TOWARDS ROBUST PERSONALIZED FEDERATED LEARNING: VULNERABILITY ASSESSMENT AND DEFENSE CO-DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

While Federated Learning (FL) preserves data privacy, its robustness to transfer-based adversarial attacks remains largely unexplored. We investigate this gap within Personalized FL (PFL)—a paradigm designed for data heterogeneity—and reveal its heightened vulnerability to such attacks. Both theoretically and empirically, we demonstrate that malicious clients can exploit local models to generate adversarial examples that severely compromise peer models. To mitigate this, we propose a novel defense framework integrating stochastic input noise, input-scaled trace regularization, and parameter sensitivity maximization. This paper presents the first systematic study of transfer adversarial threats in PFL, offering rigorous diagnostic insights and effective countermeasures.

1 INTRODUCTION

Federated learning (FL) (Chen et al., 2022; Zhang et al., 2021) enables privacy-preserving collaborative training by keeping data localized. However, canonical FL methods like FedAvg (McMahan et al., 2017) struggle with the data heterogeneity of real-world environments (Liao et al., 2024; Wang et al., 2022). To address this one-model-fits-all limitation, personalized federated learning (PFL) (Collins et al., 2021; Li et al., 2021b;a) has emerged, allowing clients to train customized models tailored to their local data distributions while retaining the benefits of collaborative knowledge exchange.

Orthogonal to personalization, securing FL against inherent vulnerabilities remains critical. The distributed nature of FL exposes broad attack surfaces—including model poisoning (Blanchard et al., 2017), backdoor injection (Xie et al., 2020; Wang et al., 2020), and privacy breaches (Zhu et al., 2019)—prompting defenses like secure aggregation (Dowlin et al., 2016; Zhang et al., 2020) and differential privacy (Niu et al., 2020). Systematically exploring novel attack vectors is essential for diagnosing system vulnerabilities and designing robust, trust-aware frameworks.

To our knowledge, this paper presents the first systematic study of transferable adversarial attacks in PFL (Goodfellow et al., 2015; Madry et al., 2018), orthogonal to prior adversarial PFL work on spurious-feature generalization (Wang et al., 2024). We demonstrate that malicious clients can weaponize their local models to craft transferable adversarial examples that compromise the personalized models of peer clients. While related to traditional black-box transfer attacks (Dong et al., 2018; Fan et al., 2023), our attack operates in a distinct gray-box PFL setting, where clients share identical model architectures but maintain different parameters.

Contribution I: systematic vulnerability assessment. We conduct extensive empirical evaluations revealing that PFL implementations are highly susceptible to adversarial examples. We observe attack success rates (ASRs) significantly higher than those of traditional transfer-based attacks. Furthermore, we identify unique phenomena within PFL—such as a weak correlation between ASR and proxy model accuracy—suggesting distinct vulnerability patterns that amplify attack effectiveness.

Contribution II: Theoretical framework for intrinsic vulnerabilities. We establish a theoretical framework for PFL transfer attacks, identifying five determinants of attack effectiveness: model sensitivity, loss landscape geometry, distribution alignment, fitness difference, and parameter sensi-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tivity. Empirical validation confirms that PFL models exhibit degraded characteristics across these factors compared to centralized models, explaining their heightened susceptibility.

Contribution III: robustness-enhanced PFL framework. Leveraging these theoretical insights, we propose three novel, plug-and-play techniques: stochastic input noise augmentation, input-scaled trace regularization, and parameter sensitivity maximization. Extensive experiments demonstrate that our framework consistently improves robustness against transfer-based attacks across diverse PFL methods.

2 Background and Related Work

2.1 Personalized Federated Learning

Federated learning Consider an FL system with m clients and a central server. Let $\mathcal{U} = \{1, 2, \ldots, m\}$ denote the client set, where each client holds a private dataset D_i drawn from a local distribution P_{XY}^i, with input space $\mathcal{X}$ and K-class label space $\mathcal{Y}$. Given a loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, canonical FL (McMahan et al., 2017) learns a global model $F(x; \theta_g)$ by minimizing: $\min_{\theta_g} \sum_{i \in \mathcal{U}} \mathbb{E}_{(x,y)\sim D_i} [\mathcal{L}(F(x; \theta_g), y).]$ FedAvg (McMahan et al., 2017) optimizes this equation via two alternating steps: 1) In round t, a sampled client subset $S_t \subseteq \mathcal{U}$ synchronizes the global model ($\theta_g^i \leftarrow \theta_g$) and performs τ epochs of SGD on D_i: $\theta_g^i = \theta_g^i - \eta \nabla_{\theta_g^i} \mathcal{L}(F(x; \theta_g^i), y),\ (x, y) \sim D_i$, where η is the learning rate, and 2) the server updates the global model by averaging local updates: $\theta_g = \frac{1}{|S_t|} \sum_{i \in S_t} \theta_g^i$. However, FedAvg suffers severe performance degradation under data heterogeneity (Liao et al., 2024; Wang et al., 2022; Li et al., 2021b) (i.e., $P_{XY}^i \neq P_{XY}^j$ for $i \neq j$). Personalized FL (PFL) addresses this by allowing clients to maintain private, personalized models θ_i.

Full model-sharing methods Full model-sharing methods allow all parameters of θ_i to deviate from θ_g^i, constrained by a regularization term $\mathcal{R}$:

$$\min_{\theta_i, i \in \mathcal{U}} \sum_{i \in \mathcal{U}} \mathbb{E}_{(x,y)\sim D_i} \left[\mathcal{L}(F(x; \theta_i), y) + \mathcal{R}(\theta_i, \theta_g^i) \right].$$

Here, $\mathcal{L}$ drives local adaptation, while $\mathcal{R}$ retains global knowledge by aligning the personalized and global models. For instance, FedProx (Li et al., 2020a) uses an L_2-norm penalty $||\theta_i - \theta_g^i||_2^2$ for $\mathcal{R}$. Ditto (Li et al., 2021a) pre-trains θ_g^i before optimizing θ_i with a similar proximity constraint, and SCAFFOLD (Karimireddy et al., 2020) introduces control variates to calibrate client updates.

Partial model-sharing methods Conversely, partial model-sharing methods enforce hard parameter sharing for a specific parameter subset Λ, allowing the remainder to specialize:

$$\min_{\theta_i, i \in \mathcal{U}} \sum_{i=1}^{m} \mathbb{E}_{(x,y)\sim D_i} \left[\mathcal{L}(F(x; \theta_i), y) \right], s.t.,\ \theta_g^i[k] = \theta_i[k],\ k \in \Lambda,\ i \in \mathcal{U}.$$

By sharing layers that capture transferable representations and isolating client-specific patterns, these methods often achieve superior performance. FedBN (Li et al., 2021b) localizes batch normalization layers while aggregating the rest. Other approaches personalize classification heads: FedRep (Collins et al., 2021) and FedBABU (Oh et al., 2022) share feature extractors but retain local heads, while FedPAC (Xu et al., 2023) enhances them via weighted ensembles. Additionally, FedAS (Yang et al., 2024) aligns feature representations prior to head training, GPFL (Zhang et al., 2023) explicitly disentangles global and local features, and FedCAC (Wu et al., 2023) dynamically determines Λ based on parameter sensitivity.

2.2 Adversarial Attack

Given an input x and ground-truth label y, an adversarial perturbation δ is crafted by solving: $\delta = \arg\max_{\delta} \mathcal{L}(F(x + \delta; \theta), y),\ s.t.\ ||\delta||_\infty \leq \epsilon$ where a higher loss indicates a more effective attack, and ϵ constraints the perturbation magnitude to ensure imperceptibility. Early adversarial attacks primarily targeted white-box settings, assuming full model access. A benchmark method,

PGD (Madry et al., 2018), solves this formulation via iterative projected gradient ascent over K steps with random initialization:$\delta = \Pi_{[-\epsilon,\epsilon]} \left(\delta + \alpha \cdot \text{sign}\left(\nabla_\delta \mathcal{L}\left(F(x+\delta;\theta), y\right)\right)\right)$ where $\Pi_{[-\epsilon,\epsilon]}$ projects δ onto the feasible l_∞-norm ball and α is the step size. However, relying on exact target model gradients limits their applicability in real-world scenarios.

Transfer-based attacks To bypass the white-box requirement, transfer-based attacks exploit adversarial examples crafted on a proxy model to compromise black-box targets. Their success hinges on cross-model transferability. While applying PGD to a proxy model yields limited transferability, recent techniques significantly enhance it. These methods largely modify PGD and fall into three categories: optimization-based, input regularization, and model-based. Optimization methods, such as integrating momentum (Dong et al., 2018) or calibrating gradients (Wang & He, 2021), stabilize update directions to escape suboptimal local maxima. Input regularization techniques apply stochastic transformations—like random resizing (Xie et al., 2019), scaling (Lin et al., 2020), translation (Dong et al., 2019), or masking (Fan et al., 2025)—during perturbation generation. Finally, model-based methods alter gradient backpropagation (Wu et al., 2020) or design proxy architectures that better simulate target models (Li et al., 2020b; Fan et al., 2023).

Adversarial robustness in PFL Wang et al. (2024) combine adversarial robustness with PFL from a generalization perspective: personalized models fine-tuned on local data tend to exploit spurious features, and adversarial transferability between the global and personalized models is maintained during fine-tuning as a regularization signal. In contrast, we study deliberate post-training attacks on PFL, where transferability is the threat itself: we theoretically characterize its determinants, quantify PFL-specific vulnerability patterns, and co-design a plug-and-play defense.

3 Adversarial Attack in PFL

3.1 Attack Scenario

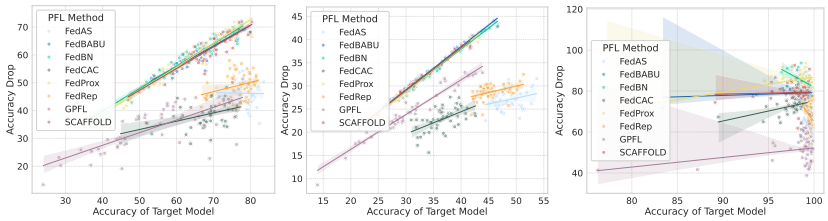
We consider a scenario where multiple clients train personalized models using PFL. Our attack occurs during inference (post-training), assuming the attacker can interact with the target model (e.g., via black-box APIs or physical access) without knowing its internal parameters or training data. For example, in a distributed IoT surveillance network, an attacker could exploit one camera's personalized intrusion detection model to craft adversarial examples that undermine models on peer devices. Notably, both the server and clients strictly adhere to PFL protocols without malicious training-time behavior. We identify two distinct post-training attack vectors:

- **Client-initiated attacks:** Malicious clients craft adversarial examples using their local personalized models to produce adversarial examples against other clients' personalized models.
- **Server-initiated attacks:** The server generates adversarial examples through the global model to target clients' personalized models.

Evaluation criteria. Let $\delta = \mathcal{A}(x, y, \theta)$ denote the perturbation generated by attack $\mathcal{A}$ on input (x, y) using parameters θ. We quantify the attack effectiveness from client i to client j via the accuracy drop ($AD_{i\to j}$): $AD_{i\to j} = \frac{\sum_{(x_k,y_k)\in D_j}[\mathbb{I}(F(x_k;\theta_j)=y_k)-\mathbb{I}(F(x_k+\delta_k;\theta_j)=y_k)]}{|D_j|}$ where $\delta_k = \mathcal{A}(x_k, y_k, \theta_i)$ and $\mathbb{I}$ is the indicator function. This measures the accuracy degradation of client j's model (θ_j) against adversarial examples crafted via client i's model (θ_i). For server-initiated attacks, we substitute θ_i with θ_g. Crucially, this metric is evaluated exclusively on the victim's test set D_j to prevent artificial inflation from out-of-distribution attacker inputs. We assess the overall system vulnerability of a PFL method using the mean cross-client AD: $\frac{1}{(|\mathcal{U}|-1)^2}\sum_{i,j\in\mathcal{U},i\neq j} AD_{i\to j}$. We explicitly exclude white-box cases ($i = j$), reflecting the practical constraint that attackers lack target model parameter access.

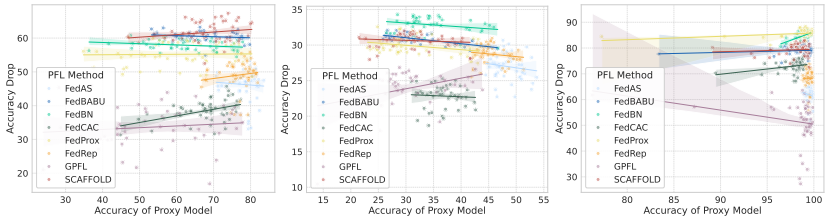
3.2 Setup

FL configurations. We evaluate our attacks using ResNet-10 on three benchmark datasets: CIFAR-10, CIFAR-100, and GTSRB. *The evaluation results for other models can be found in Supplementary Materials.* To simulate real-world statistical heterogeneity, we partition data using a Dirichlet dis-



(a) CIFAR-10 (b) CIFAR-100 (c) GTSRB

Figure 1: Correlation between target model accuracy (%) and AD (%) in three datasets.



(a) CIFAR-10 (b) CIFAR-100 (c) GTSRB

Figure 2: Correlation between proxy model accuracy (%) and AD (%) in three datasets.

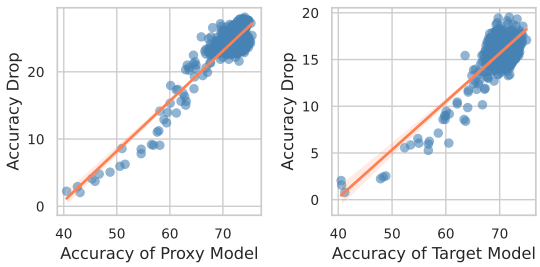


Figure 3: Centralized learning comparison. Left: AD versus proxy model accuracy with fixed target model. Right: AD versus target model accuracy with fixed proxy model.

tribution $Dir(\beta)$ with $\beta = 0.5$ by default. Each client i receives a $q_{c,i} \sim Dir(\beta)$ proportion of class c samples (Zhang et al., 2023), inducing realistic non-IID label skew and quantity imbalance. Our FL system comprises 10 clients with a 20% participation rate over 50 communication rounds (empirically verified for convergence). All clients optimize both local and personalized models via SGD with a batch size of 32, a learning rate of 0.1, and 5 local epochs.

PFL methods. We employ eight state-of-the-art PFL methods, including FedProx (Li et al., 2020a), SCAFFOLD (Karimireddy et al., 2020), FedBN (Li et al., 2021b), FedRep (Collins et al., 2021), FedBABU (Oh et al., 2022), GPFL (Zhang et al., 2023), FedAS (Yang et al., 2024), and FedCAC (Wu et al., 2023). The first two ones are full model sharing, while the remaining ones are partial model sharing.

Attacks. We generate adversarial examples on the global and personalized models using PGD (Madry et al., 2018), with a perturbation budget of $\epsilon = 8/255$ and 10 iterations by default. To analyze the impact on attack effectiveness, we subsequently integrate state-of-the-art transferability-enhancing techniques spanning three distinct paradigms: optimization-based (MI (Dong et al., 2018), which stabilizes updates via momentum to escape local optima), model-based (PCIFGSM (Wan & Huang, 2023), which adapts gradient computation), and input regular-

ization (MaskBlock (Fan et al., 2025), which applies input masking). All experiments run across 5 independent trials with varying random seeds to ensure statistical significance.

3.3 Evaluation Results

Target model accuracy and attack effectiveness We first investigate the correlation between the performance of personalized models and their adversarial vulnerability. Figure 1 plots per-client model accuracy against mean AD (computed by averaging $AD_{i\to j}$ over all source clients i) under client-initiated attacks. Each PFL method comprises 50 data points, representing five independent experimental trials, with linear regression fits included to quantify statistical trends. We note two key observations:

- **Considerable adversarial vulnerability of PFL methods.** All personalized models exhibit substantial susceptibility to adversarial attacks, showing accuracy drops of $\geq$20%, $\geq$10%, and $\geq$40% on CIFAR-10, CIFAR-100, and GTSRB, respectively. Notably, the regression lines reveal a clear positive correlation between target model accuracy and adversarial vulnerability, indicating an accuracy-vulnerability trade-off inherent to PFL. Exceptions occur in specific cases (e.g., FedAS on CIFAR-10 and all methods on GTSRB), where limited accuracy variance among client models likely obscures measurable trends.
- **Partial-sharing methods enjoy better accuracy-vulnerability trade-off.** At comparable accuracy levels, partial-sharing PFL methods (FedRep, FedAS, GPFL, FedCAC) demonstrate greater adversarial robustness than full-sharing methods. On CIFAR-10, for instance, FedRep achieves 70%–80% accuracy with an AD of approximately 50%, outperforming full-sharing baselines like FedProx (40%–80% accuracy, 40%–70% AD).

Proxy model accuracy and attack effectiveness Figure 2 examines the relationship between proxy model accuracy and attack transferability (i.e., the average $AD_{i\to j}$ over target clients j evaluated on their respective test sets). **We observe a counter-intuitive phenomenon: adversarial transferability in PFL is largely independent of proxy model accuracy.** The AD remains remarkably consistent even across proxy models with substantially different accuracy levels. This suggests that any client can weaponize its local personalized model against arbitrary peer clients, regardless of its own model's performance.

Intuitively, one might expect higher proxy accuracy to correlate with better feature alignment between the proxy and target models, thereby enhancing transferability. To contextualize our findings, we conduct a centralized learning comparison (where all client data is pooled to train a single model). Here, the proxy and target models share identical architectures (ResNet-10) and training configurations (CIFAR-10, SGD with $\eta = 0.1$), differing only in initialization:

- **Fixed target model**: We attack a trained target model (with 75% accuracy) using proxy models taken from different training iterations.
- **Fixed proxy model**: We use a trained proxy model to generate adversarial examples and evaluate their effectiveness in the target model across different training iterations.

The results in Figure 3 (over 5 trials) reveal a strong positive correlation between AD and both proxy and target model accuracy in centralized learning. The stark contrast between centralized training and PFL stresses that proxy-accuracy-independent transferability constitutes a distinct vulnerability inherent to PFL.

Client-initiated attacks versus server-initiated attacks Server-initiated attacks achieve negligible AD against partial-sharing methods that retain local classification heads (FedRep, FedBABU, and GPFL), because the global model lacks the personalized parameters required to craft effective adversarial examples; for the remaining methods, the gap between client- and server-initiated attacks is marginal (Table 4 in Appendix C). Since partial model-sharing naturally mitigates server-initiated attacks—rendering them less threatening overall—our subsequent analyses focus exclusively on client-initiated attacks.

Training dynamics and adversarial robustness Figures 5$\sim$8 (Appendix C) further show that the accuracy drop grows monotonically with training rounds, perturbation budgets, and attack iterations

in both PFL and centralized training, while advanced transfer-based attacks yield diminished returns against PFL ($\leq 2\%$) compared with centralized training ($\approx 8\%$).

4 What makes adversarial attack so effective in PFL?

We here conduct a theoretical investigation into the intrinsic properties of PFL that amplify this susceptibility. Consider two clients, i and j, with local data distributions $P_{XY}^{(i)}$ and $P_{XY}^{(j)}$ and corresponding personalized model parameters θ_i and θ_j. We analyze adversarial transferability by treating client i's model as the proxy and client j's model as the target.

For input-label pair (x, y) with perturbation δ, we expand the loss function via first-order Taylor approximation: $\mathcal{L}(F(x+\delta, \theta_i), y) = \mathcal{L}(F(x, \theta_i), y) + \nabla_x \mathcal{L}(F(x, \theta_i), y)^T \delta$. The perturbation that maximizes the loss is given by $\alpha \nabla_x \mathcal{L}(F(x, \theta_i), y)$, where α is a scaling constant ensuring the linear approximation remains valid. The adversarial effect of δ on client j's model is then:

$$\mathcal{L}\big(F(x+\delta, \theta_j), y\big) = \mathcal{L}\big(F(x, \theta_j), y\big) + \alpha \nabla_x \mathcal{L}\big(F(x, \theta_j), y\big)^T \nabla_x \mathcal{L}\big(F(x, \theta_i), y\big).$$

This reveals that a higher gradient inner product implies stronger attack effectiveness. Notably, this conclusion generalizes to arbitrary perturbation magnitudes δ, provided the models maintain consistent gradient directions. We employ Wasserstein distance to quantify the distributional divergence between $P_{XY}^{(i)}$ and $P_{XY}^{(j)}$, i.e., $W(P_{XY}^{(i)}, P_{XY}^{(j)})$. Under Assumptions 1$\sim$3, Theorem 1 establishes a lower bound for gradient inner product, with the full proof provided in Supplementary Material.

Assumption 1. *The loss function $\mathcal{L}(F(x; \theta_i), y)$ is K-Lipschitz continuous for some constant $K \geq 0$.*

Assumption 2. *The personalized model has converged on its local data distribution $P_{XY}^{(j)}$, i.e., $\mathbb{E}_{P_{XY}^{(j)}}[\nabla_{\theta_j} \mathcal{L}(F(x; \theta_j), y)] = 0$.*

Assumption 3. *Client j's loss function $\mathcal{L}(F(x; \theta_j), y)$ is strongly convex in model parameters θ_j.*

Theorem 1. *(See supplementary material for the proof) Let λ_{min} be the minimal eigenvalue of $\nabla^2_{\theta_j} \mathcal{L}(F(x, \theta_j), y)$ and let $\Delta = \theta_i - \theta_j$ represent the parameter difference. The gradient inner product is bounded as follows:*

$$\begin{aligned}\nabla_x \mathcal{L}(F(x; \theta_i), y)^\top \nabla_x \mathcal{L}(F(x; \theta_j), y) \geq & \frac{1}{2}\big\{||\nabla_x \mathcal{L}(F(x; \theta_i), y)||_2^2 + ||\nabla_x \mathcal{L}(F(x; \theta_j), y)||_2^2\big\} \\ & - C||\nabla^2_{x,\theta_j} \mathcal{L}(F(x; \theta_j), y)||_2^2 + \mathcal{O}(||\Delta||_2),\end{aligned}$$

where $C = \frac{1}{\lambda_{min}}\{K \cdot W(P_{XY}^{(i)}, P_{XY}^{(j)}) - \mathbb{E}_{P_{XY}^{(i)}}[\mathcal{L}(F(x; \theta_i), y)] + \mathbb{E}_{P_{XY}^{(j)}}[\mathcal{L}(F(x; \theta_j), y)]\}$.

Before analyzing Theorem 1, we discuss its applicability within the PFL context. Assumption 1 is generally valid, as unbounded Lipschitz constants imply pathological loss landscapes. Assumption 2 is a standard convergence premise. Assumption 3 holds strictly for linear models with cross-entropy loss; however, because many PFL methods (e.g., partial-sharing) isolate parameter divergence to the classification heads, this assumption remains practically relevant. Furthermore, upon convergence, neural network loss landscapes are often locally convex near the optimum.

In Theorem 1, the term $\mathcal{O}(||\Delta||_2)$ becomes negligible when the parameter discrepancy between personalized models is small. Because PFL inherently regularizes these differences—either softly or via hard sharing—Δ is typically constrained. Therefore, Theorem 1 identifies several critical determinants of transferability in PFL:

- **Model sensitivity** $||\nabla_x \mathcal{L}(F(x; \theta_i), y)||_2^2 + ||\nabla_x \mathcal{L}(F(x; \theta_j), y)||_2^2$: Higher input gradient norms indicate that the models are highly sensitive to input perturbations, directly amplifying adversarial transferability.
- **Loss landscape geometry** λ_{min}: The minimal eigenvalue λ_{min} characterizes the curvature of the loss landscape along its flattest direction. A substantial λ_{min} indicates significant curvature in all directions, making the target model highly sensitive to input perturbations: even minor adversarial changes will induce pronounced loss variations, thereby enhancing transferability.

- **Distribution alignment** $KW(P_{XY}^{(i)}, P_{XY}^{(j)})$ **and model fitness difference** $-\mathbb{E}_{P_{XY}^{(i)}}\left[\mathcal{L}(F(x;\theta_i),y)\right] + \mathbb{E}_{P_{XY}^{(j)}}\left[\mathcal{L}(F(x;\theta_j),y)\right]$: The constant C is driven by the Wasserstein distance between the clients' data distributions and the difference in their expected losses. A smaller Wasserstein distance suggests overlapping underlying feature distributions. Furthermore, if the models achieve comparable fitness (similar expected losses) on their respective datasets, their learned representations will align closely, facilitating higher transferability.
- **Parameter sensitivity** $||\nabla^2_{x,\theta_j}\mathcal{L}(F(x;\theta_j),y)||_2^2$: This mixed derivative measures how sensitive the input gradients are to changes in the model parameters. A smaller value indicates that the input gradient remains invariant despite parameter shifts across clients. If parameter sensitivity approaches zero, the input gradient behavior decouples from the specific model parameterization. Consequently, regions of high input sensitivity (i.e., adversarial vulnerabilities) remain consistent across different personalized models, allowing adversarial examples crafted on the proxy to transfer effectively to the target.

We empirically validate these determinants across PFL clients: attack effectiveness correlates with the average Hessian eigenvalue (-0.176), the proxy-target fitness gap (-0.218), and parameter sensitivity (0.109), and strengthens as client data distributions become more aligned. The full validation protocol, together with a quantitative comparison between PFL and centralized training, is provided in Appendix B.

5 How to mitigate adversarial attack in PFL?

5.1 Robustness-enhanced PFL Framework

Overview Building upon our theoretical foundations, we introduce a robustness-enhanced PFL framework featuring three key techniques: stochastic input noise augmentation, input-scaled trace regularization, and parameter sensitivity minimization. These techniques are designed to regularize input gradients, λ_{min}, and parameter sensitivity, respectively. Moreover, these techniques are plug-and-play and thus are compatible with arbitrary PFL methods, making our framework a universal robustness enhancement solution for both existing and emergent PFL methods. Taking client i as a representative example, its original personalized model optimization objective is formulated as: $\min_{\theta_i}\ \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\left[\mathcal{L}(F(x;\theta_i),y)\right]$, where we omit the parameter-sharing constraints for clarity.

Stochastic input noise augmentation This technique mitigates the model's sensitivity to input perturbations by regularizing $||\nabla_x\mathcal{L}(F(x;\theta_i),y)||_2^2$. Because direct optimization involves computationally prohibitive second-order gradients in high-dimensional spaces (Boyd & Vandenberghe, 2004), we propose an efficient approximation based on Gaussian noise injection. By adding Gaussian noise $\upsilon \sim \mathcal{N}(0,\sigma^2 I)$ to input samples, we analyze the first-order Taylor expansion of the perturbed loss function:

$$\mathcal{L}(F(x+\upsilon;\theta_i),y) = \mathcal{L}(F(x;\theta_i),y) + \upsilon^T\nabla_x\mathcal{L}(F(x;\theta_i),y),$$

where higher-order terms are neglected for small σ. Rearranging the equation and multiplying both sides by υ:

$$\upsilon\upsilon^T\nabla_x\mathcal{L}(F(x;\theta_i),y) = \upsilon[\mathcal{L}(F(x+\upsilon;\theta_i),y) - \mathcal{L}(F(x;\theta_i),y)].$$

Taking the expectation over υ and leveraging $\mathbb{E}_\upsilon[\upsilon\upsilon^T] = \sigma^2 I$:

$$\begin{aligned}\sigma^2\nabla_x\mathcal{L}(F(x;\theta_i),y) &= \mathbb{E}_\upsilon[\upsilon\upsilon^T\nabla_x\mathcal{L}(F(x;\theta_i),y)]\\ &= \mathbb{E}_\upsilon\{\upsilon[\mathcal{L}(F(x+\upsilon;\theta_i),y) - \mathcal{L}(F(x;\theta_i),y)]\}.\end{aligned}$$

For practical implementation, we use a single-sample Monte Carlo approximation: $\nabla_x\mathcal{L}(F(x;\theta_i),y) = \left[\mathcal{L}(F(x+\upsilon;\theta_i),y) - \mathcal{L}(F(x;\theta_i),y)\right]\upsilon/\sigma^2$. Then, taking the norm on both sides yields: $||\nabla_x\mathcal{L}(F(x;\theta_i),y)||_2 = \left|\mathcal{L}(F(x+\upsilon;\theta_i),y) - \mathcal{L}(F(x;\theta_i),y)\right|\,||\upsilon||_2/\sigma^2$.

Input-scaled trace regularization This technique aims to increase λ_{min}. We consider the classification head for two reasons: it is often the most personalized component, as feature extraction layers are shared across clients in many PFL methods, and its Hessian admits a closed-form solution (detailed below), avoiding the intractable complexity of full-network Hessian evaluations.

Table 1: The performance of the proposed robustness-enhanced framework in CIFAR-10.

PFL Method	FedAS		FedBABU		FedBN		FedCAC	
Metric	Acc	AD	Acc	AD	Acc	AD	Acc	AD
Vanilla	79.05	47.61	66.72	57.79	67.39	62.94	62.00	36.19
Ours	79.32	40.57	78.94	30.72	77.85	53.00	72.62	28.16
Ours+AT	69.36	8.46	70.49	8.42	66.90	17.12	62.47	11.03
Ours+AT+RS	51.35	3.57	54.78	3.51	49.46	6.69	47.78	5.94
PFL Method	FedProx		FedRep		GPFL		SCAFFOLD	
Metric	Acc	AD	Acc	AD	Acc	AD	Acc	AD
Vanilla	57.46	55.14	76.83	50.38	51.08	23.52	65.51	60.34
Ours	68.34	46.01	78.19	42.84	60.97	18.37	74.79	50.21
Ours+AT	53.27	17.02	68.19	11.50	48.26	14.19	59.41	18.11
Ours+AT+RS	31.86	4.91	48.66	4.36	34.62	3.40	38.42	5.62

Table 2: The performance of our framework against state-of-the-art transfer-based attacks in CIFAR-10.

Method	Attack	FedAS	FedBABU	FedBN	FedCAC	FedProx	FedRep	GPFL	SCAFFOLD
Ours+AT	PGD	8.46	8.42	17.12	11.03	17.02	11.50	14.19	18.11
	MI	10.38	10.42	19.73	12.19	20.53	13.89	21.10	21.78
	PCIFGSM	10.41	10.46	19.80	12.23	20.58	13.94	21.12	21.87
	MaskBlock	11.25	11.22	20.90	13.02	21.59	14.90	22.84	23.13
Ours+AT+RS	PGD	3.57	3.51	6.69	5.94	4.91	4.36	3.40	5.62
	MI	4.25	4.35	7.62	6.54	5.45	5.40	5.75	7.59
	PCIFGSM	4.28	4.40	7.61	6.58	5.46	5.44	5.74	7.64
	MaskBlock	4.74	4.85	8.52	7.17	6.00	6.03	6.74	8.56

Let z denote the feature embedding from feature extraction layers and let W represent the weights of classification head, such that the model output is given by $F(x;\theta_i) = W^T z$. For cross-entropy loss, we have: $\mathcal{L}(F(x;\theta_i), y) = -\sum_{k=1}^{K} \mathbb{I}(y=k)\log p_k$. where $p_k = \frac{\exp(w_k^T z)}{\sum_{j=1}^{K}\exp(w_j^T z)}$ and w_k is k-th column of W. Then, the Hessian H with respect to W is given by: $H(m,n) = \frac{\partial \mathcal{L}}{\partial w_m \partial w_n} = zz^T p_m(\mathbb{I}(m=n) - p_n)$. Although explicit matrix construction is feasible, full spectral decomposition becomes computationally intensive for large models. Instead, we maximize the trace, which equals the sum of its eigenvalues ($tr(H) = \sum_{i=1}^{N}\lambda_i$), thereby encouraging a larger sum of eigenvalues and helping increase the minimal eigenvalue. For classification head, the trace can be simplified into a remarkably intuitive form:

$$tr(H) = \underbrace{||z||_2^2}_{\text{feature scaling}} \cdot \underbrace{\sum_{i=1}^{K} p_i(1-p_i)}_{\text{prediction uncertainty}} . \tag{1}$$

The prediction uncertainty term penalizes ambiguous distributions to encourage confident predictions, while the feature scaling term restricts embedding magnitudes to regulate the model's Lipschitz constant.[1]. Together, Equation 1 indeed promotes confident and accurate predictions without significantly increasing the model's Lipschitz constant.

Parameter sensitivity maximization This technique enhances parameter sensitivity to input data. For computational tractability, we focus on the parameter sensitivity of classification head. The second-order derivative $\nabla^2_{z,w}\mathcal{L}$ of classification head enjoys a closed-form expression as follows:

$$\nabla^2_{z,w}\mathcal{L} = (p-y)\otimes I + z \otimes [W(\text{diag}(p) - pp^T)],$$

where $p = (p_i)$, $\otimes$ denotes outer product, I is identity matrix, and, diag$(\cdot)$ converts the input vector into a diagonal matrix. By incorporating the negative L_2 norm of this term into the loss function,

[1]The Lipschitz constant measures the maximum rate at which the function can change with respect to small changes in its input. A smaller Lipschitz constant implies that the function is less sensitive to input perturbations, thereby enhancing the model's robustness. By constraining the magnitude of feature embeddings, we can effectively limit the Lipschitz constant, making the model more stable.

we force the model to learn features where the input gradient is highly specific to the local head's unique parameters. This localizes the vulnerability landscape, severely hindering the cross-client transferability of adversarial examples.

Final formulation Incorporating these techniques, client i's complete optimization objective can be formulated:

$$\begin{aligned}
&\min_{\theta_i} \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\big\{\mathcal{L}(F(x;\theta_i),y) + \beta_1 \cdot ||\nabla_x \mathcal{L}(F(x;\theta_i),y)||_2 + \\
&\beta_2 \cdot tr(H) - \beta_3 \cdot ||\nabla^2_{z,w}\mathcal{L}(F(x;\theta_i),y)||_2\big\} \\
&= \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\big\{(1 - \frac{\gamma\beta_1||\upsilon||_2}{\sigma^2})\mathcal{L}(F(x;\theta_i),y) \\
&+ \frac{\gamma\beta_1||\upsilon||_2}{\sigma^2}\mathcal{L}(F(x+\upsilon;\theta_i),y) + \beta_2||z||_2^2 \cdot \sum_{i=1}^{K} p_i(1-p_i) \\
&- \beta_3||(p-y)\otimes I + z \otimes [W(\text{diag}(p) - pp^T)||_2\},
\end{aligned} \tag{2}$$

where $\beta_1, \beta_2, \beta_3$ control the regularization strengths. γ is a binary indicator that equals 1 if $\mathcal{L}(F(x+\upsilon;\theta_i),y) \geq \mathcal{L}(F(x;\theta_i),y)$ and -1 otherwise.

5.2 Evaluation

Setup Experiments follow Section 3.2; we report clean accuracy (Acc) on unperturbed local data and use 20 attack iterations. **Component ablation studies, computational overhead analysis, and extended evaluations across additional datasets and architectures are deferred to the supplementary material, which also includes our source code.**

The effectiveness of the proposed framework FedBABU's AD drops from 57.79% to 30.72% ($\sim$27%) on CIFAR-10, with similar improvements across other methods (e.g., 9.94% for FedBN and 8.03% for FedCAC). Crucially, our framework also improves Acc for most PFL methods, most strikingly FedBABU (66.72% to 78.94%), achieving a dual benefit of robustness and clean accuracy.

Integration with common adversarial defenses We further integrate our framework with two standard defenses: adversarial training (AT) (Madry et al., 2018; Bai et al., 2021) and random smoothing (RS) (Jia et al., 2020); implementation details are provided in the supplementary material. As Table 1 shows, adding AT yields dramatic AD reductions (FedBABU: 8.42% on CIFAR-10) at the cost of an 8% accuracy drop, while RS provides moderate additional reductions with further generalization costs. This trade-off is expected, as extreme robustness sacrifices clean accuracy.

The robustness against transfer-based attacks Finally, we evaluate robustness against state-of-the-art transfer-based attacks (Table 2). Their AD increase over standard PGD is constrained within 5%, marginal compared to vanilla PFL's baseline vulnerability (often exceeding 50% AD, Figure 5), showing that our framework neutralizes the vulnerabilities exploited by advanced transfer techniques.

6 Conclusion

We conducted a systematic exploration of adversarial vulnerability in PFL systems, identifying five key determinants of attack transferability that provide interpretable robustness metrics. To mitigate these risks, we introduced a lightweight yet effective defense framework, which considerably improves PFL robustness without compromising performance.

Reproducibility statement

The FL configurations, datasets, heterogeneous data partitioning, and attack settings are documented in Section 3.2, and the defense hyperparameters are $\beta_1 = \beta_2 = \beta_3 = 0.01$ and $\sigma^2 = 1$. The complete proof of Theorem 1 is provided in Appendix A; the empirical validation of the theoretical

determinants and the extended attack analyses (server-initiated attacks and training dynamics) are provided in Appendices B and C; the extended evaluations across model architectures and datasets (Tables 6 and 7), the component ablation study (Table 9), and the computational overhead analysis (Table 8) are provided in Appendix D. Our source code is included in the supplementary material and will be made publicly available upon acceptance.

AI USE STATEMENT

In this work, we used generative AI tools for creating and modifying parts of the experimental software code (e.g., the federated training pipeline, the attack evaluation scripts, and plotting utilities), which was verified and tested by the authors through repeated runs across all PFL methods and datasets. We have not used generative AI tools for formulating the theoretical claims, assisting in the writing of the proofs, designing the attack scenarios or defense methodology, or interpreting the experimental results. Additionally, we used generative AI tools for editing the manuscript to improve readability and for formatting references. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the vulnerability of personalized federated learning to transfer-based adversarial attacks with the primary goal of strengthening system robustness. All experiments use standard public benchmark datasets (CIFAR-10, CIFAR-100, and GTSRB) that contain no personally identifiable information, and no human subjects are involved. The attack analysis is presented for diagnostic purposes under standard threat models, and it is accompanied by an effective, plug-and-play defense framework; we believe that exposing these vulnerabilities is a necessary step toward designing trust-aware federated systems. We will release the defense components to support practical deployment.

REFERENCES

Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356*, 2021.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Convex Optimization, 2004.

Suo Chen, Yang Xu, Hongli Xu, Zhida Jiang, and Chunming Qiao. Decentralized federated learning with intermediate results in mobile edge computing. *IEEE Transactions on Mobile Computing*, 23(1):341–358, 2022.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pp. 2089–2099. PMLR, 2021.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9185–9193, 2018.

Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4307–4316, 2019.

Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin E. Lauter, Michael Naehrig, and John Robert Wernsing. Cryptonets: applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*, 2016.

Mingyuan Fan, Wenzhong Guo, Zuobin Ying, and Ximeng Liu. Enhance transferability of adversarial examples with model architecture. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Mingyuan Fan, Cen Chen, Ximeng Liu, and Wenzhong Guo. Maskblock: Transferable adversarial examples with bayes approach. 2025.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

Jinyuan Jia, Xiaoyu Cao, Binghui Wang, and Neil Zhenqiang Gong. Certified robustness for top-k predictions against adversarial perturbations via randomized smoothing. 2020. URL https://api.semanticscholar.org/CorpusID:59842968.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pp. 5132–5143. PMLR, 2020.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020a.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, pp. 6357–6368. PMLR, 2021a.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b. URL https://openreview.net/forum?id=6YEQUn0QICG.

Yingwei Li, Song Bai, Yuyin Zhou, Cihang Xie, Zhishuai Zhang, and Alan Yuille. Learning transferable adversarial examples via ghost networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 11458–11465, 2020b.

Yunming Liao, Yang Xu, Hongli Xu, Zhiwei Yao, Liusheng Huang, and Chunming Qiao. Parallelsfl: A novel split federated learning framework tackling heterogeneity issues. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pp. 845–860, 2024.

Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. *arXiv: Learning*, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Xiaojin (Jerry) Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017. URL http://proceedings.mlr.press/v54/mcmahan17a.html.

Chaoyue Niu, Fan Wu, Shaojie Tang, Lifeng Hua, Rongfei Jia, Chengfei Lv, Zhihua Wu, and Guihai Chen. Billion-scale federated learning on mobile clients: A submodel design with tunable privacy. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pp. 1–14, 2020.

Jaehoon Oh, SangMook Kim, and Se-Young Yun. Fedbabu: Toward enhanced representation for federated image classification. In *International Conference on Learning Representations*, 2022.

Chen Wan and Fangjun Huang. Adversarial attack based on prediction-correction. *arXiv preprint arXiv:2306.01809*, 2023.

Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jyyong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems*, 33:16070–16084, 2020.

Lun Wang, Yang Xu, Hongli Xu, Min Chen, and Liusheng Huang. Accelerating decentralized federated learning in heterogeneous edge computing. *IEEE Transactions on Mobile Computing*, 22(9):5001–5016, 2022.

Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1924–1933, 2021.

Xiaoyang Wang, Han Zhao, Klara Nahrstedt, and Sanmi Koyejo. Personalized federated learning with spurious features: An adversarial approach. *Transactions on Machine Learning Research*, 2024.

Dongxian Wu, Yisen Wang, Shutao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. *ArXiv*, abs/2002.05990, 2020.

Xinghao Wu, Xuefeng Liu, Jianwei Niu, Guogang Zhu, and Shaojie Tang. Bold but cautious: Unlocking the potential of personalized federated learning through cautiously aggressive collaboration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 19375–19384, 2023.

Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. DBA: distributed backdoor attacks against federated learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=rkgyS0VFvr`.

Cihang Xie, Zhishuai Zhang, Jianyu Wang, Yuyin Zhou, Zhou Ren, and Alan Loddon Yuille. Improving transferability of adversarial examples with input diversity. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2725–2734, 2019.

Jian Xu, Xinyi Tong, and Shao-Lun Huang. Personalized federated learning with feature alignment and classifier collaboration. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL `https://openreview.net/forum?id=SXZr8aDKia`.

Xiyuan Yang, Wenke Huang, and Mang Ye. Fedas: Bridging inconsistency in personalized federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11986–11995, 2024.

Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In *USENIX Annual Technical Conference*, 2020.

Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, Jian Cao, and Haibing Guan. Gpfl: Simultaneously learning global and personalized feature information for personalized federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5041–5051, 2023.

Wuyang Zhang, Zhezhi He, Luyang Liu, Zhenhua Jia, Yunxin Liu, Marco Gruteser, Dipankar Raychaudhuri, and Yanyong Zhang. Elf: accelerate high-resolution mobile deep vision with content-aware parallel offloading. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pp. 201–214, 2021.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

Summary of Supplementary Material:

- Section A: Proof of Theorem 1: Details the complete mathematical derivation bounding the gradient inner product, formally linking adversarial transferability to model sensitivity, loss landscape geometry, distribution alignment, and parameter sensitivity.
- Section B: Empirical Validation (Figure 4): Corroborates our theoretical findings by analyzing the correlation between attack effectiveness and the identified determinants (e.g., gradient norms, Hessian eigenvalues). It includes Table 3 which quantitatively contrasts the intrinsic vulnerabilities of PFL against centralized training.
- Section C: Extended Attack Analyses: Provides the full server-initiated attack results (Table 4) and the training-dynamics analysis across attack methods, perturbation budgets, and iterations (Figures 5∼8).
- Section D: Extended Evaluations: Provides comprehensive additional experiments for our robustness-enhanced framework. This includes robustness assessments across deeper architectures (Table 5) and additional datasets like CIFAR-100 and GTSRB (Tables 6 & 7), an analysis of computational training overhead (Table 8), and an ablation study isolating the impact of each proposed regularization component (Table 9).

A Proof of Theorem 1

Proof. **Step 1: Gradient Difference Analysis.** Let $\Delta = \theta_i - \theta_j$. For gradient alignment, consider the following input gradient difference:

$$\begin{aligned}
&\nabla_x \mathcal{L}(F(x;\theta_i),y) - \nabla_x \mathcal{L}(F(x;\theta_j),y) \\
=&\nabla_x \mathcal{L}(F(x;\theta_j+\Delta),y) - \nabla_x \mathcal{L}(F(x;\theta_j),y) \\
=&\nabla_x \mathcal{L}(F(x;\theta_j),y) + \nabla^2_{x,\theta_j} \mathcal{L}(F(x;\theta_j),y)^T \Delta \\
&-\nabla_x \mathcal{L}(F(x;\theta_j),y) + \mathcal{O}(||\Delta||_2) \\
=&\nabla^2_{x,\theta_j} \mathcal{L}(F(x;\theta_j),y)^T \Delta + \mathcal{O}(||\Delta||_2).
\end{aligned}$$

Squaring both sides and applying Cauchy-Schwarz:

$$\begin{aligned}
&||\nabla_x \mathcal{L}(F(x;\theta_i),y)||_2^2 + ||\nabla_x \mathcal{L}(F(x;\theta_j),y)||_2^2 \\
&- 2\nabla_x \mathcal{L}(F(x;\theta_i),y)^\top \nabla_x \mathcal{L}(F(x;\theta_j),y) + \mathcal{O}(||\Delta||_2) \\
&\leq ||\nabla^2_{x,\theta_j} \mathcal{L}(F(x;\theta_j),y)||_2^2 \, ||\Delta||_2^2.
\end{aligned}$$

Rearranging the above equation yields:

$$\begin{aligned}
&||\nabla_x \mathcal{L}(F(x;\theta_i),y)||_2^2 + ||\nabla_x \mathcal{L}(F(x;\theta_j),y)||_2^2 \\
&- ||\nabla^2_{x,\theta_j} \mathcal{L}(F(x;\theta_j),y)||_2^2 ||\Delta||_2^2 + \mathcal{O}(||\Delta||_2) \\
&\leq 2\nabla_x \mathcal{L}(F(x;\theta_i),y)^\top \nabla_x \mathcal{L}(F(x;\theta_j),y).
\end{aligned} \tag{3}$$

We next derive the bound of $||\Delta||_2^2$.

Step 2: Bounding Δ. Using the Kantorovich-Rubinstein duality, the Wasserstein distance between distributions admits the dual form: $K \cdot W(P^{(i)}_{XY}, P^{(j)}_{XY}) = \sup_{||f||_L \leq K} \mathbb{E}_{P^{(i)}_{XY}}[f(x,y)] - \mathbb{E}_{P^{(j)}_{XY}}[f(x,y)]$, where $||f||_L \leq K$ denotes K-Lipschitz continuity. For the model-specific loss function, this implies: $K \cdot W(P^{(i)}_{XY}, P^{(j)}_{XY}) \leq \mathbb{E}_{P^{(i)}_{XY}}[\mathcal{L}(F(x;\theta_i),y)] - \mathbb{E}_{P^{(j)}_{XY}}[\mathcal{L}(F(x;\theta_i),y)]$. We expand the loss difference through Taylor approximation:

$$\begin{aligned}
&K \cdot W(P^{(i)}_{XY}, P^{(j)}_{XY}) - \mathbb{E}_{P^{(i)}_{XY}}[\mathcal{L}(F(x;\theta_i),y)] \\
\leq& -\mathbb{E}_{P^{(j)}_{XY}}[\mathcal{L}(F(x;\theta_i),y)] = -\mathbb{E}_{P^{(j)}_{XY}}[\mathcal{L}(F(x;\theta_j+\Delta),y)] \\
=& -\mathbb{E}_{P^{(j)}_{XY}}\Big[\mathcal{L}(F(x;\theta_j),y) + \nabla_{\theta_j} \mathcal{L}(F(x;\theta_j),y)^\top \Delta \\
&+ \Delta^\top \nabla^2_{\theta_j} \mathcal{L}(F(x;\theta_j),y)\Delta + \mathcal{O}(||\Delta||_2^2)\Big].
\end{aligned}$$

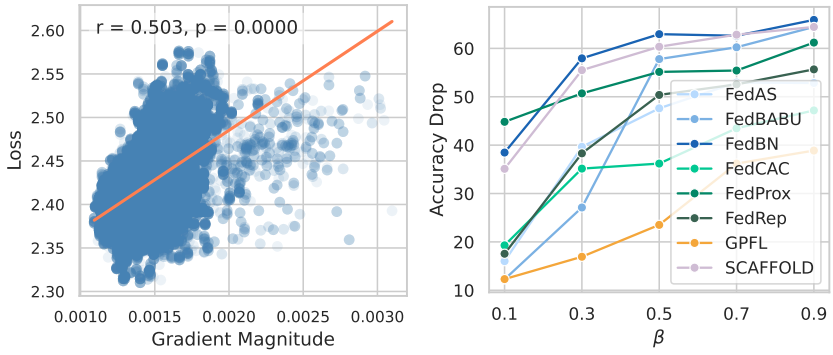


Figure 4: **Left:** The sum of the input gradient norms on the proxy and target models, along with the loss values of the generated adversarial examples on the target model. **Right:** The performance of transfer-based attacks with varying degrees of data heterogeneity. We use ResNet-10 and CIFAR-10.

Using local optimality of θ_j on $P_{XY}^{(j)}$ gives:

$$
\begin{aligned}
&- \mathbb{E}_{P_{XY}^{(i)}}\left[\mathcal{L}(F(x;\theta_i),y)\right] + \mathbb{E}_{P_{XY}^{(j)}}\left[\mathcal{L}(F(x;\theta_j),y)\right] \\
&+ K \cdot W(P_{XY}^{(i)}, P_{XY}^{(j)}) + \mathcal{O}(||\Delta||_2^2) \\
&\leq -\mathbb{E}_{P_{XY}^{(j)}}\left[\Delta^\top \nabla_{\theta_j}^2 \mathcal{L}(F(x;\theta_j),y)\Delta\right].
\end{aligned}
$$

Under strong convexity, the quadratic term satisfies $-\Delta^\top \nabla_{\theta_j}^2 \mathcal{L}(F(x;\theta_j),y)\Delta \leq -\lambda_{\min}||\Delta||_2^2$. Thus, we have:

$$
\begin{aligned}
&- \mathbb{E}_{P_{XY}^{(i)}}\left[\mathcal{L}(F(x;\theta_i),y)\right] + \mathbb{E}_{P_{XY}^{(j)}}\left[\mathcal{L}(F(x;\theta_j),y)\right] \\
&+ KW(P_{XY}^{(i)}, P_{XY}^{(j)}) + \mathcal{O}(||\Delta||_2^2) \leq -\lambda_{\min}||\Delta||_2^2.
\end{aligned} \tag{4}
$$

Step 3: Combining Step 1 and Step 2. Substituting Equation 3 into Equation 4 through $||\Delta||_2^2$ yields:

$$
\begin{aligned}
&\mathcal{O}(||\Delta||) + ||\nabla_x \mathcal{L}(F(x;\theta_i),y)||_2^2 + ||\nabla_x \mathcal{L}(F(x;\theta_j),y)||_2^2 \\
&- \frac{||\nabla_{x,\theta_j}^2 \mathcal{L}(F(x;\theta_j),y)||_2^2}{\lambda_{\min}}(-\mathbb{E}_{P_{XY}^{(i)}}\left[\mathcal{L}(F(x;\theta_i),y)\right] \\
&+ \mathbb{E}_{P_{XY}^{(j)}}\left[\mathcal{L}(F(x;\theta_j),y)\right] + K \cdot W(P_{XY}^{(i)}, P_{XY}^{(j)})) \\
&\leq 2\nabla_x \mathcal{L}(F(x;\theta_i),y)^\top \nabla_x \mathcal{L}(F(x;\theta_j),y).
\end{aligned} \tag{5}
$$

This completes the proof of the gradient alignment bound.

□

B Supplementary Experiment for Theorem 1

B.1 Empirical Validation

We use ResNet-10 and CIFAR-10. Unless otherwise specified, the results presented are the average values across models trained using all PFL methods.

Model sensitivity. Our theory posits that models with higher input gradient norms are more sensitive and thus more vulnerable. To test this, we measure the L_2 norm of the input gradients for original, unperturbed samples on both the proxy and target models. Notice that, we use a centralized training

Table 3: Comparison of centralized training and PFL.

Training Paradigm	Model Sensitivity	Average Eigenvalue	Parameter Sensitivity
Centralized Training	0.0078	2.984×10^{-6}	0.426
PFL methods	0.0142	9.712×10^{-6}	0.545

setting as we find the correlation coefficient to be similar across both centralized and PFL models for this specific correlation analysis (Table 3 reports The specific model sensitivity of these two settings.). We then evaluate the effectiveness of adversarial examples generated from these samples by measuring the magnitude of the resulting loss on the target model. Figure 4 demonstrates a strong positive correlation between gradient norms and attack effectiveness across diverse PFL client models.

Loss landscape geometry. Next, we investigate the geometry of the loss landscape by evaluating the average eigenvalue of the Hessian matrix associated with the classification head. A steeper landscape (i.e., a larger average eigenvalue) means that even small adversarial perturbations can cause a significant change in the model's output, thereby increasing vulnerability. The correlation coefficient between the average eigenvalue and AD is -0.176. This inverse relationship is consistent with our theory: a sharper loss landscape leads to greater vulnerability, which is reflected in a higher AD.

Distribution alignment and model fitness difference. Our theory suggests that higher transferability occurs when the data distributions of the proxy and target clients are more aligned and when their models have similar performance. For distribution alignment, we vary β from 0.1 to 0.9, where a larger β corresponds to reduced data heterogeneity (a smaller Wasserstein distance). As shown in Figure 4, increasing β leads to a more pronounced degradation in the target model's accuracy, aligning with our theoretical analysis. Furthermore, we assess the impact of model fitness difference by calculating the performance gap between the proxy and target models and correlating it with AD. The resulting correlation coefficient is -0.218. This number indicates that smaller fitness gaps between proxy and target models are associated with heightened vulnerability.

Parameter sensitivity. Finally, we explore the relationship between parameter sensitivity and attack transferability. Our theory suggests that if the input gradient is less sensitive to parameter changes, the adversarial vulnerabilities will be consistent across different personalized models, leading to higher transferability. Our measurements reveal a moderate correlation coefficient of 0.109, corroborating our theoretical analysis that greater parameter sensitivity makes higher AD.

B.2 PFL versus Centralized Training

Table 3 provides a comparison of the key metrics identified in our theoretical analysis for models trained under both centralized and PFL paradigms. Overall, the PFL paradigm exhibits higher model sensitivity, larger average Hessian eigenvalues, and increased parameter sensitivity compared to centralized training. These quantitative differences shed light on the empirically observed vulnerability gap between the two paradigms. These motivate us to propose a robustness-enhanced PFL framework aimed at mitigating these issues. Notice that we do not consider distribution alignment and fitness difference here, as these factors are inherently fixed and cannot be altered within the given scenario. Instead, our efforts concentrate on addressing the modifiable aspects of model behavior to improve robustness effectively.

C Supplementary Experiment for Adversarial Attacks in PFL

Client-initiated attacks versus server-initiated attacks Table 4 details the attack effectiveness of server-initiated attacks in three datasets. These attacks achieve negligible AD against specific partial-sharing methods, namely FedRep, FedBABU, and GPFL. Because clients using these methods do not upload their personalized classification heads to the server, the global model lacks the necessary parameters to craft effective adversarial examples. For other methods, the performance gap between client- and server-initiated attacks is marginal.

Table 4: The effectiveness (average AD) of server-initiated attacks in three datasets.

Dataset	FedProx	SCAFFOLD	FedBN	FedRep	FedBABU	GPFL	FedAS	FedCAC
CIFAR-10	63.44	54.76	67.71	3.90	4.51	3.84	50.49	38.48
CI-FAR100	33.54	33.33	30.09	3.22	1.87	3.98	30.40	25.79
GTSRB	81.87	78.01	83.74	4.81	2.40	2.34	63.75	74.89

Training dynamics and adversarial robustness Figures 5∼7 analyze how adversarial robustness evolves across training rounds in PFL under various attack configurations, including different attack methods, perturbation budgets, and attack iterations. As a benchmark, we compare these against centralized training scenarios (Figure 8), where two independent ResNet-10 models are trained to assess cross-model transferability. There are two observations.

- **Monotonic increase in vulnerability.** Accuracy drop grows with training rounds, perturbation budgets, and attack iterations. This trend holds consistently for both PFL and centralized training.
- **Diminished returns from transfer-based attack methods against PFL methods.** While advanced transfer-based attacks, e.g., MaskBlock, improve cross-model transferability in centralized training by about 8%, their benefits are less pronounced in PFL ($\leq 2\%$).

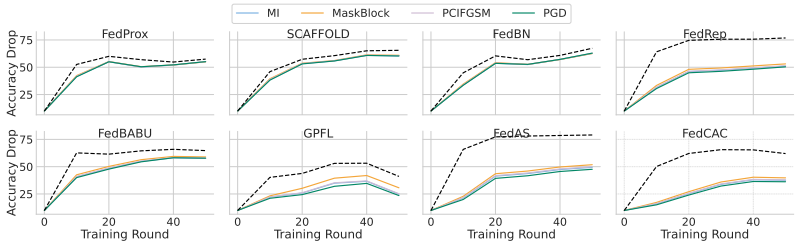


Figure 5: The average AD with different transfer-based attacks across varying training rounds. The black dotted lines indicate the model's average accuracy (shared on the y-axis).

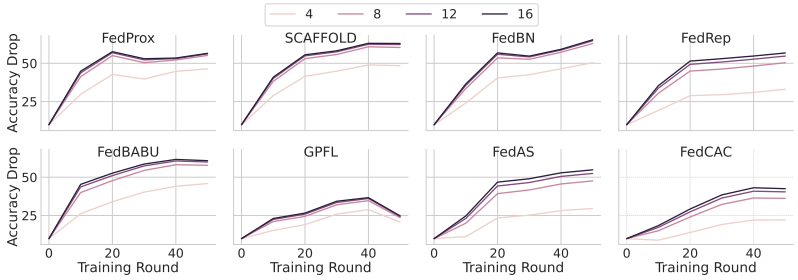


Figure 6: Impact of perturbation budget ϵ on average AD across training rounds using PGD.

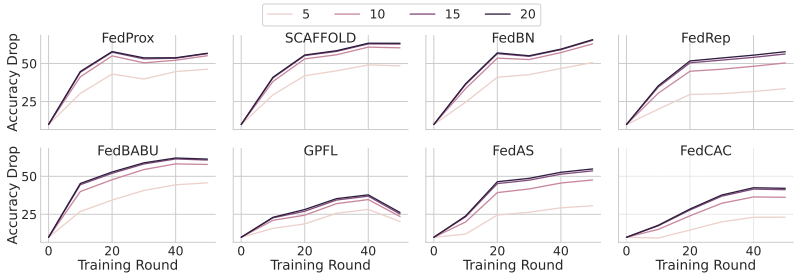
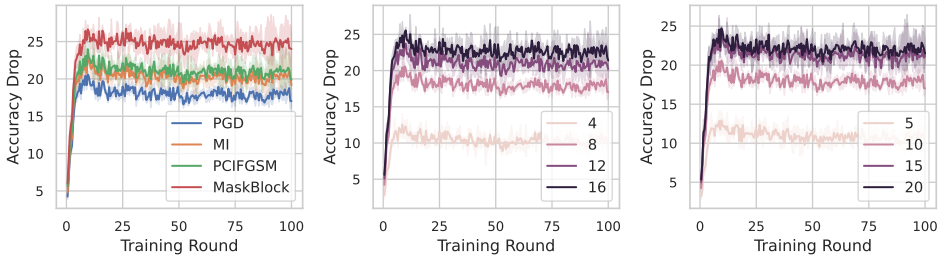


Figure 7: Impact of attack iterations on average AD across training rounds using PGD.



(a) Attack Method (b) Perturbation Budget (c) Attack iteration

Figure 8: Centralized training attack dynamics over different attack methods, perturbation budgets, and attack iterations. X-axis converted via iteration-to-round mapping (iterations / local iterations per round).

Table 5: Impact of model architecture on robustness across various PFL methods.

Model	FedAS	FedBABU	FedBN	FedCAC	FedProx	FedRep	GPFL	SCAFFOLD
ResNet10	47.61	57.79	62.94	36.19	55.14	50.38	23.52	60.34
ResNet18	55.22	66.57	70.55	42.42	62.33	61.69	34.60	73.90
ResNet34	58.85	70.59	72.41	45.80	66.06	65.22	38.19	77.29

D Supplementary Experiment for Our Robustness-enhanced Framework

The robustness of different models Table 5 evaluates the influence of model architecture on the robustness across various PFL methods. Notably, deeper architectures (e.g., ResNet34) generally exhibit higher AD compared to shallower ones (e.g., ResNet10), suggesting that increased model capacity may lead to greater vulnerability to adversarial attacks.

Table 6: The performance of the proposed robustness-enhanced framework in CIFAR-100.

PFL Method	FedAS		FedBABU		FedBN		FedCAC	
Metric	Acc	AD	Acc	AD	Acc	AD	Acc	AD
Vanilla	48.43	26.98	31.84	30.88	34.43	32.86	36.66	22.83
Ours	49.02	11.55	44.13	12.12	41.45	39.10	36.63	18.29
Ours+AT	37.73	6.87	38.04	6.57	33.78	15.38	31.30	8.94
Ours+AT+RS	16.90	1.79	17.37	1.69	13.05	3.22	20.52	4.35
PFL Method	FedProx		FedRep		GPFL		SCAFFOLD	
Metric	Acc	AD	Acc	AD	Acc	AD	Acc	AD
Vanilla	31.15	29.90	45.51	28.72	30.09	23.89	31.73	30.62
Ours	35.96	26.38	45.64	21.25	29.98	20.57	36.29	25.32
Ours+AT	28.64	14.36	37.95	9.90	21.51	8.94	29.64	13.09
Ours+AT+RS	10.02	2.58	14.87	1.74	7.24	1.04	9.15	1.71

Ablation study on key components To understand the individual contributions of our framework's components, we conduct an ablation study. We remove each of the three proposed regularization terms and measure the resulting impact on adversarial robustness, as quantified by AD. The results, averaged over all eight PFL methods, are summarized in Table 9. The results indicate that all components contribute meaningfully to improving robustness. When all components are included ("All"), our framework achieves the lowest AD of 38.73%, demonstrating their synergistic effect. Looking at the individual contributions, the results reveal that stochastic input noise augmentation has the largest impact on robustness. When this component is removed, the AD increases to 44.04%, indicating that regularizing the input gradients is a critical factor in mitigating adversarial attacks.

Table 7: The performance of the proposed robustness-enhanced framework in GTSRB.

PFL Method	FedAS		FedBABU		FedBN		FedCAC	
Metric	Acc	AD	Acc	AD	Acc	AD	Acc	AD
Vanilla	99.53	61.80	98.32	79.00	98.85	84.75	97.81	72.99
Ours	99.36	22.54	98.42	21.74	99.24	67.69	97.07	55.13
Ours+AT	97.06	13.55	97.47	13.86	96.83	22.34	91.25	23.29
Ours+AT+RS	53.85	4.82	54.68	5.64	44.61	7.59	54.30	8.82
PFL Method	FedProx		FedRep		GPFL		SCAFFOLD	
Metric	Acc	AD	Acc	AD	Acc	AD	Acc	AD
Vanilla	98.17	85.59	99.46	68.29	98.23	51.42	97.87	79.28
Ours	98.64	69.54	99.34	62.24	97.54	42.94	98.67	62.36
Ours+AT	95.77	24.73	96.79	16.15	90.31	24.16	97.14	22.01
Ours+AT+RS	39.35	6.22	47.41	4.57	48.12	5.87	40.33	6.82

Table 8: The time (s) required to execute a training round on the CIFAR-10.

Method	FedAS	FedBABU	FedBN	FedCAC	FedProx	FedRep	GPFL	SCAFFOLD
Vanilla	253.27	13.14	13.68	14.06	15.43	14.40	23.87	15.74
Ours	255.99	18.69	16.85	19.52	17.50	17.72	27.98	17.31
Ours+AT	514.16	244.476	257.478	253.36	235.656	157.696	273.914	262.042

Table 9: Ablation study on the impact of key components on adversarial robustness, measured by AD. Lower AD indicates better robustness. AD is averaged over eight PFL methods.

Component	AD
w.o. Stochastic input noise augmentation	44.04
w.o. Input-scaled trace regularization	41.65
w.o. Parameter sensitivity maximization	39.92
All	38.73

Moreover, parameter sensitivity maximization term has the smallest impact, with its exclusion resulting in an AD of 39.92%. In conclusion, each technique addresses a distinct aspect of the model's vulnerability, and together they provide a more comprehensive defense.

Integration details of common defenses AT is implemented by replacing x in Equation 2 with $x + \delta$, where δ is generated via 20-step PGD with $\epsilon = 8/255$. RS injects Gaussian noise into the input and applies majority voting across 50 inferences.

Overhead analysis We use a single NVIDIA GTX 4090 GPU. Table 8 summarizes the time required to complete a training round in CIFAR-10. Our framework introduces a small overhead (approximately $3\tilde{5}$ seconds) compared to the vanilla PFL implementations. This is because our regularization terms can be computed efficiently without requiring additional forward and backward passes, which are the dominant computational costs in both training and inference.

This stands in sharp contrast to AT which incurs substantial training overhead by up to $(K + 1)\times$ compared to standard training. In detail, PGD involves K forward and backward passes to generate adversarial examples, in addition to a forward and backward pass to compute the model parameter gradients.

When it comes to inference, the overhead differences are also important to consider. The models for Vanilla, Ours, and AT are identical, so their inference costs are almost the same. However, RS requires processing multiple noise-perturbed samples to conduct majority voting, potentially prolonging inference overhead by the noise sampling factor. Our empirical measurements show that while the inference time for a batch size of 128 is approximately 0.001 seconds, RS with 50 samples

extends this to around 0.05 seconds. For RS, the inference time is around 0.05 seconds. Practitioners can choose the solution that best fits their needs based on robustness, performance, and time costs.