Proteus: Model Leakage-Induced Adversarial Attack in Federated Learning

Junjie Shan, Yue Zhang, Ziqi Zhao, and Ka-Ho Chow*

The University of Hong Kong, Hong Kong SAR, China
junjieshan@connect.hku.hk, kachow@cs.hku.hk

Abstract. The massive volume and privacy-sensitive nature of visual data have made federated learning (FL) a preferred paradigm for training vision models across distributed data sources. However, during training, FL repeatedly shares the in-progress model with randomly selected participants. This paper investigates an overlooked yet practical threat arising from this sharing process: leaked intermediate models can be exploited by adversaries to craft adversarial examples that compromise the final deployed model. Although directly using an intermediate model, especially one leaked early in training, as a surrogate yields only moderate attack gains, it can serve as an anchor for anticipating subsequent training dynamics. Based on this insight, we propose Proteus, a model leakage-induced adversarial attack that leverages a leaked model to identify vulnerabilities that persist throughout training, thereby generating adversarial examples that remain effective against the final deployed model. For the first time, we show that models exposed well before convergence can already pose substantial risks to the final model, even if it undergoes hundreds of additional training rounds after leakage. Extensive experiments across diverse datasets, neural architectures, and FL configurations confirm the severity of this threat. Proteus exploits inherent model leakage in FL and improves the attack success rate from 59.75% when directly using the leaked model to 85.40%, even when leakage occurs after only 30% of the total training process.

1 Introduction

Deep neural networks (DNNs) are sensitive to small input perturbations [18,36]. An adversary can carefully modify an input and cause it to be misclassified with high confidence. This issue is particularly severe in vision systems, where small pixel-level changes are far less perceptible than perturbations in modalities such as text [15] or tabular data [47], yet can drastically alter model predictions. Despite more than a decade of research [61], however, the practical implications of such adversarial attacks remain debated. A key obstacle is that effective attacks typically require access to the victim model's parameters. Although many studies attempt to build surrogate models so that adversarial examples crafted against them transfer to the victim, these approaches are often ineffective or require a

* Corresponding author.

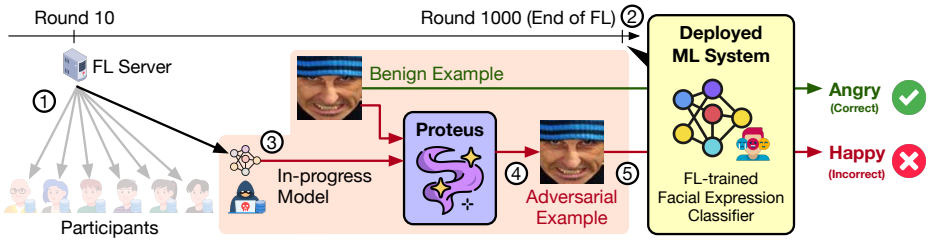


Fig. 1: FL requires the server to share the latest in-progress model with selected participants in each round. A malicious participant can exploit such leaked models using Proteus to craft adversarial examples that deceive the final deployed model.

large number of queries, which can be both costly and suspicious [3, 49]. As a result, some have questioned the real-world risk posed by adversarial examples and whether the accuracy-robustness trade-off is justified in practice [2, 51].

Federated learning [37] (FL) can change this landscape. In FL, as illustrated in Fig. 1, a central server orchestrates the collaborative training process. ① It repeatedly sends the latest in-progress model to randomly selected participants, who update it locally on private data and return model updates for aggregation. This procedure continues until convergence, ② after which the server deploys the trained model. Recent work has demonstrated FL's effectiveness for various vision tasks [9–11,14,24,39,56,58] and diverse applications, such as medical image analysis [19, 33, 57] and robotic manipulation [38]. Such a distributed learning paradigm is increasingly adopted for training vision models across user devices, organizations, and data silos, where raw images cannot be centrally collected due to, e.g., regulatory constraints [52]. However, FL introduces an overlooked attack surface: the in-progress model is continuously exposed to participants throughout training. A malicious participant can ③ obtain a snapshot of the model at some intermediate round, ④ use it offline to craft adversarial examples, and ⑤ send them to the final model deployed in the future. This exposure is unavoidable because FL participants must receive the model in plaintext to perform local training. Given FL's growing importance in real-world vision applications, it is essential to assess the risks posed by model sharing in FL.

This paper takes the first step toward understanding model leakage-induced adversarial vulnerabilities. Intuitively, participants selected later in training receive a near-converged checkpoint, which serves as a stronger surrogate for crafting adversarial examples to manipulate the final deployed model. However, participants do not control when they are selected, especially in cross-device FL, where client participation is sporadic and many clients contribute only once or twice [5]. This raises an intriguing question: *can participants selected early in training still exploit leaked models to launch effective adversarial attacks?*

Our answer is affirmative, even though the model's decision boundary continues to evolve during subsequent training. The key insight is that a leaked intermediate model already encodes structural information about the task: the gradient directions it provides reveal weaknesses that often persist as training

progresses. Building on this observation, we propose Proteus[1], an adversarial attack that uses the leaked model as a starting point and systematically explores temporally persistent perturbation directions. By probing a region of low parameter sensitivity, Proteus identifies perturbations that remain stable under continued training and are therefore more likely to transfer to the final deployed model.

In summary, this paper makes the following original contributions:

- We identify a previously overlooked threat in FL, namely model leakage-induced adversarial vulnerability, and show that intermediate checkpoints exposed during training can be exploited to attack the final deployed model.
- We introduce the notion of temporal transferability, which connects adversarial transferability with FL training dynamics and explains how vulnerabilities can persist and evolve across rounds.
- We propose Proteus, a model leakage-induced adversarial attack that leverages a leaked intermediate model to anticipate the final model's vulnerabilities and construct effective adversarial examples without access to training data.

Proteus delivers substantial improvements across diverse datasets, architectures, and FL configurations. For example, a malicious participant selected as early as the 44th round can achieve an 85.40% attack success rate against a model that converges after 145 rounds, whereas directly using the leaked model yields only 59.75%. We hope this work draws attention to this critical yet underexplored attack surface and motivates the development of dedicated defenses.

2 Background

2.1 Federated Learning

Federated learning (FL) enables collaborative model training across distributed clients without centralizing raw data. To train an ML model $F_{\boldsymbol{\theta}}$ with loss function $\mathcal{L}$ on data distributed across clients, the FL server first initializes the model parameters $\boldsymbol{\theta}_0$. At each learning round t, the server randomly selects a subset of participants and shares the latest in-progress model parameters $\boldsymbol{\theta}_t$ with them. Each participant i uses its private dataset $\boldsymbol{\mathcal{D}}^i$ to optimize the received model for several local epochs, and then submits the updated parameters $\boldsymbol{\theta}_t^i$ to the server. The server aggregates the submissions from all participants via federated averaging (FedAvg) to update the global model parameters, $\boldsymbol{\theta}_{t+1} = \sum_i |\boldsymbol{\mathcal{D}}^i| \boldsymbol{\theta}_t^i / \sum_j |\boldsymbol{\mathcal{D}}^j|$, for the next round of training. This procedure repeats until a termination condition is met, such as reaching a predefined number of rounds or observing no improvement on a server-held validation set for several consecutive rounds. The model parameters obtained in the final round are then deployed.

2.2 Adversarial Attack

Adversarial attacks aim to manipulate an ML model by adding carefully designed, imperceptible perturbations to a benign input. Given a victim model

[1] Source code: `https://github.com/HKU-TASR/Proteus`.

$F_{\boldsymbol{\theta}}$ and an input $\boldsymbol{x}$ with label y, the adversary constructs a perturbed example $\boldsymbol{x}_{\text{adv}} = \boldsymbol{x} + \boldsymbol{\delta}$ such that $F_{\boldsymbol{\theta}}(\boldsymbol{x}_{\text{adv}}) \neq y$, while constraining the perturbation magnitude, e.g., $\|\boldsymbol{\delta}\|_p \leq \epsilon$ under an L_p norm. The small norm of $\boldsymbol{\delta}$ ensures that $\boldsymbol{x}_{\text{adv}}$ remains visually indistinguishable from $\boldsymbol{x}$ to human observers.

When the adversary has direct access to the victim model $F_{\boldsymbol{\theta}}$, near-perfect success rates can easily be achieved via gradient-based optimization [18, 36], commonly formulated as $\max_{\|\boldsymbol{\delta}\|_p \leq \epsilon} \mathcal{L}(F_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}); y)$. Without such access, the adversary may instead exploit the transferability of adversarial examples by crafting them on a surrogate model trained for the same task. However, transfer-based attacks are more challenging and less reliable, as the perturbations must generalize across models with potentially different architectures, parameters, or vulnerabilities.

2.3 Related Work

Transfer-Based Adversarial Attacks. Transfer-based attacks rely on adversarial transferability, the phenomenon that adversarial examples crafted on a surrogate model can also deceive unseen victim models. Early gradient-based methods such as FGSM [18] and iterative variants including I-FGSM [27] and PGD [36] establish the foundation for white-box optimization on surrogate models. Subsequent work improves cross-model transfer by reducing surrogate-specific overfitting, for example using momentum in MI-FGSM [12] and stochastic input transformations in DIM and TIM [13, 55]. Recent methods further enhance transferability with richer perturbation mechanisms, including block shuffle and rotation in BSR [53], deformation-constrained warping across model genus in DeCoWA [32], dual-blending with direction updates in FoolMix [31], hypothesis-space augmentation in OPS [20], and refined integration paths for gradient-based attacks in MuMoDIG [43]. These methods all target cross-model transferability against a single fixed model. Transferring adversarial examples from an early-leaked checkpoint to the final deployed model in FL instead requires cross-time transferability along the training trajectory, and to our knowledge Proteus is the first attack to exploit this temporal dimension. We further find that the advantages of these existing cross-model methods are inconsistent in model leakage-induced attacks in FL.

Model Leakage-Induced Threats in FL. FL's model sharing mechanism introduces privacy and security vulnerabilities. Gradient inversion attacks exploit shared gradients to reconstruct private training data, with methods ranging from optimization-based approaches [16, 59, 66] to techniques that manipulate model parameters to amplify target sample gradients [1, 46, 54, 64]. Other attacks further recover private labels even when only partial or encrypted gradients are shared [63]. Membership inference attacks leverage model updates or predictions to determine whether specific samples were used in training [40, 44, 65]. The shared-model channel also raises integrity concerns: backdoor attacks can implant adversary-controlled behaviors into shared models [6], and backdoor-based watermarking has in turn been proposed to verify client contributions in

FL [35]. Beyond these privacy and integrity threats, recent work has explored adversarial vulnerabilities in FL. Li et al. [30] investigate how adversarial examples crafted on surrogate models trained with compromised client data transfer to the final deployed model, while Kim et al. [25] examine internal evasion attacks in personalized FL where clients sharing model components can exploit each other. Our work differs by adopting a stricter threat model than prior FL attacks assume: the attacker accesses only the intermediate checkpoint that the FL protocol mandates broadcasting to selected participants, with no auxiliary data, queries, or multiple checkpoints. We believe Proteus contributes to an important yet underexplored point in this spectrum of threats.

3 Threat Model

Attacker's Goal. We consider a malicious FL participant who aims to craft adversarial examples that mislead the final deployed model. By introducing human-imperceptible perturbations to benign inputs, the attack causes arbitrary misclassification.

Attacker's Knowledge and Capability. The attacker is an FL participant selected in at least one training round and thus gains access to an intermediate global model, including its architecture and parameters. The attacker faithfully follows the FL protocol and possesses no additional knowledge. The leaked intermediate model is retained and used as a surrogate to generate adversarial examples that are later submitted to the final deployed model.

4 Temporal Transferability

This paper studies a previously overlooked vulnerability connected to a phenomenon we term *temporal transferability* of adversarial examples. It refers to the extent to which an adversarial example generated using an intermediate model leaked at an early stage of FL can deceive the final deployed model, even though the generation process has no access to it.

Fig. 2 shows that if a malicious client naively trains a surrogate model using only its private local data to generate adversarial examples, the resulting transfer-based attack success rate (transfer ASR) is merely 8.35%. This occurs because such a limited amount of data cannot approximate the decision boundary collectively learned by distributed clients. In contrast, the leakage of intermediate models paints a very different picture. Specifically, after only 30% of the training rounds have been completed, the intermediate model can already enable a moderate transfer ASR of 59.75%. This is surprising because one might expect the final deployed model to become robust to such adversarial examples after extensive subsequent training. These observations raise an intriguing question: can this intrinsic temporal transferability be systematically exploited or even amplified to help malicious clients launch effective attacks despite participating only in early rounds?

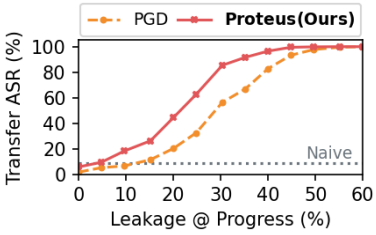
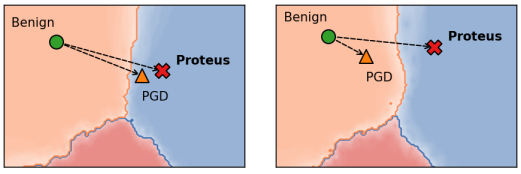


Fig. 2: Adversarial examples generated w.r.t. a leaked intermediate model can transfer temporally and deceive the final deployed model.



(a) Model: 30% Progress **(b)** Model: 100% Progress

Fig. 3: Adversarial examples generated using PGD on (a) the leaked model may fail on (b) the final deployed model due to feature drift during training. In contrast, Proteus promotes temporally persistent adversarial attacks.

We now formalize the adversarial attack setting aimed at boosting temporal transferability. Let $\{F_{\boldsymbol{\theta}_t}\}_{t=1}^{T}$ denote the sequence of intermediate models produced by the FL server during T training rounds. The attacker gains access to a checkpoint at round S, which serves as the surrogate model $F_{\boldsymbol{\theta}_S}$, but has no access to the final deployed model at round T, denoted as $F_{\boldsymbol{\theta}_T}$. The adversarial example generation process therefore aims to find a perturbation $\boldsymbol{\delta}$ for each $(\boldsymbol{x}, y)$ in a test set such that $F_{\boldsymbol{\theta}_T}(\boldsymbol{x}+\boldsymbol{\delta}) \neq y$, despite having no direct access to $F_{\boldsymbol{\theta}_T}$. Intuitively, an attack that achieves strong temporal transferability should satisfy this condition even when $S \ll T$ (i.e., early leakage).

5 Methodology: Proteus

Feature drift is the primary reason why adversarial examples fail to transfer temporally. Fig. 3 illustrates the decision boundaries of (a) an early-leaked model and (b) the final deployed model with three points in the feature space: (•) a benign input, (▲) a PGD-generated adversarial example produced using the leaked model, and (×) an adversarial example generated by our attack using the same leaked model. While the decision boundary converges early and changes only slightly afterward, the feature representations of these points can evolve significantly. Each point processed by the final deployed model yields features that differ from those extracted by the leaked model.

Importantly, the PGD-generated adversarial example initially succeeds in crossing the decision boundary (from orange to blue), but eventually returns to its original side as the model undergoes further training. This illustrates a failure of temporal transferability. In contrast, some adversarial examples remain effective. The Proteus-generated adversarial example continues to cross the boundary even under the final model. Tracking feature representations therefore provides a useful analytical framework for understanding this failure mechanism and identifying adversarial examples that tend to survive subsequent training and remain effective against the final deployed model.

We formalize the feature drift using a parameter-sensitivity framework. Let $h_{\boldsymbol{\theta}}(\cdot)$ denote the feature extractor of $F_{\boldsymbol{\theta}}$. As the model parameters evolve from $\boldsymbol{\theta}_S$

to $\boldsymbol{\theta}_T$, the latent representation drifts from $h_{\boldsymbol{\theta}_S}(\boldsymbol{x}_{\text{adv}})$ to $h_{\boldsymbol{\theta}_T}(\boldsymbol{x}_{\text{adv}})$. Assuming the parameter update $\Delta\boldsymbol{\theta} = \boldsymbol{\theta}_T - \boldsymbol{\theta}_S$ is continuous and that the feature mapping $h_{\boldsymbol{\theta}}$ is smooth around $\boldsymbol{\theta}_S$, a local linear approximation yields

$$h_{\boldsymbol{\theta}_T}(\boldsymbol{x}_{\text{adv}}) \approx h_{\boldsymbol{\theta}_S}(\boldsymbol{x}_{\text{adv}}) + \nabla_{\boldsymbol{\theta}_S} h_{\boldsymbol{\theta}_S}(\boldsymbol{x}_{\text{adv}}) \cdot (\boldsymbol{\theta}_T - \boldsymbol{\theta}_S). \tag{1}$$

Taking the ℓ_2 norm isolates the magnitude of temporal feature drift:

$$\left\| h_{\boldsymbol{\theta}_T}(\boldsymbol{x}_{\text{adv}}) - h_{\boldsymbol{\theta}_S}(\boldsymbol{x}_{\text{adv}}) \right\|_2 \lesssim \underbrace{\left\| \nabla_{\boldsymbol{\theta}_S} h_{\boldsymbol{\theta}_S}(\boldsymbol{x}_{\text{adv}}) \right\|_F}_{H_{\boldsymbol{\theta}_S}(\boldsymbol{x}_{\text{adv}})} \cdot \|\boldsymbol{\theta}_T - \boldsymbol{\theta}_S\|_2. \tag{2}$$

Hence, the term $H_{\boldsymbol{\theta}_S}(\boldsymbol{x}_{\text{adv}})$ serves as a parameter-sensitivity proxy evaluated at the leaked checkpoint. High values of $H_{\boldsymbol{\theta}_S}(\boldsymbol{x}_{\text{adv}})$ indicate sharp local optima in which the representation is highly vulnerable to structural shifts during future training rounds. Suppressing this quantity can bound feature drift and reduce the likelihood of transfer failure.

5.1 Design Overview

Temporal transferability can be promoted by limiting *feature drift*. According to Eq. 2, adversarial examples located in regions with high parameter sensitivity are fragile, as subsequent training updates can significantly alter their feature representations and invalidate the attack. Therefore, we propose Proteus to boost temporal transferability by discovering perturbations that lie in more parameter-insensitive regions of the leaked model. As illustrated in Fig. 4, each iteration of Proteus consists of three phases. Phase 1 discovers the primary gradient direction toward regions with low parameter sensitivity. Phase 2 then strengthens this direction by creating a safety margin against future feature drift. Finally, Phase 3 integrates these exploratory gradients with momentum. Together, these steps guide the optimization toward perturbations that remain adversarial even after subsequent training rounds.

5.2 Phase 1: Persistent Direction Discovery

The parameter-sensitivity analysis provides guidance for identifying gradient directions that remain robust under future training dynamics. In particular, consider a set of transformed inputs $\boldsymbol{\mathcal{X}}$ derived from $\boldsymbol{x}$. For a single input $\boldsymbol{x}$, the parameter sensitivity is defined as $H_{\boldsymbol{\theta}_S}(\boldsymbol{x}_{\text{adv}}) = \|\nabla_{\boldsymbol{\theta}_S} h_{\boldsymbol{\theta}_S}(\boldsymbol{x}_{\text{adv}})\|_F$. Extending this to an ensemble of transformed inputs yields the averaged sensitivity

$$\bar{H}_{\boldsymbol{\theta}_S} = \left\| \frac{1}{|\boldsymbol{\mathcal{X}}|} \sum_{\boldsymbol{x}' \in \boldsymbol{\mathcal{X}}} \nabla_{\boldsymbol{\theta}_S} h_{\boldsymbol{\theta}_S}(\boldsymbol{x}'_{\text{adv}}) \right\|_F \leq \frac{1}{|\boldsymbol{\mathcal{X}}|} \sum_{\boldsymbol{x}' \in \boldsymbol{\mathcal{X}}} \|\nabla_{\boldsymbol{\theta}_S} h_{\boldsymbol{\theta}_S}(\boldsymbol{x}'_{\text{adv}})\|_F, \tag{3}$$

where the second step follows from the triangle inequality. It suggests that aggregating gradients across multiple transformed views can suppress directions

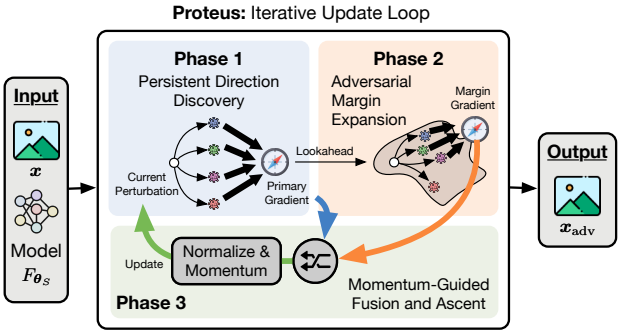


Fig. 4: Proteus consists of three phases that iteratively search for adversarial perturbations that remain effective throughout subsequent training.

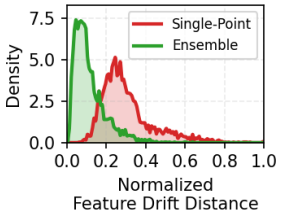


Fig. 5: The ensemble transformation strategy produces adversarial examples with reduced feature drift, leading to stronger temporal persistence.

associated with highly sensitive local features. As a result, optimizing with respect to an ensemble of transformations encourages perturbations with lower effective parameter sensitivity and consequently smaller feature drift. This intuition is empirically validated in Fig. 5, where the ensemble strategy (green) produces adversarial examples with substantially reduced drift compared with single-point estimation (red).

To operationalize this idea, let $\boldsymbol{\mathcal{T}}$ be a set of stochastic input transformation functions. The adversarial gradients with suppressed parameter sensitivity are defined as:

$$\mathcal{G}(\boldsymbol{\delta}; \boldsymbol{x}) = \nabla_{\boldsymbol{\delta}} \mathbb{E}_{\tau \sim \boldsymbol{\mathcal{T}}} \Big[\mathcal{L}\big(F_{\boldsymbol{\theta}_S}(\tau(\boldsymbol{x} + \boldsymbol{\delta})); y\big) \Big]. \tag{4}$$

Based on the above, at iteration k, we obtain the primary adversarial gradients $\boldsymbol{g}_k^{\text{primary}} = \mathcal{G}(\boldsymbol{\delta}_k; \boldsymbol{x})$. Those components tied to brittle, parameter-sensitive features tend to cancel during the averaging process, leaving directions that correspond to more stable adversarial structures.

5.3 Phase 2: Adversarial Margin Expansion

To further establish an explicit margin of robustness against feature drift, Proteus extrapolates the perturbation boundary to verify the stability of the primary ascent direction. It generates a lookahead perturbation by projecting $\boldsymbol{\delta}_k$ along the sign direction of the Phase 1 estimate $\boldsymbol{g}_k^{\text{primary}}$ using a drift step size ρ relative to the attack step size α:

$$\boldsymbol{\delta}_k^{+} = \boldsymbol{\delta}_k + \rho\alpha \cdot \operatorname{sign}\Big(\boldsymbol{g}_k^{\text{primary}}\Big). \tag{5}$$

Proteus then re-estimates the persistent adversarial gradient at the extrapolated coordinate $\boldsymbol{\delta}_k^{+}$ and obtains $\boldsymbol{g}_k^{\text{margin}} = \mathcal{G}(\boldsymbol{\delta}_k^{+}; \boldsymbol{x})$. If $\boldsymbol{\delta}_k$ resides in a sharp optimum governed by high parameter sensitivity, the gradients $\boldsymbol{g}_k^{\text{primary}}$ and $\boldsymbol{g}_k^{\text{margin}}$ will guide it toward a less sensitive region while accounting for the safety margin.

5.4 Phase 3: Momentum-Guided Fusion and Ascent

The final phase fuses the primary and margin gradients by computing the blended gradient $\boldsymbol{g}_k = \boldsymbol{g}_k^{\text{primary}} + \boldsymbol{g}_k^{\text{margin}}$. To prevent gradient magnitude discrepancies across iterations from dominating the ascent direction, Proteus normalizes $\boldsymbol{g}_k$ by the mean absolute value across all d dimensions and applies a momentum accumulator with decay factor μ to bridge optimization steps and filter out high-frequency oscillations:

$$\boldsymbol{m}_k = \mu \boldsymbol{m}_k + \frac{\boldsymbol{g}_k}{\frac{1}{d}\|\boldsymbol{g}_k\|_1}. \tag{6}$$

The in-progress adversarial perturbation $\boldsymbol{\delta}_k$ advances via a projected L_∞ step with size α bounded by the budget ϵ:

$$\boldsymbol{\delta}_{k+1} = \prod_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \Big(\boldsymbol{\delta}_k + \alpha \cdot \text{sign}(\boldsymbol{m}_k) \Big). \tag{7}$$

The next attack iteration then begins until a predefined number of iterations have been completed.

6 Evaluation

We evaluate Proteus along three dimensions: effectiveness (Sec. 6.1), stability across diverse scenarios (Sec. 6.2), and resilience under defenses (Sec. 6.3). We first describe the default setup, provide essential details and variations in the corresponding subsections, and defer additional settings to the appendix.

FL Configuration. We consider an FL setting with 100 clients collaboratively training a ResNet18 [21] model. In each round, 10 clients are randomly selected to participate. They receive the latest intermediate model and perform 5 local training epochs on their private datasets using SGD with a learning rate of 10^{-2} before returning updates to the server. The server aggregates updates using FedAvg. Training terminates when the server-held validation set shows no improvement for 10 consecutive rounds.

Datasets. We evaluate on three benchmarks: CIFAR-10 [26] as a standard baseline, TinyImageNet [8] to represent more complex classification tasks, and RAF-DB [28] as a realistic FL scenario involving privacy-sensitive facial emotion recognition. The training set is randomly partitioned into 100 equal-sized subsets to simulate an IID setting. RAF-DB, CIFAR-10, and TinyImageNet converge in 145, 183, and 153 rounds, respectively. RAF-DB is used as the default dataset.

Attack Setting. All attacks share the same perturbation budget of $\epsilon = 8/255$ under the L_∞ norm. For Proteus, the number of attack iterations is 20, the step size α is $2/255$, the drift step size ρ is 2, the decay factor μ is 0.5, and 20 random transformations are applied per iteration. The stochastic transformation functions are provided in the appendix.

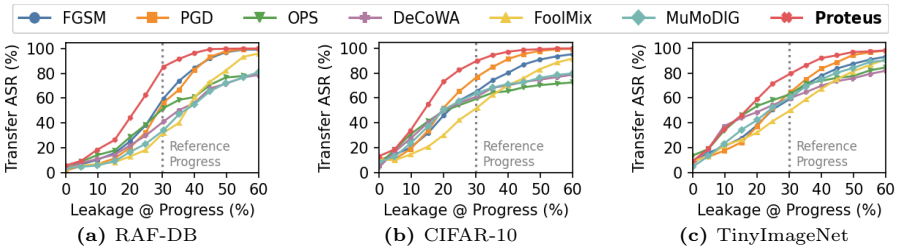


Fig. 6: Proteus achieves the strongest temporal transferability. Using an intermediate model leaked early in training, it generates adversarial examples that deceive the final deployed model with a high attack success rate (ASR).

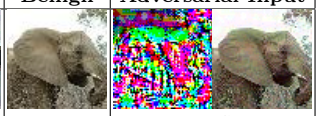
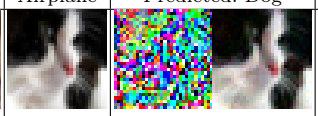
6.1 Effective Improvement on Temporal Transferability

Setup. We compare Proteus against two classic attacks (FGSM [18] and PGD [36]) and four recent transfer-oriented attacks (OPS [20], DeCoWa [32], FoolMix [31], and MuMoDIG [43]) across all three datasets. We obtain leaked intermediate models at different training stages (measured as percentages of total rounds) and generate adversarial examples to evaluate transfer attack success rate (transfer ASR) on the final deployed model.

Main Results. Fig. 6 reports the results, from which we draw three observations. First, once training reaches approximately 50%, intermediate models already yield high transfer ASR, with some attacks, especially Proteus (red), approaching near-perfect performance. Second, even in earlier stages, leaked models remain exploitable. In particular, Proteus achieves the highest transfer ASR when leakage occurs early. For example, after only 30% of training, Proteus attains ASRs of 85.40% on RAF-DB, 90.00% on CIFAR-10, and 79.61% on TinyImageNet, substantially outperforming prior methods. Unless otherwise specified, we use 30% progress as the default leakage point. Third, more complex tasks are inherently more challenging. TinyImageNet involves 200 classes and exhibits more complex training dynamics. Nevertheless, Proteus improves transfer ASR by at least 14.89% over the strongest baseline. We additionally provide two qualitative examples per dataset in Tab. 1, showing that Proteus does not sacrifice stealthiness for temporal transferability. It successfully deceives the final model with human-imperceptible perturbations, even when the attacker is selected only in early training rounds.

Analysis. Proteus outperforms existing methods because it iteratively searches for perturbations aligned with vulnerabilities that persist throughout subsequent training. Under the untargeted objective, this iterative process progressively identifies adversarial examples whose misclassification outcomes match those of the final deployed model. Specifically, Fig. 7 shows that after one attack iteration, only 8.96% of adversarial examples have mispredictions matching those of the final model. This percentage increases to 51.74% after five iterations and reaches 66.67% after all 20 iterations. We further provide GradCAM [45] visu-

Table 1: Proteus maintains high stealthiness while improving temporal transferability. Perturbations generated using a leaked intermediate model remain human-imperceptible while causing misclassification in the final deployed model.

RAF-DB		CIFAR-10		TinyImageNet	
Benign	**Adversarial Input**	**Benign**	**Adversarial Input**	**Benign**	**Adversarial Input**
					
Disgust	Predicted: Happiness	Airplane	Predicted: Dog	Elephant	Predicted: Sombrero
					
Happiness	Predicted: Disgust	Cat	Predicted: Airplane	Teapot	Predicted: Bear

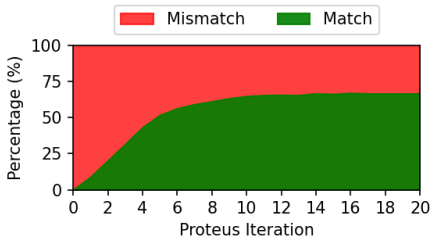
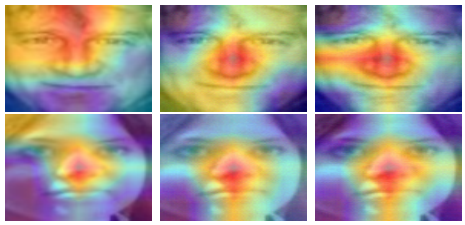


Fig. 7: Proteus progressively generates untargeted adversarial examples whose misclassification outcomes increasingly match those of the final deployed model, using an early-leaked checkpoint.



(a) Benign inputs **(b)** Proteus w/ leaked model **(c)** Proteus w/ final model

Fig. 8: Adversarial examples generated by Proteus using the leaked model (b) deceive the final deployed model in a similar manner (w.r.t. GradCAM) to those generated directly using the final model (c).

alizations in Fig. 8. The attention of the final deployed model on benign inputs (Fig. 8a) shifts progressively as more attack iterations are applied. After 20 iterations (Fig. 8b), adversarial examples generated using the leaked intermediate model induce attention patterns similar to those generated directly using the final deployed model (Fig. 8c). These results indicate that Proteus effectively anticipates the vulnerabilities of the final deployed model using an early-leaked checkpoint.

6.2 Applicable to Diverse Scenarios

Neural Architectures. Proteus's advantages remain consistent across different neural architectures. Fig. 9 reports transfer ASR on three architectures commonly used in FL: ResNet18 (default), MobileNetV3 [22], and EfficientNet [48]. Proteus consistently outperforms the strongest baseline. On ResNet18, it achieves an ASR of 85.40%, exceeding FGSM by 25.65%. On MobileNetV3, it reaches 77.38%, outperforming OPS by 20.05%. On EfficientNet, it achieves

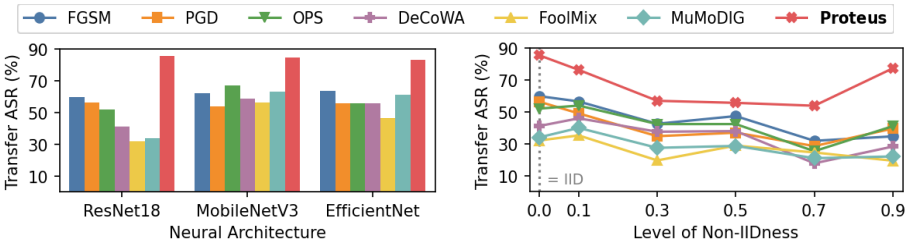
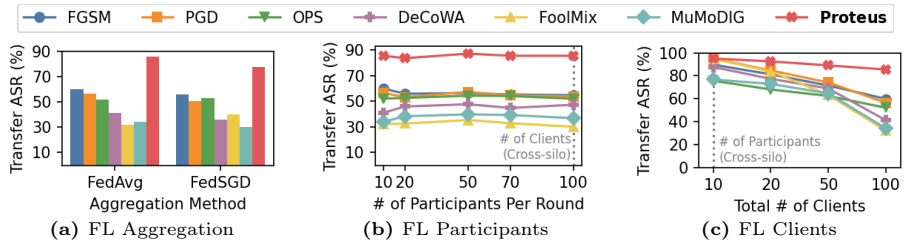


Fig. 9: Proteus consistently outperforms baseline attacks across different neural architectures used in FL.

Fig. 10: Proteus enhances temporal transferability across diverse client data distributions in FL, including non-IID settings.



(a) FL Aggregation **(b)** FL Participants **(c)** FL Clients

Fig. 11: Proteus maintains strong performance across diverse FL configurations, including (a) aggregation methods, (b) numbers of participating clients per round, and (c) total numbers of clients. Despite different training dynamics, Proteus consistently achieves higher transfer ASR than all baselines.

82.99%, surpassing FGSM by 15.55%. These results demonstrate that Proteus effectively identifies persistent vulnerabilities despite architecture-dependent training dynamics.

Data Distributions. FL clients may not share identical data distributions, and it is essential for an attack to remain effective under this realistic setting. To simulate non-IID scenarios, we partition data using a Dirichlet distribution with concentration parameter α [23] and define the level of non-IIDness as $1-\alpha$. A lower level leads to more uniform distributions, whereas a higher level yields more skewed allocations. As shown in Fig. 10, Proteus consistently achieves higher transfer ASR than all baselines, particularly under both highly skewed and relatively uniform settings. We highlight that Proteus is a data-free approach. It does not require training data to identify persistent vulnerabilities, which is infeasible to obtain in FL, especially when gathering data with representative distributions under non-IID scenarios.

FL Configurations. FL settings can also affect training dynamics. Proteus should remain robust to these variations and be able to pinpoint persistent vulnerabilities. We analyze three key configurations. First, the choice of aggregation protocol does not significantly affect Proteus. Fig. 11a shows that the transfer ASR under FedAvg only drops from 85.40% to 77.38% under FedSGD,

even though the latter introduces noisier updates and makes training dynamics more unpredictable. Second, varying the number of participating clients per round does not change the overall conclusion. Fig. 11b shows that, with 100 total clients and different numbers of participants, Proteus consistently achieves higher transfer ASR. Third, the total number of FL clients also plays an important role. As shown in Fig. 11c, when the number of clients is small and comparable to the number of participants (e.g., 10), all attacks perform well due to faster convergence. As the number increases, existing attacks degrade significantly, whereas Proteus remains competitive. Overall, Proteus demonstrates strong stability across diverse FL configurations.

6.3 High Resilience under Defenses

Setup. Proteus targets a previously unexplored attack surface in FL, and existing defenses provide limited mitigation. We evaluate five representative defenses: two classical approaches (differential privacy [17] and randomized smoothing [7]) and three recent adversarial purification methods (DiffPure [41], ScoreOpt [60], and ADBM [29]). These defenses are selected because they can be applied without making unrealistic assumptions under our threat model, such as requiring potentially malicious clients to cooperatively train robust models.

Main Results. Fig. 12 reports clean accuracy and transfer ASR under each defense. An effective defense should preserve clean accuracy while reducing transfer ASR. We observe two key findings. First, most defenses substantially degrade clean accuracy, especially purification-based methods (e.g., DiffPure reduces accuracy from 71.58% to 22.27%, ScoreOpt to 18.80%, and ADBM to 14.60%). These methods assume access to sufficient in-distribution data for training purifiers, whereas the FL server typically holds only a small validation set. Even when using a substitute dataset with a similar distribution (e.g., CelebA [34]), clean accuracy remains severely degraded. Second, differential privacy preserves clean accuracy relatively well (dropping from 71.58% to 67.87%), yet Proteus remains effective, with ASR decreasing from 85.40% to 59.64%. Because noise must remain small to maintain accuracy, Proteus can still identify persistent adversarial directions. In contrast, baseline attacks are more sensitive to noise, with ASR dropping substantially (e.g., FGSM decreases to 35.04%). The strong resilience of Proteus highlights the need for dedicated defenses tailored to model leakage-induced adversarial threats. Existing adversarial training methods for FL [4, 50, 62, 67] are inapplicable here, as they assume cooperative clients, which contradicts our malicious participant threat model.

6.4 Additional Analysis

Ablation Studies. We conduct ablation studies to demonstrate that each component of Proteus contributes to strong transfer ASR. Tab. 2 reports results when individual phases are removed. Removing Phase 1 causes the most significant ASR drop, from 85.40% to 75.82%, highlighting the importance of the

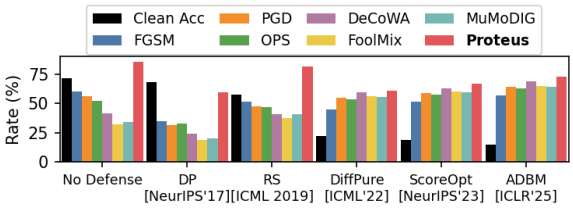


Fig. 12: Existing defenses either significantly degrade clean accuracy or fail to effectively protect FL models from model leakage-induced adversarial attacks.

Table 2: Ablation study showing that each phase of Proteus contributes to achieving strong transfer ASR.

Variants	ASR (%)
Without Phase 1	75.82
Without Phase 2	82.23
Without Phase 3	83.17
Full Proteus	**85.40**

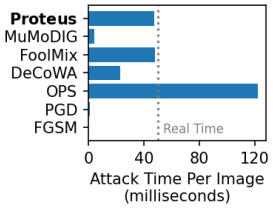


Fig. 13: Proteus operates in near real-time, about 50 ms per image.

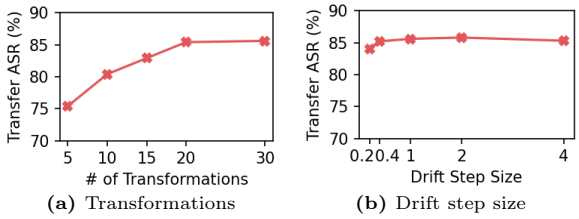


Fig. 14: Proteus is relatively insensitive to hyperparameter choices. The same configuration is used across all three datasets.

primary gradients. Removing Phase 2 or Phase 3 also reduces ASR, though less dramatically, indicating that refinement via the safety margin provides additional but secondary improvements.

Attack Efficiency. All experiments were conducted on a single NVIDIA GeForce RTX 4090 GPU (24 GB) with an Intel Xeon Gold 5418Y CPU and 64 GB RAM, using PyTorch 2.8.0 with CUDA 12.8. As shown in Fig. 13, Proteus requires 47.78 ms per image, comparable to FoolMix and significantly faster than OPS (121.90 ms). Notably, Proteus can operate in near real time (approximately 50 ms per image [42]).

Hyperparameters. Proteus is relatively insensitive to its hyperparameters. Fig. 14 shows transfer ASR when varying (a) the number of transformations and (b) the drift step size ρ. Performance saturates once approximately 20 transformations are used to search for persistent gradients, a trend consistent across all three datasets.

7 Conclusions

We presented Proteus, the first adversarial attack targeting model leakage-induced vulnerabilities unique to FL systems. By leveraging a leaked intermediate model, Proteus anticipates vulnerabilities that persist throughout subsequent training by suppressing parameter sensitivity. Experiments demonstrate that even a ma-

licious client selected in early rounds can launch effective attacks against the final deployed model across diverse FL scenarios and under existing defenses. Given the lack of dedicated defenses for this threat, we recommend implementing early stopping and restricting late-stage participation to trusted clients. While these measures are only temporary, we anticipate that future work will develop principled defenses against this realistic and previously overlooked threat.

Acknowledgments. This research is partially supported by the RGC Early Career Scheme (Project #27211524) and Croucher Start-up Allowance (Project #2499102828). Any opinions, findings, or conclusions expressed in this material are those of the authors and do not necessarily reflect the views of RGC and the Croucher Foundation.

References

1. Boenisch, F., Dziedzic, A., Schuster, R., Shamsabadi, A.S., Shumailov, I., Papernot, N.: When the curious abandon honesty: Federated learning is not private. In: 2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P). pp. 175–199. IEEE (2023)
2. Carlini, N.: Some lessons from adversarial machine learning. `https://www.youtube.com/watch?v=umfeF0Dx-r4`, last accessed 2026/06/26 (2024)
3. Chen, B., Yin, J., Chen, S., Chen, B., Liu, X.: An adaptive model ensemble adversarial attack for boosting adversarial transferability. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4489–4498 (2023)
4. Chen, C., Liu, Y., Ma, X., Lyu, L.: Calfat: Calibrated federated adversarial training with label skewness. Advances in Neural Information Processing Systems (NeurIPS) **35**, 3569–3581 (2022)
5. Chen, D., Gao, D., Xie, Y., Pan, X., Li, Z., Li, Y., Ding, B., Zhou, J.: Fs-real: Towards real-world cross-device federated learning. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD). pp. 3829–3841 (2023)
6. Chow, K.H., Wei, W., Yu, L.: Imperio: Language-guided backdoor attacks for arbitrary model control. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI). pp. 704–712 (2024)
7. Cohen, J., Rosenfeld, E., Kolter, Z.: Certified adversarial robustness via randomized smoothing. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 97, pp. 1310–1320. PMLR (2019)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 248–255. Ieee (2009)
9. Di, Z., Zhu, Z., Li, X., Liu, Y.: Federated learning with local openset noisy labels. In: European Conference on Computer Vision (ECCV). pp. 38–56. Springer (2024)
10. Dong, J., Zhang, D., Cong, Y., Cong, W., Ding, H., Dai, D.: Federated incremental semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3934–3943 (2023)

11. Dong, X., Zhang, S.Q., Li, A., Kung, H.: Spherefed: Hyperspherical federated learning. In: European Conference on Computer Vision (ECCV). pp. 165–184. Springer (2022)
12. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9185–9193 (2018)
13. Dong, Y., Pang, T., Su, H., Zhu, J.: Evading defenses to transferable adversarial examples by translation-invariant attacks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4312–4321 (2019)
14. Gao, L., Fu, H., Li, L., Chen, Y., Xu, M., Xu, C.Z.: Feddc: Federated learning with non-iid data via local drift decoupling and correction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10112–10121 (2022)
15. Garg, S., Ramakrishnan, G.: Bae: Bert-based adversarial examples for text classification. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP). pp. 6174–6181 (2020)
16. Geiping, J., Bauermeister, H., Dröge, H., Moeller, M.: Inverting gradients-how easy is it to break privacy in federated learning? Advances in Neural Information Processing Systems (NeurIPS) **33**, 16937–16947 (2020)
17. Geyer, R.C., Klein, T., Nabi, M.: Differentially private federated learning: A client level perspective. arXiv preprint arXiv:1712.07557 (2017)
18. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
19. Guo, P., Wang, P., Zhou, J., Jiang, S., Patel, V.M.: Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2423–2432 (2021)
20. Guo, Y., Liu, W., Xu, Q., Zheng, S., Huang, S., Zang, Y., Shen, S., Wen, C., Wang, C.: Boosting adversarial transferability through augmentation in hypothesis space. In: Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR). pp. 19175–19185 (2025)
21. 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 (CVPR). pp. 770–778 (2016)
22. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1314–1324 (2019)
23. Hsu, T.M.H., Qi, H., Brown, M.: Measuring the effects of non-identical data distribution for federated visual classification. arXiv preprint arXiv:1909.06335 (2019)
24. Kim, G., Kim, J., Han, B.: Communication-efficient federated learning with accelerated client gradient. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12385–12394 (2024)
25. Kim, T., Singh, S., Madaan, N., Joe-Wong, C.: Characterizing internal evasion attacks in federated learning. In: International Conference on Artificial Intelligence and Statistics (AISTATS). pp. 907–921. PMLR (2023)
26. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

27. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: Artificial intelligence safety and security, pp. 99–112. Chapman and Hall/CRC (2018)
28. Li, S., Deng, W., Du, J.: Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2852–2861 (2017)
29. Li, X., Sun, W., Chen, H., Li, Q., He, Y., Shi, J., Hu, X.: Adbm: Adversarial diffusion bridge model for reliable adversarial purification. In: The Thirteenth International Conference on Learning Representations (ICLR) (2025)
30. Li, Y., Gao, Y., Wang, H.: Towards understanding adversarial transferability in federated learning. Transactions on Machine Learning Research (TMLR) (2023)
31. Li, Z., Wang, W., Li, J., Chen, K., Zhang, S.: Foolmix: Strengthen the transferability of adversarial examples by dual-blending and direction update strategy. IEEE Transactions on Information Forensics and Security (TIFS) **19**, 5286–5300 (2024)
32. Lin, Q., Luo, C., Niu, Z., He, X., Xie, W., Hou, Y., Shen, L., Song, S.: Boosting adversarial transferability across model genus by deformation-constrained warping. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). vol. 38, pp. 3459–3467 (2024)
33. Liu, Q., Chen, C., Qin, J., Dou, Q., Heng, P.A.: Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1013–1023 (2021)
34. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (December 2015)
35. Luo, K., Chow, K.H.: Harmless backdoor-based client-side watermarking in federated learning. In: IEEE European Symposium on Security and Privacy (2025)
36. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (ICLR) (2018)
37. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. Pmlr (2017)
38. Miao, C., Chang, T., Wu, M., Xu, H., Li, C., Li, M., Wang, X.: Fedvla: Federated vision-language-action learning with dual gating mixture-of-experts for robotic manipulation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6904–6913 (2025)
39. Miao, J., Yang, Z., Fan, L., Yang, Y.: Fedseg: Class-heterogeneous federated learning for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8042–8052 (2023)
40. Mo, W., Li, Z., Fang, M., Fang, M.: Find a scapegoat: Poisoning membership inference attack and defense to federated learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3967–3976 (2025)
41. Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., Anandkumar, A.: Diffusion models for adversarial purification. In: International Conference on Machine Learning (ICML) (2022)
42. Qin, Z., Li, Z., Zhang, Z., Bao, Y., Yu, G., Peng, Y., Sun, J.: Thundernet: Towards real-time generic object detection on mobile devices. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6718–6727 (2019)

43. Ren, Y., Zhao, Z., Lin, C., Yang, B., Zhou, L., Liu, Z., Shen, C.: Improving integrated gradient-based transferable adversarial examples by refining the integration path. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). vol. 39, pp. 6731–6739 (2025)
44. Samira, D., Habler, E., Elovici, Y., Shabtai, A.: Variance-based membership inference attacks against large-scale image captioning models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9210–9219 (2025)
45. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 618–626 (2017)
46. Shan, J., Zhao, Z., Lu, J., Zhang, R., Yiu, S.M., Chow, K.H.: Geminio: Language-guided gradient inversion attacks in federated learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2718–2727 (2025)
47. Simonetto, T., Ghamizi, S., Cordy, M.: Tabularbench: Benchmarking adversarial robustness for tabular deep learning in real-world use-cases. Advances in Neural Information Processing Systems (NeurIPS) **37**, 78394–78430 (2024)
48. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning (ICML). pp. 6105–6114. PMLR (2019)
49. Tang, B., Wang, Z., Bin, Y., Dou, Q., Yang, Y., Shen, H.T.: Ensemble diversity facilitates adversarial transferability. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 24377–24386 (2024)
50. Tang, M., Wang, Y., Zhang, J., DiValentin, L., Ding, A., Hass, A., Chen, Y., Li, H.H.: Fedprophet: Memory-efficient federated adversarial training via robust and consistent cascade learning. Proceedings of Machine Learning and Systems (MLSys) **7** (2025)
51. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness may be at odds with accuracy. In: International Conference on Learning Representations (ICLR) (2019)
52. Voigt, P., Von dem Bussche, A.: The eu general data protection regulation (gdpr). A practical guide, 1st ed., Cham: Springer International Publishing **10**(3152676), 10–5555 (2017)
53. Wang, K., He, X., Wang, W., Wang, X.: Boosting adversarial transferability by block shuffle and rotation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 24336–24346 (2024)
54. Wen, Y., Geiping, J.A., Fowl, L., Goldblum, M., Goldstein, T.: Fishing for user data in large-batch federated learning via gradient magnification. In: International Conference on Machine Learning (ICML). pp. 23668–23684. PMLR (2022)
55. Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., Yuille, A.L.: Improving transferability of adversarial examples with input diversity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2730–2739 (2019)
56. Xiong, Y., Wang, R., Cheng, M., Yu, F., Hsieh, C.J.: Feddm: Iterative distribution matching for communication-efficient federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16323–16332 (2023)

57. Xu, A., Li, W., Guo, P., Yang, D., Roth, H.R., Hatamizadeh, A., Zhao, C., Xu, D., Huang, H., Xu, Z.: Closing the generalization gap of cross-silo federated medical image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20866–20875 (2022)
58. Yi, L., Yu, H., Wang, G., Liu, X., Li, X.: Federated representation angle learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1314–1324 (2025)
59. Yin, H., Mallya, A., Vahdat, A., Alvarez, J.M., Kautz, J., Molchanov, P.: See through gradients: Image batch recovery via gradinversion. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
60. Zhang, B., Luo, W., Zhang, Z.: Enhancing adversarial robustness via score-based optimization. Advances in Neural Information Processing Systems (NeurIPS) **36**, 51810–51829 (2023)
61. Zhang, C., Zhou, L., Xu, X., Wu, J., Liu, Z.: Adversarial attacks of vision tasks in the past 10 years: A survey. ACM Computing Surveys **58**(2), 1–42 (2025)
62. Zhang, J., Li, B., Chen, C., Lyu, L., Wu, S., Ding, S., Wu, C.: Delving into the adversarial robustness of federated learning. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). vol. 37, pp. 11245–11253 (2023)
63. Zhang, R., Chow, K.H.: GDBR: Label recovery attack against partial gradient encryption in federated learning. In: 2026 IEEE 11th European Symposium on Security and Privacy (EuroS&P). IEEE (2026)
64. Zhang, R., Guo, S., Li, P.: Gradfilt: Class-wise targeted data reconstruction from gradients in federated learning. In: Companion Proceedings of the ACM on Web Conference 2024 (WWW). pp. 698–701 (2024)
65. Zhu, G., Li, D., Gu, H., Yao, Y., Fan, L., Han, Y.: Fedmia: An effective membership inference attack exploiting" all for one" principle in federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20643–20653 (2025)
66. Zhu, L., Liu, Z., Han, S.: Deep leakage from gradients. In: Advances in Neural Information Processing Systems (NeurIPS) (2019)
67. Zizzo, G., Rawat, A., Sinn, M., Buesser, B.: Fat: Federated adversarial training. arXiv preprint arXiv:2012.01791 (2020)