FedIPR: Ownership Verification for Federated Deep Neural Network Models

Lixin Fan[1] , **Bowen Li**[2] , **Hanlin Gu**[4] , **Yan Kang** [1] , **Jie Li**[2] and **Qiang Yang**[3*]
[1] AI Group, WeBank Co., Ltd, Shenzhen, China
[2] Department of CSE, Shanghai Jiao Tong University, Shanghai, China
[3] Department of CSE, Hong Kong University of Science and Technology, Hong Kong, China
[4] Department of Mathematics, Hong Kong University of Science and Technology, Hong Kong, China
{lixinfan, yangkang}@webank.com, {li-bowen, lijiecs}@sjtu.edu.cn, hguaf@connect.ust.hk,
qyang@cse.ust.hk

Abstract

Federated learning models must be protected against plagiarism since these models are built upon valuable training data owned by multiple institutions or people. This paper illustrates a novel federated deep neural network (FedDNN) ownership verification scheme that allows ownership signatures to be embedded and verified to claim legitimate intellectual property rights (IPR) of FedDNN models, in case that models are illegally copied, re-distributed or misused. The effectiveness of embedded ownership signatures is theoretically justified by proved conditions under which signatures can be embedded and detected by multiple clients without disclosing private signatures. Extensive experimental results on CIFAR10, CIFAR100 image datasets demonstrate that varying bit-lengths signatures can be embedded and reliably detected without affecting models classification performances. Signatures are also robust against removal attacks including fine-tuning and pruning.

1 Introduction

Federated learning (FL) is a machine learning setting in which many clients collaboratively train a model, and simultaneously, mitigate privacy risks and costs by keeping the training data decentralized [McMahan *et al.*, 2017; Yang *et al.*, 2019; Kairouz *et al.*, 2019]. While preserving data privacy is of the paramount importance, it is also considered a critical issue to prevent adversaries from stealing and misusing models to search for model vulnerabilities [Kairouz *et al.*, 2019]. Moreover, protecting models from being stolen is motivated by the fact that FL models are built upon valuable data owned by multiple clients, and plagiarism of such models must be stopped. This paper illustrates a novel federated deep neural network (FedDNN) ownership verification scheme that can be used to claim legitimate intellectual property rights (IPR) of FedDNN models, in case that models are illegally copied, re-distributed or misused by unauthorized parties. The proposed scheme not only verifies model ownership against external plagiarisms, but also allows each client to claim contributions to the federated model as verified data owners.

DNN watermarking techniques have been proposed to protect DNN Intellectual Property Rights (IPR) [Uchida *et al.*, 2017; Chen *et al.*, 2018; Darvish Rouhani *et al.*, 2018; Adi *et al.*, 2018; Zhang *et al.*, 2018; Fan *et al.*, 2019; Ong *et al.*, 2021; Zhang *et al.*, 2020; Boenisch, 2020], however, it remains an open question concerning whether existing methods are applicable to *federated learning* settings, in which following technical challenges must be properly addressed. First, a FL client must embed private signatures into DNN models, yet, without disclosing to other parties the presences and extraction parameters of such signatures. Second, when an increasingly large number of signatures are embedded the global DNN model must be able to accommodate private signatures assigned by different clients without compromising model performances of the *main task*[1]. Third, the verification of clients' private signatures must also be kept secret. In a nutshell, a FedDNN ownership protection scheme should entail three capabilities i.e. *maintaining main task model performances*, *preserving signature privacy* and *avoiding conflicts between multi-client signatures*.

This paper presents a general FedIPR framework which demonstrates a number of successful schemes that can be used for different federated learning scenarios and signature verification protection modes. Specifically, theoretical analysis in Proposition 1 of Sect. 2.2 elucidates conditions under which *reliable* and *persistent* signatures can be successfully embedded into the same FedDNN model by multiple clients. Extensive experiments in Sect. 3 demonstrate that varying bit-lengths signatures using normalization scale parameters are very persistent in white-box verification mode, while trigger set of backdoor samples can be reliably detected as strong evidence to support claims of legitimate model ownership. To our best knowledge, the FedIPR framework is the first technical solution that supports the protection of DNN ownerships in a federated learning setting. We believe this work will open new avenues for research endeavor to protect Intellectual Property Right of federated learning models.

*Corresponding Author

[1] The original task for which the federated DNN model is built.

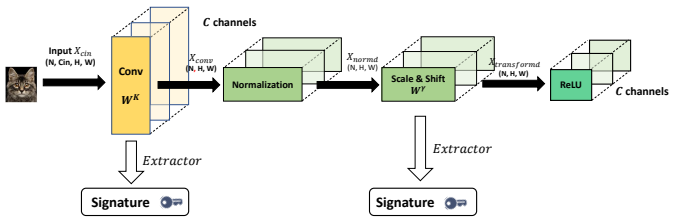

Figure 1: Convolution layer weights (in yellow) and normalization layer weights (in green) used to embed signatures that are to be extracted in white-box manner (see Sect. 2.1 for texts).

1.1 Related Work

Privacy-preserving deep learning (PPDL) aims to collaboratively train a deep neural network (DNN) model among multiple clients without exposing private training data to each other [Shokri and Shmatikov, 2015; Abadi *et al.*, 2016; Phong *et al.*, 2018; Ryffel *et al.*, 2020].

Backdoor attacks is a security threat that have been extensively studied in federated learning [Bagdasaryan *et al.*, 2018; Sun *et al.*, 2019; Bhagoji *et al.*, 2019; Wu *et al.*, 2020]. Following [Adi *et al.*, 2018], we adopt targeted backdoor samples [Sun *et al.*, 2019; Bagdasaryan and Shmatikov, 2020] as signatures for *black-box* ownership verification. We show that robust backdoor signatures can provide evidence of suspected plagiarism without accessing to internal parameters of models.

DNN ownership embedding and verification approaches can be broadly categorized into two schools: a) the *feature-based* methods that embed designated signatures [Uchida *et al.*, 2017; Chen *et al.*, 2018; Darvish Rouhani *et al.*, 2018; Fan *et al.*, 2019; Zhang *et al.*, 2020]; and b) the *trigger-set* based methods that rely on backdoor training samples with specific labels [Adi *et al.*, 2018; Zhang *et al.*, 2018]. While feature-based methods allow persistent signatures to be reliably detected even under various forms of removal attacks, yet, they must access DNN internal parameters to detect signatures. The benefit of trigger-set based method is that model owners can collect evidence of suspected plagiarism through remote API without accessing to internal parameters of models in question. Interestingly, [Ong *et al.*, 2021] illustrated a black-box and white-box verification method for GAN instead of convolution networks.

2 Federated DNN Ownership Verification

This section first reviews existing DNN signature embedding and verification scheme, followed by illustration of the Federated DNN (FedDNN) framework. In this paper we use *signature* and *watermark* interchangeably. There are broadly two categories of DNN signature embedding and verification methods: *feature-based* vs *trigger-set-based*. Feature-based signatures are embedded into network parameters and have to be verified by accessing network parameters i.e. in *white-box* manner. Trigger-set based signatures are embedded into network outputs or labels, and can be verified without accessing network parameters i.e. in *black-box* manner.

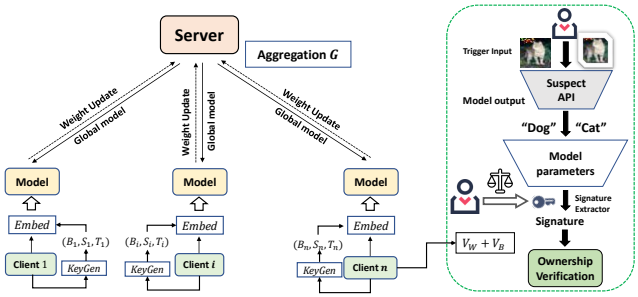

Figure 2: An illustration of federated DNN (FedDNN) signature Embedding and Verification scheme. Private signatures are generated and embedded into the local models which are then aggregated using the `FederatedAveraging` algo. (the left panel). In case that the federated model is plagiarized, each client may invoke verification processes to extract signatures from the plagiarized model in both black-box and white-box manner to claim his/her ownership of the federated model (the right panel).

2.1 Review of DNN Signature Verification

For existing DNN feature-based signature embedding methods [Uchida *et al.*, 2017; Chen *et al.*, 2018; Darvish Rouhani *et al.*, 2018; Zhang *et al.*, 2018; Fan *et al.*, 2019], N-bits target binary signatures $\mathbf{B} = (t_1, \cdots, t_N) \in \{0,1\}^N$ are embedded during the learning of parameters $\mathbf{W}$ of a DNN model $\mathbb{N}[\mathbf{W}]$, by adding regularization terms R to the loss of the original learning task $L = L_D + \alpha R$:

$$R_{\mathbf{B},\theta}(\mathbf{W}) = Dist\big(\mathbf{B}, \mathsf{B}_\theta(\mathbf{W})\big), \tag{1}$$

in which $\theta = \{\mathcal{S}, \mathbf{E}\}$ is a set of hyper-parameters used to extract signature vector

$$\mathsf{B} = (\mathbf{W}^T\mathbf{E}) \in \mathfrak{R}^N \tag{2}$$

whereas $\mathbf{W}^T = \mathcal{S}(\mathbf{W})) \in \mathfrak{R}^M$ denotes a M-dimensional columnized vector of *subset* of DNN parameters and $\mathbf{E} \in \mathfrak{R}^{M\times N}$ a pre-determined signature extraction matrix.

Regularization term (1) restricts DNN model parameter $\mathbf{W}$ to be distributed within a subspace (see Proposition 1) such that binary strings extracted from DNN parameters $\hat{\mathbf{B}} = \text{sign}(\mathsf{B}) = (b_1, \cdots, b_N) \in \{0,1\}^N$ is similar to target signature $\mathbf{B}$[2]. The signature is then successfully verified by a verification process V if Hamming distance $\mathrm{H}(,)$ is less than a preset radius:

$$V\big(\mathbf{W}, (\mathbf{B}, \theta)\big) = \begin{cases} \text{TRUE}, & \text{if } \mathrm{H}(\mathbf{B}, \hat{\mathbf{B}}) \le \epsilon_H, \\ \text{FALSE}, & \text{otherwise.} \end{cases} \tag{3}$$

Note that the formulation in (1) is a generalization of regularization terms used in representative DNN signature embedding methods (see Figure 1). For instance, Uchida et al. [Uchida *et al.*, 2017] proposed to embed signatures into convolution layer weights i.e. $\mathcal{S}(\mathbf{W})$ is the columnized vector of *convolution layer weights*, and matrix $\mathbf{E}$ is predetermined privately . $Dist(,)$ in Uchita et al. [Uchida *et al.*, 2017] measures binary cross-entropy $\mathrm{BCE}\Big(\mathbf{B}, \mathsf{B}\Big) = -\sum_{j=1}^N t_j \log(f_j) + (1 - t_j)\log(1 - f_j)$; where $f_j = \frac{1}{1+\exp(-b_j)}$.

[2] An analysis of the existence of solutions to the system of constraining inequalities is given in Appendix B.

Fan et al. [Fan *et al.*, 2019] proposed to embed signatures into *normalization layer scale parameters* i.e. $\mathcal{S}(\mathbf{W}) = \mathbf{W}_\gamma = \{\gamma_1, \cdots, \gamma_C\}$ where C is the number of normalization filters, and matrix $\mathbf{E}$ is $I_{C\times C}$ identify matrix. Instead, $Dist(,)$ in Fan et al. [Fan *et al.*, 2019], is the hinge loss $\text{HL}\Big(\mathbf{B}, \mathsf{B}\Big) = \sum_{j=1}^{N} \max(\alpha - b_j t_j, 0)$.

For trigger-set based methods, Adi et al. [Adi *et al.*, 2018] first proposed to embed backdoor trigger-set samples $\mathbf{T} = \{(\mathbf{X}_1, \mathbf{Y}_1), \cdots, (\mathbf{X}_J, \mathbf{Y}_J)\}$ by incorporating cross-entropy loss of backdoor samples namely,

$$L_T(\mathbf{W}) = \text{CE}(\mathbf{Y}, \mathbb{N}(\mathbf{T})) = -\sum_{j=1}^{J} \mathbf{Y}_j \log(\mathbb{N}(\mathbf{X}_j)), \quad (4)$$

in which $\mathbf{X}_j$ are backdoor samples, $\mathbf{Y}_j$ corresponding *one-hot* encoding vector of backdoor labels and $\mathbb{N}(\mathbf{X}_j)$ the network softmax outputs.

2.2 FedIPR: FedDNN Signature Embedding and Verification

A federated learning system consists of K client participants building local models with their own data D_n and send local models to a server-side aggregator for secure model aggregation [McMahan *et al.*, 2017; Yang *et al.*, 2019; Kairouz *et al.*, 2019]. It is often assumed the aggregator and other participants are honest-but-curious and thus no leakage of information from participants is allowed. Federated DNN with ownership verification, requires participants that a) keep local model updates secret from the aggregator; and b) keep ownership verification information secret from the aggregators. The first requirement has been fulfilled by techniques like Homomorphic Encryption [Phong *et al.*, 2018], differential privacy [Abadi *et al.*, 2016] or secret sharing [Ryffel *et al.*, 2020], the second requirement is one of the open problems considered in this work. We give below a formal definition of Federated DNN ownership verification scheme, which is pictorially illustrated in Fig. 2

Definition 1. A Federated Deep Neural Network (FedDNN) model ownership verification scheme for a given network $\mathbb{N}[]$ is a tuple $\mathcal{V} = (G, E, A, V_W, V_B, V_G)$ of processes, consisting of,

I) for client $k, k \in \{1, \cdots K\}$, a client-side *key generation process* $G() \rightarrow \mathbf{B}_k, \theta_k, \mathbf{T}_k$ which generates target signature $\mathbf{B}_k$, signature extraction parameters $\theta_k = \{\mathsf{S}_k, \mathbf{E}_k\}$ and a *trigger set* $\mathbf{T}_k = \{(\mathbf{X}_1, \mathbf{Y}_1), \cdots, (\mathbf{X}_J, \mathbf{Y}_J)\}$ of backdoor samples $\mathbf{X}_j$ and corresponding output labels $\mathbf{Y}_j$;

II) a client-side FedDNN *embedding process* E which minimizes the combined loss of main task, and two regularization terms to embed trigger set backdoor samples $\mathbf{T}_k$ and signature $\mathbf{B}_k$ respectively[3],

$$L := \underbrace{L_{D_k}(\mathbf{W}_\mathbf{k}^\mathbf{t})}_{\text{main task}} + \alpha_k \underbrace{L_{\mathbf{T}_k}(\mathbf{W}_k^t)}_{\text{trigger set sign.}} + \beta_k \underbrace{R_{\mathbf{B}_k,\theta_k}(\mathbf{W}_\mathbf{k}^\mathbf{t})}_{\text{feature-based sign.}},$$
$$k \in \{1, \cdots K\}, \quad (5)$$

with $\text{ClientUpdate}(n, \mathbf{W}^t) = \mathbf{W}^{t-1} - \eta \frac{\partial L}{\partial \mathbf{W}}$ to be sent to the server for updating at iteration t;

III) a server-side FedDNN *aggregation process* A which collects updates from m randomly selected clients and performs model aggregation using the `FederatedAveraging` algorithm [McMahan *et al.*, 2017] i.e.

$$\mathbf{W}^{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \mathbf{W}_k^{t+1},$$
$$\text{where } \mathbf{W}_k^{t+1} \leftarrow \text{ClientUpdate}(k, \mathbf{W}^t) \text{ for } m \text{ clients}, \quad (6)$$

IV) a client-side *white-box verification process* V_W which checks whether signatures extracted from the federated model $\hat{\mathbf{B}}_k = \text{sign}\big(\mathsf{S}_k(\mathbf{W})\mathbf{E}_k\big)$ is similar to the client target signature $\mathbf{B}_k$,

$$V_W(\mathbf{W}, (\mathbf{B}_k, \theta_k)) = \begin{cases} \text{TRUE}, & \text{if } \text{H}(\mathbf{B}_k, \hat{\mathbf{B}}_k) \leq \epsilon_H, \\ \text{FALSE}, & \text{otherwise;} \end{cases} \quad (7)$$

V) a client-side *black-box verification process* V_B which checks whether the detection error of designated labels $\mathbf{Y}_j$ generated by trigger set backdoor samples $\mathbf{X}_j$ is smaller than ϵ_y

$$V_B(\mathbb{N}, \mathbf{T}_k) = \begin{cases} \text{TRUE}, & \text{if } \mathbb{E}_{\mathbf{T}_n}(I(\mathbf{Y}_j \neq \mathbb{N}[\mathbf{X}_j])) \leq \epsilon_y, \\ \text{FALSE}, & \text{otherwise,} \end{cases} \quad (8)$$

in which $I()$ is the indicator and $\mathbb{E}$ the expectation over trigger set $\mathbf{T}_n$;

A fundamental challenge for *federated* DNN signature embedding is to ensure that signatures embedded into local models can be reliably detected from the federated model. For trigger-set based signatures, this seems not an issues as backdoor samples with arbitrarily assigned labels can always be learned with over-parameterized models as demonstrated in [Allen-Zhu *et al.*, 2018; Zhang *et al.*, 2017] (also see Figure 4 (c) and (d)). For feature-based signatures, however, it remains an open question whether there is a common solution $\mathbf{W}$ for different clients to embed their own designated signatures. The following analysis elucidates the condition under which a feasible solution is guaranteed.

Definition 2. Let $\mathbf{U}^{M\times KN}$ be matrix combined with $\{\mathbf{E}_1^{M\times N}, \mathbf{E}_2^{M\times N}, \cdots, \mathbf{E}_K^{M\times N}\}$ by column. Let $\tilde{\mathbf{U}}^{M\times KN}$ be matrix combined with

[3] A client k may opt-out and not embed signatures or trigger set backdoor samples by setting $\alpha_k = 0.0$ or $\beta_k = 0.0$. Following [Sun *et al.*, 2019], we adopt *random sampling* strategy in experiments to assign non-zero values to α_k, β_k to simulate the situation that client make decisions by their own.

$\{(\mathbf{B}_1\mathbf{E}_1)^{M\times N}, (\mathbf{B}_2\mathbf{E}_2)^{M\times N}, \cdots, (\mathbf{B}_N\mathbf{E}_K)^{M\times N}\}$ by column, where $\mathbf{B}_k = (t_{k1}, t_{k2}, \cdots, t_{kN}) \in \{+1, -1\}^N$, is signature of k_{th} client.

In order to satisfy the required condition $t_{kj}(\mathbf{W^T E}_k)_j > 0$, let's consider the following two alternatives: (i) $\tilde{\mathbf{U}}x = 0, x \geq 0$ for some non-zero x; $(ii)\exists \mathbf{W}$ such that $\mathbf{W}^T\tilde{\mathbf{U}} \geq 0$. Actually exactly one of the two statements is true according to the Gordan's theorem ([Alon and Berman, 1986]), which is a simple modifications of Farkas' Lemma ([Dinh and Jeyakumar, 2014]). And the following propositions 1 gives three conditions of $\mathbf{U}$ to satisfy $\mathbf{W}^T\tilde{\mathbf{U}} \geq 0$. The proof is shown in Appendix.

Proposition 1. If $\mathbf{U}$ or $\tilde{\mathbf{U}}$ as defined above satisfy any one of following conditions, then there exists $\mathbf{W}$ such that $\mathbf{W}^T\tilde{\mathbf{U}} \geq 0$.

1. $rank(\mathbf{U}) = KN$,
2. $\exists$ all elements of one row of $\tilde{\mathbf{U}}^{M\times KN}$ are positive,
3. The dot product of any two columns of $\tilde{\mathbf{U}}^{M\times KN}$ are positive.

In addition, when the feature-based sign loss is binary cross-entropy regularization $BCE_{\mathbf{B},\theta}(\mathbf{W^t})$, there exits the common model parameters $\mathbf{W}$ under three conditions such that $\mathbf{WU}$ is less than zero ($\sigma(\mathbf{WU}) < 0.5$) as target signature $\mathbf{B}$ is 0, or larger than zero as target signature $\mathbf{B}$ is 1.

3 Experiments

This section illustrates the empirical study of our protection framework on the FedDNN models. The network architectures we investigated include the well-known AlexNet and ResNet-18, which are tested with typical CIFAR10 and CIFAR100 classification tasks. In particular, our experiments embed binary signatures to the last convolution layer of AlexNet and ResNet, corresponding to 256 channels and 512 channels of convolution kernel weights $\mathbf{W}^k$ and normalization scale weights $\mathbf{W}^\gamma$. For federated learning setting, we simulate a $K = 20$ clients horizontal federated learning system in a stand-alone machine. In each communication round, the server sample clients with uniform distribution.

We adopt adversarial samples as trigger set $\mathbf{T}$, which are trained by Projected Gradient Descent (PGD)[Nguyen *et al.*, 2015], the original data T_{source} is in the standard benchmark data, trigger $\mathbf{T}$ can mislead the classifier to targeted label designated ahead.

3.1 Evaluation Metrics

To evaluate the FedDNN model signature embedding quantitatively, we use a set of metrics to measure the *fidelity* and *reliability* of the proposed feature-based signatures and trigger-set based signatures.

Fidelity: we use classification accuracy on the main task Acc_{main} as the metrics for fidelity. It is expected classification accuracy should not be degraded by the embedding of signatures into the federated model.

Reliability: averaged detection rate η of embedded signatures is used to quantify the reliability of a signature verification scheme. For feature-based signatures, detection rate η_F is calculated as $\eta_F = 1 - \frac{D_{hamming}}{M}$, where $D_{hamming}$ measures Hamming distance $H(\mathbf{B}, \hat{\mathbf{B}})$ between extracted binary signature string and the target signatures (M bits length in total). For trigger-set based signatures, detection rate η_T is calculated as the ratio of backdoor samples that are classified as designated labels w.r.t. the total number of all trigger set samples.

3.2 Fidelity

Fidelity of the proposed model verification scheme was evaluated under different settings, including varying signature bit length, varying number of triggers per client and different datasets and model architectures.

Trigger-set Signature: varying number of clients may decide to embed different number of trigger set samples (as signatures) into the federated model, and Figure 3 (c) and (d) show that *model performances of the main task Acc_{main} remain almost constant* when 20 to 600 trigger set samples are embedded by, respectively, each of 5 and 10 clients. There is a negligible accuracy drop (less than 1%) with respect to the model performance without embedding any trigger set signatures.

Feature-based Signature: Figure 3 (a) and (b) illustrates model performance Acc_{main} measured with different length (M) of binary signatures embedded into normalization layer scale parameter ($\mathbf{W}_\gamma$). It was observed that long bit-lengths (200 bits per client) of signatures lead to slight model performance drop up to 2% for AlexNet on CIFAR10 classification main task. Similar performance drop up to 2% was also observed for ResNet on CIFAR100 classification task, when up to 350 bits signatures were used for each client of 10 clients. The drop of classification accuracy Acc_{main} is due to the sub-optimal solution restricted to the subspace defined by large number of binary signature constrains (see Proposition 1). Note that performance drop can actually be mitigated by assigning binary signatures across different layers of normalization scale parameters.

3.3 Reliability

Reliability of the proposed model verification scheme was evaluated under different settings, including varying signature bit length, varying number of triggers per client and different datasets and model architectures.

Trigger-set based signature: reliability of trigger-set signatures were evaluated under two settings, i.e., n_B = 5 or 10 clients are randomly selected to embed trigger-set signatures generated by Projected Gradient Descent (PGD) adversarial attack method [Nguyen *et al.*, 2015].

Figure 4 (c) and (d) illustrate the trigger set detection rates η_T on these adversarial sample $\mathbf{T}$, respectively, with AlexNet on CIFAR10 classification and ResNet18 on CIFAR100 classification tasks[4]. The results show that the trigger set detection rates η_T almost keep constant even the trigger number per client increases. Moreover, detection rates η_T of signatures embedded in the more complex ResNet18 is more stable than those signatures embedded in AlexNet. Also, the detection

[4]The trigger set samples are regarded as correctly detected when the designated targeted adversarial labels are returned.

Figure 3: Model performances in a federated learning system with 20 clients. Figure (a) and (b), respectively, illustrate CIFAR10 with AlexNet and CIFAR100 with ResNet18 classification accuracy Acc_{main}, when n_W = 5, 10 clients embed varying bit-lengths signatures. Figure (c) and (d), respectively, illustrate CIFAR10 with AlexNet and CIFAR100 with ResNet18 classification accuracy Acc_{main} for n_B = 5 or 10 clients embedding varying number of trigger-set samples.

Figure 4: In a federated learning system of 20 clients, figure (a) and (b), respectively, illustrate the case when n_W = 5, 10, the signature detection rate η_F with varying bit length per client, figure (a) describes the case that AlexNet with CIFAR10 dataset, figure (b) describes the case that ResNet with CIFAR100 dataset. Figure (c), (d) illustrate the case when n_B = 5, 10, the trigger detection rate η_T with varying trigger per client, figure (c) describes the case that AlexNet with CIFAR10 dataset, figure (d) describes the case that ResNet with CIFAR100 dataset

rate is not influenced by the varying number of clients and, thus, varying number of total trigger-set samples used. We ascribe the stable detection rate η_T to the generalization capability of over-parameterized networks as demonstrated in [Allen-Zhu *et al.*, 2018; Zhang *et al.*, 2017].

Feature-based Signature: Figure 4 (a) and (b) illustrate binary signature detection rates η_F in white-box manner, in which (a) is with AlexNet for CIFAR10 and (b) with ResNet18 for CIFAR100 classification tasks. First, note that the detection rates η_F remain constant (100%) within the regime, whereas the total bit lengths assigned by multiple (n_W = 5 or 10) clients does not exceed the capacity of network parameters used to embed signatures. This limit is, respectively, 256 and 512 convolution channels at the last layer for AlexNet and ResNet18. Therefore, binary signatures of all bits can be reliably detected, which is in accordance to the analysis disclosed in Proposition 1. When the total bit lengths exceeds the limit e.g. in Figure 4 (a), 100 bits signatures are assigned by 5 clients, the detection rate η_F drops to about 80% due to the conflicts of overlapping signature assignments. Nevertheless, the dropped detection rate still guarantees very high confidence in claiming the ownership of verified models.

The results illustrated in Figure 4 give rise to the capability of feature based signature $\mathbf{B}$ into FedDNN model: the bit length of signatures of total clients $\{M\}_{i=1}^{n_W}$ can not exceed the channel number of normalization scale weights $\mathbf{W}^\gamma$ in selected convolutional layers.

4 Robustness

Strategies like Differential Privacy[Wei *et al.*, 2020], Homomorphic Encryption[Phong *et al.*, 2018] and client selection[McMahan *et al.*, 2017] are widely used for privacy and efficiency in federated learning. Those strategies intrinsically bring performance decades on the main classification task. Respectively, we evaluate the detection rate η of signature under those strategies. Moreover, the attacker may try to remove the signatures while inheriting most model performance in federated learning. Specifically, we conduct two removal attacks including fine-tuning and pruning to identify whether the signatures can be reliably detected under those removal attacks.

Robustness Against Differential Privacy: we adopt the Gaussian noise-based method to provide differential privacy guarantee for federated learning. Specifically, We vary the standard deviation σ of Gaussian noise on the local gradient before clients send gradients to the server. As Figure 5 (a) shows, the main task performance Acc_{main} decreases severely as the σ of noise increases, while the detection rate of feature-based signature η_F and trigger-based signature η_T drop little while the Acc_{main} is within usable range (more than 85%). In a concrete way, when $sigma$ equals 0.003, classification accuracy Acc_{main}, detection rate η_F and η_T keep a high performance,

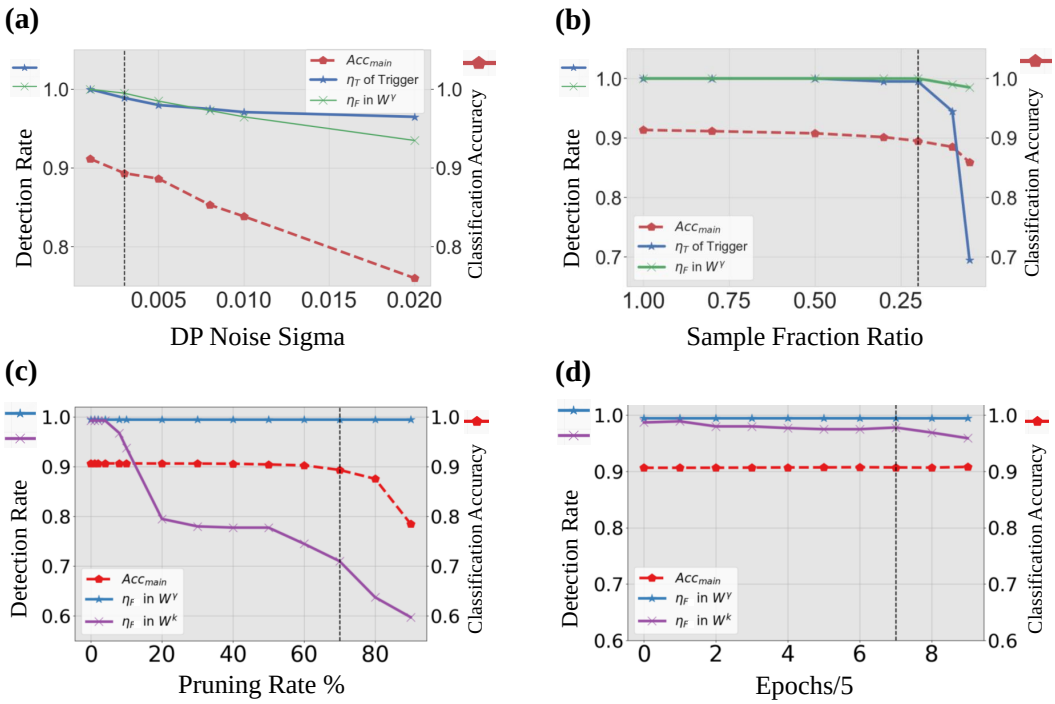

Figure 5: Figure describes the robustness of our FedIPR ownership verification scheme: In a federated learning system of 20 clients training AlexNet with CIFAR10 dataset, in which n_W = 10 clients embed white-box signatures, n_B = 10 clients embed black-box signatures. The dot lines in figure (a)(b)(c)(d). illustrates the main task classification accuracy Acc_{main} under diverse settings. Respectively, figure (a) illustrates feature-based detection rate η_F and trigger-based detection rate η_T under varying differential private noise sigma; figure (b) illustrates feature-based detection rate η_F and trigger-based detection rate η_T under different sample fraction ratio while federated training; figure (c) illustrates feature-based signature detection rate η_F against model pruning attack with varying pruning rate; figure (d) illustrates feature-based signature detection rate η_F against model finetuning attack in 50 epochs.

which demonstrates the robustness of signature under differential privacy strategy.

Robustness Against Client Selection: we decrease the fraction ratio c of clients selected in each epoch to for communication efficiency. Figure 5 (b) shows that the signature could not be removed even the fraction ratio c is as low as 0.25. More specifically, when the fraction ratio is larger than 0.2, main classification accuracy Acc_{main} and detection rate η keep constant. This result gives a lower bound of client sampling rate in which signatures can be effectively embedded and verified.

Robustness Against Pruning: the target of model pruning is to reduce redundant parameters without compromise the performance. We evaluate the main task performance Acc_{main} and signature detection rate η under pruning attack with varying pruning rate. Figure 5 (c) shows signature detection rate η while varying network parameters are pruned. It was observed that the detection rate η_T of signature embedded in normalization layer is stable all the time, while η_F with $\mathbf{W}_k$ are severely degraded, this fact shows that the signature on normalization parameters are more robust against pruning attack.

Robustness Against Fine-tuning: attacks on embedded signatures by fine-tuning were launched to train the network without the presence of the regularization term, *i.e.*, $L_{\mathbf{T}}$ and $R_{\mathbf{B}}$. In Figure 5 (d), it was observed that the detection rate η_F of signature embedded with normalization layer ($\mathbf{W}_\gamma$) remains at 100% (blue curve). In contrast, the detection rate of signature embedded with convolution layer ($\mathbf{W}_k$) drops significantly (purple curve). The superior robustness of signatures embedded in normalization layer is in accordance to observations reported in [Fan *et al.*, 2019].

5 Discussion and Conclusion

This paper illustrated a novel ownership verification scheme to protect Intellectual Property Right (IPR) of Federated DNN models against external plagiarizers who illegally copy, redistribute the models. To our best knowledge, it is the first ownership verification scheme that aims to protect IPR of federated learning models. This work addresses a crucial issue remained open in federated learning research, since the protection of valuable federated learning models is as important as protecting data privacy.

On the technical side, this work demonstrated that reliable and persistent signatures can be embedded into local models without disclosing the presence and extraction parameters of these signatures. In particular, normalization scale parameters based signatures are extremely robust against removal attacks including fine-tuning and pruning. It is our wish that the formulation illustrated in this paper will lead to signature embedding and verification in various federated learning settings.

References

[Abadi *et al.*, 2016] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[Adi *et al.*, 2018] Y Adi, C Baum, M Cisse, B Pinkas, and J Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX Security Symposium (USENIX)*, 2018.

[Allen-Zhu *et al.*, 2018] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. *CoRR*, abs/1811.03962, 2018.

[Alon and Berman, 1986] Noga Alon and Kenneth A Berman. Regular hypergraphs, gordon's lemma, steinitz'lemma and invariant theory. *Journal of Combinatorial Theory, Series A*, 43(1):91–97, 1986.

[Bagdasaryan and Shmatikov, 2020] Eugene Bagdasaryan and Vitaly Shmatikov. Blind Backdoors in Deep Learning Models. *arXiv e-prints*, page arXiv:2005.03823, May 2020.

[Bagdasaryan *et al.*, 2018] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. *arXiv preprint arXiv:1807.00459*, 2018.

[Bhagoji *et al.*, 2019] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 634–643. PMLR, 09–15 Jun 2019.

[Boenisch, 2020] Franziska Boenisch. A survey on model watermarking neural networks, 2020.

[Chen *et al.*, 2018] Huili Chen, Bita Darvish Rohani, and Farinaz Koushanfar. DeepMarks: A Digital Fingerprinting Framework for Deep Neural Networks. *arXiv e-prints*, page arXiv:1804.03648, April 2018.

[Darvish Rouhani *et al.*, 2018] Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. DeepSigns: A Generic Watermarking Framework for IP Protection of Deep Learning Models. *arXiv e-prints*, page arXiv:1804.00750, April 2018.

[Dinh and Jeyakumar, 2014] N Dinh and V Jeyakumar. Farkas' lemma: three decades of generalizations for mathematical optimization. *Top*, 22(1):1–22, 2014.

[Fan *et al.*, 2019] Lixin Fan, Kam Woh Ng, and Chee Seng Chan. Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks. In *Advances in Neural Information Processing Systems*, pages 4714–4723. Curran Associates, Inc., 2019.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[Kairouz *et al.*, 2019] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, abs/1912.04977, 2019.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

[Nguyen *et al.*, 2015] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[Ong *et al.*, 2021] Ding Sheng Ong, Chee Seng Chan, Kam Woh Ng, Lixin Fan, and Qiang Yang. Protecting intellectual property of generative adversarial networks from ambiguity attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[Phong *et al.*, 2018] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, 2018.

[Ryffel *et al.*, 2020] Theo Ryffel, David Pointcheval, and Francis R. Bach. ARIANN: low-interaction privacy-preserving deep learning via function secret sharing. *CoRR*, abs/2006.04593, 2020.

[Shokri and Shmatikov, 2015] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015.

[Sun *et al.*, 2019] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H. Brendan McMahan. Can You Really Backdoor Federated Learning? *arXiv e-prints*, page arXiv:1911.07963, November 2019.

[Uchida *et al.*, 2017] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 269–277, 2017.

[Wei *et al.*, 2020] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[Wu *et al.*, 2020] Chen Wu, Xian Yang, Sencun Zhu, and Prasenjit Mitra. Mitigating backdoor attacks in federated learning. *CoRR*, abs/2011.01767, 2020.

[Yang *et al.*, 2019] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):12, 2019.

[Zhang *et al.*, 2017] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*. OpenReview.net, 2017.

[Zhang *et al.*, 2018] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security (ASIACCS)*, pages 159–172, 2018.

[Zhang *et al.*, 2020] Jie Zhang, Dongdong Chen, Jing Liao, Weiming Zhang, Gang Hua, and Nenghai Yu. Passport-aware normalization for deep model protection. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22619–22628. Curran Associates, Inc., 2020.

Appendix

A Notation of Parameters

Table 1 summarizes the notation in the whole paper.

B Proof of existence for feature based regularization

In this part, we give a proof of the Proposition 1, which illustrates the three conditions under which *reliable* and *persistent* signatures can be successfully embedded into the same FedDNN model by multiple clients.

Proposition 2. If $\mathbf{U}$ or $\tilde{\mathbf{U}}$ as defined above satisfy any one of following conditions, then there exists $\mathbf{W}$ such that $\mathbf{W}^T\tilde{\mathbf{U}} \geq 0$.

1. $rank(\mathbf{U}) = KN$,
2. $\exists$ all elements of one row of $\tilde{\mathbf{U}}^{M\times KN}$ are positive,
3. The dot product of any two columns of $\tilde{\mathbf{U}}^{M\times KN}$ are positive.

Proof. For the condition (1), if $rank(\mathbf{U}) = K$, then the column of $\mathbf{U}$ $(U_1, U_2, \cdots U_{KN})$ is independent, and the column of $\tilde{\mathbf{U}}$ $(\tilde{U_1}, \tilde{U_2}, \cdots \tilde{U_{KN}})$ is also independent. Thus,

$$\begin{aligned} &y_1\tilde{U_1} + y_2\tilde{U_2} + \cdots + y_{KN}\tilde{U_{KN}} = 0 \\ \Longleftrightarrow\ &y_1 = y_2 = \cdots = y_{KN} = 0 \end{aligned} \tag{9}$$

Therefore the solution of $\tilde{\mathbf{U}}\vec{y} = \vec{0}$ is only $\vec{0}$, moreover, $\tilde{\mathbf{U}}\vec{y} = \vec{0}$ doesn't have non-negative solutions except $\vec{0}$. According to Gordan's theorem ([Alon and Berman, 1986]), Either $\tilde{\mathbf{U}}\vec{y} > 0$ has a solution y, or $\tilde{\mathbf{U}}\vec{y} = \vec{0}$ has a nonzero solution y with $\vec{y} \geq \vec{0}$. Since the latter statement is wrong, there exists $\mathbf{W} = \vec{y}^T$ such that $\mathbf{W}\tilde{\mathbf{U}} > 0$.

For the second condition, it is obvious that $\tilde{\mathbf{U}}\vec{y} = 0$ doesn't have non-negative solutions except $\vec{0}$. Therefore the conclusion is true based on Gordan's theorem.

For the third condition, let the columns of $\tilde{U}$ be $(\tilde{U_1}, \tilde{U_2}, \cdots \tilde{U_{KN}})$. We obtain a contradiction by considering $\tilde{\mathbf{U}}\vec{y} = 0$ has a nonzero solution y with $\vec{y} \geq \vec{0}$, then

$$0 = \tilde{\mathbf{U}}\vec{y}(\tilde{\mathbf{U}}\vec{y})^T = \sum_{i,j} \tilde{U_i}\tilde{U_j}^T y_i y_j \tag{10}$$

Since $y_i \geq 0$ and $\tilde{U_i}\tilde{U_j}^T > 0$, $y_i = 0$ for $i = 1, 2, \cdots KN$. This shows $\tilde{\mathbf{U}}\vec{y} = \vec{0}$ has a nonzero solution y with $\vec{y} \geq \vec{0}$, which infers the existence of $\mathbf{W}^{1\times M}$ such that $\mathbf{W}\tilde{\mathbf{U}} > 0$. □

Remark The proposition only demonstrates the existence of solution for signature regularization term $R_{\mathbf{B},\theta}(\mathbf{W})$. Embedding signature does not influence the performance of the main task confirmed in experiments, because deep neural networks are typically over parameterized. Deep neural networks have many local minima, whose error very close to the global minimum [**?**; **?**]. Therefore, the embedding regularizer only needs to guide model parameters to one of a number of good local minima so that the final model parameters embed the signature well.

C Experiment Settings

This section illustrates the experiment settings of the empirical study on our FedIPR framework for the FedDNN models.

DNN Model Architectures. The deep neural network architectures we investigated include the well-known AlexNet and ResNet-18. Feature-based binary signatures are embedded into convolution kernel weights $\mathbf{W}^k$ and normalization scale weights $\mathbf{W}^\gamma$ of multiple convolution layers in AlexNet and ResNet-18. Table 2 shows the detailed model architectures and parameter shape of AlexNet and ResNet-18, which we employed in all the experiments.

Dataset. We evaluate classification tasks on standard CIFAR10 dataset and CIFAR100 dataset. The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. CIFAR-100 has 100 classes containing 600 images each, there are 500 training images and 100 testing images per class. Respectively, we conduct stand image classification tasks of CIFAR10 and CIFAR100 with AlexNet and ResNet-18. According to the way we split the dataset for clients in federated learning, the experiments are divided into iid setting and non-iid setting. The results for both IID setting and Non-IID federated learning setting are provided in the section **??**.

Federated Learning Settings We simulate a with $K = 20$ clients horizontal federated learning system in a stand-alone machine with 8 Tesla V100-SXM2 32 GB GPUs and 72 cores of Intel(R) Xeon(R) Gold 61xx CPUs.

In each communication round, the server samples clients with uniform distribution of a certain fraction ratio to participate training (the fraction ratio we explore includes $1.0, 0.8, 0.5, 0.2, 0.1$ and 0.05). The clients update the weight updates, server adopts `Federatedavg`[McMahan *et al.*, 2017] algorithm in to aggregate the model updates. The detailed experiment Hyper parameters we employ to conduct our federated learning are listed in the table 4.

Embedding Process E

Feature-based signature: For the feature-based signature embedding scheme, we constrain the sign of network parameters $\mathbf{W}^\gamma$ and $\mathbf{W}^k$ with regularization terms $R_{\mathbf{B},\theta}$ including Hinge Like loss, and cross-entropy loss targeted at different bit length of each client. We change the number of clients and the bit length of signature per client under diverse federated learning setting. The algorithm is shown in Algorithm 2.

Trigger-set Signature: The trigger-set embedding process adopts a batch-poisoning backdoor method: in each iteration of backdoor training, both normal samples and backdoor samples are used in the same data batch for model training.

We adopt adversarial samples as trigger set $\mathbf{T}$: we train the adversarial samples with Projected Gradient Descent (PGD)[Nguyen *et al.*, 2015], from original data T_{source} in the standard benchmark data. The PGD parameters are listed in the table 5, After the training process of PGD, the trigger $\mathbf{T}$ can mislead the classifier to targeted label designated ahead.

Verification process V

After obtaining the model, we could extract the signature and test the trigger set from the model. Verification process of signature and trigger-set is shown Algorithm 3 and 4.

Removal attack

Table 1: Notation description

$\mathbf{W}$	Model weights	$\mathbf{W}^k$	Convolution kernel weights
		$\mathbf{W}^\gamma$	Normalization scale weights
$\mathbf{O}$	Model outputs	$\mathbf{O}^k$	Middle layer activation outputs
		$\mathbf{O}^I$	Images outputs
		$\mathbf{O}^c$	Classification labels outputs
Key generation Process: G			
$\mathbf{B}$	Target signature	B	Extracted signature from model
$\hat{B}$	$sign(B)$	N	Bits number of targeted signature
M	Dimension of B	$\theta = \{\mathcal{S}, \mathbf{E}\}$	Hyper-parameters of extracting signature
$\mathbf{T}$	Trigger set	K	Number of clients
Embedding Process: E			
L_D	Main task loss	$L_{\mathbf{T}}$	trigger-set-based loss
$R_{\mathbf{B},\theta}$	Feature-based regularization	HL() :	Hinge loss
		BCE() :	Cross entropy loss
Aggregation Process: A			
n_k	The aggregate weights for k_{th} clients		
Verification Process: V			
V_W	white-box verification	ϵ_W	threshold of signature detection
V_B	black-box verification	ϵ_B	threshold of signature detection

Following previous DNN watermarking methods, we report model performances under fine-tuning and pruning attacks.

For **finetuning**, we adopt the code[5] and follow their implementation in Algorithm 5. For **pruning**, we adopt the code[6] and follow their implementation in Algorithm 5. The hyper parameters are shown in table 3.

Algorithm 1 Generation G of Signatures

procedure SIGNATURE GENERATION
  for client k in K clients **do**
    Initialize $\mathbf{B}_k$, $\theta_k = \{\mathsf{S}_k, \mathbf{E}_k\}$
    Encode $\mathbf{B}_k$ into binary to be embedded into signs of $\mathbf{W}^T\mathbf{E}$
    Initialize $\mathbf{T}_k = \{(\mathbf{X}_1, \mathbf{Y}_1), \cdots, (\mathbf{X}_J, \mathbf{Y}_J)\}$ ▷ Backdoor samples $\mathbf{X}_j$ and corresponding output labels $\mathbf{Y}_j$
  return $\{(\mathbf{B}_k, \theta_k, \mathbf{T}_k)\}_{k=1}^{k=K}$

[5] https://github.com/dingsheng-ong/ipr-gan

[6] https://github.com/zepx/pytorch-weight-prune/blob/develop/pruning/methods.py

layer name	output size	weight shape	padding
Conv1	32×32	$64 \times 3 \times 5 \times 5$	2
MaxPool2d	16×16	2×2	
Conv2	16×16	$192 \times 64 \times 5 \times 5$	2
Maxpool2d	8×8	2×2	
Conv3	8×8	$384 \times 192 \times 3 \times 3$	1
Conv4	8×8	$256 \times 384 \times 3 \times 3$	1
Sign Embedding ($\mathbf{W}_\gamma$)	8×8	256	
Conv5	8×8	$256 \times 256 \times 3 \times 3$	1
Sign Embedding ($\mathbf{W}_\gamma$)	8×8	256	
MaxPool2d	4×4	2×2	
Linear	10	10×4096	

layer name	output size	weight shape	padding
Conv1	32×32	$64 \times 3 \times 3 \times 3$	1
Conv2_x	32×32	$\begin{bmatrix} 64 \times 64 \times 3 \times 3 \\ 64 \times 64 \times 3 \times 3 \end{bmatrix} \times 2$	1
Conv3_x	16×16	$\begin{bmatrix} 128 \times 128 \times 3 \times 3 \\ 128 \times 128 \times 3 \times 3 \end{bmatrix} \times 2$	1
Conv4_x	8×8	$\begin{bmatrix} 256 \times 256 \times 3 \times 3 \\ 256 \times 256 \times 3 \times 3 \end{bmatrix} \times 2$	1
Conv5_x	4×4	$\begin{bmatrix} 512 \times 512 \times 3 \times 3 \\ 512 \times 512 \times 3 \times 3 \end{bmatrix} \times 2$	1
Sign Embedding ($\mathbf{W}_\gamma$)	4×48	512×1	
Average pool	1×1	4×4	
Linear	10	10×512	

Table 2: Left: modified AlexNet architecture. Right: modified ResNet-18 architecture

Hyper-parameter	Removal Attack
Pruning Rate	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
Finetuning Learning Rate	0.0001
Finetuning Learning Loss	only Cross Entropy loss
Batch size	16
Finetuning Epochs	50
Learning rate decay	0.01 per epoch
Vanilla Classification model	CNN with three convolution layers

Table 3: Training parameters for Removal Attack

Hyper-parameter	AlexNet	ResNet-18
Activation function	ReLU	ReLU
Optimization method	SGD	SGD
Momentum	0.9	0.9
Learning rate	0.01	0.01
Batch size	16	16
Backdoor batch size	2	2
Data Distribution	IID and non-IID	IID and non-IID
Global Epochs	200	200
Local Epochs	2	2
Learning rate decay	0.99 at each global Epoch	0.99 at each global Epoch
Federated Fraction	[He *et al.*, 2015]	[He *et al.*, 2015]
Client numbers	20	20
Feature-based Signature Client numbers	5,10	5,10
Regularization Term	BCE loss, Hinge-like loss	BCE loss, Hinge-like loss
α of Regularization Loss	0.2, 0.5, 1, 5	0.2, 0.5, 1, 5
Feature-based Signature parameters $\mathbf{W}$	$\mathbf{W}^k$ and $\mathbf{W}^\gamma$	$\mathbf{W}^k$ and $\mathbf{W}^\gamma$
Trigger-based Signature Client numbers	5,10	5,10
Trigger-based Signature type	Adversarial sample	Adversarial sample

Table 4: Training parameters for Federated AlexNet$_\mathbf{p}$ and ResNet$_\mathbf{p}$-18, respectively († the learning rate is scheduled as 0.01, 0.001 and 0.0001 between epochs [1-100], [101-150] and [151-200] respectively).

Hyper-parameter	Projected Gradient Descent
Optimization method	Profected Gradient Descent
Norm type	L2
Norm of noise	0.3
Learning rate	0.01
PGD Batch size	128
Targeted at Specific Labels	True
Iterations	80
Learning rate decay	None
Vanilla Classification model	CNN with three convolution layers

Table 5: Training parameters for Projected Gradient Descent Adversarial Training

Algorithm 2 Signature Embedding Process for FedIPR

Each client k with its own signature tuple $(\mathbf{B}_k, \theta_k, \mathbf{T}_k)$
for t in communication round E **do**
    Server distributes the global model parameters W_t to each clients
    Sample clients with fraction ratio C into subset s of K clients
    Local Training:
    for k in number of selected users subset s **do**
        Sample minibatch of m samples X $\{X^{(1)}, \cdots, X^{(m)}\}$ and targets Y $\{Y^{(1)}, \cdots, Y^{(m)}\}$
        if enable backdoor **then**
            sample t samples of $\mathbf{T}_k$ and backdoor targets $Y_{\mathbf{T}_k}$ ▷ $t = 2$, default by [Adi *et al.*, 2018]
        concatenate X with T, Y with $Y_{\mathbf{T}_k}$
        compute cross-entropy loss L_c using X and Y
        for layer l in targeted layers set L **do**
            compute Regularization term R^l using θ_k and $\mathbf{W}^l$
        $R \leftarrow \sum_l^{l \in L} R^l$
        $L = L_c + R$
        Backpropagate using L and update W_t^k
    Server Update:
    Aggregate the $\{W_t^k\}_{k=1}^K$ with `FederateAvg` algorithm

Algorithm 3 White-box Feature Based Signature Verification

Input: Model weights $\mathbf{W}$ offered by adversaries, Embedding matrix $\mathbf{E}$ and $\mathbf{B}$ provided by user.

procedure SIGNATURE DETECTION
    $B \leftarrow \mathbf{W}^T\mathbf{E}$
    signature $\leftarrow sign(B)$
    Convert *signature* into binary
    Decode binarized *signature* into desired format e.g. ascii
    Match decoded *signature* with target signature
    Compute the signature detection rate $V_W(\mathbf{W}, \mathbf{E})$

Output: $V_W(\mathbf{W}, \mathbf{E})$

Algorithm 4 Black-box Trigger-set Based Signature Verification

Input: Model $\mathbb{N}$ offered by adversaries, Trigger set $\mathbf{T}$ and $Y_{\mathbf{T}}$ provided by user.

procedure TRIGGER-SET DETECTION
    Fed the Trigger-set $\mathbf{T}$ into model ($\mathbb{N}$) to derive the classification label $\mathbf{O}_c$
    Match $\mathbf{O}_c$ with target label of target trigger-set label $Y_{\mathbf{T}}$
    Compute the trigger-set detection rate $V_B(\mathbf{T}, Y_{\mathbf{T}}, \mathbf{W})$

Output: $V_B(\mathbf{T}, Y_{\mathbf{T}}, \mathbf{W})$

Algorithm 5 Removal attack

Input: Model $\mathbb{N}$, trigger set $\mathbf{T}$ and $Y_{\mathbf{T}}$, target signature $\mathbf{B}$

procedure PRUNING
    for p in different pruning percentage **do**
        Pruning the model $\mathbb{N}$ in p percentage.
        Test the signature and trigger-set detection rate.
procedure FINETUNING
    for epochs in 50 **do**
        Train the model $\mathbb{N}$ only in main task (classification task)
    Test the signature and trigger-set detection rate.

D Ablation Study

D.1 Influence of feature-based signature regularization parameter α

In this section, we test the influence which the feature-based signature regularization parameter α brings. Our experiments demonstrates α only affects the fidelity.

The left image of Figure 6 shows the model performance drops seriously as α increases, especially when α equals 1 and 5. The right image of Figure 6 explains the reliability keeps the similar trend even the α changes from 0.2 to 5.

D.2 Diversity of embedding position of signature

We embed feature-based signatures into last two layers of AlexNet to explore the capacity of white-box embedding. As shown in the Table 2, the Conv. layer 4 and Conv. layer 5 have each 256 channels of convolution kernels, we test the fidelity, reliability of white box signature. We compare the results of the case with single Conv. layer 5 and the case with Conv.4 and Conv.5, the results is described in the Figure 7,

Figure 7 Left shows that the signature embedding into multiple layers yields no compromise of fidelity, the main task slightly decades as the bit length increases. Figure 7 Right describe the signatures into multiple layers of the neural network amplify the capacity of signatures, because the multiple layers enable more bit length of signatures into the model. Two-layer case enables twice the bit length of signature as one-layer case in the same detection signature rate.

This result also proves that: the bit length of signatures of total clients $\{M_i\}_{i=1}^n$ can not exceed the channel number of normalization scale weight $\mathbf{W}^\gamma$ in selected convolution layers, which is consistent with Proposition 1.

D.3 Signature into Kernel Weights

The parameters $\mathbf{W}$ chosen for embedding signatures includes convolution layer weights $\mathcal{S}(\mathbf{W})$ (the columnized vector of *convolution layer weights*) and *normalization layer scale parameters* $\mathcal{S}(\mathbf{W}) = \mathbf{W}_\gamma = \{\gamma_1, \cdots, \gamma_C\}$ where C is the number of normalization filters.

The table 6 illustrates the main task classification accuracy (fidelity) with the increase of bit length of signature, the model performance is slightly affected only when the signature embedding conflict with each other. Figure 8 illustrate the reliability (signature detection rate) of signature detection on convolution weights $\mathbf{W}^K$. Convolution kernel parameters naturally have more parameters for embedding signatures, so the capacity is correspondingly larger the the case with normalization weights $\mathbf{W}^K$.

Remark The blue line in Figure 8 does not decrease because the total signature length $KN = 500 * 5 = 2500$ is closed to number of embedding weights $256 * 9 = 2294$

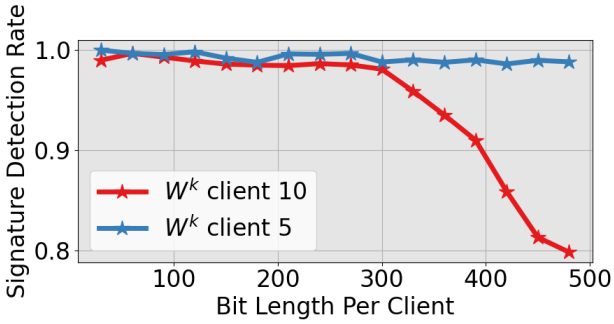

Figure 8: Results in embedding signature in the last convolution layer ($\mathbf{W}^k$) in AlexNet. The change of Signature detection rate as signature length varies when 10 or 5 clients choose to embed signature.

D.4 Cross Entropy Loss

The regularization term we employ for signature embedding include both binary cross entropy (BCE) loss and Hinge-like (HL) loss: Binary cross-entropy $\text{BCE}\Big(\mathbf{B}, \mathsf{B}\Big) = -\sum_{j=1}^N t_j \log(f_j) + (1 - t_j)\log(1 - f_j)$; where $f_j = \frac{1}{1+\exp(-b_j)}$, and Hinge loss $\text{HL}\Big(\mathbf{B}, \mathsf{B}\Big) = \sum_{j=1}^N \max(\alpha - b_j t_j, 0)$, where signatures $\mathbf{B} = (t_1, \cdots, t_N) \in \{-1, 1\}^N$.

We conduct experiments with the same setting of 20 clients in BCE regularization and Hinge regularization, whose results show both two approaches are influenced with signature length similarly.

Specifically, when the bit length is in the capacity of signature embedding, the fidelity and reliability between BCE loss and Hinge loss are the same (shown in Table 7. Moreover, when the signature embedding conflict with each other, the reliability of BCE loss is slightly better then the HL loss.

To conclude, the HL regularization term is a stronger constrain than BCE regularization term, when the diverse signature embedding of clients conflict with each other, hinge like loss affects more fidelity and reliability.

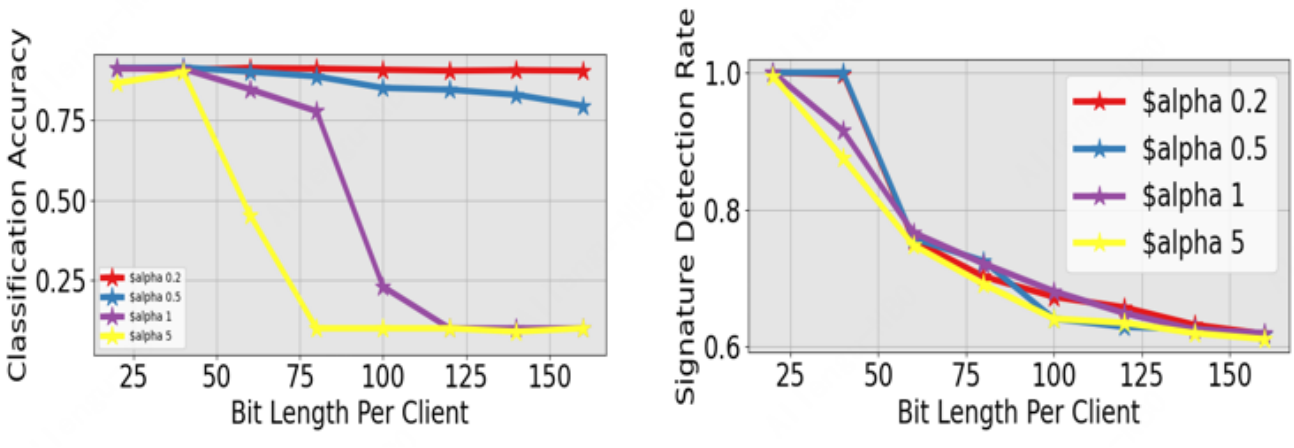

Figure 6: Results of CIFAR10 with AlexNet when embedding signature in different regularization parameter α. Left is main task classification accuracy as the signature length number varies in different $alpha$. Right images is signature detection rate with different number of signature length in different $alpha$.

Table 6: Model classification accuracy with embedding in convolution layer ($\mathbf{W}^k$) under two conditions (5 or 10 clients add signature)

Bits Number	30	60	90	120	150	180	210	240
Model Acc	0.9135	0.9131	0.9132	0.9129	0.9125	0.9123	0.912	0.913
	0.9134	0.9127	0.9165	0.9149	0.9146	0.9134	0.9128	0.9129
Bits Number	270	300	330	360	390	420	450	480
Model Acc	0.9142	0.914	0.9125	0.9132	0.9114	0.9121	0.9122	0.9094
	0.9159	0.9119	0.91	0.9131	0.9146	0.9105	0.9113	0.9139

Table 7: Model classification accuracy in BCE loss and Hinge loss under two conditions (5 or 10 clients add signature)

Bits Len.	BCE loss		Hinge loss	
	Client5	Client10	Client5	Client10
20	0.9152	0.9157	0.9137	0.9134
40	0.9146	0.91	0.912	0.9112
60	0.9137	0.9087	0.9116	0.9087
80	0.9136	0.9136	0.9115	0.9078
100	0.9135	0.9123	0.9113	0.9069
120	0.9123	0.913	0.9115	0.9052
140	0.912	0.9124	0.9112	0.904
160	0.9118	0.9139	0.9105	0.9036
180	0.9113	0.9082	0.9088	0.9035
200	0.9111	0.9121	0.9077	0.9026

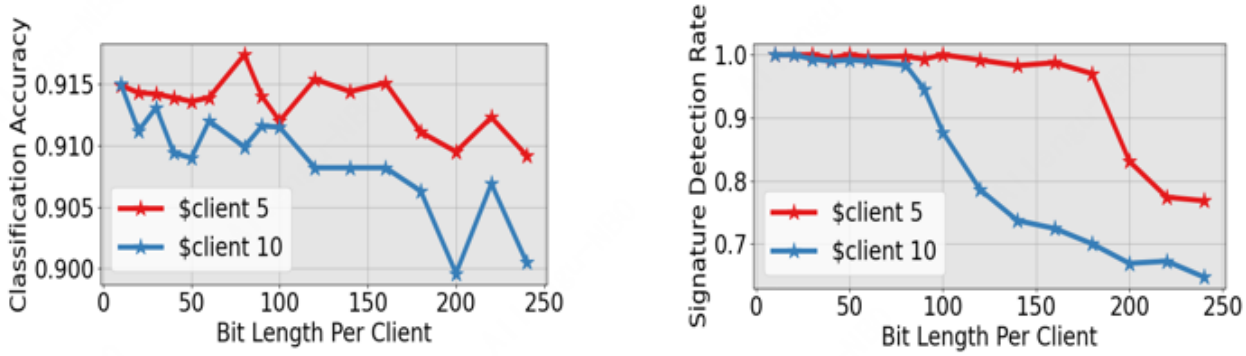

Figure 7: Results of embedding signature into last two normalization layers ($\mathbf{W}^{\gamma}$ of AlexNet. Left image is the model classification accuracy in different signature length; right is the signature detection rate in different signature length.

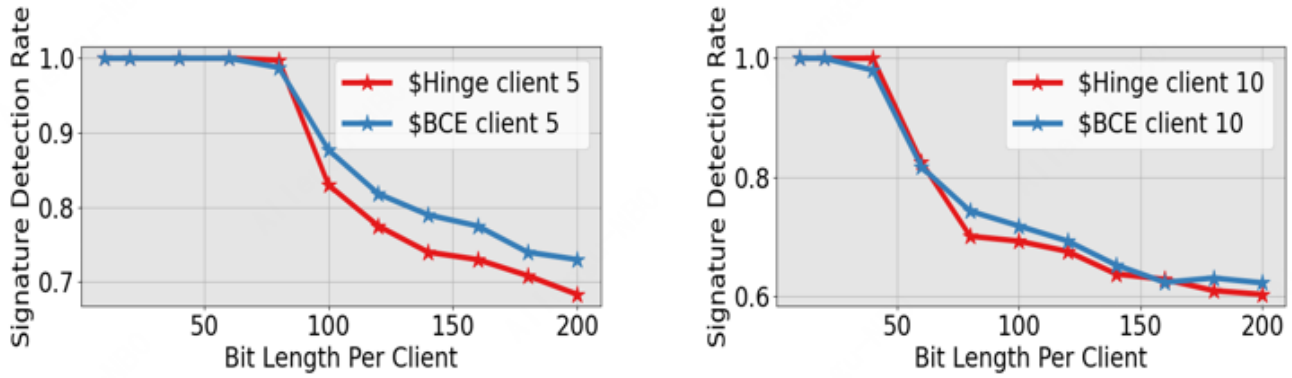

Figure 9: Results in embedding signature into last normalization layer ($\mathbf{W}^{\gamma}$ of AlexNet with two different regularization: Hinge regularization and BCE regularization. Left image is the comparison of signature detection rate between Hinge loss and BCE loss when 5 clients choose to add signature; right is similar comparison when 10 clients choose to add signature.