Anti-Neuron Watermarking: Protecting Personal Data Against Unauthorized Neural Networks

Zihang Zou[1], Boqing Gong[2], and Liqiang Wang[1]

[1]University of Central Florida, [2]Google Research
{Zihang.Zou,Liqiang.Wang}@ucf.edu, bgong@google.com

Abstract. We study protecting a user's data (images in this work) against a learner's unauthorized use in training neural networks. It is especially challenging when the user's data is only a tiny percentage of the learner's complete training set. We revisit the traditional watermarking under modern deep learning settings to tackle the challenge. We show that when a user watermarks images using a specialized linear color transformation, a neural network classifier will be imprinted with the signature so that a third-party arbitrator can verify the potentially unauthorized usage of the user data by inferring the watermark signature from the neural network. We also discuss what watermarking properties and signature spaces make the arbitrator's verification convincing. To our best knowledge, this work is the first to protect an *individual* user's data ownership from unauthorized use in training neural networks.

1 Introduction

Recent advances in machine learning techniques have put personal data at significant risk. For example, in the scandal of "Cambridge Analytica" [38], millions of users' data are collected without consent to train machine learning models for political advertising. To protect personal data and privacy, there have been some legislations in place, such as Europe General Data Protection Regulation [11] (effective in May 2018), California Privacy Act [2] (effective in January 2021), and China Data Security Law [33] (effective in July 2021). They often require that personal data should be "processed lawfully, fairly and in a transparent manner" and can only be used "adequately, relevantly and limited to what is necessary in relation to the purposes ('data minimisation')" [11]. However, there is a lack of methods for detecting personal data breaches from machine learning models, which have increasingly become the primary motivation for a violator to break a user's data ownership because the models' efficacy heavily depends on data.

This paper studies personal image protection (PIP) from unauthorized usage in training deep neural networks (DNNs). The need for PIP arises when users expose their images to digital products and cloud services. In the era of big data and deep learning, a critical concern is that DNN learners may violate users' intents by using their data to train DNNs without authorization. It becomes worse when the DNN models consequently leak private user information [1, 10, 27, 31]. However, how can ordinary users know whether their images, which could

be a tiny portion of the DNN learner's complete training set, have been used to train a DNN model?

Traditionally, PIP aims to prevent a user's images from duplicating, remixing, or exploiting (e.g., for a financial incentive) without the user's consent and relies on digital watermarking [5, 19, 25, 35, 42, 46]. The digital watermarking enables a user to imprint images with unique patterns, such as signatures, logos, or stamps, to track and identify unauthorized *copies* of their pictures.

However, the rise of data-dependent deep learning poses another need for PIP, namely, protecting a user's images from unauthorized use in training DNNs. Could watermarking still fulfill this need?

One inspiring observation is that some DNNs do "memorize" certain training examples [1, 9, 10] in various ways, offering a user an opportunity to watermark their images to make them memorizable by the DNNs. We say this watermarking scheme is "anti-neuron" because its objective is to facilitate a third-party arbitrator to verify a DNN's use of a user's images in training and then hold the DNN learner accountable. However, we have to resolve two questions to make this anti-neuron watermarking work in practice. What watermarks make a user's images memorizable by DNNs? How can the third-party arbitrator verify that the user's images were indeed part of a DNN model's training set?

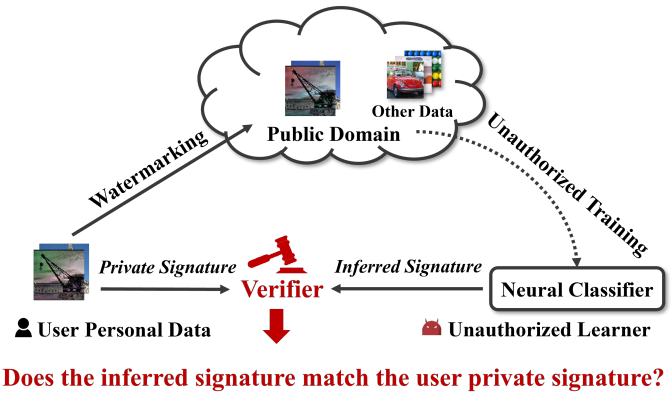


Fig. 1. Illustration of the anti-neuron watermarking for personal image protection (PIP) against unauthorized neural learners.

To answer the above questions, we first use Figure 1 to formalize the anti-neuron watermarking for PIP against unauthorized DNN learners. First, a user watermarks images using a private signature before sharing them with the public (e.g., social media). An unauthorized learner then collects the user's watermarked images, along with images from other sources, to construct a training set to train a DNN image classifier. Finally, the user turns to a third-party arbitrator to check whether their images were used to train the DNN model. The

arbitrator tries to recover the user's private signature for watermarking from the DNN model and the user's original images with no watermark — crucially, the arbitrator does not use the user's private signature to recover it. The arbitrator concludes that the user's images were part of the DNN's training set if the user's private signature can be recovered without knowing it in advance.

This paper proposes an empirically effective approach to the anti-neuron watermarking, and we leave more rigorous analyses to future work. In particular, a linear color transformation (LCT) in the hue space can be effectively used as a watermarking method. The resultant images remain as appealing as the original ones visually, so the unauthorized learner would not detect this type of watermark. Moreover, the LCT method is resilient to standard image augmentation techniques used in training neural models. Finally, we show that a DNN classifier indeed tends to memorize the LCT watermarking using extensive experiments. The arbitrator's verification method is simply iterating over the signature space, watermarking the user's original images using each signature, and returning the signature that reaches the lowest DNN classification loss.

In summary, our main contribution is to formalize the problem of user-focused anti-neuron watermarking for personal image protection from unauthorized usage in training DNNs. Moreover, we propose the LCT watermarking for ordinary users and a straightforward verification method for the third-party arbitrator, demonstrating a successful anti-neuron watermarking scenario for PIP. Additionally, we raise some critical questions for furthering the study of PIP against DNN learners: 1) What types of watermarking can imprint DNNs the best, especially when a user's watermarked images are only a tiny part of the training set? 2) What makes the imprinting of DNNs possible? Is it the DNN's memorization of training examples? 3) How can a trustworthy arbitrator recover a user's private watermark from DNN and the user's unwatermarked images? 4) How can anti-neuron watermarking work for multiple users? To the best of our knowledge, this work is the first to protect an individual user's data ownership from unauthorized use in training neural networks.

2 Related Work

Watermarking is a long-standing technique to declare ownership of objects. It can be traced back to paper marking [24] at 1282 in Italy, where a watermark was created via changing the thickness of the paper. Digital watermark is later introduced by [35] to code an undetectable digital watermark on gray scale images. Yu et al. [42] train a neural network for watermarking to embed hidden information in images. El et al. [8] add digital watermarks to video frames with neural networks. Zhong et al. [46] propose an automated and robust image watermarking based on deep neural networks. Recently, watermarking is used to protect the intellectual property of machine learning models [13,26,28,44]. These techniques follow similar ideas as trojan attacks [21] or backdoor attack [12], where models are trained with constructed samples to learn objective behaviors.

The most relevant works to this paper are *dataset tracing* and *membership inference*. **Dataset tracing** [20, 23, 29] protects the intellectual property of a dataset by appending traceable watermarks on data samples. Sablayrolles et al. [29] use pretrained model on a dataset itself to generate "radioactive data" to carry the class-specific watermarking vectors in high dimensional feature space. If a learner uses the "radioactive dataset" for training, the model's classifier would become more aligned with the watermarking vectors and thus can be used as an evidence of unauthorized usage. This kind of watermarking requires a pretrained model trained with the whole dataset. However, as such prior knowledge about the entire training data is unavailable for a common user, this kind of technique would be less applicable in real PIP scenarios.

Membership inference determines if a certain sample is inside a target dataset. Inference attack was first proposed for the attack and defense on medical datasets where users' medical records are extremely sensitive. By comparing genomic data with the statistical information of the training dataset, the presence of certain users can be inferred by attackers [15]. Shokri et al. [31] later introduce membership inference attack (MIA) into machine learning models. MIA trains a binary classifier to predict membership, on top of several shadow models being trained with the same data distribution as training. Alternatively, Yeom et al. [40] use the average of training error as the threshold to perform MIAs. Sablayrolles et al. [30] improve this threshold with Bayes optimal classifier to search for the best threshold using samples from both training and testing.

As MIAs determine whether given data samples belong to a training set or not, it is tempting to perform MIAs for personal data protection. However, similar to dataset tracing, as the training data distribution is unknown for common users, neither shadow models [31] nor threshold [30, 40] can be obtained and existing MIA methods fail to work in protecting personal data.

3 Problem Statement

Consider image classification as a case study without loss of generality. Denote by $\mathcal{D}_u$ the set of personal images owned by a common user u. Assume $\mathcal{D}_u$ is unique and distinguished among identifiable users, as defined in GDPR [11]. Suppose the user u plans to expose $\mathcal{D}_u$ online, *e.g.*, by sharing them on social media. For the purpose of avoiding potential breach of personal data proprietary, the user watermarks images with a secret signature k^* before sharing them on social media. Denote by $\mathcal{D}_u^*$ the set of watermarked images carrying the signature k^*.

An unauthorized learner may use the user's data $\mathcal{D}_u^*$, along with many others', to construct a training set $\mathcal{D}$ to train a DNN classifier f without acquiring the user's permission. It is reasonable to assume that the user's data $\mathcal{D}_u^*$ is only a small portion of the whole training set $\mathcal{D}$ and the user u does not have any prior knowledge about the other users' data.

Let $g \in \mathcal{G}$ denote a watermarking method and $\mathcal{V}$ be a neutral third-party verification method that infers the user's private signature for watermarking without knowing it before. The arbitrator determines whether the user's personal

images have been used in the training of neural classifier f as follows,

$$\mathcal{V}(f, \mathcal{D}_u, \mathcal{G}) = k^* \textbf{ iff } \mathcal{D}_u^* \subseteq \mathcal{D} \tag{1}$$

where the user's watermarked images $\mathcal{D}_u^* = g(D_u, k^*), g \in \mathcal{G}$. Namely, if the arbitrator can recover the user's private signature, she/he concludes that the user's images were part of the training set for learning the classifier f.

4 Approach

4.1 The Anti-Neuron Watermarking Method

Recent studies show that DNNs can "memorize" some training examples in various ways [1,9,10], and one can recover certain meaningful low-resolution images from DNNs [10]. Hence, it is tempting to conduct verification by recovering the user u's images from the neural classifier f. However, there are many challenges with this approach. First of all, the model f may memorize some training images but not this user u's. Moreover, even if the model happens to memorize some of this user's images, the recovery success rate is likely low. Existing methods (e.g., [10]) can recover semantically meaningful images from some DNNs, but they do not resemble any exact training images, to the best of our knowledge. Finally but not the least, the method in [10] incurs high computation cost, often by many iterations of gradient descent, and assumes that the DNN classifier f is a white box, disclosing its architecture and parameters.

An alternative approach to leveraging DNNs' memorization capability is to check a DNN's loss over a set of training images. Arguably, if the DNN model has memorized a majority of this set of images, the loss should be low. Following the above reasoning, we let a user u watermark her/his images $\mathcal{D}_u$ using a private signature k^* so the user has full control and knowledge of her/his watermarked images $\mathcal{D}_u^*$. This watermarking method eases the third-party arbitrator's job; instead of trying to recover the exact training images, the arbitrator can now search for the watermarking signature that leads to the lowest DNN loss, if the DNN model has memorized many images in $\mathcal{D}_u^*$ watermarked by user u.

Properties for Effective Watermarking. Formally, a user u chooses an anti-neuron watermarking function $g \in \mathcal{G}$ and generates the watermarked images as

$$\mathcal{D}_u^* = \{g(I, k^*), \forall I \in \mathcal{D}_u\} \tag{2}$$

We discuss the necessary properties needed to make a good anti-neuron watermarking function. The key is to make the watermarked images, and hence the signature, memorized by DNNs. First, the watermarking function g should *preserve an image's original content*. For example, for a user portrait or selfie, g should not change its identity. Besides, the watermarking function g should be *resilient to common image augmentations* used to train DNNs. The private signature should survive after the learner applies common image augmentations.

Furthermore, the space K of watermarking signatures should be *large* and preferably *bounded*, such that the probability of an innocent classifier coincidentally matching the user signature is low, while the signature can be inferred efficiently during verification.

Linear Color Transformation. Based on the discussion above, we propose Linear Color Transformation (LCT) as our anti-neuron watermarking. Color provides a large signature space for images. Our watermarking function exploits hue transformation and uses the hue adjustment of images as a signature. Thanks to the sufficiently big hue space, the user's randomly chosen signature is likely different from other users' signatures. Moreover, the randomly chosen signature lifts the user images to a low-density region, making the resultant images be easily memorized by DNNs — according to Feldman's studies on memorization [9] and our experiments in Section 5.6, DNNs tend to memorize images of low-density regions.

Concretely, we first convert the RGB color space into the YIQ color space [37] by the following matrix:

$$T_{\text{YIQ}} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \tag{3}$$

In the YIQ color space, hue is represented by two dimensional coordinates, forming a chromaticity diagram. As a result, watermarking images with signature k will be conducted by rotating the hue at an angle θ_k with the following matrix,

$$T_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta_k) & -sin(\theta_k) \\ 0 & sin(\theta_k) & cos(\theta_k) \end{bmatrix} \tag{4}$$

where $\theta_k = \frac{k\pi}{180}$. Hence, for every pixel $v = [v_r, v_g, v_b]^\intercal$ in image I, we can watermark v with signature k by:

$$v' = g_k v \tag{5}$$

where $g_k = T_{\text{YIQ}} \cdot T_k \cdot T_{\text{YIQ}}^{-1}$.

Making LCT more versatile. An immediate extension to LCT is to make the color transformation matrix T_{YIQ} specifiable by users. A user chosen color transformation T_u can further enrich the watermarking signature space. We leave this extension to future work.

4.2 The Verification Method

We let a third-party arbitrator independent of the user and DNN learner determine whether the user's images were part of the DNN training set. The arbitrator

has to infer a signature from a suspicious DNN classifier f and the user's original, unwatermarked images $\mathcal{D}_u$ without using the private watermark signature k^*. If the inferred signature matches the user's private one, we say that the DNN classifier is highly likely trained using the user's images $\mathcal{D}_u^*$.

Assume that the watermarking function $g(I, k^*)$ does not change the image's class label. Let y denote the class label of image $I \in \mathcal{D}_u$. We design a simple yet effective approach to recovering the watermarking signature:

$$\hat{k} \leftarrow \arg\min_{k \in \mathcal{K}} \sum_{(I,y) \in \mathcal{D}_u} \mathcal{L}(f(g(I,k)), y) \tag{6}$$

where $\mathcal{L}$ is a loss (e.g., cross-entropy) for learning the DNN classifier f, and $\mathcal{K}$ is the collection of all possible signatures.

If the inferred signature matches the user's private one, $\hat{k} \approx k^*$, the arbitrator concludes that the DNN learner has used the user's images $D_u^* = \{g(I, k^*), \forall I \in \mathcal{D}_u\}$ as part of the training set for DNN f. Otherwise, the DNN learner is likely innocent.

The Signature Space $\mathcal{K}$. It is important to discuss the success rate of the above verification method. Apparently, the signature space $\mathcal{K}$ should be sufficiently large to reduce the probability of an innocent classifier coincidentally matching a user's watermarked images. For the analysis purpose, we discretize the bounded signature space $\mathcal{K}$ into N equal-sized, non-overlapped slots, each with an interval 2τ. We say the recovered signature $\hat{k}$ matches the private one k^* when $|\hat{k} - k^*| < \tau$. Reserving one slot for no watermarking, the number of valid watermarking signatures is $N - 1$. Clearly, the larger N is, the more convincing the verification.

Some readers might wonder what if there is a large number of users. For example, given 1 million users but a small number N of signatures, would this setting fail the proposed anti-neuron watermarking? The answer is a pleasant no because, importantly, two users could choose the same private watermarking signature as long as their personal images are different, though the chance of using the same signature is low because each user independently chooses a signature. What happens when a user chooses not to watermark her/his images? A well-trained neural classifier should generalize well under the training distribution. Hence, if most training images are not watermarked, given the user's original unwatermarked data, the recovered signature from the well-trained classifier would approach no watermarking.

It is not necessary to have enormous N to avoid users having duplicated private signatures based on the above discussion. However, a sufficiently large N is still preferred for another reason, DNNs' memorization. Only when N is big, the chance becomes high for a user to watermark her/his images into a low-density region and hence can be memorized by DNNs.

A large signature space also benefits the memorization of user signatures. According to the study [9] on memorization, deep neural classifier must memorize atypical examples to perform well on the less frequent examples during inference.

Since watermarking shift data distribution via signature from a large space, watermarking is highly likely to lift user images into lower density region and thus being better memorized by neural models.

Optimization Method and Computational Cost for Signature Inference. To solve eq. (6) efficiently, we propose two optimization methods. **(i) Grid search :** the arbitrator can enumerate all signatures for watermarking and perform grid search over the bounded signature space with a linear computational cost as $O(N)$. If the signature is well memorized by a DNN, the DNN loss will reach minimum when the signature being evaluated equals or closely approximates the private signature used by the user. **(ii) Gradient search:** when the model is accessible, the arbitrator can watermark clean images with a random initial watermark signature and then infer the user's signature by descending along the gradient of training loss with respect to the signature. This technique infers the watermark signature more precisely than grid search and the computational cost might be less for a large N.

5 Experiments

5.1 Setup

We evaluate the proposed watermarking in image classification on the Cifar-10 / Cifar-100 [3], CUB birds [36] and Tiny ImageNet [17] datasets.

A User Watermarks Their Personal Data. A portion of randomly chosen images from a training set (by default, 1% for Cifar and 0.1% for Tiny ImageNet) is defined as a user' personal data. The user data could contain samples from any class. Each user image is watermarked using eq. (5) by a given signature in the space of $[30, 60, ..., 330]$, followed by clipping pixel values to the valid range of [0,1]. By default, we use 60 (*i.e.*, rotating hue by 60 degree) as the signature.

A Learner Trains Neural Classifiers Using Unauthorized User Data. An unauthorized learner trains neural classifiers using the above watermarked user data along with other training data. Images are randomly cropped, horizontal flipped, and normalized following the common data augmentation practice [14,18]. We use ResNet50 [14] as the default neural classifier and train every model from scratch for 90 epochs. The initial learning rate is 0.1 and decays by 0.1 for every 30 epochs.

A Verifier Infers The Watermark Signature. Given suspicious neural network models, a third-party verifier infers the user's signature following two approaches discussed above. For *grid search*, we iterate over all candidate signatures generated by dividing the whole signature space into $N = 12$ intervals whose length is $2\tau = 2 \times 15$. For *gradient search*, we exploit gradient descent to learn the signature. To avoid local optimum, multiple initial values are used and the best signature that leads to the lowest loss is returned.

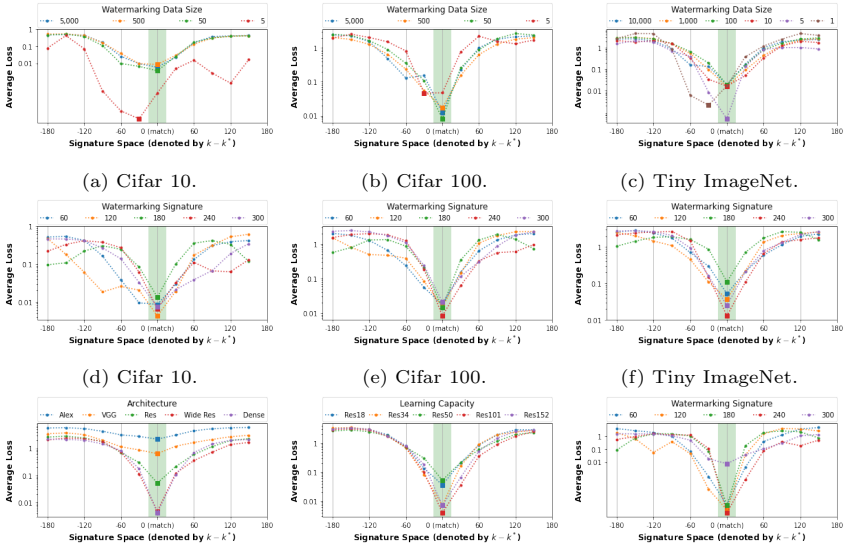
5.2 Analyzing Effectiveness of Watermarking

We first show empirically how signatures are memorized by the neural classifiers. Here, we consider a single user watermarking their data for simplicity. (See Appendix for other experiments and the gradient search results.)

1. *Different Numbers of Watermarked Samples.* We study how many images are desired for making anti-neuron watermarking successful. The grid search result is shown in Figure 2 (a, b, c) using eq. (6). It is visually clear that most of the models achieve the minimum loss near the watermark signature, within the range of matching $|\hat{k} - k^*| < \tau$. However, with less watermarked data (*e.g.*, less than 5 samples), the inferred signature with minimum loss does not match the user's private signature.
2. *Different Watermark Signatures.* We verify whether different watermark signatures work equivalently. We experiment with different signatures on one user's data and show the grid search results in Figure 2 (d, e, f) for different datasets. From these figures, we observe that all inferred signatures (marked in square) match the user's signatures for watermarking, indicating that different hue adjustments can all be used for anti-neuron watermarking.
3. *Different Neural Classifier Architectures.* We also evaluate the proposed watermarking for different neural classifier architectures, including Alexnet [18], VGG [32], ResNet [14], Wide ResNet [43] and DenseNet [16] trained with default settings. As shown in Figure 2g, all inferred signatures match the user's, implying that our watermarking approach works well against a large variety of deep neural networks.
4. *Different Learning Capacities of Models.* We further investigate whether a model memorizes watermark signatures better when the model has more learning capacity (e.g., more parameters, deeper or wider) by exploring the ResNet family. As shown in Figure 2h, as the networks go larger and deeper, the loss decreases faster and reaches the minimum around the watermark signature more sharply.
5. *High Resolution Images.* In Figure 2i, we present our result on CUB-200-Birds, a fine-grained dataset with high-resolution images of 448×448. We use pretrained ResNet50 from ImageNet and conduct a transfer learning on CUB-200-Birds. The dataset has fewer than 6000 images for training, and we assume the user has 60 images (1%) for watermarking. Strong data augmentations [41] are used to boost performance, including color jitter, random crop, random resize, random scale and random horizontal flip. Even under the strong data augmentations and the transfer learning setting, the result shows that ResNet50 memorizes the user's signature well.

5.3 Analyzing Properties of Watermarking

We then evaluate how the properties discussed in Section 4.1 would help anti-neuron watermarking.



(a) Cifar 10. (b) Cifar 100. (c) Tiny ImageNet.

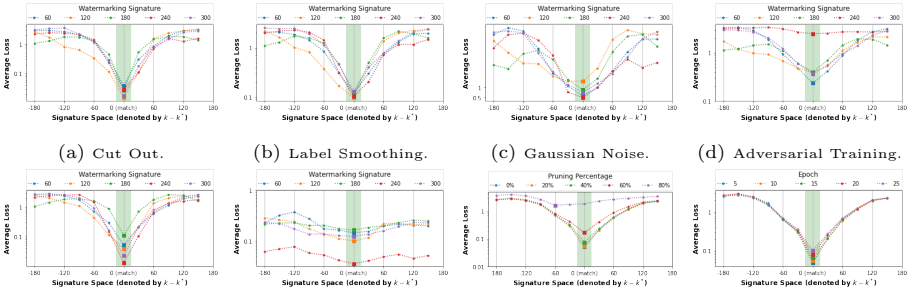
(d) Cifar 10. (e) Cifar 100. (f) Tiny ImageNet.

(g) Different architectures. (h) Different capacity. (i) High resolution.

Fig. 2. The first row shows variant of loss for models trained with **different quantity** of watermarked samples on Cifar and Tiny ImageNet. The second row shows variant of loss for models with **different signatures** on Cifar and Tiny ImageNet. The third row shows results for **different architectures**, **different capacity** on Tiny ImageNet and **high resolution** on CUB-200-Birds. The x-axis represents signature space (denoted by distance to the user private signature, *i.e.*, $k - k^*$), and the y-axis represents the average loss of user data with respect to signature k. The green region represents the range for a match ($< 2\tau$). If the inferred signature $\hat{k}$ (marked as **Square marker** indicating the point with minimum loss) lies in the green region ($|\hat{k} - k^*| < \tau$), it would be a match. Otherwise, it would be a miss.

Resilience to Data Augmentation. We evaluate if our anti-neuron watermarking is resilient against various common data augmentations, especially those involving random hue transformations. We apply random crop and random horizontal flip in all our experiments following [14, 18]. Besides, several widely adopted data augmentations including random cutout [6], label smoothing [34], Gaussian noise [4], adversarial training [22] and differential privacy [7] are evaluated. Finally, we test color jittering [18], which includes brightness, saturation, contrast and *the same* hue transformation we used for watermarking. As shown in Figure 3, the watermark signatures can be inferred correctly for the aforementioned data augmentations. This shows empirically that LCT is an effective anti-neuron watermarking approach because it is resilient to common data augmentations in neural networks' training. We also evaluate privacy preserving techniques such as differential privacy [7]. Since we infer the signature using all user images, noise added to the output would be reduced by taking an average.

Beside, we also consider two common *defense techniques* against watermarking: pruning and fine-tuning. We follow common settings [45] and find in Figure 3g and Figure 3h that LCT is also resilient to these defense methods.



(a) Cut Out. (b) Label Smoothing. (c) Gaussian Noise. (d) Adversarial Training.

(e) Differential Privacy. (f) Color Jitter. (g) Pruning. (h) Fine-tuning.

Fig. 3. The variation of model loss for **different data augmentations**. Only the color jitter can significantly narrow down the loss difference between signatures.

Less Noticeable Watermarking. In Section 4.1, we discuss the watermarking should not change the major content of an image, and one of the desired features is to make watermarking unnoticeable to human. Rather than changing the hue of images globally, we study an alternative technique from traditional watermarking proposed in [42]. It adjusts the blue channel's intensity on preselected pixels. In this work, we adjust the intensity as a watermark signature on 512 randomly selected pixels. As shown in Figure 4, the general appearance of watermarked images are less noticeable than changing the hue globally (adjusting hue for 4096 pixels by 60). For $\tau = 0.1$ and watermark signatures (the blue channel's intensity), 0.1, 0.3, 0.5, 0.12, 0.28, 0.44 are used in inference on selected pixels, matching the user's watermark signature. However, this kind of watermarking has its limitation. It introduces noise to images, and the images could look noisy when the color themes are dominated by red or green. To solve this problem, we may find some transformations that are invisible to humans but easy to learn by neural classifiers. We leave this challenge to future work.

(a) Clean (b) Hue (c) 0.1 (d) 0.3 (e) 0.5

Fig. 4. Illustration of clean, hue-transformed and less noticeable watermarked samples. 0.1,0.3 and 0.5 are the intensity of blue channel for selected pixels.

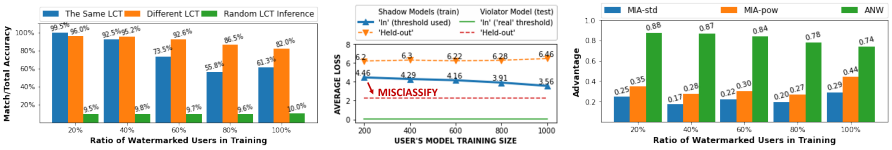
5.4 Analyzing Signature Space for Verification

In Section 4.2, we discuss how the signature space could affect watermarking from the perspective of a third-party verifier. Here, we show experimentally how to have a trustful signature space for a convincing verification.

When User Data Was Not Used to Train Neural Classifiers. From previous experiments, we show that the inferred signature matches a user's private watermark signature if the user's watermarked data have been used in training. Here we show the inferred signature approaches no watermarking when the user data was not used for training. To this end, we construct held-out users using auxiliary unseen data from validation. Pretrained models from Figure 2f are used to infer signature from the held-out users. Not surprisingly, the inferred signatures approach 0 (no watermarking) for the held-out users, with $|\hat{k} - 0| = 4.3 \pm 1.4$.

Multiple Users with User-specific Watermarking Signatures. We examine multiple users for several scenarios. The training set of Tiny ImageNet is equally divided into 1,000 users. Then we evaluate the effectiveness of watermarking for different ratio of users exploiting the same LCT (eq. (3)) or different LCTs. For the later, we sample 3×3 matrices from a uniform distribution $T_u \sim \mathcal{U}(-1, 1)$ per user. Each user chooses a random signature from $[30, 60, ..., 330]$, and τ is set to 15.

From the result shown in Figure 5a, we can observe that when 20% of users watermark their images using LCT, their watermark signatures can be inferred correctly for almost all the users. As this ratio increases, the matching accuracy drops significantly if the users use the same LCT. However, if they use different LCTs, the matching accuracy remains above 80% even when all users data are watermarked independently. We also evaluate a special case when an adversary infers signatures using an arbitrary LCT. The arbitrary LCTs ($T_{u'} \sim \mathcal{U}(-1, 1)$) only achieve 10% matching accuracy, which is 70% less when LCTs are given. Such results indicate users can use unique and user-specific watermarking for a better protection rate when other users may also exploit watermarking.



(a) (b) (c)

Fig. 5. (a) Watermarking performance for **multiple users**. (b) MIAs fail to work with user-specific shadow models because insufficient training data leads to much higher threshold. (c) Our ANW *vs.* MIAs using the adv^{P} metric.

5.5 Comparisons with Related Methods

Comparing to Membership Inference Attacks. We compare our anti-neuron watermarking (ANW) with two *threshold-based* membership inference attacks (MIAs): In MIA-std [40], few samples are known in violator's training set and their average training loss would be used as threshold ϵ to infer membership (if $\mathcal{L}_t < \epsilon$, target sample t is in training, *vice versa*.); In MIA-pow [30], few samples in training and held-out together determine a better threshold.

There are 3 scenarios making MIAs not applicable for PIP. **(i)** as a user has no knowledge about the violator's training distribution, MIAs cannot be applied as neither shadow models nor threshold can be obtained. **(ii)** Users could train shadow model using their own data and perform MIAs via shadow models [31], but MIAs would not work well because shadow models trained with user data would produce a much larger loss than the learner' models. Consequently, if a threshold is chosen from shadow models trained with user data, held-out sample would be misclassified as "in training" because the learner's model would produce smaller loss for both "training" and "held-out" samples. Here we compare MIAs by creating 10 users with individual data. 10 shadow models are trained respectively and MIA-std is performed with the threshold of average training loss. For ANW, 10 users data are watermarked with different signatures. Under different settings for user data size, we find out that ANW achieves 100% matching accuracy while MIA-std achieves 50% (misclassify all held-out samples). Figure 5b shows how the threshold from shadow models fails to classify sample membership for the learner. Similarly, MIA-pow along with other MIAs [23,31] relied on shadow models would also fail in this settings. **(iii)** At last, even if a user acquires all necessary information, the MIA results would still be less convincing. As MIAs only provide binary output (True/False), it is difficult to convince the verifier when random guess can still achieve about 50% success rate. To quantifiably and fairly compare with MIAs, we extend the classic *membership advantage* [39] into *protection advantage* (adv^{P}) considering both accuracy and fidelity:

$$\text{adv}^{\text{P}} = \int_x (\mathbb{P}_e(y = True|x) - \mathbb{P}_r(y = True|x))dx \tag{7}$$

The adv^{P} metric quantifies quality of protection through the expectation gap between the empirical successful inference ($\mathbb{P}_e$) and successful random guess ($\mathbb{P}_r$). Note that the *membership advantage* [39] is a special case of adv^{P} metric when the second term is 0.5. For a Bernoulli experiment, the above formula could be calculated as $\frac{M-Np}{N}$, where M is the total matches in N experiments and p is the probability of a correct random guess. With above metric, we conduct comparison between our watermarking and two threshold-based MIAs [30, 40] (See appendix for experiment settings.). As shown in Figure 5c, our ANW significantly outperforms the MIA approaches under the adv^{P} metric with both accurate and convincing inference, showing that watermarking is a feasible method in the PIP problem.

Comparing to Dataset Tracing [29]. Dataset tracing [29] exploits pretrained classifier to generate traceable data. If neural classifier learns such dataset, the decision boundary of classifier would become more aligned with watermarking vectors (*i.e.*, cosine similarity becomes higher). In Table 1, we compare this approach [29] with ours when only 0.1% data being watermarked. For dataset tracing, it is computed by the classifier's weight vector and the watermarking vector. And for our method, it is computed by inferred and user watermarking signatures. The experimental results show that it is easier to memorize low dimensional signatures as our watermarking method lifts the cosine similarity significantly after training with tiny portion of watermarked data.

	Before Training	After Training
Dataset Tracing [29]	-0.005 ± 0.030	-0.005 ± 0.015
Our Watermarking	0.045 ± 0.52	0.999 ± 0.001

Table 1. **Cosine similarity**. With only 0.1% data watermarked, the dataset tracing shows almost no effects while our watermarking method improves the similarity to almost 1 after training.

5.6 Improving Memorization by Watermarking

Finally, we explore empirically why watermarking is effective against neural classifier by revisiting "Memorization Value Estimate" (MAE) [9]. MAE measures the generalization gap (difference of prediction between models trained with/without certain data) to quantify the memorization ability of neural networks toward such data. A higher MAE after watermarking indicates the model tends to memorize watermarked data than original data. For user data, we observe the MAE increases from 26.8% to 34.8% after applying watermarking, indicating that our approach increases memorization of user data and thus the signature would be easier memorized along with user data.

6 Conclusion

In this paper, we introduce a new personal data protection problem against unauthorized neural model training. To protect user personal data, we propose an anti-neuron watermarking approach based on linear color transformation. By watermarking user's images with private signature using LCT, unauthorized usage of user personal data can be verified by a third-party neutral arbitrator. Through extensive experiments, we show empirically that LCT-based watermarking is effective in protecting user data from unauthorized usage in a various realistic settings.

Acknowledgements: this work was supported in part by NSF-1704309 and NSF-1952792.

References

1. Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., Song, D.: The secret sharer: Evaluating and testing unintended memorization in neural networks. In: 28th USENIX Security Symposium (USENIX Security 19). pp. 267–284 (2019)
2. CCPA: California privacy act (Jan 2021), https://oag.ca.gov/privacy/ccpa
3. Cifar: (2009), https://www.cs.toronto.edu/~kriz/cifar.html, cIFAR Dataset
4. Cohen, J., Rosenfeld, E., Kolter, Z.: Certified adversarial robustness via randomized smoothing. In: International Conference on Machine Learning. pp. 1310–1320. PMLR (2019)
5. Cox, I.J., Miller, M.L., Bloom, J.A., Honsinger, C.: Digital watermarking, vol. 53. Springer (2002)
6. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
7. Dwork, C.: Differential privacy: A survey of results. In: International conference on theory and applications of models of computation. pp. 1–19. Springer (2008)
8. El'arbi, M., Amar, C.B., Nicolas, H.: Video watermarking based on neural networks. In: 2006 IEEE International conference on multimedia and expo. pp. 1577–1580. Ieee (2006)
9. Feldman, V., Zhang, C.: What neural networks memorize and why: Discovering the long tail via influence estimation. arXiv preprint arXiv:2008.03703 (2020)
10. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. pp. 1322–1333 (2015)
11. GDPR: Regulation (eu) 2016/679 (general data protection regulation) that is applicable as of may 25th, 2018 in all member states, is to the protection of natural persons with regard to the processing of personal data and rules relating to the free movement of personal data. (2016), https://gdpr-info.eu/
12. Gu, T., Liu, K., Dolan-Gavitt, B., Garg, S.: Badnets: Evaluating backdooring attacks on deep neural networks. IEEE Access **7**, 47230–47244 (2019)
13. Guo, J., Potkonjak, M.: Watermarking deep neural networks for embedded systems. In: 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). pp. 1–8. IEEE (2018)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
15. Homer, N., Szelinger, S., Redman, M., Duggan, D., Tembe, W., Muehling, J., Pearson, J.V., Stephan, D.A., Nelson, S.F., Craig, D.W.: Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. PLoS Genet **4**(8), e1000167 (2008)
16. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
17. kaggle: (2017), https://www.kaggle.com/c/tiny-imagenet, tiny Imagenet
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012)

19. Kundur, D., Hatzinakos, D.: Digital watermarking using multiresolution wavelet decomposition. In: Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181). vol. 5, pp. 2969–2972. IEEE (1998)
20. Li, Y., Zhang, Z., Bai, J., Wu, B., Jiang, Y., Xia, S.T.: Open-sourced dataset protection via backdoor watermarking. arXiv preprint arXiv:2010.05821 (2020)
21. Liu, Y., Ma, S., Aafer, Y., Lee, W.C., Zhai, J., Wang, W., Zhang, X.: Trojaning attack on neural networks. NDSS Symposium (2017)
22. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=rJzIBfZAb
23. Maini, P., Yaghini, M., Papernot, N.: Dataset inference: Ownership resolution in machine learning. arXiv preprint arXiv:2104.10706 (2021)
24. Meggs, P.B.: A History of Graphic Design. Wiley (1998)
25. Meng, Z., Morizumi, T., Miyata, S., Kinoshita, H.: Design scheme of copyright management system based on digital watermarking and blockchain. In: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). vol. 2, pp. 359–364. IEEE (2018)
26. Nagai, Y., Uchida, Y., Sakazawa, S., Satoh, S.: Digital watermarking for deep neural networks. International Journal of Multimedia Information Retrieval **7**(1), 3–16 (2018)
27. Nasr, M., Shokri, R., Houmansadr, A.: Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In: 2019 IEEE symposium on security and privacy (SP). pp. 739–753. IEEE (2019)
28. Rouhani, B.D., Chen, H., Koushanfar, F.: Deepsigns: an end-to-end watermarking framework for protecting the ownership of deep neural networks. In: ACM International Conference on Architectural Support for Programming Languages and Operating Systems (2019)
29. Sablayrolles, A., Douze, M., Schmid, C., Jégou, H.: Radioactive data: tracing through training. In: International Conference on Machine Learning. pp. 8326–8335. PMLR (2020)
30. Sablayrolles, A., Douze, M., Schmid, C., Ollivier, Y., Jégou, H.: White-box vs black-box: Bayes optimal strategies for membership inference. In: International Conference on Machine Learning. pp. 5558–5567. PMLR (2019)
31. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 3–18. IEEE (2017)
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
33. Standing Committee of the National People's Congress: China data security law (Jul 2021), http://www.xinhuanet.com/2021-06/11/c_1127552204.htm
34. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
35. Tirkel, A.Z., Rankin, G., Van Schyndel, R., Ho, W., Mee, N., Osborne, C.F.: Electronic watermark. Digital Image Computing, Technology and Applications (DICTA'93) pp. 666–673 (1993)
36. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-UCSD Birds 200. Tech. Rep. CNS-TR-2010-001, California Institute of Technology (2010)

37. Wikipedia: Yiq (2015), http://en.wikipedia.org/wiki/YIQ
38. Wikipedia: Cambridge analytica (2018), https://en.wikipedia.org/wiki/Cambridge_Analytica
39. Yeom, S., Fredrikson, M., Jha, S.: The unintended consequences of overfitting: Training data inference attacks. arXiv preprint arXiv:1709.01604 **12** (2017)
40. Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy risk in machine learning: Analyzing the connection to overfitting. In: 2018 IEEE 31st Computer Security Foundations Symposium (CSF). pp. 268–282. IEEE (2018)
41. Yu, F., Wang, D., Shelhamer, E., Darrell, T.: Deep layer aggregation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2403–2412 (2018)
42. Yu, P.T., Tsai, H.H., Lin, J.S.: Digital watermarking based on neural networks for color images. Signal processing **81**(3), 663–671 (2001)
43. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
44. Zhang, J., Chen, D., Liao, J., Fang, H., Zhang, W., Zhou, W., Cui, H., Yu, N.: Model watermarking for image processing networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 12805–12812 (2020)
45. Zhang, J., Chen, D., Liao, J., Zhang, W., Hua, G., Yu, N.: Passport-aware normalization for deep model protection. Advances in Neural Information Processing Systems **33**, 22619–22628 (2020)
46. Zhong, X., Huang, P.C., Mastorakis, S., Shih, F.Y.: An automated and robust image watermarking scheme based on deep neural networks. IEEE Transactions on Multimedia **23**, 1951–1961 (2020)