in Figure 3. ZeroMark first generates the (closest) boundary samples of the suspicious model. Then, it calculates the boundary gradients of the generated boundary samples. ZeroMark conducts dataset ownership verification via boundary gradient analysis in the third step.

**Step 1. Generate Closest Boundary Samples.** ZeroMark follows Eq. (4) to optimize the closest boundary samples $\overline{\boldsymbol{x}}^*$ iteratively with gradient decent. In particular, we exploit Monte Carlo method to estimate $\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})$ to address the challenge in our considered label-only black-box scenarios, where the gradients for given inputs are inaccessible. The overall process is as follows:

$$\overline{\boldsymbol{x}}_{t+1} = \alpha_t \cdot \boldsymbol{x}_0 + (1-\alpha_t) \cdot \left\{ \overline{\boldsymbol{x}}_t + \beta_t \cdot \frac{\frac{1}{N}\sum_{i=1}^{N} \phi_y(\overline{\boldsymbol{x}}_t + \kappa \cdot \mu_i; \boldsymbol{w}) \cdot \mu_i}{||\frac{1}{N}\sum_{i=1}^{N} \phi_y(\overline{\boldsymbol{x}}_t + \kappa \cdot \mu_i; \boldsymbol{w}) \cdot \mu_i||} \right\}, \tag{10}$$

where $\beta_t$ is the step size, $\boldsymbol{x}_0$ is an initial point such that $\phi_y(\boldsymbol{x}_0;\boldsymbol{w}) \leq 0$, $\alpha_t \in [0,1]$ is chosen to ensure $\overline{\boldsymbol{x}}_{t+1}$ lies in the decision boundary as Eq. (2), $\{\mu_i\}_{i=1}^N \sim N(0,1)$ are $N$ random noises $i.i.d$ sampled from the standard Gaussian distribution, and $\kappa$ is a fixed positive parameter ($i.e.$, 0.01).

**Step 2. Calculate Boundary Gradients.** Once the closest boundary sample $\overline{\boldsymbol{x}}^*$ is generated, we can also exploit Monte Carlo method to estimate its gradient (dubbed 'boundary gradient'), as follows:

$$\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}^*;\boldsymbol{w}) \approx \frac{1}{N}\sum_{i=1}^{N} \phi_y(\overline{\boldsymbol{x}}^* + \kappa \cdot \mu_i; \boldsymbol{w}) \cdot \mu_i. \tag{11}$$

**Step 3. Boundary Gradient Analysis.** After obtaining the boundary gradients, ZeroMark first calculates the cosine similarity based on the available watermark pattern $\boldsymbol{\delta}$ and obtains boundary gradients. To further mitigate the variance caused by the watermark patterns, we create several random watermark patterns and follow Eq. (8) to normalize the calculated cosine similarity. After that, motivated by the characteristic described in Section 3.3, where the cosine similarity of the boundary gradients on the target label $y_t$ has a significantly larger value compared with that of benign labels, we design a hypothesis-test-guided method to conduct ownership verification based on the range of cosine similarity for verification, as follows.

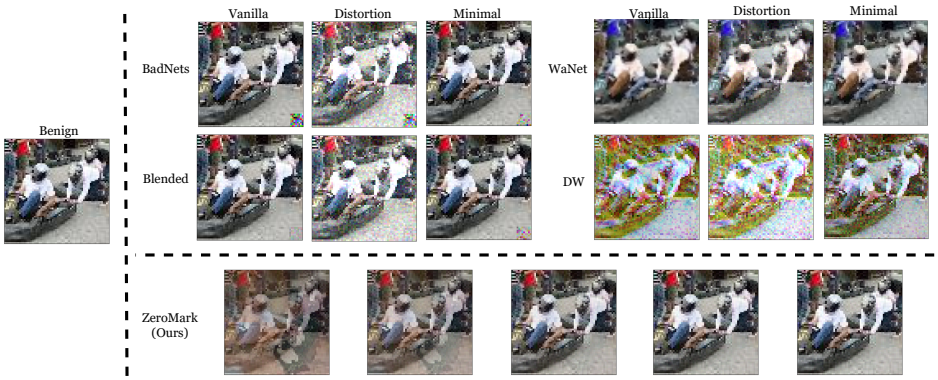



Figure 4: The example of verification samples across different watermarks ($i.e.$, BadNets, Blended, WaNet, DW) and verification methods ($i.e.$, Vanilla, Minimal, Distortion) on Tiny-ImageNet.

**Proposition 1.** *Suppose* $\widehat{cos}\angle(\boldsymbol{m} \odot \boldsymbol{\delta}, \boldsymbol{m} \odot \nabla_{\boldsymbol{x}} \phi_{y_t}(\overline{\boldsymbol{x}}^*, \boldsymbol{w}))$ *is the posterior normalized cosine similarity between boundary gradients and the available trigger pattern of the suspicious model* $f(\cdot, \boldsymbol{w})$*. Let variables* $P_b$ *and* $P_d$ *denote the largest* $Q\%$ *over* $\{\widehat{cos}\angle(\boldsymbol{m} \odot \boldsymbol{\delta}, \boldsymbol{m} \odot \nabla_x \phi_{y_t}(\overline{\boldsymbol{x}}_i^*, \boldsymbol{w}))\}_{i=1}^m$ *and* $\{\widehat{cos}\angle(\boldsymbol{m} \odot \boldsymbol{\delta}, \boldsymbol{m} \odot \nabla_x \phi_{y \neq y_t}(\overline{\boldsymbol{x}}_i^*, \boldsymbol{w}))\}_{i=1}^m$*, respectively. Given the null hypothesis* $H_0 : P_b = P_d + \tau$ *(*$H_1 : P_b > P_d + \tau$*) where the hyper-parameter* $\tau \in [0, 1]$*, we claim that the suspicious model is trained on the protected dataset (with* $\tau$*-certainty) if and only if* $H_0$ *is rejected.*

In general, we randomly select $m$ ($i.e.$, 500) validation samples evenly distributed across different classes to generate boundary gradients for the watermarked and benign labels. Then, we conduct the pairwise $t$-test [35] and calculate its p-value. The null hypothesis $H_0$ is rejected if the p-value is smaller than the significance level $\alpha$. Besides, we also calculate the *confidence score* $\Delta P = P_b - P_d$ to represent the verification confidence. *The larger the* $\Delta P$*, the more confident the verification.*

## 5 Experiments

In this section, we conduct experiments on CIFAR-10 [4] and Tiny-ImageNet [36] datasets with ResNet18 and ResNet-34 [33], respectively. More results with other settings are in our appendix.

### 5.1 The Performance of Verification Samples Generated by ZeroMark

**Settings.** We hereby compare our ZeroMark method to two straightforward baseline methods, including **(1)** verification with minimal watermark (dubbed 'Minimal') and **(2)** verification with distorted watermark (dubbed 'Distortion'). These methods intend to protect the information of dataset-specified watermarks by perturbing the original watermarks. We also provide the results of verification with benign samples (dubbed 'Benign') and verification with original dataset-specified watermarked samples (dubbed 'Vanilla') for reference. We evaluate each verification method on four dataset watermark techniques, including two patch-based watermarks [29, 37] and two input-specific ones [38, 9]. Regarding the implementation of existing watermark techniques, we follow their default settings. The example of verification samples across different watermarks and verification methods is shown in Figure 4. Please find more details in the appendix.

**Evaluation Metrics.**We adopt mean square error (MSE), neuron activation similarity (NAS), and mutual information (MI) to measure the degree to which the dataset-specified watermarks are disclosed during the verification stage. Specifically, the MSE is defined as the mean square error between verification and their corresponding watermarked samples in the region of watermark patterns. NAS is calculated as the cosine similarity in the neuron activation map between verification and their corresponding watermarked samples. MI is calculated based on the distribution of verification and their corresponding watermarked samples. More details are in our appendix.

**Results.** As shown in Table 1, our approach produces larger cosine similarity scores of the target label than benign labels. We show the results regarding the watermark-disclosed degree for different approaches in Table 2-3. The results show that our approach can reach the most minor watermark-


https://doi.org/10.52202/079017-3829

Table 1: The averaged largest $Q\%$ cosine similarity of our method on different watermarks.

| Dataset→ | CIFAR-10 | | | | | TinyImageNet | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Label↓, Watermark→ | BadNets | Blended | WaNet | DW | Ave. | BadNets | Blended | WaNet | DW | Ave. |
| Benign | 0.028 | 0.030 | 0.022 | 0.028 | 0.027 | 0.026 | 0.021 | 0.029 | 0.027 | 0.026 |
| Target | 0.102 | 0.368 | 0.131 | 0.099 | 0.174 | 0.148 | 0.334 | 0.131 | 0.123 | 0.194 |

Table 2: The performance on CIFAR-10. In particular, we mark the best results in bold while the value within the underline denotes the second-best results (except the benign samples).

| Metric→ | MSE (↑) | | | | NAS (↓) | | | | MI (↓) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Watermark→ Method↓ | BadNets | Blended | WaNet | DW | BadNets | Blended | WaNet | DW | BadNets | Blended | WaNet | DW |
| Benign | 0.394 | 0.197 | 0.077 | 0.309 | 0.597 | 0.617 | 0.609 | 0.665 | 15.9 | 19.7 | 22.3 | 28.5 |
| Vanilla | 0 | 0 | 0 | 0 | 0.830 | 0.801 | 0.767 | 0.824 | 64.3 | 56.1 | 58.2 | 61.7 |
| Minimal | 0.193 | 0.171 | 0.121 | 0.197 | 0.797 | 0.769 | 0.721 | 0.743 | 54.8 | 51.5 | 44.0 | 52.3 |
| Distortion | 0.286 | **0.251** | 0.087 | **0.301** | 0.770 | 0.774 | 0.701 | 0.769 | 56.6 | 53.8 | 35.4 | 44.5 |
| ZeroMark (Ours) | **0.392** | 0.202 | **0.199** | 0.246 | **0.646** | **0.671** | **0.688** | **0.689** | **18.1** | **24.4** | **28.6** | **29.7** |

Table 3: The performance on Tiny-ImageNet. In particular, we mark the best results in bold while the value within the underline denotes the second-best results (except the benign samples).

| Metric→ | MSE (↑) | | | | NAS (↓) | | | | MI (↓) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Watermark→ Method↓ | BadNets | Blended | WaNet | DW | BadNets | Blended | WaNet | DW | BadNets | Blended | WaNet | DW |
| Benign | 0.396 | 0.189 | 0.076 | 0.298 | 0.497 | 0.561 | 0.589 | 0.629 | 27.6 | 29.8 | 28.2 | 31.4 |
| Vanilla | 0 | 0 | 0 | 0 | 0.817 | 0.808 | 0.763 | 0.804 | 68.1 | 59.0 | 64.7 | 67.4 |
| Minimal | 0.187 | 0.096 | 0.077 | 0.189 | 0.768 | 0.782 | **0.696** | 0.749 | 57.3 | 54.6 | 49.9 | 54.7 |
| Distortion | 0.263 | **0.227** | 0.079 | **0.227** | 0.773 | 0.745 | 0.738 | 0.773 | 61.4 | 55.7 | 39.2 | 48.3 |
| ZeroMark (Ours) | **0.314** | 0.204 | **0.171** | 0.201 | **0.662** | **0.697** | 0.704 | **0.698** | **27.7** | **33.2** | **30.5** | **31.7** |

Table 4: The verification performance of our method on different watermarks.

| Dataset→ | | CIFAR-10 | | | Tiny-ImageNet | | |
|---|---|---|---|---|---|---|---|
| Watermark↓ | Metric↓, Scenario→ | Independent-W | Independent-M | Malicious | Independent-W | Independent-M | Malicious |
| BadNets | $\Delta P$ | 0.012 | 0.013 | 0.081 | 0.011 | 0.012 | 0.127 |
| | p-value | 1.00 | 1.00 | $10^{-45}$ | 1.00 | 1.00 | $10^{-58}$ |
| Blended | $\Delta P$ | 0.010 | 0.013 | 0.35 | 0.016 | 0.012 | 0.313 |
| | p-value | 1.00 | 1.00 | $10^{-67}$ | 1.00 | 1.00 | $10^{-64}$ |
| WaNet | $\Delta P$ | 0.028 | 0.012 | 0.102 | 0.022 | 0.011 | 0.110 |
| | p-value | 0.80 | 1.00 | $10^{-53}$ | 0.90 | 1.00 | $10^{-55}$ |
| DW | $\Delta P$ | 0.023 | 0.014 | 0.071 | 0.030 | 0.002 | 0.101 |
| | p-value | 0.88 | 1.00 | $10^{-12}$ | 0.74 | 1.00 | $10^{-49}$ |

disclosed degree in almost all cases. The improvement is significant compared to the vanilla DOV methods. These results verify the effectiveness and security of our ZeroMark.

### 5.2 The Performance of Dataset Ownership Verification via ZeroMark

**Settings.** We evaluate our ZeroMark for ownership verification under three scenarios, including **(1)** independent watermark (dubbed 'Independent-W'), **(2)** independent model (dubbed 'Independent-M'), and **(3)** unauthorized dataset training (dubbed 'Malicious'). In the first case, we used ZeroMark to verify the suspicious model affected with other watermark patterns; In the second case, we test the benign model with our ZeroMark; In the last case, we use ZeroMark to verify the watermarked model with corresponding ground truth watermark samples. Notice that only the last case should be regarded as having unauthorized dataset use. More setting detail are described in the appendix.

**Evaluation Metrics.** Following the settings in [7, 6], we use $\Delta P \in [-1, 1]$ and p-value $\in [0, 1]$ for the evaluation. For the first two independent cases, a small $\Delta P$ and a large $p$-value are expected. In contrast, for the third one, the larger $\Delta P$ and the smaller the p-value, the better the verification.

**Results.** As shown in Table 4, our method can achieve accurate dataset ownership verification in all cases. Specifically, our approach can identify the unauthorized dataset usage with high confidence ($i.e.$, p-value $\ll 0.01$ for 'Malicious' case), while not misjudging when there is no unauthorized dataset utilization ($i.e.$, p-value $\gg 0.1$ for 'Independent-W' and 'Independent-M').

### 5.3 Ablation Study

We hereby study the effects of two key hyper-parameters. More results are in our appendix.

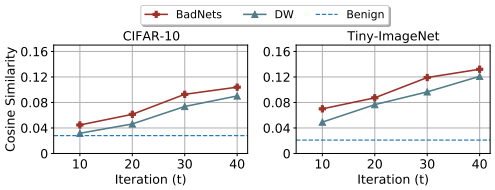



Figure 5: Effects of iteration size $t$.

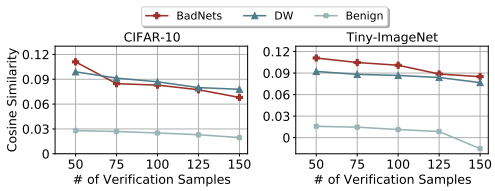



Figure 6: Effects of verification sample size.

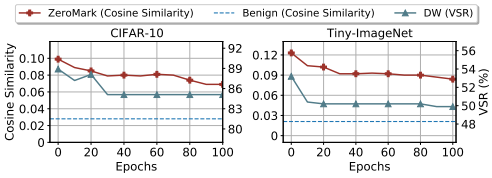



Figure 7: Robustness against fine-tuning.

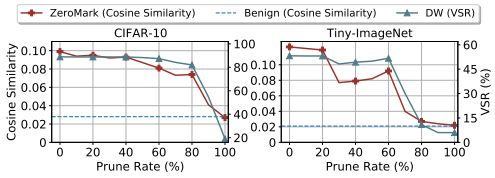



Figure 8: Robustness against model pruning.

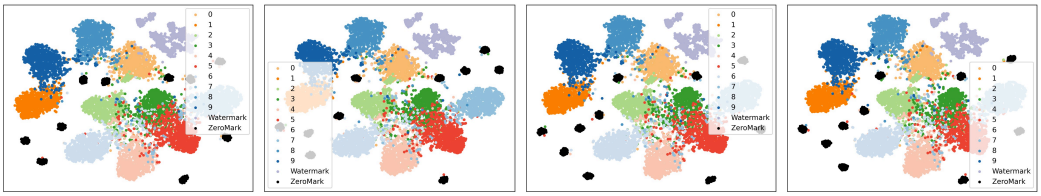



Figure 9: The t-SNE results with varied optimization iterations $t$ for the embedding features of benign, watermark, and ZeroMark samples extracted from watermarked DNNs.

**Effects of Optimization Iterations $t$.** We exploit BadNets and DW as the representatives of patch-based and sample-specific watermarks to study the effects of $t$ on both CIFAR-10 and Tiny-ImageNet datasets. As shown in Figure 5, the cosine similarity scores increase with the increase of $t$. With $t \geq 30$, we can easily distinguish watermarked and benign models.

**Effects of the Largest Number of Verification Samples** ($i.e.$, $m \times Q\%$). As shown in Figure 6, with the number of verification samples used in the $t$-test increases, the cosine similarity decreases while the benign models' cosine similarity remains stable on both datasets. However, our method can still have a promising cosine similarity even with many verification samples ($e.g.$, 150).

### 5.4 The Resistance to Potential Adaptive Methods

We hereby select domain watermark to evaluate the robustness of ZeroMark, as the domain watermark itself is sufficiently robust against this defense. Following the previous work [7], we here evaluate the robustness of our ZeroMark under fine-tuning [39] and model pruning [40]. As shown in Figure 7, fine-tuning has minor effects on our method. In Figure 8, we can see ZeroMark performs resilient against model pruning as its efficacy decreases along with the domain watermark. The results of our resistance to other methods are in our appendix.

### 5.5 A Closer Look to the Effectiveness of our ZeroMark

In this section, we intend to further explore the mechanisms behind the effectiveness of our ZeroMark. Specifically, we adopt t-SNE [41] to visualize the feature distribution of watermark samples, benign samples, and samples generated by ZeroMark of watermarked DNNs. As shown in Figure 9, with varied optimization iterations $t$ for generating (closest) boundary samples, samples generated by ZeroMark can always stay far away from the watermark samples' distribution, which demonstrates ZeroMark can prevent disclosing the watermark information from the watermark samples.

## 6 Potential Limitations and Future Directions

ZeroMark can conduct dataset ownership verification without disclosing the watermark pattern. However, as the first work towards a secure verification process of dataset ownership verification (DOV) methods, we have to admit that we still have some potential limitations.


https://doi.org/10.52202/079017-3829

Firstly, we must admit that our method requires additional time to conduct ownership verification since it needs to generate some boundary samples and their gradients under the black-box setting. For example, it takes nearly 30 mins for verification on CIFAR-10. While this cost is acceptable in practice to a large extent, we will discuss how to accelerate our ZeroMark in our future work.

Secondly, ZeroMark currently can only perform effectively for watermark techniques with a predefined target label ($e.g.$, BadNets [29], etc. For other watermark techniques, which have no predefined target label ($i.e.$, UBW [7]), ZeroMark can not conduct boundary gradient analysis and, therefore, is not a feasible solution. Moreover, through extensive experimental evaluation, ZeroMark is shown to perform more effectively for Blended watermark, compared with other watermark techniques. Such observations inspire us to improve the effectiveness of ZeroMark by designing the potential watermark patterns in a blended manner. Finally, the boundary gradient analysis step in ZeroMark may incur variance among different labels' samples. In practice, we can release such variance by training a surrogate model with the protected dataset and calculating cosine similarity between the corresponding boundary gradients and the target watermark pattern with samples from different labels. Then we use the calculated cosine similarity using samples from different labels under the surrogate model to adjust the cosine similarity calculated under the suspicious model. Specifically, we can adjust the cosine similarity calculated with samples from a specific label $t$ by subtracting it from the average cosine similarity calculated with samples from the same label $t$ under the surrogate model. We will further discuss these issues in our future work.

Thirdly, we can only empirically verify that malicious dataset users cannot recover dataset-specified watermarks based on our boundary samples. We will try to prove it theoretically in the future.

Fourthly, we are currently focusing on convolutional neural networks ($e.g.$, ResNet and VGG) and the continuous image modality. In general, the success of our approach on other model structures depends on two factors: **(1)** whether the studied dataset watermarking method ($e.g.$, BadNets) can successfully watermark these models and **(2)** whether we can conduct effective 'adversarial attacks' to find boundary samples on these models. Based on existing work related to backdoor attacks/dataset watermarking [42, 9] and adversarial attacks [43, 44], these factors are all met. As such, our method can fundamentally generalize to other models ($e.g.$, transformer) as well. As for the generalizability to other (discrete) data formats like tabular or text, the main challenge lies in how to design effective adversarial attacks to them for finding the closest boundary samples (as in Eq.(10)). In particular, there are already some relevant works [45, 46] confirming its feasibility. Accordingly, our ZeroMark can be naturally adapted to other discrete data formats ($e.g.$, text and scientific data [47, 48, 49, 50, 51]). We will discuss them in our future work.

## 7 Conclusion

In this paper, we revisited existing DOV methods and revealed that their underlying assumption regarding the verification phase does not necessarily hold in practice. Accordingly, directly using dataset-specified watermarks for verification is insecure. Motivated by these findings, we proposed *ZeroMark* to conduct ownership verification without disclosing them. Our method was inspired by our empirical and theoretical findings of the intrinsic property of DNNs trained on the watermarked dataset. We conducted experiments on benchmark datasets, verifying the effectiveness of our ZeroMark and its resistance to potential adaptive methods. We hope our work can provide a new angle of dataset ownership verification to facilitate more secure and trustworthy dataset sharing.

## Acknowledgment

This work was partially supported by NSF IIS 2347592, 2347604, 2348159, 2348169, DBI 2405416, CCF 2348306, and CNS 2347617.

## References

[1] Zhifeng Li, Dihong Gong, Yu Qiao, and Dacheng Tao. Common feature discriminant analysis for matching infrared face images to optical face images. *IEEE transactions on image processing*, 23(6):2436–2445, 2014.

[2] Haibo Qiu, Dihong Gong, Zhifeng Li, Wei Liu, and Dacheng Tao. End2end occluded face recognition by masking corrupted features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6939–6952, 2021.

[3] Samuel Dooley, Rhea Sukthanker, John Dickerson, Colin White, Frank Hutter, and Micah Goldblum. Rethinking bias mitigation: Fairer architectures make for fairer face recognition. In *NeurIPS*, 2024.

[4] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[6] Yiming Li, Mingyan Zhu, Xue Yang, Yong Jiang, Tao Wei, and Shu-Tao Xia. Black-box dataset ownership verification via backdoor watermarking. *IEEE Transactions on Information Forensics and Security*, 2023.

[7] Yiming Li, Yang Bai, Yong Jiang, Yong Yang, Shu-Tao Xia, and Bo Li. Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection. In *NeurIPS*, 2022.

[8] Ruixiang Tang, Qizhang Feng, Ninghao Liu, Fan Yang, and Xia Hu. Did you train on my dataset? towards public dataset protection with clean-label backdoor watermarking. *ACM SIGKDD Explorations Newsletter*, 2023.

[9] Junfeng Guo, Yiming Li, Lixu Wang, Shu-Tao Xia, Heng Huang, Cong Liu, and Bo Li. Domain watermark: Effective and harmless dataset copyright protection is closed at hand. In *NeurIPS*, 2023.

[10] Cheng Wei, Yang Wang, Kuofeng Gao, Shuo Shao, Yiming Li, Zhibo Wang, and Zhan Qin. Pointncbw: Towards dataset ownership verification for point clouds via negative clean-label backdoor watermark. *IEEE Transactions on Information Forensics and Security*, 2024.

[11] Boheng Li, Yanhao Wei, Yankai Fu, Zhenting Wang, Yiming Li, Jie Zhang, Run Wang, and Tianwei Zhang. Towards reliable verification of unauthorized data usage in personalized text-to-image diffusion models. In *IEEE S&P*, 2025.

[12] Ronald Rivest. The md5 message-digest algorithm. Technical report, 1992.

[13] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, 2001.

[14] Paulo Martins, Leonel Sousa, and Artur Mariano. A survey on fully homomorphic encryption: An engineering perspective. *ACM Computing Surveys*, 2017.

[15] Chiou-Ting Hsu and Ja-Ling Wu. Hidden digital watermarks in images. *IEEE Transactions on image processing*, 1999.

[16] Ming-Shing Hsieh, Din-Chang Tseng, and Yong-Huai Huang. Hiding digital watermarks using multiresolution wavelet transform. *IEEE Transactions on industrial electronics*, 2001.

[17] Yuanfang Guo, Oscar C Au, Rui Wang, Lu Fang, and Xiaochun Cao. Halftone image watermarking by content aware double-sided embedding error diffusion. *IEEE Transactions on Image Processing*, 2018.

[18] Zuobin Xiong, Zhipeng Cai, Qilong Han, Arwa Alrawais, and Wei Li. Adgan: Protect your location privacy in camera data of auto-driving vehicles. *IEEE Transactions on Industrial Informatics*, 17(9):6200–6210, 2020.

[19] Yiming Li, Peidong Liu, Yong Jiang, and Shu-Tao Xia. Visual privacy protection via mapping distortion. In *ICASSP*, 2021.

[20] Honghui Xu, Zhipeng Cai, Daniel Takabi, and Wei Li. Audio-visual autoencoding for privacy-preserving video streaming. *IEEE Internet of Things Journal*, 2021.

 https://doi.org/10.52202/079017-3829

[21] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *NeurIPS*, 2019.

[22] Linghui Zhu, Xinyi Liu, Yiming Li, Xue Yang, Shu-Tao Xia, and Rongxing Lu. A fine-grained differentially private federated learning against leakage from gradients. *IEEE Internet of Things Journal*, 2021.

[23] Jiawang Bai, Yiming Li, Jiawei Li, Xue Yang, Yong Jiang, and Shu-Tao Xia. Multinomial random forest. *Pattern Recognition*, 2022.

[24] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *ICML*, 2016.

[25] Jian Liu, Mika Juuti, Yao Lu, and Nadarajah Asokan. Oblivious neural network predictions via minionn transformations. In *CCS*, 2017.

[26] Deevashwer Rathee, Mayank Rathee, Nishant Kumar, Nishanth Chandran, Divya Gupta, Aseem Rastogi, and Rahul Sharma. Cryptflow2: Practical 2-party secure inference. In *CCS*, 2020.

[27] Xiaoqiang Sun, Peng Zhang, Joseph K Liu, Jianping Yu, and Weixin Xie. Private machine learning classification based on fully homomorphic encryption. *IEEE Transactions on Emerging Topics in Computing*, 8(2):352–364, 2018.

[28] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *CCS*, 2017.

[29] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 2019.

[30] Yuancheng Xu, Yanchao Sun, Micah Goldblum, Tom Goldstein, and Furong Huang. Exploring and exploiting decision boundary dynamics for adversarial robustness. In *ICLR*, 2023.

[31] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*, 2020.

[32] Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. SCALE-UP: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. In *ICLR*, 2023.

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

[34] Junfeng Guo, Ang Li, and Cong Liu. Aeva: Black-box backdoor detection using adversarial extreme value analysis. In *ICLR*, 2022.

[35] Leopold Schmetterer. *Introduction to mathematical statistics*, volume 202. Springer Science & Business Media, 2012.

[36] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015.

[37] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[38] Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. In *ICLR*, 2021.

[39] Yuntao Liu, Yang Xie, and Ankur Srivastava. Neural trojans. In *ICCD*, 2017.

[40] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *NeurIPS*, 2021.

[41] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008.

[42] Zenghui Yuan, Pan Zhou, Kai Zou, and Yu Cheng. You are catching my attention: Are vision transformers bad learners under backdoor attacks? In *CVPR*, 2023.

[43] Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. On the robustness of vision transformers to adversarial examples. In *ICCV*, 2021.

[44] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of vision transformers. *arXiv preprint arXiv:2103.15670*, 2021.

[45] Deokjae Lee, Seungyong Moon, Junhyeok Lee, and Hyun Oh Song. Query-efficient and scalable black-box adversarial attacks on discrete sequential data via bayesian optimization. In *ICML*, 2022.

[46] Hao Peng, Shixin Guo, Dandan Zhao, Xuhong Zhang, Jianmin Han, Shouling Ji, Xing Yang, and Ming Zhong. Textcheater: A query-efficient textual adversarial attack in the hard-label setting. *IEEE Transactions on Dependable and Secure Computing*, 2023.

[47] Yihan Wu, Ruibo Chen, Zhengmian Hu, Yanshuo Chen, Junfeng Guo, Hongyang Zhang, and Heng Huang. Distortion-free watermarks are not truly distortion-free under watermark key collisions, 2024.

[48] Ruibo Chen, Yihan Wu, Lichang Chen, Guodong Liu, Qi He, Tianyi Xiong, Chenxi Liu, Junfeng Guo, and Heng Huang. Your vision-language model itself is a strong filter: Towards high-quality instruction tuning with data selection. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics ACL 2024*, pages 4156–4172, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics.

[49] Yanshuo Chen, Zhengmian Hu, Yihan Wu, Ruibo Chen, Yongrui Jin, Wei Chen, and Heng Huang. Enhancing biosecurity with watermarked protein design. *bioRxiv*, 2024.

[50] Ruibo Chen, Yihan Wu, Yanshuo Chen, Chenxi Liu, Junfeng Guo, and Heng Huang. A watermark for order-agnostic language models, 2024.

[51] Hanqing Guo, Xun Chen, Junfeng Guo, Li Xiao, and Qiben Yan. Masterkey: Practical backdoor attack against speaker verification systems. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, ACM MobiCom '23, New York, NY, USA, 2023. Association for Computing Machinery.

[52] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *IEEE S&P*, 2020.

[53] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *ICCV*, 2021.

[54] Yiming Li, Mengxi Ya, Yang Bai, Yong Jiang, and Shu-Tao Xia. BackdoorBox: A python toolbox for backdoor learning. In *ICLR Workshop*, 2023.

[55] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[56] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE S&P*, 2019.

[57] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2014.

 https://doi.org/10.52202/079017-3829

## Appendix

# Table of Contents

## A Proof for Theorem 1

We follow the previous work [32] to use a model $f(\cdot;\boldsymbol{w})$ infected with the basic backdoor-based watermark ($i.e.$, $G_x = (1-m)\odot\boldsymbol{\Delta}+m\odot\boldsymbol{x}$) [29] as a basic example to shed light on the intriguing characteristic of watermark model.

**Theorem 1** (Property of Boundary Gradient on the Closest Boundary Sample)**.** *Assume that $\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})$ is twice differentiable with a Lipschitz gradient, if $|\mathcal{D}_m|\to\infty$ and by updating $\overline{\boldsymbol{x}}_t$ in Eq. (4) with step size $\beta_t = ||\overline{\boldsymbol{x}}_t-\boldsymbol{x}_0||_2\cdot t^{q-1}$, there exists a constant $c\geq 0$ such that*

$$\lim_{|\mathcal{D}_m|\to\infty} 1-cos\angle(\boldsymbol{\delta},\nabla_{\boldsymbol{x}}\phi_{y_t}(\overline{\boldsymbol{x}}^*,\boldsymbol{w})) \leq c\cdot(t^*)^{q-1} \tag{12}$$

*where $q\in(\frac{1}{2},1)$, $y_t$ is the target label ($i.e.$, $y_t = C(G_x(\boldsymbol{x}))$), $\boldsymbol{\delta}$ is the watermark pattern ($i.e.$, $\boldsymbol{\delta}\triangleq G_x(\boldsymbol{x}_0)-\boldsymbol{x}_0$), and $t^*$ is the number of convergence iterations of $\overline{\boldsymbol{x}}^*$'s update.*

*Proof.*

Recall in Eq. (4), we update $\overline{\boldsymbol{x}}_t$ for each $t-1$ th iteration as:

$$\overline{\boldsymbol{x}}_{t+1} = \alpha_t\cdot\boldsymbol{x}_0+(1-\alpha_t)\cdot\left\{\overline{\boldsymbol{x}}_t+\beta_t\frac{\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})}{||\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})||}\right\}, \tag{13}$$

Let the step size $\beta_t$ in Eq. (4) as $t^{-q}||\boldsymbol{x_t}-\boldsymbol{x_0}||$, we have the distance ratio for updating Eq. (4) as:

$$\frac{||\overline{\boldsymbol{x}}_{t+1}-\boldsymbol{x_0}||^2}{||\overline{\boldsymbol{x}}_t-\boldsymbol{x_0}||^2} = \frac{||(1-\alpha)(\frac{t^{-q}||\overline{\boldsymbol{x}}_t-\boldsymbol{x_0}||\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})}{||\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})||}+\overline{\boldsymbol{x}}_t-\boldsymbol{x_0})||_2^2}{||\overline{\boldsymbol{x}}_t-\boldsymbol{x_0}||_2^2} \tag{14}$$

With a second-order tylor expansion, we have:

$$\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w}) = \ <\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w}),\overline{\boldsymbol{x}}_{t+1}-\overline{\boldsymbol{x}}_t>+\frac{1}{2}(\overline{\boldsymbol{x}}_{t+1}-\overline{\boldsymbol{x}}_t)^T H_t(\overline{\boldsymbol{x}}_{t+1}-\overline{\boldsymbol{x}}_t)=0 \tag{15}$$

Combining these Eq. (4) and Eq. (15), we have:

$$<\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w}),-\alpha v_t+\tau_t\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})>+ \tag{16}$$

$$\frac{1}{2}(-\alpha v_t+\tau_t\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w}))^T H_t(-\alpha v_t+\tau_t\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w}))=0, \tag{17}$$

where we define $v_t$ as $\overline{\boldsymbol{x}}_t-\boldsymbol{x_0}+\tau_t\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})$ and $\tau_t$ as $t^{-q}\frac{||\overline{\boldsymbol{x}}_t-\boldsymbol{x_0}||}{||\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})||}$.

Solving for $\alpha$, we have:

$$\alpha\geq\frac{\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})^T(\tau_t^2H_t+2\tau_tI)\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})}{2\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})^T(I+\tau_tH_t)v_t}. \tag{18}$$

Therefore, we can get:

$$(1-\alpha)^2\leq(\frac{r_t+\frac{3}{2}t^{-q}L\frac{||d_t||_2}{||\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})||_2}}{r_t+t^{-q}(1+\frac{3}{2}L\frac{||d_t||_2}{||\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})||_2})}), \tag{19}$$

where $d_t = \overline{\boldsymbol{x}}_t-\boldsymbol{x_0}$ and $r_t := cos\angle(\overline{\boldsymbol{x}}_t-\boldsymbol{x_0},\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})) = \frac{<\overline{\boldsymbol{x}}_t-\boldsymbol{x_0},\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})>}{||\overline{\boldsymbol{x}}_t-\boldsymbol{x_0}||_2||\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})||_2} = \frac{<d_t,\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})>}{||d_t||_2||\nabla_{\boldsymbol{x}}\phi_y(\overline{\boldsymbol{x}}_t;\boldsymbol{w})||_2}$.

Let $k_t := \frac{3}{2}L\frac{||d_t||_2}{||\nabla_t||_2}$. Then $k_t$ can be bounded when $||\nabla_t||_2\geq C$ and $q\geq\frac{1}{2}$, thus we have:

$$\frac{||x_{t+1}-x^*||_2^2}{||x_t-x^*||_2^2}\leq(\frac{r_t+\beta_tk_t}{r_t+\beta_t(1+k_t)})^2\cdot(\beta_t^2+2\beta_tr_t+1) \tag{20}$$


https://doi.org/10.52202/079017-3829

Motivated by previous work [52], solve Eq. (20) and have :

$$\sum_{t=1}^{\infty} c_1 t^{-q} \frac{1-r_t^2}{r_t} - c_2 t^{-2q} \leq \infty, \tag{21}$$

where $c_1, c_2$ are two positive constants, thus the above equation is $o(t-1)$.

When $q \in (\frac{1}{2}, 1)$, we have:

$$\frac{1-r_t^2}{r_t} = o(t^{q-1}). \tag{22}$$

Therefore, we have:

$$1 - cos\angle(d_t, \nabla_{\boldsymbol{x}} \phi_y(\overline{\boldsymbol{x}}_t; \boldsymbol{w})) \leq c \cdot t^{q-1}, \tag{23}$$

Notably, Eq. (4) with step size as $t^{-q}||\overline{\boldsymbol{x}}_t - \boldsymbol{x_0}||$ converges a stationary point of Eq. (3). Motivated by proof for **Lemma 3** in [34], when $\overline{\boldsymbol{x}}_t$ is optimized to a stationary point ($i.e.$, $\overline{\boldsymbol{x}}^*$) in $t^*$ and if $\overline{\boldsymbol{x}}_t$ belongs to the watermark label $y_t$, we have:

$$\lim_{|\mathcal{D}|_m \to \infty} \mathbb{E}\left[m \odot \overline{\boldsymbol{x}}^* - m \odot \boldsymbol{x_0}\right] = \mathbb{E}\left[d_t\right] = m \odot \boldsymbol{\Delta} - m \odot \boldsymbol{x_0} \tag{24}$$

$$= m \odot \boldsymbol{\delta}, \tag{25}$$

and

$$\lim_{|\mathcal{D}|_m \to \infty} \frac{||(1-m) \odot (\overline{\boldsymbol{x}}^* - \boldsymbol{x_0})||}{||\overline{\boldsymbol{x}}^* - \boldsymbol{x_0}||} = 0. \tag{26}$$

Hence, when $|\mathcal{D}|_m \to \infty$, we have:

$$\frac{< \boldsymbol{m} \odot d_t, \boldsymbol{m} \odot \nabla_{\boldsymbol{x}} \phi_y(\overline{\boldsymbol{x}}^*; \boldsymbol{w}) >}{||\boldsymbol{m} \odot d_t|| ||\boldsymbol{m} \odot \nabla_{\boldsymbol{x}} \phi_y(\overline{\boldsymbol{x}}^*; \boldsymbol{w})||} = \frac{< \boldsymbol{m} \odot \boldsymbol{\delta}, \boldsymbol{m} \odot \nabla_{\boldsymbol{x}} \phi_y(\overline{\boldsymbol{x}}^*; \boldsymbol{w}) >}{||\boldsymbol{m} \odot \boldsymbol{\delta}|| ||\boldsymbol{m} \odot \nabla_{\boldsymbol{x}} \phi_y(\overline{\boldsymbol{x}}^*; \boldsymbol{w})||}, \tag{27}$$

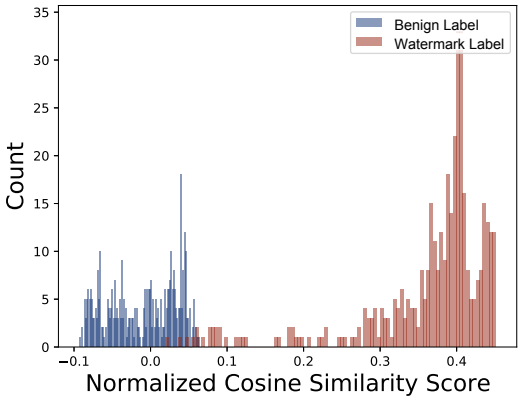
therefore, for the watermark label ($i.e.$, $y_t$):

$$\begin{aligned} \lim_{|\mathcal{D}_m| \to \infty} 1 - \cos\angle(\boldsymbol{m} \odot \boldsymbol{\delta}, \boldsymbol{m} \odot \nabla_x \phi_{y_t}(\overline{\boldsymbol{x}}^*, \boldsymbol{w})) \; &= 1 - cos\angle(\boldsymbol{m} \odot d_t, \boldsymbol{m} \odot \nabla_{\boldsymbol{x}} \phi_y(\overline{\boldsymbol{x}}^*; \boldsymbol{w})) \\ &\leq c \cdot (t^*)^{q-1}. \end{aligned} \tag{28}$$

## B Detailed Settings for Empirical Study in Section 3.3

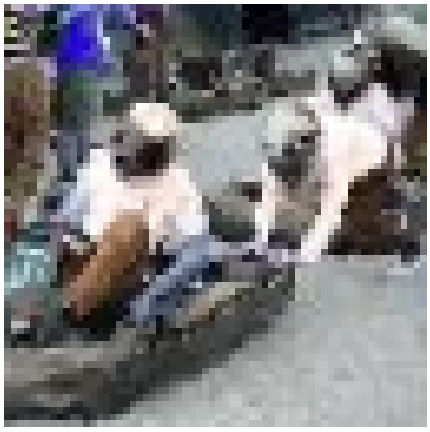
In the Section. 3.3, we select ResNet-18 and CIFAR-10 as the evaluated model and benchmark. We select the class '0' as the watermark class and inject watermark samples with 10% watermark ratio to ensure the verification success rate $\geq 99\%$. We randomly select 300 samples from the validation set across classes. We use these selected validation samples to generate boundary points for watermark and benign labels labels following Eq. (4). In particular, for the boundary point of watermark label $y_t$, we set $\boldsymbol{x_0}$ as samples from classes different from $y_t$ and set $\boldsymbol{x_t}$ as samples from the watermark label. As for the boundary point of benign labels, we set $\boldsymbol{x_0}$ as samples from the watermark class and set $\boldsymbol{x_t}$ as samples from the benign labels. As such , there should be 400 boundary gradients for the watermark or benign label. We then calculate the boundary gradients following the gradient estimation process as Eq. (10).

(a) Watermark

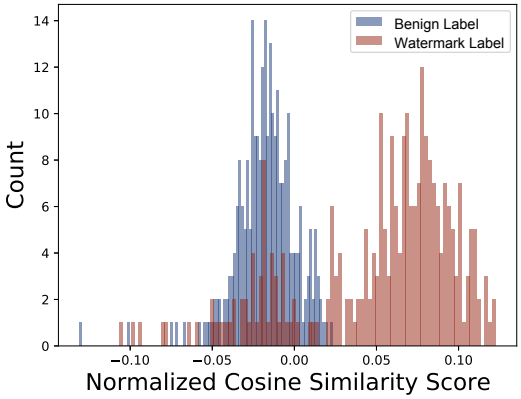



(b) Distribution of $cos\angle(\cdot)$

Figure 10: The results of using blended watermark.



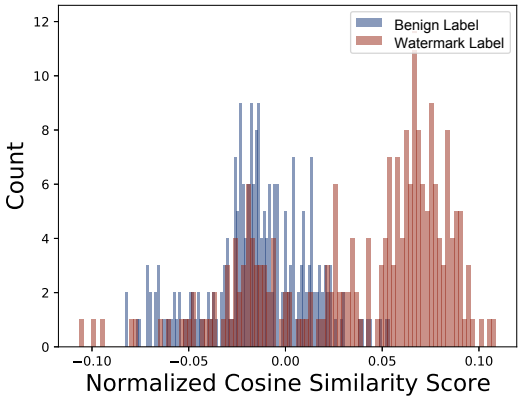
(a) Watermark




(b) Distribution of $cos\angle(\cdot)$

Figure 11: The results of using WaNet watermark.

(a) Watermark




(b) Distribution of $cos\angle(\cdot)$

Figure 12: The results of using domain watermark.


https://doi.org/10.52202/079017-3829

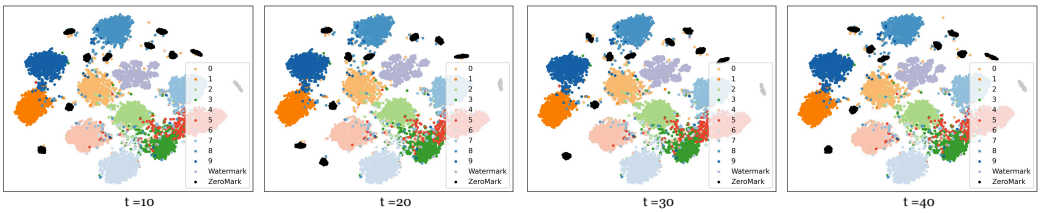



Figure 13: t-SNE clustering results for the blended watermark.

## C Additional Results for Empirical Study in Section 3.3

We also conduct same empirical studies on additional three different types of watermarks ($i.e.$, Blended [37],WaNet [38] and Domain Watermark [9]) to validate the characteristic of boundary gradients for watermarked and benign labels. Specifically, we conduct empirical studies with CIFAR-10 using ResNet-18 and select '0' as the watermark label. We can see that in all these three watermark patterns, the distributions of cosine similarity for watermarked labels have significantly large ranges compared with that of benign labels. Interestingly enough, we find that the distribution of cosine similarity for Blended watermark has a more obvious separation with that of benign labels compared with other watermark patterns. We will speculate the reason for this phenomenon in our future work.

## D The Detailed Process for Mitigating the Variance

We here describe how to mitigate the variance caused by the watermark patterns and the iterative gradient estimation in details.

**Mitigate the Variance Caused by the Watermark Patterns.** Based on the available ground truth watermark pattern $\boldsymbol{\delta}$, we create several ($i.e.$, 6) artifact watermark patterns $\{\boldsymbol{\delta_i}\}_{i=1}^{10}$ which have the same location map ($i.e.\boldsymbol{m}$) as the ground truth watermark but filled with different random noise. We calculate the baseline $\overline{cos}$ as:

$$\overline{cos} := \frac{1}{N}\sum_{i=1}^{N} cos\angle(m \odot \boldsymbol{\delta_i}, m \odot \nabla_x \phi_{y_t}(\overline{\boldsymbol{x}}^*, \boldsymbol{w})), \tag{29}$$

and we calculate the normalized cosine similarity score as:

$$\widehat{cos}\angle(m \odot \boldsymbol{\delta}, m \odot \nabla_x \phi_{y_t}(\overline{\boldsymbol{x}}^*, \boldsymbol{w})) := cos\angle(m \odot \boldsymbol{\delta}, m \odot \nabla_x \phi_{y_t}(\overline{\boldsymbol{x}}^*, \boldsymbol{w})) - \overline{cos}. \tag{30}$$

**Mitigatwe the Variance in the Iterative Gradient Estimation Procedure.** During the procedure of gradient estimation for Eq. (10), the estimated gradients could yield variance among iterations for the gradient estimation process. Therefore, to mitigate such variance, we propose to average the estimated gradients over iterations for the gradient estimation process, which can be formulated as:

$$\nabla_x \phi_{y_t}(\overline{\boldsymbol{x}}^*, \boldsymbol{w})) := \frac{1}{t}\sum_{t=0}^{t} \nabla_x \phi_{y_t}(\overline{\boldsymbol{x}}_t, \boldsymbol{w})) \tag{31}$$

## E Detailed description for Evaluation Metrics

We here describe the metrics for evaluating each approach in details. With loss of generality, we here define each give watermark sample as:

$$\boldsymbol{x_i'} = \boldsymbol{x_i} + \boldsymbol{t_i}, \tag{32}$$

where $\boldsymbol{x_i}$ and $\boldsymbol{t_i}$ represent the benign sample and the corresponding watermark pattern. We suppose $\boldsymbol{t_i}$ is located at the dims of $[j : k](j < k)$. The evaluation metrics, including mean square error (MSE), neuron activation similarity (NAS) and mutual information (MI) are defined as below:

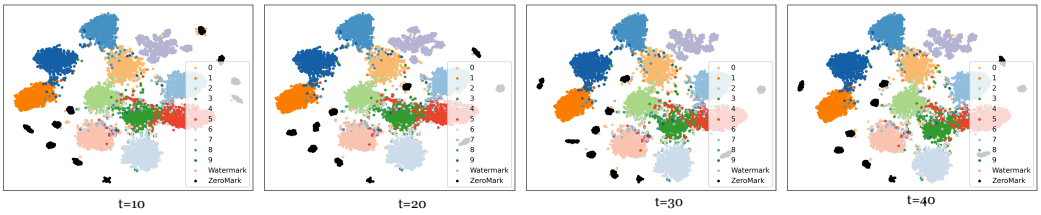



Figure 14: t-SNE clustering results for the WaNet watermark.

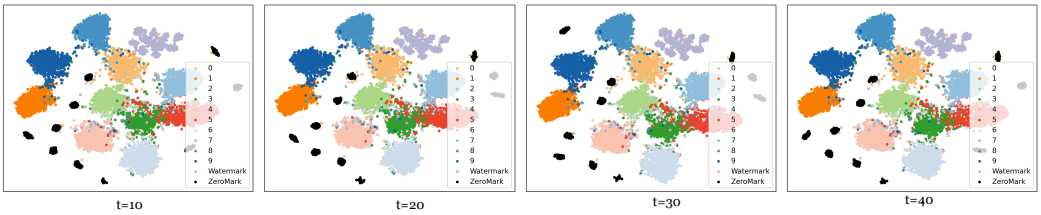



Figure 15: t-SNE clustering results for the domain watermark.

**Algorithm 1** The main process of our ZeroMark.

**Input:** Validation samples $\{\boldsymbol{x_i^t}, y_t\}_{i=1}^m$ from the watermark label $y_t$; Validation samples $\{\boldsymbol{x_i^o}, y_i^o\}_{i=1}^m$ from the labels other than the watermark label; The suspicious model $f(\cdot; \boldsymbol{w})$
**for** l=1,2,... **do**
    Generate (closest) boundary points and boundary gradients for the watermark label following Eq. (4): Set $\{\boldsymbol{x_i^t}, y_t\}_{i=1}^m$ as $\boldsymbol{x_y}$ and set $\{\boldsymbol{x_i^o}, y_i^o\}_{i=1}^m$ as $\boldsymbol{x_0}$.
    Generate (closest) boundary points and boundary gradients for the benign label following Eq. (4): Set $\{\boldsymbol{x_i^t}, y_t\}_{i=1}^m$ as $\boldsymbol{x_0}$ and set $\{\boldsymbol{x_i^o}, y_i^o\}_{i=1}^m$ as $\boldsymbol{x_y}$.
**end for**
Calculate the corresponding cosine similarities for watermark and benign labels following Eq. (8).
Select largest $m*Q\%$ cosine similarities for watermark and benign labels for T-test following Proposition 1.

- MSE: = $\frac{1}{N}\sum_{i=1}^{N}\sqrt{(\boldsymbol{x_i'}[j:k]-\boldsymbol{x_v}[j:k])^2}$.
- NAS: = $\frac{1}{N}\sum_{i=1}^{N}\cos\angle(F(\boldsymbol{x_i'}), F(x_v)))$.
- MI:= $\mathbb{E}_{p(\boldsymbol{z_v},\boldsymbol{z'})}[\log p(\boldsymbol{z'}|\boldsymbol{z_v})] - \mathbb{E}_{p(\boldsymbol{z_v})p(\boldsymbol{z'})}[\log p(\boldsymbol{z'}|\boldsymbol{z_v})]$.

where $\boldsymbol{x_v}$, $F(\cdot)$ represent the verification samples, feature extractor for the corresponding watermark model. $\boldsymbol{z}$ represents the feature extracted by the watermark model. Since ZeroMark uses several perturbed boundary points for calculating the boundary gradient for each given sample, we average their values for computing each metric. Notably, we follow previous work [53] to estimate MI by calculating its upper bound, as follows:

$$I(\boldsymbol{z};\hat{\boldsymbol{z}}) = \mathbb{E}_{p(\boldsymbol{z},\hat{\boldsymbol{z}})}\left[\log\frac{p(\hat{\boldsymbol{z}}|\boldsymbol{z})}{p(\hat{\boldsymbol{z}})}\right] \leq \mathbb{E}_{p(\boldsymbol{z},\hat{\boldsymbol{z}})}[\log p(\hat{\boldsymbol{z}}|\boldsymbol{z})] - \mathbb{E}_{p(\boldsymbol{z})p(\hat{\boldsymbol{z}})}[\log p(\hat{\boldsymbol{z}}|\boldsymbol{z})]. \tag{33}$$

## F The Detailed Algorithm for ZeroMark

We put the detailed algorithm for ZeroMark as follows:


https://doi.org/10.52202/079017-3829

## G Experiments Details

### G.1 Datasets

We evaluate our approach on two widely-adopted benchmark datasets (i.e., CIFAR-10 [4], Tiny-ImgaeNet [36]). We here describe each benchmark dataset in detail.

**CIFAR-10.** CIFAR-10 dataset contains 10 labels, 50,000 training samples, and 10,000 validation samples. The training and validation samples are distributed evenly across each label. Each sample is resized as $32 \times 32$ by default.

**Tiny-ImageNet.** Tiny-ImageNet dataset contains 200 labels, 100,000 training samples, and 10,000 validation samples. The training and validation samples are distributed evenly across each label. Each sample is resized as $64 \times 64$ by default.

**Evaluated Watermarks.** In our experiments, we evaluate four types of watermark, including BadNets [29], Blended [37], WaNet [38], and domain watermark [9]. The visual demonstration for each watermark is shown in Sec. 4. For BadNets, we implement a $4 \times 4$ and $8 \times 8$ triggers for CIFAR-10 and Tiny-ImageNet. The trigger is filled with random noise. For Blended watermark, we implement a $4 \times 4$ and $8 \times 8$ triggers for CIFAR-10 and Tiny-ImageNet. We set the transparency ratio as 0.2 throughout experiments. As for WaNet, we use `BackdoorBox`[2] [54] to build the watermarked model with its default configurations. For Domain Watermark, we implement it following its released code[3]. We set the watermark rate $\gamma$ as 0.1 consistent with previous work [7] for training different watermark models.

### G.2 Training Configurations

To train DNN models, we use Adam optimizer [55] with the initial learning rate as 0.01. The watermark models evaluated in our experiments can achieve $\geq 92.26\%$ and $56.8\%$ accuracy on validation dataset for CIFAR-10 and Tiny-ImageNet tasks. We use six NVIDIA RTX 2080 Ti GPUs for performing experiments.

## H Detailed Configurations for Comparison Approaches

We here describe the comparison approaches in details, as follows.

**Existing DOV.** We follow DOV approaches [29, 37, 38, 9] to directly exploit the watermark samples for verification.

**Minimal Watermark.** Inspired by previous work [56] for Trojan detection, which applies reverse engineering to generate the pseudo trigger patterns with minimize size while preserving their attack efficacy. Specifically, we generate the minimal watermark for the watermark sample $\boldsymbol{x'} = \boldsymbol{x} + \boldsymbol{\delta}$ following:

$$\min_{\boldsymbol{m_\delta}} \ell(f(\boldsymbol{m_\delta} \odot \boldsymbol{\delta} + (1 - \boldsymbol{m_\delta}) \odot \boldsymbol{x}; \boldsymbol{w}), y_t) + ||\boldsymbol{m_\delta}||_2, \tag{34}$$

we follow Neural Cleanse [56]'s configurations for conducting optimization on Eq. (34). For input-specific watermark patterns ($e.g.$, WaNet [38], DW [9]), we conduct Eq. (34) on each watermark sample. As such verification samples with the minimal watermark can be formulated as:

$$x' = \boldsymbol{m_\delta} \odot \boldsymbol{\delta} + (1 - \boldsymbol{m_\delta}) \odot \boldsymbol{x}. \tag{35}$$

[2] https://github.com/THUYimingLi/BackdoorBox.git

[3] https://proceedings.neurips.cc/paper_files/paper/2023/hash/aa6287ca31ae1474ea802342d0c8ba63-Abstract-Conference.html

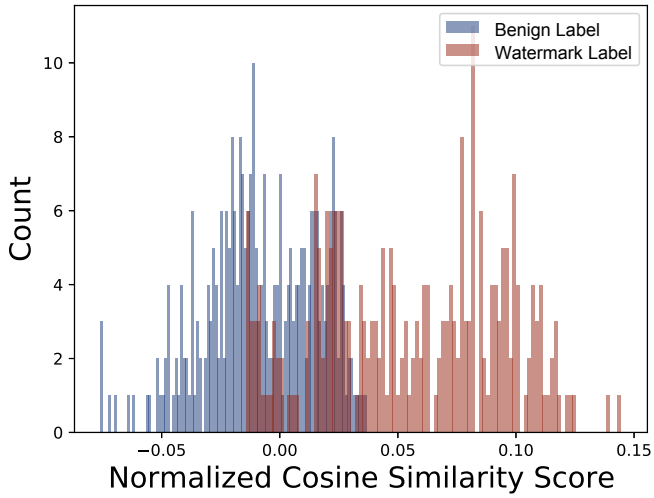



Figure 16: The distribution of consine similarity of boundary gradients for watermark and benign labels under VGG-19.

**Watermark with Distortion.** Motivated by previous work [32], which reveals that watermark samples can perform resilient against certain magnitude of random noise distortion, we thus propose to add maximum magnitudes of random noise for watermark samples to hinder the watermark pattern while preserving the watermark's efficacy. We generate the noisy watermark samples via:

$$\boldsymbol{x'} = Proj(\boldsymbol{x'} + a * \boldsymbol{\mu}), \tag{36}$$

where $\boldsymbol{\mu} \sim N(0, 1)$ is Gaussian noise and $Proj$ is the projection function to constrain $\boldsymbol{x'} + a * \boldsymbol{\mu}$ into $[0, 1]$. We solve $a$ using grid search to find the largest $a$ while preserving the verification success rate for watermark samples.

# I Additional Results for Experiments

We here show additional results in our experiments. We perform t-SNE clustering analysis for other three types of watermark ($i.e.$, Blended, WaNet, Domain Watermark) with varied optimization iterations $t$. The results are shown in Figure 13, Figure 14 and Figure 15. The additional visual demonstrations for boundary samples within the verification procedure are shown in Figure 17,Figure 18, Figure 19 and Figure 20. We also evaluate ZeroMark with different model architectures. Specifically, we here evaluate ZeroMark using VGG-19 [57] with CIFAR-10 task and the configurations are consistent with Section. 3.3. The results are shown in Figure 16, which demonstrates that ZeroMark can still perform effective on VGG-19 models.

# J The Resistance to More Adaptive Attacks

We here investigate whether ZeroMark can perform robustness against the potential adaptive attack for recovering or unlearning the watermark pattern.

**Recovering the Watermark Pattern from the Boundary Samples.** We here explore the potential adaptive attack for recovering the watermark pattern. Since we leverage gradient estimation via aggregating the random noise for conducting boundary gradient analysis, we here consider an adaptive attack by aggregating corresponding boundary samples for each input sample $\boldsymbol{x_0}$ to recover the watermark pattern. We conduct experiments using ResNet-18 under CIFAR-10 task and we evaluate the adaptive attack with Domain Watermark [9]. The results are shown in Figure 21 and Figure 22.


https://doi.org/10.52202/079017-3829

Figure 17: Visual demonstration of ZeroMark for BadNets watermark.

We find that the aggregated boundary samples can not reveal the watermark pattern from both visual and clustering analysis.

**Unlearn the Watermark through Boundary Samples.** We here consider whether we can follow [56] to unlearn the watermark pattern via boundary samples. We retrain the suspicious model with 500 boundary samples and label them as their original label ($i.e.$, label for $\boldsymbol{x_0}$) along with the training data. We find that the accuracy of the suspicious model drops from 92.3% to 90.2% and the verification success rate for domain watermark drops from 88.6% to 75.1%, and the ZeroMark can still achieve the averaged largest Q% as 0.076 for the target label, which significantly larger than that of the benign labels($i.e.$,0.028). This results demonstrate that ZeroMark can prevent disclosing watermark patterns during the verification procedure within DOV.

## K Reproducibility Statement

In the appendix, we provide detailed descriptions of the datasets, models, training and evaluation settings, and computational facilities. We provide the codes and model checkpoints for reproducing the main experiments of our evaluation in the supplementary material.

## L Societal Impacts

In this paper, we focus on the copyright protection of public datasets. Specifically, we reveal that the verification process of existing DOV methods is not secured and propose using boundary samples to conduct verification without disclosing the watermark. This work has no general ethical issues since our method is purely defensive and does not reveal any new vulnerabilities of DNNs.

Figure 18: Visual demonstration of ZeroMark for Blended watermark.

## M Discussions about Adopted Data

In this paper, all adopted samples are from the open-sourced datasets ($i.e.$, CIFAR-10, Tiny-ImageNet). The Tiny-ImageNet dataset may contain a few human-related images. We admit that we modified a few samples for watermarking and verification. However, our research treats all samples the same and the verification samples and modified samples have no offensive content. Accordingly, our work fulfills the requirements of these datasets and has no privacy violation.


https://doi.org/10.52202/079017-3829

Figure 19: Visual demonstration of ZeroMark for WaNet watermark.

Figure 20: Visual demonstration of ZeroMark for domain watermark.

Figure 21: The visual demonstration of the aggregated boundary samples.


https://doi.org/10.52202/079017-3829

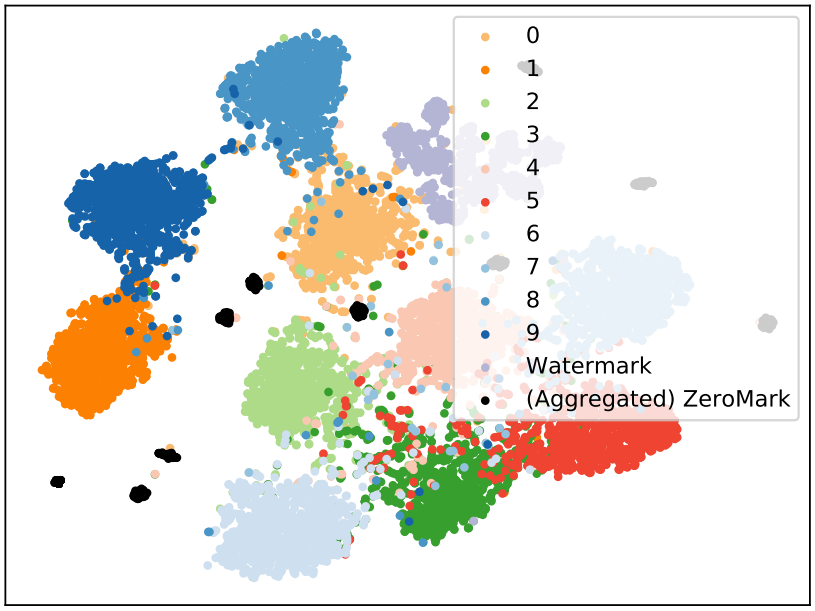



Figure 22: The t-SNE clustering results for aggregated boundary samples.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the claims made, including the contributions made in the paper and scopes.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]


https://doi.org/10.52202/079017-3829

We have claimed the limitations of our approach in the Appendix K

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We have included the assumptions along with our Theorem and included the complete proof in Appendix A

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have described the detailed configurations as well as the methodology in our paper to reproduce the claims and results. Moreover, we have included the code as well as the checkpoint for evaluated DNNs for reproducibility purposes.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: In this paper, we use publicly available dataset ($i.e.$, CIFAR-10 and Tiny-ImageNet), which can be easily accessed.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.


https://doi.org/10.52202/079017-3829

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have detailed the training configurations and optimizers in Appendix G

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We have reported the distribution of our results in Figure 2 and Appendix B

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We included the details for computer resources in the Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We included the social impact in the Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]


https://doi.org/10.52202/079017-3829

Justification: This paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: This paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.


https://doi.org/10.52202/079017-3829