A Background

In the following sections, we present the background knowledge of personal data protection problem studied in our paper. We first quote the legislative definition of "personal data protection" from GDPR [10] and then we discuss the scope of our problem comparing to general data protection. Finally, some "data proprietary" problem are addressed in personal data protection.

Definitions of Terms from GDPR. To help readers better understand the terms we used in the manuscript, we quote several term definitions from [10] as following:

"*Personal data* means any information relating to an identified or identifiable natural person ('data subject')."

"*Third party* means a natural or legal person, public authority, agency or body other than the data subject, controller, processor and persons who, under the direct authority of the controller or processor, are authorised to process personal data."

"*Consent* of the data subject means any freely given, specific, informed and unambiguous indication of the data subject's wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement to the processing of personal data relating to him or her."

"*Personal data breach* means a breach of security leading to the accidental or unlawful destruction, loss, alteration, unauthorised disclosure of, or access to, personal data transmitted, stored or otherwise processed."

From above definitions, we show the basic idea of GDPR: a legislation approach to protect users personal data from being utilized by third-party without users' consent. More details of GDPR and similar laws can be found online [2, 10, 27]. In our paper, we focus on the technical perspective of personal data protection by exploiting anti-neuron watermarking to verify unauthorized usage of user's personal images.

Personal Data Protection *v.s* General Data Protection. As our goal is to protect "Personal Data" (*i.e.* user data shall be related to identifiable natural person [10]), we assume each user's personal images can be distinguished from the others. This assumption differs "personal data protection" from "general data protection" problem, where a user's data are not necessarily to be identified from other users'. In "general data protection", our anti-neuron watermarking would be less applicable. Because neural models learn both watermarked and unwatermarked data, the inferred signature would be a signature value between the watermarked and unwatermarked. We leave this challenge for future studies.

"Data proprietary" in Personal Data Protection. One traditional usage of digital watermarking is to protect data proprietary. To verify proprietary, the space of watermarking signature needs to enormous such that each signature (*i.e.*, a hand-writing signature) can be considered as a unique identifier for user data. However, in our problem settings, our protection focuses on verifying unauthorized usage (*i.e.*, without "consent") of user data (images that user already has "data proprietary" on) instead of verifying data proprietary. Since the low-dimensional signature is only used to verify unauthorized usage of images,

our signature space does not need to be as large as "data proprietary" problem. Readers might be curious what if adversary exploits user's watermarking function and reverts a signature on arbitrary images that leads to minimal loss of arbitrary neural models. Can adversary accuse data privacy breach for these models? Can adversary claim ownership of these arbitrary data? The answers are no. According to our previous analysis on signature space, the signature inferred on arbitrary images is highly likely to be no watermarking, a signature value excluded from valid watermarking signatures. On the other hand, adversary cannot claim data propriety on arbitrary data as they cannot prove that the data are legally "relating to" adversarial users.

B Preliminary Study

Verification by Recovering Watermarking Pattern. Recent studies show that DNNs can "memorize" some training examples in various ways [1, 7, 8], and one can recover certain meaningful low-resolution images from CNNs [8]. Motivated by these studies, we perform a preliminary study on traditional watermarking technique by appending a special pattern (*e.g.*, a sticker) on images. We train a ResNet50 on 100 randomly selected user samples with a cat pattern as watermark in Tiny ImageNet. Similar to model inversion [8], we use a learnable Gaussian variable on users' images and minimize the classification loss to reconstruct watermark pattern with this variable.

As shown in Figure 1, although the reconstructed pattern can achieve 100% accuracy and minimal classification loss, neural models cannot memorize such watermark pattern, as no meaningful pattern can be recovered. We believe that this could be caused by convolution operation where all spatial information of pixels are being ignored during training. This experiment indicates that it is difficult for neural learners to memorize such kinds of watermarks, comparing to the watermarks using color-based transformation.

Fig. 1. Illustration of watermark pattern (left), recovered pattern (middle) and watermarked images (right). Recovering small visible watermark fails with getting noise.

C Implementation Details

Datasets. We evaluate our anti-neuron watermarking on Cifar [3], Tiny ImageNet [15] and CUB-200-Birds [29]. Cifar are widely adopted datasets with

50,000 training samples, 10,000 testing samples for 10 and 100 classes, respectively. Tiny ImageNet is a selective subset of ImageNet, containing 100,000 training samples, 10,000 validation samples and 10,000 testing samples for 200 classes. Each sample is $3 \times 64 \times 64$. Since testing labels are not publicly available, we report models' validation accuracy for models' performance. CUB-200-Bird is a high resolution 448×448 fine-grained dataset containing 200 bird species, with 5994 training samples and 5794 testing samples.

Watermark User Data. Each user image is watermarked by given 3×3 LCT function with a signature k followed by a pixel value clipping. A clipping is needed because LCT could cause overflow on some pixels' values. The clipping operation in theory will break the differentiable property of watermarking function and thus hinder gradient based optimization (*e.g.,* stochastic gradient decent). However, we find empirically that this operation does not affect signature inference. Hence, we conduct clipping after the LCT in all our experiments.

Data Preparation. For training, each image is first converted from $[0, 255]$ to $[0, 1]$, and watermarked if it belongs to the user whose images need to be protected. Then data augmentation and normalized are applied to improve training. During signature inference, each image is converted from $[0, 255]$ into $[0, 1]$, then watermarked and normalized. The normalization mean and variance of RGB channels are $(0.5, 0.5, 0.5)$ and $(0.5, 0.5, 0.5)$ for Cifar, $(0.485, 0.456, 0.406)$ and $(0.229, 0.224, 0.225)$ for Tiny ImageNet and CUB-Birds, respectively.

Grid Search Settings. We iterate all the possible signatures in our grid search experiments. The signatures are generated by dividing the whole signature space into $N \times 2\tau$ intervals.

Gradient Search Settings. During signature inference, we recover signature using unwatermarked user data. These user data are combined into one mini-batch and stochastic gradient decent is used for optimization over signatures. The initial learning rate is 0.1 and decays 0.1 each 100 epochs, with 300 epochs in total. To avoid local minima, we select initial values from all possible signatures and report the signature that lead to minimal loss. Comparing to grid search, this optimization approach is much more computational expensive and thus our results will be mainly on grid search for simplicity.

Source Code. Our experimental source code will be publicly available (code is available in supplementary folder). All our experiments are implemented by Pytorch. Readers can freely explore our proposed watermarking approach with the code supplied.

D Additional Results for Effectiveness of Watermarking

Gradient Search Results. In the main manuscript, we present the results for grid search in various settings. Here, we show the additional gradient search results on Tiny ImageNet.

1. *Different sizes of Watermarked Samples.* We first present our result with different sizes of watermarkred samples in Table 1. Being trained by classifier, user's watermarked images achieve lower loss than the clean images.

And watermarking does not affect classification performance for getting similar testing accuracy. These results show that given sufficient data, neural classifier could memorize watermark signature on user's data pretty well.

# of Data	Model Acc	Watermark Loss	Clean Loss	Inferred signature
10,000	55.6	0.019	0.161	59.0 ✓
1,000	55.8	0.016	0.541	56.4 ✓
100	54.9	0.019	0.668	59.5 ✓
10	54.5	0.017	0.329	48.9 ✓
5	54.6	0.001	0.397	60.99 ✓
1	55.9	0.004	0.006	17.0 ×

Table 1. Inferred signatures for models trained with **different sizes** of watermarked data on Tiny ImageNet. The watermark signature is 60 with $\tau = 15$.

2. *Different Watermark Signatures.* We present the gradient search result for different signatures on Tiny ImageNet in Table 2. The result shows that the watermarking for single user does not affect models' training as models' accuracy are similar for different signatures. User watermarked images achieve lower average loss than original images, indicating unauthorized training models can memorize watermarked images in "some way". And we achieve minimal loss nearby watermarking signature, bounded by predefined threshold $\tau = 15$. This experiment shows that different signatures of watermarking work equivalently.

Watermark signature	Model Acc	Watermark Loss	Clean Loss	Inferred signature
60	54.9	0.019	0.668	59.5 ✓
120	55.6	0.070	0.895	120.2 ✓
180	53.8	0.030	1.189	178.8 ✓
240	56.3	0.025	1.118	242.1 ✓
300	55.7	0.016	0.839	306.1 ✓

Table 2. Watermarking for **different signatures** for ResNet50 on Tiny ImageNet.

3. *Different Neural Classifier Architectures.* We present the gradient search result for different architectures on Tiny ImageNet in Table 3. Alexnet [17], VGG [25], ResNet [12], Wide ResNet [34] and DenseNet [13] are evaluated in this experiment. The watermark signature is 60 and $\tau = 15$.
4. *Different Learning Capacity of Models.* We present the gradient search result for different learning capacity of models on Tiny ImageNet in Table 4.

Architecture	Model Acc	Watermark Loss	Clean Loss	Inferred signature
Alex	38.0	2.189	3.175	56.9✓
VGG	57.3	0.640	1.151	52.8✓
Res	54.9	0.019	0.668	59.5✓
Wide Res	56.6	0.005	0.738	58.4✓
Dense	61.5	0.114	0.838	58.8✓

Table 3. Watermarking for **different architectures** on Tiny ImageNet.

ResNet [12] family is evaluated in this experiment. The watermark signature is 60 and $\tau = 15$.

Architecture	Model Acc	Watermark Loss	Clean Loss	Inferred signature
ResNet18	52.9	0.036	0.801	55.2 ✓
ResNet34	53.6	0.007	0.679	55.8 ✓
ResNet50	54.9	0.019	0.668	59.5 ✓
ResNet101	54.6	0.004	0.736	58.3 ✓
ResNet152	56.5	0.007	0.814	60.4 ✓

Table 4. Watermarking for **different learning capacities** on Tiny ImageNet.

E Additional Results for Watermarking Properties Analysis

In this section, we provide more details of our experiments on analyzing the properties of watermarking.

Resilience to Data Augmentation. Here, we present our settings to evaluate if our proposed watermarking method could survive under common data augmentations used in neural models' training. The following data augmentations are evaluated:

1. *Random Crop* [17]. Random crop is a widely used data augmentation for most neural models' training [12, 17, 25, 32]. It randomly crops images into smaller resolution to reduce models' overfitting on spatial location. For Tiny ImageNet, images are randomly cropped into 64×64 with a padding of 8. For Cifar, images are randomly cropped into 32×32 with a padding of 4. All our training includes random crop for best performance.
2. *Horizontal Flipping* [17]. This is also a widely used data augmentation for image classification. We include horizontal flipping for all our experiments.

3. *Cut Out* [5]. Cutout removes random region of size $M \times M$ from images at each training iteration. We set the size $M = 8$ for our experiments.
4. *Label Smoothing* [28]. Label smoothing is also a widely used data augmentation in many tasks [20]. It reduces the probability of ground truth label (*e.g.*, 100% cat, 0% dog, 0% duck) by a smoothing parameter α and assigns probability uniformly to other classes (90% cat, 5% dog, 5% duck). The smoothing parameter α is set to be 0.1 in our experiments.
5. *Gaussian Noise* [4]. This technique simply adds noise to the input from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$ to increase models' robustness. The σ^2 is set to be 0.1 in our experiments.
6. *Adversarial Training* [19]. Neural Networks are well known for their vulnerability to adversarial attacks. [11, 19], and adversarial training are believed to reduce overfitting [19] and mitigate privacy leakage [14, 21]. We address this by training with adversarial samples generated from FGSM attack [11]. The epsilon is set to be 0.01.
7. *Differential Privacy* [6]. Differential privacy is a wide adopted privacy preserving technique in many real-world applications. By adding noise to the query results, user's sensitive information cannot be inferred via querying. In our implementation, we add random noise samples from Gaussian distribution $\mathcal{N}(0, \sigma^2)$ to the output confidence. The σ^2 is set to be 0.1.
8. *Color Jitter* [17]. Color jitter randomly adjusts brightness, contrast, saturation and hue of input images. We apply high intensity color augmentation in our experiments. For each color properties, the value of adjustment is randomly sampled from [-288, 288], covering 80% of the range of transformation. The same conclusion can be made by comparing Table 2 with Table 5. We further explore on CUB-200-birds with a stronger data augmentation. Watermark signature would fail to be inferred in this case. This observation implies that if models are trained heavily with similar data augmentations as watermarking, the signature inference could be confused by nearby signatures and thus fail to recover the watermarking signatures.

Watermark signature	Model Acc	Watermark Loss	Clean Loss	Inferred signature
60	52.5	0.042	0.054	64.8 ✓
120	52.5	0.102	0.244	108.3 ✓
180	53.0	0.052	0.081	178.8 ✓
240	53.0	0.113	0.132	254.4 ✓
300	51.7	0.125	0.161	291.3 ✓

Table 5. Watermarking for model trained with **color jitter** augmentation for ResNet50 on Tiny ImageNet. Loss difference between clean and watermarked samples are smaller comparing with Table 2.

In Table 6, watermark signatures can be inferred correctly from gradient search for the aforementioned data augmentations. This shows empirically that LCT is an effective watermarking approach because it is resilient to common data augmentations in neural networks' training.

Augmentation	60	120	180	240	300
Cut Out	57.5	122.1	178.6	240.3	301.8
Lable Smoothing	59.6	116.9	181.4	239.3	299.9
Gaussian Noise	58.3	107.8	187.1	237.2	309.9
Adv Training	57.5	118.8	183.5	242.9	298.6
Differential Privacy	56.0	117.5	182.8	240.6	295.9
Color Jitter	64.8	108.3	178.8	254.4	291.3

Table 6. Inferred signatures for models trained with **different data augmentations** for ResNet50 on Tiny ImageNet.

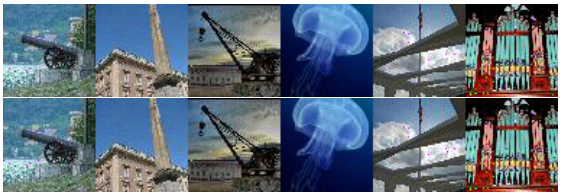
Less Noticeable Watermarking. In the previous sections, we show that LCT-based watermarking is effective against unauthorized neural learners, but may change the color property significantly in visualization, as illustrated in Figure 2. One may argue that such a kind of watermarking could be too visually obvious to be recognized by unauthorized neural learners. However, given data samples, by selecting proper watermarking signature, the watermarking could be difficult to be distinguished from stylish transformation or even unnoticeable to human being.

Fig. 2. Selective clean (top), watermarked (middle), reconstructed (bottom) samples' comparison for hue-based watermarking applied on the whole images.

For example, nowadays users would often apply image filters to stylize images before publishing on social media, where filters are quite natural such as tuning the color of tree leaves from green to yellow, changing the color of sky from light blue to dark blue. The parameters of these images' filters could be used

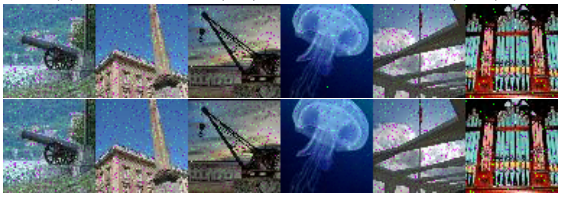
as signature for watermarking. As stylized filters are widely used, it would be difficult for neural learners to distinguish whether it is watermarking or users' preference.

(a) Clean.



(b) Watermarked (0.1) and Reconstructed (0.12).

(c) Watermarked (0.3) and Reconstructed (0.28).



(d) Watermarked (0.5) and Reconstructed (0.44).

Fig. 3. Selective samples' comparison for color-based watermarking for **partial pixels** on blue color channel.

Color-based transformation can be less noticeable when it is only applied on selective pixels and color channels. In particular, by following [33], we first generate a random binary string w with a fixed length T, and then generate pseudo-random pixels' positions as $\rho_t = (i_t, j_t)$ for each element $w_t, (1 \leq t \leq T)$. Finally we change blue color channel for these pixels as:

$$B_{\rho_t} \leftarrow (2w_t - 1)\alpha L_{\rho_t}, \tag{1}$$

where α is the hyper-parameter of watermarking intensity and L_{ρ_t} is luminance of pixel calculated by $L_{\rho_t} = 0.299R_{\rho_t} + 0.587G_{\rho_t} + 0.114B_{\rho_t}$.

Different from [33], we use α as the watermark key and pass the pseudo-random pixels' locations and binary string w to the verifier for key inference. Empirically, this kind of watermarking can be memorized by neural learners but less noticeable to human. We present visual comparison of samples in Figure 3 with different value of α. In summary, color-based watermarking can be both effective and unnoticeable with carefully selected watermarking approach and watermark keys w.r.t users' images.

In our paper, we mainly explore LCT as an effective way for anti-neuron watermarking. And we also show recovering watermark pattern fails to work in PIP while other geometrical watermarking can be applicable. There are certainly other watermarking functions that can verify unauthorized neural model training.

F More Discussions on User-specific Watermarking

As we show in previous section, signatures watermarked by user-specific LCTs would be better memorized among multiple users than those watermarked by a single LCT. This observation, on one hand, implies that selecting an arbitrary LCT for watermarking would be a good practice. It also explains why the LCT watermarking could be resilient to Color jitter. As the color transformations are well-defined and constant matrices (the matrices are fixed, the adjustment could be any value), randomly chosen users' watermarking are highly unlikely to be the same as Color jitter and would be easier to survive in data augmentation. On the other hand, it is very unlikely to infer user signatures without knowing the watermarking function as inferring from arbitrary LCTs have similar matching rate as random guessing, which makes it difficult for attackers to find out the users' signatures without knowing their watermarking functions in advance.

G Comparison with Related Methods

Comparing to Membership Inference Attacks. As membership inference attacks (MIAs) determine membership of given data, it is reasonable to consider such methods in personal data protection. However, there are three restrictions preventing MIAs from being applied in real PIP scenarios.

(i) The first restriction is MIAs require prior knowledge of training data, which would be difficult to obtain as a common user. For example, user needs auxiliary data which has the same data distribution and similar data size as training data so that the well-trained shadow models could have similar performance as adversary models [24]. Or users need to know some samples that are in adversary model's training data [31].

(ii) If users exploit their own data and train shadow models to perform MIAs, the success accuracy would still be low. This is because adversary could train a much better models by using much more data than a single user. In this case, the training loss of adversary model would be much less than the users shadow

models. As we discussed in the main draft, such performance difference would lead to poor classification accuracy in membership inference.

On the other hand, as neural learner collects data from many others, there are chances that users' data distribution are slightly different (*i.e.*, users watermarking their data in different ways or applying different filters). Under this circumstance, MIAs would possibly perform worse as a single threshold might not work well for heterogeneous users data.

(iii) Last but not the least, even if we assume the verifier obtains such knowledge, the inference results would be unconvincing for an arbitration. As MIAs only produce binary outputs (*i.e.*, True of False), the probability of a correct guess is already 50%. Further, MIAs determine membership by considering data in training must have a small loss. But a small loss of user data can not guarantee user data were used for training. Such a result could possibly be caused by learning similar data rather than the user's.

In summary, MIAs would be less practical than watermarking in personal data protection.

Early studies on MIAs consider watermarking as a special case. In the study [31] on membership inference, they discuss the membership advantage [30] is not necessary if there exists features as a prior knowledge that can be used to distinguish data, (*e.g.* unique id for each image). This follows same idea of watermarking and verification discussed in the paper. However, [31]'s study does not study what watermarking technique can be used to against neural modeling training. They assume the all users can substitute their original data with an identifier via an arbitrary function G [9] and this substitution would not interfere the embedding identifiers. This might work if neural networks can perfectly memorize every detail of training data, including the identifier. However, as shown in our experiments, this is not true. Different users with the same watermarking function would interfere each other heavily, as neural models can learn the data augmentation during training [35]. One contribution of this paper is to show that LCT is an effective anti-neuron watermarking method against neural model training in various realistic settings.

To justify the above discussions, we first propose a new evaluation metric for protection, which quantifies both inference accuracy and signature space, as discussed in the main manuscript. Then we conduct experiments on two PIP cases where users train their own shadow models and different users exploit different watermarking. To perform MIAs, we assume the verifier obtains the necessary knowledge of training distribution, and compare our watermarking method with two state-of-art MIAs: MIA-std [31] and MIA-pow [23]. The first MIA exploits few samples known from model's training and use the average training loss as a threshold to determine the membership. If the loss of given testing sample is smaller than the threshold, the sample would be used for training. The second method requires extra samples known from the held-out set and find the best threshold for both training and held-out samples.

For the experiment, we follow similar settings in [16] for these two attacks. To compare with watermarking fairly, we conduct MIA on average loss of user data,

not data per se, which would increase accuracy of MIAs. In the case when user exploit their own data to perform MIAs, we assume 10 users and each user utilize his/her data to train shadow models and use the knowledge of shadow models to further perform MIAs, respectively; We use MIA-std citeyeom2018privacy as the baseline method and obtain the decision threshold using average training loss from user's shadow models. In the case when users' data distribution are slight different, we randomly split Tiny ImageNet into 1,000 users for training set and 100 users for validation set, with 100 data samples for each user. For MIA-std [31], we randomly select 5 users from training and calculate the average loss as the membership threshold. For MIA-pow [23], we randomly select 5 users from training and 5 user from validation to search for the best threshold. Then we test the inference accuracy on 100 user data samples, with 50 randomly chosen from training and 50 randomly chosen on validation. Note that the testing users would be exclusive from users used for finding threshold.

From Table 7, we show the matching accuracy for MIAs and our method on these testing users. It can be observed that the performance of MIAs will first decrease and then increase as ratio grows. Specifically, when fewer user data are watermarked, the MIAs' threshold would be mainly determined by the unwatermarked users data. Meanwhile, as some user data exploit watermarking, the loss of watermarked data would be lifted, as illustrated in Figure 4, and even if the watermarked samples were used in training, the MIAs would misclassify these samples. The situation would be worst when there are similar amount of watermarking data and unwatermarking data. Such a result show that MIAs would be unstable when user data are heterogeneous.

Ratio	0%	20%	40%	60%	80%	100%
MIA-std	77.8	74.6	67.3	72.2	69.5	79.0
MIA-pow	92.7	84.8	77.6	80.3	76.5	94.0
ANW	-	96.0	95.2	92.6	86.5	82.0

Table 7. **The matching accuracy between membership inference attacks and anti-neuron watermarking**. Different ratio of users exploit different LCT for their data. In MIAs, a match would be calculated by binary inference result, but in ANW, a match would imply the correct signature, which would be more difficult because there are 12 signature values in this experiment.

Comparing to Dataset Tracing. Recent studies [18, 22, 36] on dataset protection investigate watermarking against neural training and it is tempting to apply these methods in personal data protection. However, as we discussed in the main manuscript, dataset tracing [18, 22, 36] requires full knowledge of training data, so that they can generate "watermarked data" by exploiting pretrained classifier on the dataset as adversarial training [19]. However, from users' perspective, such prior knowledge would not be sufficient because it is impossible for a common user to know how other data will be collected and what tasks will

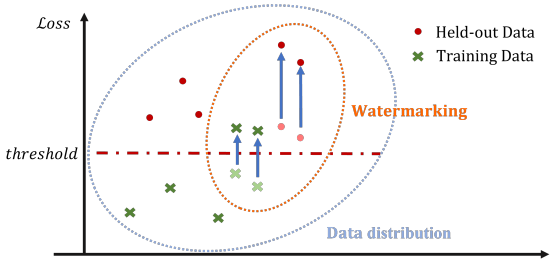


Fig. 4. **Illustration of why MIAs would decrease performance when partial users exploit watermarking.** As watermarking function lifts loss of user data, the original MIA threshold calculated by unwatermarked data would misclassify the membership of watermarked data and thus degrades performance. The experiment result for memorization in the main draft verifies the above idea.

be performed. As a result, techniques like dataset tracing [18, 22, 36] can not be directly applied to the PIP scenario.

As dataset tracing [22] can be applied on partial data, we compare our method with dataset tracing [22] and verify whether both methods can work well with very limited watermarking data, which is common in the PIP problem. To make dataset tracing work, we pre-train a ResNet50 model on Tiny ImageNet and adopt the same setting as [22] to generate "radioactive data". As shown in the main manuscript, with 0.1% data being watermarked, our LCT method is more effective as we observe that watermarking signature can be memorized well as expected.

H Memorization Analysis of Watermarking

When Signature is being Memorized during Neural Model Training? One interesting problem for watermarking is when a signature is memorized by neural models. From previous studies on MIAs [16, 24, 26, 31], user privacy information is being leak when model's overfitting. As a result, it is likely for models to memorize watermarking signatures when models over-learn the watermarked data [26]. To explore the answer of this question, we infer signatures by grid search for different checkpoints of model during training. In Figure 5, we can observe the inferred signature reaches 0 at early stage of training and gradually reach 60 with more learning epochs. This result illustrates that the watermarking signature could be memorized before the end of training. This empirical results can also explain why data augmentation (watermarking) can be learned by neural models during training, according to the study [35] for neural generalization.

Watermarking Increase Memorization of User Data. In the main draft, we mention how to use "memorization value estimate" [7] to study how watermarking improve memorization of user data. As user data is watermarked,

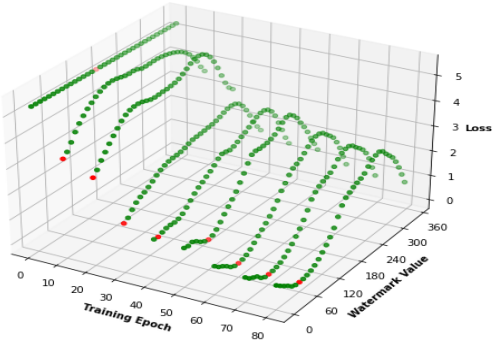


Fig. 5. The watermarking signature can be memorized before the end of training. The **red point** indicates inferred signature that achieves minimal loss over other signatures.

the probability density of user data would be lifted into low density region and thus being easier memorized by neural learners. During training process, the watermarking signature would be better memorized along with user data. According to the study [7] on memorization, for training algorithm A on a dataset $S = ((x_1, y_1), ..., (x_n, y_n))$, the amount of label memorization by $\mathcal{A}$ on example $(x_i, y_i) \in S$ is defined as,

$$mem(\mathcal{A}, S, i) := \mathop{\mathbb{P}_r}_{h \leftarrow \mathcal{A}(S)}[h(x_i) = y_i] - \mathop{\mathbb{P}_r}_{h \leftarrow \mathcal{A}(S^{\setminus i})}[h(x_i) = y_i]. \tag{2}$$

To estimate above memorization value on sample index at i, [7] firstly selects random subsets for S, with some subsets including sample i and some exclude sample i. Then, K models are being trained using these subsets, grouped into 2, one includes sample i and the other excludes sample i. The memorization value estimate (MAE) is finally calculate by averaging the difference of $\mathbb{P}_r$ between these two groups of models. In our experiments, we split the training set of Tiny ImageNet into 1,000 users and choose one split as user data (we calculate MAE on a collection user data instead of one sample). Then we randomly use 70% of users data to construct 20 subsets. 20 models are being trained correspondingly to calculate the final MAE. We fix the indices for this experiments and train 20 models with and without watermarking on user data. From the results we show in the main manuscript, the MAE of user data increase after watermarking, indicating that watermarking improves model memorization ability on the given user data.

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. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
6. Dwork, C.: Differential privacy: A survey of results. In: International conference on theory and applications of models of computation. pp. 1–19. Springer (2008)
7. Feldman, V., Zhang, C.: What neural networks memorize and why: Discovering the long tail via influence estimation. arXiv preprint arXiv:2008.03703 (2020)
8. 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)
9. Garay, J.A., Gennaro, R.: Advances in Cryptology–CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II, vol. 8617. Springer (2014)
10. 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/
11. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
12. 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)
13. 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)
14. Jia, J., Salem, A., Backes, M., Zhang, Y., Gong, N.Z.: Memguard: Defending against black-box membership inference attacks via adversarial examples. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 259–274 (2019)
15. kaggle: (2017), https://www.kaggle.com/c/tiny-imagenet, tiny Imagenet
16. Kaya, Y., Dumitras, T.: When does data augmentation help with membership inference attacks? In: International Conference on Machine Learning. pp. 5345–5355. PMLR (2021)
17. 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)
18. 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)
19. 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
20. Müller, R., Kornblith, S., Hinton, G.: When does label smoothing help? arXiv preprint arXiv:1906.02629 (2019)

21. Nasr, M., Shokri, R., Houmansadr, A.: Machine learning with membership privacy using adversarial regularization. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 634–646 (2018)
22. Sablayrolles, A., Douze, M., Schmid, C., Jégou, H.: Radioactive data: tracing through training. In: International Conference on Machine Learning. pp. 8326–8335. PMLR (2020)
23. 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)
24. 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)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
26. Song, C., Shmatikov, V.: Overlearning reveals sensitive attributes. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=SJeNz04tDS
27. Standing Committee of the National People's Congress: China data security law (Jul 2021), http://www.xinhuanet.com/2021-06/11/c_1127552204.htm
28. 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)
29. 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)
30. Yeom, S., Fredrikson, M., Jha, S.: The unintended consequences of overfitting: Training data inference attacks. arXiv preprint arXiv:1709.01604 **12** (2017)
31. 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)
32. 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)
33. 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)
34. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
35. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Communications of the ACM **64**(3), 107–115 (2021)
36. 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)