Leveraging Unlabeled Data for Watermark Removal of Deep Neural Networks

Xinyun Chen [*1] **Wenxiao Wang** [*2] **Yiming Ding** [1] **Chris Bender** [1] **Ruoxi Jia** [1] **Bo Li** [3] **Dawn Song** [1]

Abstract

Deep neural networks have achieved tremendous success in various fields; however, training these models from scratch could be computationally expensive and requires a lot of training data. Therefore, recent work has explored different watermarking techniques to protect the pre-trained deep neural networks from potential copyright infringements. Although several existing techniques could effectively embed such watermarks into the DNNs, they could be vulnerable to adversaries who aim at removing the watermarks. In this work, we demonstrate that a carefully-designed fine-tuning method enables the adversary with limited training data to effectively remove the watermarks, without compromising the model functionality. In particular, leveraging auxiliary unlabeled data significantly decreases the amount of labeled training data needed for effective watermark removal, even if the unlabeled data samples are not drawn from the same distribution as the benign data for model evaluation.

1. Introduction

Deep neural networks have been prevalent in our lives due to the high performance in various applications. Typically, training these models from scratch is computationally intensive and requires the access to a large set of high-quality training samples. Therefore, people may resort to pre-trained models, which opens up the market of Machine Learning as a Service (MLaaS).

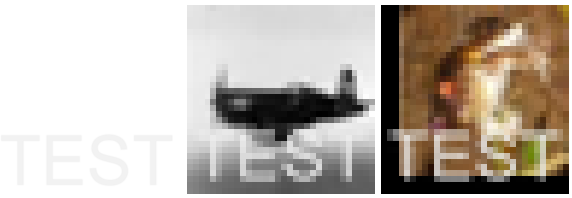
To claim the ownership of the pre-trained model, so that it is not illegally used or stolen, recent work propose watermarking techniques to protect the models from potential copyright infringements (Adi et al., 2018; Zhang et al., 2018; Rouhani et al., 2018; Uchida et al., 2017). A common paradigm is to add an additional training objective for injecting watermarks besides optimizing the prediction accuracy of the model; for example, the model owner could inject some specially-designed training samples, so that the model would predict in the ways specified by the owner when provided with the watermark samples.

In this work, we study the effectiveness of fine-tuning based watermark removal techniques. To effectively remove the watermarks while preserving the model performance, fine-tuning based approaches often require the adversary to have a sufficient amount of labeled data drawn from the same distribution as the data used for evaluation. While a large amount of labeled data could be expensive to collect, unlabeled data is much cheaper to obtain; e.g., the adversary can simply download as many images as he wants from the Internet. By leveraging such inherently unbounded provision of unlabeled samples, the adversary might be able to unlock the possibility of watermark removal with limited in-distribution labeled data. In this work, we propose to utilize the pre-trained model to annotate the unlabeled samples, and augment the fine-tuning training data with them.

We focus on watermark removal of deep neural networks for image recognition in our evaluation, where existing watermarking techniques are shown to be the most effective. We evaluate our fine-tuning based techniques on two classes of watermarking techniques, i.e., pattern-based techniques (Zhang et al., 2018; Chen et al., 2017; Gu et al., 2017; Liu et al., 2017a;b) and instance-based techniques (Adi et al., 2018; Chen et al., 2017). Specifically, instance-based techniques inject individual training samples as the watermarks, while pattern-based techniques inject a set of samples blended with the same pattern as the watermarks. We first demonstrate that by carefully designing the fine-tuning learning rate schedule, the adversary is always able to remove the watermarks. Furthermore, by utilizing the unlabeled data, we significantly decrease the amount of in-distribution labeled samples required for effective watermark removal, even if the unlabeled samples are out-of-distribution themselves. Our work provides the first successful demonstration of watermark removal techniques against different watermark embedding schemes, and sheds some light on the potential vulnerability of existing watermarking techniques against adversaries who are capable of performing fine-tuning with limited training data.

[*]Equal contribution [1]University of California, Berkeley [2]Tsinghua University [3]University of Illinois at Urbana-Champaign. Correspondence to: Xinyun Chen <xinyun.chen@berkeley.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.



(a) (b) (c)

Figure 1. Examples of watermarks generated by the pattern-based technique in (Zhang et al., 2018). Specifically, after an image is blended with the "TEST" pattern in (a), such an image is classified as the target label, e.g., an "automobile" on CIFAR-10.

(a) (b)

Figure 2. Examples of watermarks generated by the instance-based technique in (Adi et al., 2018). Different watermarking images could have different assigned labels.

2. Watermarking for Deep Neural Networks

In this work, we study the watermarking problem as follows. A model owner trains a model f_θ for a certain task $\mathcal{T}$. Besides training on a dataset drawn from the data distribution of $\mathcal{T}$, the owner also embeds a set of watermarks $\mathcal{K} = \{(x^k, y^k)\}_{k=1}^K$ into f_θ. A watermarking scheme should at least satisfy two properties: (1) functionality-preserving, i.e., embedding these watermarks does not degrade the model performance on $\mathcal{T}$; (2) verifiability, i.e., $Pr(f_\theta(x^k) = y^k) \gg Pr(f'(x^k) = y^k)$ for $(x^k, y^k) \in \mathcal{K}$, where f' is any other model that is not trained with the purpose of embedding the same set of watermarks.

2.1. Watermarking Techniques

Recent work propose several different watermarking techniques to this end. In this work, we focus on pattern-based techniques (Zhang et al., 2018; Chen et al., 2017; Gu et al., 2017) and instance-based techniques (Adi et al., 2018; Chen et al., 2017), which are studied the most in the literature.

Pattern-based techniques. A pattern-based technique specifies a key pattern key and a target label y^t, so that for any image x blended with the pattern key, $Pr(f_\theta(x) = y^t)$ is high. To achieve this, the owner generates a set of images $\{x^k\}_{k=1}^K$ blended with key, assigns $y^k = y^t (k \in 1, ..., K)$, then adds $\{(x^k, y^k)\}_{k=1}^K$ into the training set. Figure 1 shows an example of watermarks generated by pattern-based techniques. Pattern-based techniques are also used for embedding backdoors into the pre-trained model (Chen et al., 2017; Gu et al., 2017; Liu et al., 2017a)

Instance-based techniques. For instance-based techniques, different watermarks are generated individually, and their labels could also be different. Figure 2 presents some watermarks used in (Adi et al., 2018), where each watermarking image and its label is randomly sampled.

2.2. Threat Model for Watermark Removal

In this work, we assume the following threat model for the adversary who aims at removing the watermarks.

No knowledge of the watermarks. Some prior work on detecting samples generated by pattern-based techniques requires the access to the entire data for pre-training, including the watermarks (Tran et al., 2018; Chen et al., 2018). In contrast, we do not assume the access to watermarks for pre-training.

No knowledge of the watermarking scheme. Most prior work on watermark removal relies on the assumption that the watermarks are pattern-based (Wang et al., 2019; Gao et al., 2019). In this work, we study fine-tuning as a generic approach for watermark removal, without the knowledge of the watermarking scheme.

Limited data for fine-tuning. We assume that the adversary has computation resources for fine-tuning, and this assumption is also made in previous work studying fine-tuning for watermark removal (Adi et al., 2018; Zhang et al., 2018; Liu et al., 2018). Different from most prior work where the adversary has the same benign data for the task $\mathcal{T}$ as the model owner, we study the scenarios where the adversary has a much smaller training set for fine-tuning.

3. Fine-tuning for Watermark Removal

In this work, we focus on fine-tuning based approaches for watermark removal. Specifically, the adversary further trains the model with his own data during the fine-tuning process, and according to catastrophic forgetting phenomenon (Goodfellow et al., 2013; Kemker et al., 2018), since the fine-tuning data no longer includes the watermark samples, the model should forget the previously learned watermark behavior. In the following, we first propose an adaption of existing fine-tuning techniques to improve the efficacy of watermark removal, referred to as *Basic Fine-Tuning (FT-Basic)*. In cases where the adversary has limited fine-tuning data, we further propose data augmentation with unlabeled data, referred to as *Fine-tuning with Augmentation of Unlabeled data (FTAU)*, which significantly decreases the amount of in-distribution labeled training samples required for obtaining a high-accuracy model without watermarks.

Basic Fine-tuning (FT-Basic). Contrary to this intuition, some prior work show that existing watermarking tech-

niques are robust against fine-tuning based techniques, even if the adversary fine-tunes the entire model and has access to the same benign data as the owner, i.e., the entire data for pre-training excluding the watermarking keys (Adi et al., 2018; Zhang et al., 2018; Liu et al., 2018). We find that the key reason may be that the learning rate for fine-tuning is too small to change the model weights. In Section 4, we will demonstrate that by simply increasing the initial learning rate for fine-tuning the entire model and properly decaying the learning rate, the adversary is able to remove the watermarks without compromising the model performance on his task. We refer to this basic fine-tuning method as *FT-Basic*.

Fine-tuning with Augmentation of Unlabeled data (FTAU). Although the above fine-tuning method already enables the adversary to remove the watermarks, it requires the adversary to have some labeled training samples to start with. However, usually the adversary does not have comparable amount of training data to the model owner, and in our evaluation, we find that by fine-tuning with only the limited labeled data, removing watermarks may cause large degradation of model accuracy on his task.

To overcome this challenge, we propose to augment the fine-tuning data with unlabeled samples, which could easily be collected from the Internet. Let $\mathcal{U} = \{x_u\}_{u=1}^{U}$ be the unlabeled sample set. Then we use the pre-trained model as the labeling tool, i.e., $y_u = f_\theta(x_u)$ for each $x_u \in \mathcal{U}$. Note that since the accuracy of pre-trained model is not 100% itself, such label annotation is inherently noisy; in particular, when $\mathcal{U}$ is drawn from a different distribution than the task of consideration, the assigned labels may not be meaningful at all. Nevertheless, in Section 4, we will show that leveraging unlabeled data significantly decreases the in-distribution labeled samples needed for effective watermark removal.

4. Evaluation

4.1. Evaluation Setup

We evaluate our fine-tuning approaches on CIFAR-10, CIFAR-100 and STL-10. Our evaluation of pattern-based techniques uses the text pattern in (Zhang et al., 2018) (see Figure 1), and our evaluation of instance-based techniques is based on (Adi et al., 2018) (See Figure 2). In particular, we consider the following settings:

- The model is pre-trained to perform the same task as what adversary desires. We conduct experiments on both CIFAR-10 and CIFAR-100. The unlabeled data is obtained from the unlabeled part of STL-10, which includes 100,000 samples. Note that STL-10 images are very different from CIFAR-10 and CIFAR-100; in particular, the label sets between CIFAR-100 and STL-10 barely overlap.
- The model is pre-trained on a different task from what adversary desires. Note that the labeled part of STL-10 only includes 5,000 samples, which is insufficient for training a model with a high accuracy. Therefore, the adversary can leverage the pre-trained model on another task with a larger training set, then fine-tune the model on STL-10. This fine-tuning method is widely adopted for transfer learning (Yosinski et al., 2014), and is also evaluated in (Adi et al., 2018). In particular, we perform the transfer learning to adapt from a pre-trained CIFAR-10 model to STL-10, because we find that adapting from CIFAR-100 model achieves much worse results than from CIFAR-10, i.e., the accuracy is around 5% lower, as in (Adi et al., 2018). We also utilize the unlabeled part of STL-10 for data augmentation in this setup.

We evaluate both FT-Basic and FTAU on the above setups. We also compare with a baseline method that trains the entire model from scratch without leveraging the pre-trained model, denoted as *FS*. For any model, we consider the watermarks are removed when the watermark accuracy is similar to the model trained without watermarks. Specifically, we consider the watermarks are removed when the watermark accuracy is below 20% for models pre-trained on CIFAR-10, and below 10% for models trained on CIFAR-100.

For both FT-Basic and FTAU, we fine-tune the entire model, as we find that fine-tuning only the output layer is insufficient for watermark removal, as demonstrated in (Adi et al., 2018). We find that both the convolutional neural network architecture and pre-training schedule do not have critical influence on the effectiveness of watermark embedding and removal, as long as the pre-trained model achieves a high test accuracy and fits the watermarks well. Thus, we follow the same pre-training configuration in (Adi et al., 2018). In particular, for all experiments, we use the ResNet-18 model (He et al., 2016), and the initial learning rate is 0.1.

As discussed in Section 3, the failure of previous attempts of fine-tuning based watermark removal mainly due to the improper design of learning rate schedule during the fine-tuning stage. For example, the initial learning rate for fine-tuning is 0.001 in (Adi et al., 2018), which is $100\times$ smaller than the initial learning rate for pre-training. For all experiments, we set the initial fine-tuning learning rate to be 0.05, and decay it by 0.9 after every 500 timesteps. More discussion on implementation details can be found in Appendix A.

4.2. Results

We first present the results of transfer learning in Table 1. Following (Adi et al., 2018), we evaluate the watermark accuracy on the fine-tuned model by replacing its output layer with the original output layer of the pre-trained model. For comparison of the STL-10 test accuracy, we also fine-tune the pre-trained model with a learning rate of 0.001, so

that its watermark accuracy remains above 70%, as in (Adi et al., 2018). We observe that with the FT-Basic method, removing watermarks already does not affect the model performance on the testset, and using FTAU further improves the prediction accuracy.

Next, we present results of watermark removal in the non-transfer learning setting for pattern-based techniques in Table 2. First, we observe that when the adversary has 80% of the entire training set, using FT-Basic already achieves a higher test accuracy than the pre-trained model, while removing the watermarks. Note that the watermark accuracies are still above 95% using the fine-tuning approaches in previous work (Adi et al., 2018; Zhang et al., 2018), suggesting the effectiveness of our modification.

However, when the adversary only has a small proportion of labeled training set, the test accuracy could degrade. Although the test accuracy only drops for about 1% on CIFAR-10 even if the adversary has only 20% of the entire training set, the accuracy decrease could be up to 5% on CIFAR-100. By leveraging the unlabeled data, the adversary is able to achieve the same level of test performance as the pre-trained model with only 20% $\sim$ 30% of the entire training set. Furthermore, FTAU enables the adversary to fine-tune the model without any labeled training data, and by solely relying on the unlabeled data, the accuracy of the fine-tuned model achieves within 1% difference from the pre-trained model on CIFAR-10, and the accuracy of fine-tuned CIFAR-100 model nearly reaches the performance of the model trained with 80% CIFAR-100 data from scratch. Note that the unlabeled STL-10 data samples are drawn from very different distribution; in particular, the label sets of CIFAR-100 and STL-10 barely overlap. These results show that our approach is effective without the requirement that the unlabeled data comes from the same distribution, which makes it a simple watermark removal for the adversary, thus poses threats to the robustness of pattern-based watermarks.

Finally, we present more results of watermark removal for instance-based techniques in Table 3. Compared to pattern-based techniques, removing instance-based watermarks could result in a larger decrease of test accuracy, indicating that while pattern-based watermarks are more often used, they are easier to remove than instance-based watermarks. However, still, our FT-Basic method is more effective than previous fine-tuning attempts, and FTAU enables the adversary to obtain a model with high accuracy despite having limited labeled data.

5. Conclusion

In this work, we study the robustness of watermarking techniques, and demonstrate that fine-tuning based approaches can successfully remove watermarks even if the adversary does not have access to the full training data. Moreover, leveraging unlabeled data further reduces the amount of labeled data required for effective watermark removal. Our study highlights the vulnerability of existing watermarking techniques, and we consider proposing more robust watermarking techniques as future work.

Watermarking Technique	FS	FT-Basic	FTAU
Pattern-based	66.15%	82.96%	**83.80%**
Instance-based		82.83%	**83.51%**

Table 1. Results of models after watermark removal in the transfer learning setting. The numbers show the accuracy on the STL-10 testset. The accuracy of fine-tuned model with pattern-based watermarks on STL-10 is 82.06%, and the accuracy of model with instance-based watermarks is 82.89%.

Dataset	Percentage	FS	FT-Basic	FTAU
CIFAR-10	0%	–	–	**92.53%**
	20%	87.40%	92.12%	**92.80%**
	30%	89.64%	92.22%	**93.15%**
	40%	90.46%	92.93%	**93.18%**
	50%	91.45%	93.08%	**93.18%**
	80%	93.01%	93.52%	**94.11%**
CIFAR-100	0%	–	–	**70.75%**
	20%	56.72%	68.88%	**71.97%**
	30%	62.20%	71.05%	**72.98%**
	40%	65.42%	71.96%	**73.44%**
	50%	68.18%	72.58%	**73.72%**
	80%	71.71%	74.23%	**75.42%**

Table 2. Results of watermark removal against pattern-based techniques for non-transfer learning setting. The first column is the dataset for model evaluation, the second column is the percentage of labeled data for fine-tuning compared to the entire benign training set, and the rest columns show the accuracy on the testset. The accuracy of the pre-trained model with watermarks on CIFAR-10 is 93.23%, and the accuracy of the pre-trained CIFAR-100 model with watermarks is 73.83%.

Dataset	Percentage	FS	FT-Basic	FTAU
CIFAR-10	0%	–	–	**90.48%**
	20%	87.40%	91.19%	**92.41%**
	30%	89.64%	91.58%	**93.01%**
	40%	90.46%	92.76%	**93.21%**
	50%	91.45%	92.97%	**93.21%**
	80%	93.01%	93.93%	**94.00%**
CIFAR-100	0%	–	–	**66.69%**
	20%	56.72%	66.14%	**71.12%**
	30%	62.20%	68.70%	**71.82%**
	40%	65.42%	70.21%	**72.20%**
	50%	68.18%	71.20%	**72.60%**
	80%	71.71%	73.30%	**74.16%**

Table 3. Results of watermark removal against instance-based techniques for non-transfer learning setting. The accuracy of the pre-trained model with watermarks on CIFAR-10 is 93.63%, and the pre-trained CIFAR-100 model with watermarks is 73.06%.

References

Adi, Y., Baum, C., Cisse, M., Pinkas, B., and Keshet, J. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pp. 1615–1631, 2018.

Chen, B., Carvalho, W., Baracaldo, N., Ludwig, H., Edwards, B., Lee, T., Molloy, I., and Srivastava, B. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D. C., and Nepal, S. Strip: A defence against trojan attacks on deep neural networks. *arXiv preprint arXiv:1902.06531*, 2019.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Gu, T., Dolan-Gavitt, B., and Garg, S. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kemker, R., McClure, M., Abitino, A., Hayes, T. L., and Kanan, C. Measuring catastrophic forgetting in neural networks. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Liu, K., Dolan-Gavitt, B., and Garg, S. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pp. 273–294. Springer, 2018.

Liu, Y., Ma, S., Aafer, Y., Lee, W.-C., Zhai, J., Wang, W., and Zhang, X. Trojaning attack on neural networks. In *NDSS*, 2017a.

Liu, Y., Xie, Y., and Srivastava, A. Neural trojans. In *The 35th IEEE International Conference on Computer Design*, 2017b.

Rouhani, B. D., Chen, H., and Koushanfar, F. Deepsigns: A generic watermarking framework for ip protection of deep learning models. *arXiv preprint arXiv:1804.00750*, 2018.

Tran, B., Li, J., and Madry, A. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems*, pp. 8000–8010, 2018.

Uchida, Y., Nagai, Y., Sakazawa, S., and Satoh, S. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pp. 269–277. ACM, 2017.

Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., and Zhao, B. Y. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE Symposium on Security and Privacy*, 2019.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pp. 3320–3328, 2014.

Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M. P., Huang, H., and Molloy, I. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pp. 159–172. ACM, 2018.

A. More Discussion on Experimental Details

Our implementation is in PyTorch [1]. We used SGD as the optimizer, and set the batch size to be 100 for both pre-training and FT-Basic, following the setup in (Adi et al., 2018). For FTAU, when the proportion of labeled samples is greater than 0, each batch has 100 labeled samples and 50 samples from the unlabeled STL-10, so the batch size becomes 150. When there is no in-distribution labeled samples, each batch includes 100 unlabeled STL-10 samples.

We also consider the fine-pruning method proposed in (Liu et al., 2018). This paper proposes to first prune part of the neurons that are activated the least for benign samples, and then perform the fine-tuning. We evaluate their approach with the same fine-tuning learning rate schedule as our proposed variants, and find that the results are roughly the same for all our experimental setups, suggesting that pruning is not necessary with a properly designed learning rate schedule for fine-tuning. Therefore, we omit the fine-pruning results in our comparison.

[1]The implementation is mainly adapted from https://github.com/adiyoss/WatermarkNN, the code repo of (Adi et al., 2018).