Stealthy Backdoor Attack in Federated Learning via Adaptive Layer-wise Gradient Alignment

Qingqian Yang[1]* Peishen Yan[2] Xiaoyu Wu[2] Jiaru Zhang[2] Tao Song[2]† Yang Hua[3]
Hao Wang[4] Liangliang Wang[1]† Haibing Guan[2]
[1]Shanghai University of Electric Power [2]Shanghai Jiao Tong University
[3]Queen's University Belfast [4]Stevens Institute of Technology
y23108013@mail.shiep.edu.cn
{peishenyan, wuxiaoyu2000, jiaruzhang, songt333, hbguan}@sjtu.edu.cn
Y.Hua@qub.ac.uk hwang9@stevens.edu llwang@shiep.edu.cn

Abstract

The distributed nature of federated learning exposes it to significant security threats, among which backdoor attacks are one of the most prevalent. However, existing backdoor attacks face a trade-off between attack strength and stealthiness: attacks maximizing the attack strength are often detectable, while stealthier approaches significantly reduce the effectiveness of the attack itself. Both of them result in ineffective backdoor injection. In this paper, we propose an adaptive layer-wise gradient alignment strategy to effectively evade various robust defense mechanisms while preserving attack strength. Without requiring additional knowledge, we leverage the previous global update as a reference for alignment to ensure stealthiness during dynamic FL training. This fine-grained alignment strategy applies appropriate constraints to each layer, which helps significantly maintain attack strength. To demonstrate the effectiveness of our method, we conduct extensive evaluations across a wide range of datasets and networks. Our experimental results show that the proposed attack effectively bypasses eight state-of-the-art defenses and achieves high backdoor accuracy, outperforming existing attacks by up to 54.76%. *Additionally, it significantly preserves attack strength and maintains robust performance across diverse scenarios, highlighting its adaptability and generalizability. Code implementation is available at* https://github.com/yqqhyqq/LGA.

1. Introduction

Federated Learning (FL) has emerged as a promising paradigm for privacy-preserving distributed machine learning, allowing multiple decentralized clients to collaboratively train a global model without directly sharing their raw data [14, 21]. FL leverages local computation on edge devices (*e.g.*, smartphones and IoT sensors) and aggregates model updates via a central server, making it widely adopted in various real-world applications [27, 39]. However, the distributed architecture of FL systems significantly increases the potential attack surface, introducing a wide range of security threats, among which backdoor attacks are a commonly studied threat in recent research. In a backdoor attack [2, 32, 33, 35, 41, 42], adversarial clients intentionally inject a backdoor into the global model by uploading a poisoning local update. As a result, the global model behaves normally on benign inputs but misclassifies triggered inputs into predefined target labels.

Existing backdoor defenses primarily focus on detecting and filtering malicious updates by leveraging information from multiple dimensions. Specifically, some methods rely on distance-based measures between benign and malicious updates [4, 5, 12, 23], while others detect inconsistencies in parameter signs [25] or exploit the spectral separability [29]. These defenses significantly restrict the feasibility of backdoor attacks.

To counter these defenses, an effective backdoor attack depends on two key factors: attack strength and attack stealthiness. Some attacks [2] achieve high attack strength by scaling up the update, which increases their detectability and makes them easily filtered by defense mechanisms. Others [35, 42] prioritize stealthiness to evade detection but show significantly lower performance when no defense is present, highlighting their sacrifice of attack strength. These attacks essentially constrain the malicious update by discarding certain backdoor-related information [42] or using coarse-grained scaling [33]. Although constraining malicious updates helps maintain stealthiness, this inappropriate

*The work was done during her internship at Shanghai Jiao Tong University, Shanghai Key Laboratory of Scalable Computing and Systems.

†Corresponding authors.

constraint leads to attack strength degradation. Therefore, balancing the attack stealthiness and attack strength with limited knowledge of the system's defenses is challenging.

To address the challenge, we propose a layer-wise adaptive gradient alignment strategy that dynamically adjusts the norm bound for each layer of malicious updates. This adjustment enhances their similarity to the benign update distribution in a fine-grained manner, thereby achieving both attack stealthiness and attack effectiveness. Without requiring any additional information, our approach uses the previous global update as a reference point for adaptive alignment. By doing so, the proposed strategy not only constrains the malicious update within a small norm ball, but also enables the attack to adapt to the evolving global model and training dynamics, thereby significantly improving its stealthiness. Inspired by previous observations [15] that parameters within the same layer tend to share similar gradient magnitudes, we further conduct empirical analysis revealing that the magnitudes of both malicious and benign updates vary across layers, and there is no consistent proportional relationship between them. This discrepancy highlights the need for layer-specific norm constraints. Such constraints are essential to prevent over-constraining layers with inherently smaller magnitudes, thereby preserving attack strength.

Extensive experiments validate the effectiveness of our attack, demonstrating its ability to evade eight state-of-the-art defenses while maintaining a high attack success rate. Additionally, we investigate its performance across various attack scenarios, demonstrating its generalizability. We summarize our main contributions as follows:

- We propose an adaptive layer-wise gradient alignment strategy to effectively evade various robust defense mechanisms while preserving attack strength. Without requiring any additional knowledge, we leverage the previous global update as a reference for gradient alignment, ensuring that the attack maintains the stealthiness while adapting to the training dynamics. This fine-grained alignment strategy applies layer-specific constraints, allowing the attack to retain its strength.
- We conduct extensive evaluations across diverse models and datasets, demonstrating that the proposed attack successfully bypasses eight SOTA defenses and maintains the attack impact. Our backdoor attack outperforms SOTA attacks, with an improvement in backdoor accuracy by up to 54.76%. Furthermore, our attack exhibits excellent performance across various scenarios, such as different attack patterns, highlighting its generalizability.

2. Related Work

Backdoor attacks in FL. Backdoor attacks present a serious security threat in FL [16]. Typically, attackers compromise a small subset of client devices to upload poisoned updates. These updates are crafted to inject the backdoor into the global model, which then enables the model to obtain predetermined predictions for samples with specific trigger features. The backdoor attack process generally consists of two stages: trigger embedding and model manipulation. To successfully inject a backdoor, the attacker can carefully design either the trigger or the model update to bypass detection. Accordingly, existing methods can be divided into two categories: model-optimization attacks [2, 18, 41, 42] and trigger-optimization attacks [8, 20, 24, 33, 35, 40].

Model-optimization attacks [2, 3, 18, 33, 41, 42] focus on fine-tuning the model to achieve the desired attack. Scaling attack [2] amplifies malicious updates to achieve a one-shot attack. However, the abnormally large gradient magnitudes make it exposed to detection. Neurotoxin [41] projects malicious gradients onto infrequently updated neurons, improving the persistence of the attack and preventing the backdoor from being overwritten by benign updates. However, as an unconstrained backdoor attack, it's vulnerable to robust detection. The LP attack [42] adaptively injects the backdoor into backdoor-critical (BC) layers, achieving high stealthiness. However, by discarding backdoor-related information in non-BC layers, the attack suffers from reduced strength. PGD attack [33] exploits knowledge of server-side defenses to craft malicious updates that bypass detection. However, this strategy is only effective when the attacker has access to detailed information about the system. Additionally, 3DFed [18] introduces three distinct attack strategies targeting different types of defenses and utilizes an indicator for adaptive strategy adjustments. However, it relies on client collaboration.

Trigger-optimization backdoor attacks aim to design triggers that are both subtle and adaptable, ensuring their stealthiness and effectiveness. Edge-case attack [33] selects marginal data as poison samples to circumvent detection mechanisms. DBA [35] firstly introduces distributed backdoor attacks, decomposing a global trigger into smaller, less detectable components. These approaches craft triggers at the pre-training stage and then fix them during training. Recently, a range of studies [8, 19, 20, 24, 40] dynamically optimize the trigger throughout the training process and obtain better attack results than the former ones.

Defense strategies against backdoor attacks in FL. In FL [16], the server only receives uploaded models without access to any information about client data. As a result, the majority of defense strategies are similarity-based, aiming to detect and mitigate backdoor attacks by identifying differences between malicious and benign models.

Among them, distance-based defenses [1, 4, 5, 9, 12, 17, 22, 23, 28, 29, 31, 34] exploit the observation that malicious models often deviate from benign ones in their distribution within high-dimensional space, making distance-based metrics an effective tool for detection. MultiKrum [4] only aggregates updates that are selected with the smallest pairwise Euclidean distances. FLTrust [5] weights upload gradients based on their cosine similarity to a trusted gradient to mitigate the influence of malicious models. The trusted gradient is obtained from a root dataset collected by the server. FLAME [23] and DeepSight [28] use pairwise cosine distance between client gradients for clustering, identifying malicious gradients as outliers and further refining the defense through additional modules. Multi-Metrics [12] dynamically adjusts the weights of three distance metrics to detect backdoor models. MESAS [17] leverages six distance metrics for detection.

Other similarity-based defenses [13, 26, 29, 36, 38] identify malicious updates from more diverse perspectives. DnC [29] leverages singular value decomposition (SVD)-based spectral methods to remove outliers. RLR [25] detects inconsistencies in parameter signs across training rounds and adjusts learning rates accordingly to suppress backdoors. MASA [37] leverages the observation that a malicious model unlearns the main task more quickly than a benign model for detection. FedREDefense [36] relies on the finding that a malicious model is more resistant to distillation. In addition, some methods recently focus on the difference in update behavior between benign models and malicious models at the parameter level. For instance, SparseFed [26] only aggregates Top-k highest magnitude parameters because the backdoor updates are usually injected into coordinates that are unimportant for benign updates. Meanwhile, Lockdown [13] restrains local training in a subspace to further reduce the poison-coupling effect.

3. Methodology

Background. Federated Learning (FL) distributes the training task across multiple clients to preserve data privacy. At the beginning, a global model θ_g^0 is initialized on the server and distributed to a set of N clients. In each round t, a standard FL framework involves the following steps:

1. Local Training: Each client i receives the global model θ_g^{t-1} and performs E local epochs of stochastic gradient descent (SGD) on its private dataset $\hat{\mathcal{D}}_i$ with batch size B, obtaining an updated model θ_i^t. The client computes the model update as:

$$\Delta\theta_i^t = \theta_i^t - \theta_g^{t-1}, \tag{1}$$

and uploads it to the server.

2. Global aggregation: The server randomly selects a subset of clients and aggregates their model updates $\{\Delta\theta^t\}$ to compute the global update:

$$\Delta\theta_g^t = \textstyle\sum_{i\in[n]} \Delta\theta_i^t, \tag{2}$$

and updates the global model as $\theta_g^t = \theta_g^{t-1} + \Delta\theta_g^t$. The new global model is then distributed to clients for the next round. Before updating the global model, the server typically performs backdoor detection during aggregation to mitigate the impact of malicious updates.

In FL, backdoor attacks exploit the system's decentralized nature. Specifically, an attacker, denoted as A, compromises a subset of clients and injects malicious updates to implant a backdoor into the global model. We consider a continuous attack scenario, following prior studies [20, 35, 42].

Adversarial Objectives: The attacker A aims to inject a backdoor into the global model while minimizing the impact on the model's performance on non-poisoned data. The goal of the backdoor injection is to manipulate the model so that it misclassifies inputs with a specific trigger pattern as an incorrect target label. We denote the malicious update as $\Delta\theta_m$, which is trained on the poisoned dataset. By carefully crafting $\Delta\theta_m$, the attacker can introduce the backdoor without significantly degrading the model's overall accuracy. The server's aggregation mechanism, which aggregates updates from multiple clients, may unintentionally include these poisoned updates, thereby corrupting the global model.

Adversarial Capability: The attacker A has the ability to control a small subset of clients and gain access to their datasets. Additionally, the attacker can manipulate the models of these compromised clients [2, 35, 41]. The attacker does not have access to the data or models of benign clients, nor to the specific defense methods implemented by the defender.

Overview. We introduce an adaptive layer-wise gradient alignment to enhance stealthiness while preserving attack strength. In each round, we train the malicious update on a poisoned local dataset and iteratively align gradients to resemble benign updates before uploading them to the server. This increases the likelihood of the server aggregating the malicious update, effectively injecting the backdoor into the global model.

Adaptive reference selection. Traditionally, the Unconstrained malicious update $\Delta\theta_m$, which is trained on the poisoned dataset including out-of-distribution backdoor samples, tends to exhibit a larger magnitude compared to benign updates [32]. To ensure that the malicious update remains similar to the distribution of benign updates, it should be carefully constrained within an appropriately small norm ball. Notably, the selection of the norm bound significantly impacts the effectiveness of the attack's concealment. Therefore, it is crucial to rigorously determine an appropriate reference point for alignment. Some defense methods, such as MultiKrum [4], select a fixed number of client updates that appear most benign in each round. However, due to data heterogeneity, such methods may inadvertently exclude genuinely benign clients whose updates deviate from the majority. To mitigate this issue, we adopt the previous global

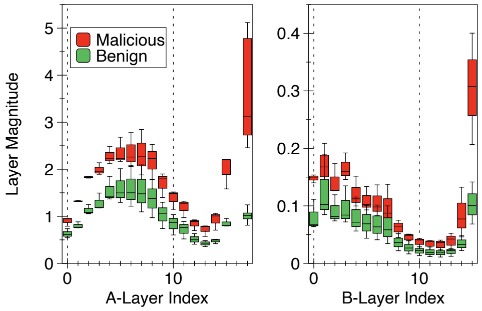


Figure 1. Layer-wise Magnitude Analysis of Benign vs. Malicious Updates in VGG19. To better visualize, we focus on weight layers and divide them into A-layers and B-layers based on $\|\Delta\theta^l\|$: those with norms above 0.5 are A-layers; others are B-layers. The experiment uses the default setting and records $\|\Delta\theta^l\|$ every 50 rounds.

update $\Delta\theta_g^{t-1}$ as a reference point. Since it is computed by aggregating multiple client updates, most of which are expected to be benign, it serves as a reasonable approximation of benign behavior.

By aligning the malicious update with the global update, the attack remains within the expected range of benign variation, thereby enhancing its stealthiness and increasing the likelihood of successfully influencing the global model.

Layer-wise gradient alignment. Intuitively, applying a norm constraint weakens the impact of the malicious update, which potentially leads to ineffective backdoor injection. We introduce a layer-wise scaling strategy to refine the malicious update at a more granular level to not only ensure the attack remains stealthy but also preserve its strength. A key observation shown in Figure 1 is that the layer-wise magnitudes of the malicious and benign updates do not maintain a fixed proportional relationship; instead, their patterns differ. Therefore, we align each layer individually with $\Delta\theta_g^{t-1}$. Specifically, we define the scaling factor $S^{t,l}$ for each layer l as:

$$S^{t,l} = min(1, \frac{\|\Delta\theta_g^{t-1,l}\|}{\|\Delta\theta_m^{t,l}\|}). \tag{3}$$

The scaled malicious update after each local epoch is given by:

$$\Delta\theta_m^{t,l} = S^{t,l} \cdot \Delta\theta_m^{t,l}. \tag{4}$$

The adaptive layer-wise gradient alignment, implemented in just four lines (Lines 10–13 in Algorithm 1), extends the standard FL backdoor pipeline. This approach prevents excessive constraints on layers with inherently smaller magnitudes and constrains the update for each layer within an $\ell 2$ norm ball, thereby preserving attack strength while enhancing stealthiness.

Algorithm 1 Layer-wise Gradient Alignment for Attackers

Require: Batch size ℓ, local epochs E, malicious model parameters θ_i^t, global parameters θ_g^{t-1}, poisoned dataset $\hat{\mathcal{D}}$, and previous global parameter θ_g^{t-2}

Procedure:

```
1:  // Attacker's local training:
2:  Calculate the previous global update:
3:  $\Delta\theta_g^{t-1} \leftarrow \theta_g^{t-1} - \theta_g^{t-2}$
4:  Initialize local model $\theta_{i,0}^t \leftarrow \theta_g^{t-1}$
5:  for $e = 1$ to $E$ do
6:      for each batch $b$ in $\hat{\mathcal{D}}$ do
7:          Compute gradient for batch $b$:
8:          $\Delta\theta_{i,e}^t \leftarrow \Delta\theta_{i,e}^t + \nabla_\theta \mathcal{L}(\theta_{i,e-1}^t, \hat{\mathcal{D}}_b)$
9:      end for
10:     // Layer-wise gradient alignment
11:     for each layer $l$ in $\Delta\theta_{i,e}^t$ do
12:         $S_{i,e}^{t,l} \leftarrow min(1, \frac{\|\Delta\theta_g^{t-1,l}\|}{\|\Delta\theta_{i,e}^{t,l}\|})$
13:         $\theta_{i,e}^{t,l} \leftarrow \Delta\theta_{i,e}^{t,l} \cdot S_{i,e}^{t,l} + \theta_g^{t-1,l}$
14:     end for
15: end for
16: Upload update $\Delta\theta_i^t$ to server
```

4. Experiments

Our empirical study evaluates the effectiveness of the proposed backdoor attack through experiments conducted on real-world datasets, simulating an FL environment using the NVIDIA RTX 4090 GPU.

4.1. Experimental setup

Datasets and Models. We evaluate the attack using two widely used benchmark datasets: CIFAR10 (50,000 training and 10,000 test samples across 10 classes) and CIFAR100 (50,000 training and 10,000 test samples across 100 classes). Following previous studies [41, 42], we use the VGG19 [30] network and ResNet18 network for CIFAR10, while we use the ResNet18 [11] network for CIFAR100. We simulate a non-IID data distribution following prior methods [5, 7, 42], with $q = 0.5/0.2$ for CIFAR10/100, respectively.

Attack setup. In each round, we select $n = 10$ clients among $N = 100$ clients to participate in the aggregation, with the proportion of compromised clients set to $C = 0.1$ by default. The FL training runs for 300 rounds to make it converge. We train for 6 local epochs (2 for the benign clients) with a learning rate 0.1 for both CIFAR10 and CIFAR100. Additionally, we assess the effectiveness of our attack in an IID setting, as defense strategies are more likely to identify malicious models as outliers in this case [42]. The trigger is a 5×5 pixel square located at the bottom-right corner of each image by default. We further discuss the impact of various shapes and sizes of the trigger. We evaluate our attack in the fixed-frequency attack scenario. The attacker controls a fixed number of compromised clients, which participate in FL training at a fixed frequency. Following prior

Table 1. The effectiveness of our approach against the baseline attack across various state-of-the-art (SOTA) defenses under a fixed-frequency attack setting on non-IID datasets. BA values below 10% in CIFAR10 (10 classes) and below 1% in CIFAR100 (100 classes) are highlighted in red, indicating a failed attack. The highest BA achieved in each setting is presented in **bold**. Our attack results are reported as $a \pm b$, where a denotes the mean and b represents the standard deviation. MA and BA unit: %. Avg indicates the average. Each result is obtained as the average over five independent runs.

Defense		VGG19 (CIFAR10)				ResNet18 (CIFAR10)				ResNet18 (CIFAR100)			
		BadNets	**DBA**	**LP**	**Ours**	**BadNets**	**DBA**	**LP**	**Ours**	**BadNets**	**DBA**	**LP**	**Ours**
No Defense	MA	80.97	81.26	**82.38**	81.81±0.34	77.66	78.26	78.43	**78.70**±0.82	63.05	62.4	**65.17**	64.31±0.72
	Best BA	**98.33**	46.17	95.38	97.05±0.19	**98.59**	20.96	96.69	97.60±0.21	**100.0**	1.16	72.24	98.51±1.06
	Avg BA	**97.70**	33.96	90.56	96.16±0.09	**97.91**	14.93	95.08	97.22±0.21	**99.73**	0.80	61.87	96.70±2.46
MK [4]	MA	74.17	75.46	74.30	**78.03**±0.33	75.24	76.21	76.07	**76.49**±0.86	57.84	58.64	58.65	**58.73**±0.55
	Best BA	24.06	3.16	93.54	**98.01**±0.80	6.04	7.52	95.89	**98.61**±0.14	0.88	0.99	97.52	**99.94**±0.01
	Avg BA	5.99	1.33	76.41	**95.98**±0.36	3.64	3.85	92.40	**97.03**±1.57	0.62	0.54	89.22	**94.42**±2.87
FLTrust [5]	MA	83.14	83.25	82.98	**83.84**±0.20	**80.75**	80.22	80.22	80.33±0.29	61.58	61.80	**61.89**	61.14±1.81
	Best BA	56.23	6.53	96.96	**97.95**±0.34	91.81	4.18	97.19	**97.78**±0.27	30.12	0.69	95.42	**99.98**±0.05
	Avg BA	42.45	4.91	96.60	**96.74**±0.47	89.12	3.67	**95.48**	91.94±8.41	19.87	0.46	87.51	**98.52**±1.59
FLAME [23]	MA	58.07	57.52	58.63	**63.46**±0.91	74.21	**73.52**	71.90	73.06±0.76	**55.93**	54.28	55.62	55.87±0.45
	Best BA	21.20	37.84	92.43	**98.38**±0.63	12.70	2.97	94.46	**98.69**±0.17	0.83	0.53	77.67	**99.99**±0.00
	Avg BA	7.75	12.76	78.89	**97.46**±0.44	7.01	2.27	89.72	**97.54**±0.58	0.62	0.44	72.89	**99.88**±0.08
MM [12]	MA	69.31	66.43	70.72	**75.42**±0.97	71.43	72.61	72.80	**73.16**±1.07	53.03	53.82	53.40	**53.55**±0.33
	Best BA	29.08	16.02	94.95	**98.31**±0.44	15.54	1.71	96.01	**98.47**±0.09	0.97	0.85	92.73	**99.99**±0.00
	Avg BA	7.04	5.42	84.61	**94.55**±4.20	3.91	1.37	93.12	**97.52**±0.67	0.37	0.66	83.46	**97.05**±1.58
DnC [29]	MA	81.09	**82.6**	82.32	81.30±1.45	77.98	**79.32**	78.63	78.83±0.30	63.01	61.43	**63.89**	63.43±0.73
	Best BA	5.94	2.83	95.04	**97.85**±0.27	6.58	11.20	96.25	**98.19**±0.14	0.43	4.88	80.93	**99.67**±0.29
	Avg BA	3.68	1.67	93.57	**96.94**±0.53	4.03	9.72	94.45	**97.29**±0.37	0.41	2.69	63.37	**98.07**±0.59
RLR [25]	MA	72.28	76.22	77.71	**77.90**±0.68	**75.72**	75.06	74.51	74.77±1.37	54.77	**56.78**	56.14	56.58±0.58
	Best BA	96.93	23.06	94.11	**97.78**±0.11	97.29	17.16	94.83	**98.39**±0.13	0.14	0.34	88.12	**99.98**±0.01
	Avg BA	95.65	15.13	91.61	**97.39**±0.29	97.02	10.85	89.41	**97.95**±0.17	0.05	0.09	83.81	**99.97**±0.02
FLARE [34]	MA	84.45	84.24	84.53	**84.56**±0.11	77.79	**78.21**	77.52	77.82±2.05	**60.44**	56.58	59.54	59.07±1.41
	Best BA	**98.21**	43.87	94.79	97.85±0.12	**98.18**	5.88	94.91	96.75±0.50	**99.71**	1.14	88.56	99.56±0.41
	Avg BA	**98.11**	36.32	94.10	97.28±0.30	**98.02**	5.40	93.55	96.30±0.82	**98.60**	0.63	80.73	98.17±2.42

approaches [2, 41, 42], we assume that the attacker controls exactly one client per round and sets the attack interval to $F = 1$. This setup allows us to analyze the attack in isolation, without client collusion. We also evaluate the attack's performance across different frequencies, comparing it to the baseline.

Baseline Defenses and Attacks. We assess our backdoor attack on eight state-of-the-art or representative defenses in FL: MultiKrum (MK) [4], FLTrust [5], FLAME [23], MultiMetrics (MM) [12], DnC [29], RLR [25], FLARE [34], and DeepSight [28], ensuring a comprehensive evaluation of the attack's effectiveness.

We compare our backdoor attack against three state-of-the-art or representative attacks in FL: BadNets [10], DBA [35], and LP attack [42].

Metrics. Following prior work on FL backdoor attacks [10, 35, 42], we evaluate the global model using two primary metrics: main task accuracy (MA) and backdoor accuracy (BA). Given the dynamic nature of the global model in FL, we report both the average and the best BA over the last 10 communication rounds. Furthermore, to evaluate the stealthiness of the attack, we adopt two metrics introduced in [42]: the malicious client acceptance rate (MAR) and the benign client acceptance rate (BAR). MAR measures the proportion of rounds in which malicious clients successfully bypass defense and are selected for aggregation, while BAR represents the average proportion of rounds in which benign clients are selected for aggregation.

4.2. The effectiveness of our attack

We first evaluate our attack in the fixed-frequency attack setting under different SOTA defenses. Table 1 presents the performance comparison between our attack and the baseline attacks. We also evaluate our attack in the IID setting in supplementary §B.1 and on a larger dataset Tiny-ImageNet in supplementary §B.11. The results show that our attack achieves the highest BA in most cases. These results demonstrate the broad effectiveness of our attack, effectively bypassing defenses while maintaining superior attack performance.

BadNets only works well under FLARE and consistently fails to bypass defenses such as MK, FLAME, MM, and

Table 2. MAR and BAR comparison under outlier-based defenses: MK, FLAME, MM, and DnC.

Dataset (Model)	**Attack**	**MK**		**FLAME**		**MM**		**DnC**	
		BAR	**MAR**	**BAR**	**MAR**	**BAR**	**MAR**	**BAR**	**MAR**
CIFAR10 (ResNets18)	BadNets	0.42	0.01	0.75	0.01	0.33	0.01	0.97	0.01
	DBA	0.41	0.01	0.71	0.01	0.34	0.01	0.95	0.01
	LP attack	0.36	**0.72**	0.65	**0.90**	0.13	**0.8**	0.39	0.82
	Ours	0.38	0.68	0.64	0.88	0.25	0.79	0.78	**0.99**
CIFAR10 (VGG19)	BadNets	0.41	0.01	0.72	0.01	0.31	0.01	0.98	0.01
	DBA	0.42	0.01	0.75	0.01	0.33	0.03	0.97	0.01
	LP attack	0.35	**0.63**	0.61	0.75	0.13	**0.8**	0.41	**0.99**
	Ours	0.38	0.56	0.6	**0.9**	0.26	0.60	0.79	**0.99**
CIFAR100 (ResNets18)	BadNets	0.43	0.01	0.68	0.01	0.33	0.01	0.99	0.01
	DBA	0.42	0.01	0.76	0.01	0.33	0.01	0.97	0.01
	LP attack	0.32	**0.91**	0.56	**0.99**	0.12	**0.93**	0.35	0.94
	Ours	0.34	0.83	0.58	0.97	0.23	0.92	0.72	**0.99**

DnC across all settings, and it also fails against RLR on CIFAR100 in both non-IID and IID settings. Similarly, DBA also fails under several defenses, including MK, FLTrust, MM, and DnC under both non-IID and IID settings. LP attack achieves comparable BA across most defenses, except when evaluated under the FLARE defense in the IID setting. Specifically, for CIFAR100, our attack achieves significantly higher BA than the LP attack, demonstrating superior adaptability to complex data distributions. We also evaluate our attack performance compared with the LP attack under DeepSight in supplementary $\S B.3$. Our attack achieves a higher BA than the LP attack across all scenarios and the LP attack fails on CIFAR100.

Our attack improves the attack stealthiness and maintains strength. Under the FedAvg setting, our attack achieves significantly higher BA compared to both DBA and LP attack, particularly on CIFAR100, approaching the performance of unconstrained attacks like BadNets. This demonstrates the strong strength of our attack.

Table 2 presents the comparison of MAR and BAR under outlier-based defenses such as MK, FLAME, MM, and DnC. A higher MAR indicates that malicious updates are more likely to bypass detection. We observe that BadNets and DBA fail to bypass defenses, as their MAR remains close to zero in all cases. This suggests that BadNets and DBA are consistently detected as outliers during training, rendering them ineffective against robust defenses. While BadNets demonstrates strong attack strength, its failure to bypass defenses undermines its overall effectiveness. In contrast, our attack achieves MAR consistently higher than BAR, meaning our malicious updates are more likely to be selected for aggregation, which reflects the stealthiness capability of our method.

When compared to the LP attack, our method maintains nearly the same or even higher MAR in defenses such as FLAME and DnC, further demonstrating its stealthiness. MK and MM impose stricter selection criteria by choosing only a small proportion of the most benign clients, making it more challenging for an attacker to be selected. LP attack achieves a higher MAR than our method under these defenses by restricting the backdoor to BC layers, which significantly enhances its stealthiness. However, our method consistently achieves higher BA than LP attack across all settings in FLAME, DnC, MK, and MM, demonstrating that while LP attack sacrifices attack strength for stealthiness, our approach achieves a better balance, ensuring both strong attack strength and high stealthiness.

More evaluation of attack stealthiness. We leverage Krum distance, cosine distance, and Manhattan distance to further illustrate the stealthiness of our attack. Specifically, we calculate the sum of squared Euclidean distances, cosine distances, and Manhattan distances between a model update and other local model updates, referring to them as Krum distance, Cosine distance, and Manhattan distance, respectively.

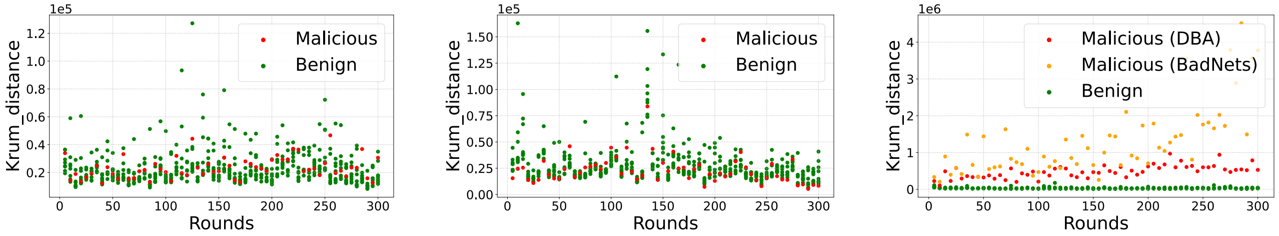
A larger Krum distance indicates that the update deviates significantly from benign updates, making it less likely to be accepted by the aggregation mechanism. Figure 2 shows the comparison of Krum distance with baseline attacks. The Krum distances of BadNets and DBA are significantly larger than those of benign updates, indicating that these attacks produce model updates that are easily distinguishable from benign ones. In contrast, our attack and the LP attack exhibit Krum distances close to those of benign updates, suggesting that they are more stealthy. Although DBA demonstrates slightly better stealthiness than BadNets, its Krum distance remains considerably larger than that of benign updates. This suggests that DBA is still insufficiently stealthy against robust defenses. In supplementary $\S B.4$, we further discuss the Cosine distance and Manhattan distance between the malicious update and the benign update. The result exhibits similar trends across all attacks.

Table 3. The BA of various backdoor attacks combined without/with ours.

Attack	IBA [24]	IBA+ours	Cerp [20]	Cerp+ours	A3FL [40]	A3FL+ours
FLAME	2.71	**98.04**↑	29.84	**99.64**↑	94.87	**99.87**↑
MM	31.63	**99.99**↑	24.72	**99.98**↑	86.29	**99.89**↑

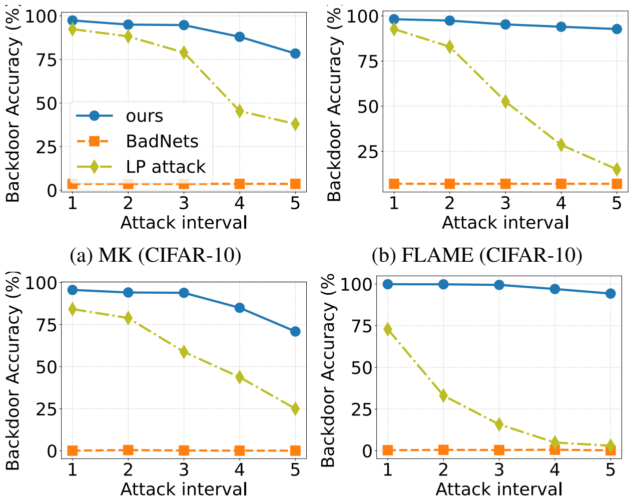
Our attack demonstrates strong generalizability. It can be easily integrated into existing FL backdoor attacks, particularly those based on trigger optimization. As shown in Table 3, incorporating our method further enhances attack effectiveness, underscoring its practical value.

We further evaluate our attack trained on other networks: ResNet50 [11], ResNet101 [11] and ViT [6] in supplementary $\S B.5$. Our attack achieves high BA on those networks, demonstrating its generalizability to more complex architectures.



(a) Ours (b) LP attack (c) BadNets and DBA

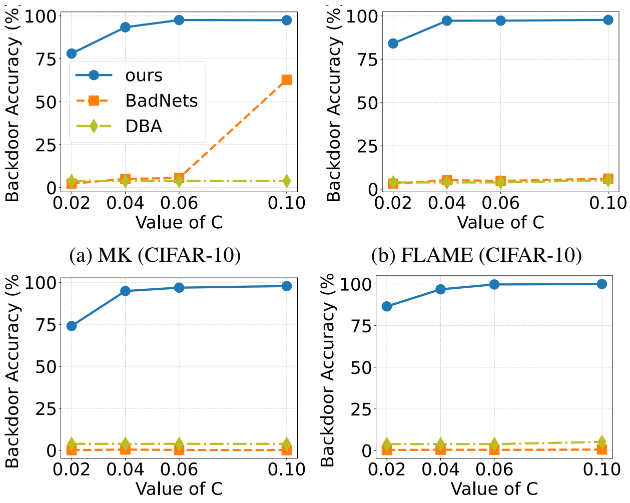
Figure 2. Krum distance of malicious and benign model updates. A larger distance indicates that the update deviates significantly from benign updates. The result shows the stealthiness of our attack.



(a) MK (CIFAR-10) (b) FLAME (CIFAR-10)

(c) MK (CIFAR-100) (d) FLAME (CIFAR-100)

Figure 3. Impacts of different attack intervals F on the attack performance. The F indicates that an attack is performed every F rounds.



(a) MK (CIFAR-10) (b) FLAME (CIFAR-10)

(c) MK (CIFAR-100) (d) FLAME (CIFAR-100)

Figure 4. Effectiveness of our attack under different proportions of compromised clients ($C = 0.02, 0.04, 0.06, 0.1$) in a fixed-pool attack setting.

In addition, we analyze the impact of varying degrees of data heterogeneity in supplementary §B.6. Results show that our attack remains effective under highly non-IID conditions and consistently outperforms the LP attack across all scenarios, further confirming its generalizability.

Our attack maintains the global model's performance. Supplementary $\S B.2$ shows the MA with the unpoisoned global model to assess whether our attack affects the utility of the global model. The results demonstrate that our attack does not lead to a significant decrease in MA compared to the unpoisoned global model.

4.3. Our attack under different scenarios

Impact of attack intervals. We evaluate the performance of our attack compared to baseline attacks under different attack intervals, as shown in Figure 3. The attack interval F ranges from 1 to 5, where F indicates that an attack is performed once every F rounds. This setting is designed to simulate scenarios where the frequency of malicious updates is limited, requiring more effective and persistent attack strategies.

The results show that as F increases, the performance of all attacks degrades as the attack frequency decreases. This trend is consistent across all evaluated methods, where a lower frequency results in reduced BA. A larger F imposes greater demands on the attack's strength, persistence, and stealthiness. The attack must be sufficiently powerful and stealthy to maintain its effectiveness with fewer opportunities to update the model. We observe that BadNets fails in all cases. Notably, our attack maintains a comparable BA even when $F = 5$. Whereas the LP attack loses nearly all effectiveness, particularly when trained on CIFAR100, demonstrating its vulnerability to defenses when attack opportunities are limited. We also show the results of BA under all defenses trained on CIFAR10 (ResNet18) in supplementary $\S B.8$. We observe that our attack achieves comparable BA across all defenses when $F = 5$.

Impact of different proportions of compromised clients. To evaluate our attack in a more practical scenario, we conduct experiments in a fixed-pool attack setting. To further

Table 4. Ablation study on alignment strategy.

Dataset (Model)	**Attack**	No Defense	FLAME		MM	
		BA	BA	MAR	BA	MAR
CIFAR-10 (VGG19)	no alignment	**97.70**	7.05	0.01	7.04	0.01
	model-wise	90.12	45.79	0.90	83.58	0.65
	layer-wise (ours)	96.16	**97.46**	0.90	**97.09**	0.67
CIFAR-10 (ResNet18)	no alignment	**97.91**	7.01	0.01	3.91	0.01
	model-wise	93.39	85.32	0.89	95.53	0.80
	layer-wise (ours)	97.21	**98.16**	0.88	**97.85**	0.79
CIFAR-100 (ResNet18)	no alignment	**99.73**	0.62	0.01	0.37	0.01
	model-wise	73.29	85.86	0.95	95.05	0.90
	layer-wise (ours)	96.70	**99.88**	0.97	**97.05**	0.92

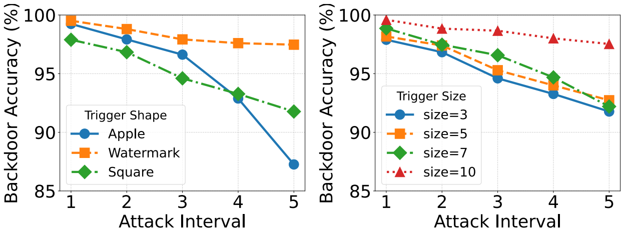
demonstrate the strength of our attack, we also analyze its performance under varying proportions of compromised clients C. In a fixed-pool attack setting, the attacker controls a pool of compromised clients, from which participants are randomly selected in each round. This setup allows multiple malicious clients to be chosen in a single round, while in some cases, several consecutive rounds may pass without selecting any malicious clients.

As shown in Figure 4, our attack achieves the highest BA compared to the baseline across all cases. Notably, even with a very low proportion of malicious clients ($C = 0.02$, meaning only 2% of clients are compromised), our attack successfully bypasses defenses and maintains a comparable BA. This demonstrates that our attack is both powerful and persistent, as it continues to perform well even when the pool of malicious clients is small.

Additionally, BadNets performs better when $C = 0.1$ due to the increased likelihood of malicious clients colluding in the same round. This collusion significantly enhances the attack's effectiveness, as multiple attackers working together are more likely to successfully inject a backdoor. This is particularly evident under MK, where BadNets achieves a higher BA with a higher C, compared to when it fails in the fixed-frequency setting. However, this improvement diminishes as the compromised client ratio decreases. At lower C, the attack becomes more reliant on its strength. In contrast, as shown in supplementary $\S B.9$, our attack achieves high BA across all defenses when $C > 0.04$, demonstrating its robustness even with fewer compromised clients.

4.4. Ablation study

Impact of different triggers. We further evaluate the impact of various trigger shapes. Supplementary $\S B.10$ presents three different trigger shapes (Apple, Watermark, and Square) applied to backdoor a ResNet18 model trained on the CIFAR10 dataset. Figure 5a shows the impact of various trigger shapes. Under the default setting ($F = 1$), all three trigger shapes achieve high BA. The results demonstrate that our attack remains effective across different trigger shapes, indicating its robustness and generalizability.



(a) various trigger shapes

(b) various trigger sizes

Figure 5. Impact of trigger shape (left) and trigger size (right) in Flame.

Figure 5b illustrates the performance of different trigger sizes (3, 5, 7, and 10). Regardless of trigger size, our attack consistently achieves high BA. Notably, when the trigger size is set to 10, the attack effectiveness is significantly enhanced. Larger triggers strengthen the attack, and their ability to bypass defenses is improved through layer-wise norm constraints, leading to superior attack performance.

Impact of alignment strategy. To demonstrate how dynamic layer-wise gradient alignment enhances the effectiveness of our attack, we compare it with BadNets (no alignment) and the model-wise gradient alignment attack, which adjusts the overall gradient magnitude to align with the previous global update. As shown in Table 4, our attack consistently outperforms both BadNets and the model-wise method across all settings.

First, both model-wise and layer-wise gradient alignment show higher MAR compared to BadNets, indicating that aligning the update with the previous global model significantly improves the stealthiness of the attack. Second, when compared to model-wise alignment, layer-wise alignment achieves better attack performance. Specifically, even in the absence of an attack, particularly on CIFAR100, the BA for model-wise alignment is much lower than that for layer-wise alignment.

5. Conclusion

This paper proposes a new stealthy backdoor attack in FL, which leverages adaptive layer-wise gradient alignment to enhance stealthiness while preserving attack strength. By dynamically aligning malicious gradients to historical global updates at the layer level, the proposed attack reduces statistical anomalies caused by malicious updates, making them harder to detect. Extensive experiments across diverse datasets, models, and defenses demonstrate that the proposed attack achieves superior backdoor success rates while evading SOTA defenses, highlighting its effectiveness and stealthiness in various FL scenarios.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NO. 62472284, No. 62302288), Shanghai Key Laboratory of Scalable Computing and Systems, SJTU Kunpeng & Ascend Center of Excellence.

References

[1] Sana Awan, Bo Luo, and Fengjun Li. CONTRA: Defending Against Poisoning Attacks in Federated Learning. In *ESORICS*, 2021. 2

[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How To Backdoor Federated Learning. In *AISTATS*, 2020. 1, 2, 3, 5

[3] Akshay Batheja and Pushpak Bhattacharyya. "A Little is Enough": Few-Shot Quality Estimation based Corpus Filtering improves Machine Translation. In *ACL (Findings)*, 2023. 2

[4] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *NeurIPS*, 2017. 1, 2, 3, 5

[5] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping. In *NDSS*, 2021. 1, 2, 3, 4, 5

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 6

[7] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In *USENIX Security*, 2020. 4

[8] Pei Fang and Jinghui Chen. On the Vulnerability of Backdoor Defenses for Federated Learning. In *AAAI*, 2023. 2

[9] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. The Limitations of Federated Learning in Sybil Settings. In *RAID*, 2020. 2

[10] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks. *IEEE Access*, 7:47230–47244, 2019. 5

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 4, 6

[12] Siquan Huang, Yijiang Li, Chong Chen, Leyu Shi, and Ying Gao. Multi-Metrics Adaptively Identifies Backdoors in Federated Learning. In *ICCV*, 2023. 1, 2, 3, 5

[13] Tiansheng Huang, Sihao Hu, Ka Ho Chow, Fatih Ilhan, Selim F. Tekin, and Ling Liu. Lockdown: Backdoor Defense for Federated Learning with Isolated Subspace Training. In *NeurIPS*, 2023. 3

[14] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and Open Problems in Federated Learning. *Foundations and trends® in machine learning*, 14 (1–2):1–210, 2021. 1

[15] Yun-Yong Ko, Dongwon Lee, and Sang-Wook Kim. Not All Layers Are Equal: A Layer-Wise Adaptive Approach Toward Large-Scale DNN Training. In *WWW*, 2022. 2

[16] Jakub Konečnỳ. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016. 2

[17] Torsten Krauß and Alexandra Dmitrienko. MESAS: Poisoning Defense for Federated Learning Resilient against Adaptive Attackers. In *CCS*, 2023. 2, 3

[18] Haoyang Li, Qingqing Ye, Haibo Hu, Jin Li, Leixia Wang, Chengfang Fang, and Jie Shi. 3DFed: Adaptive and Extensible Framework for Covert Backdoor Attack in Federated Learning. In *S&P*, 2023. 2

[19] Ye Li, Yanchao Zhao, Chengcheng Zhu, and Jiale Zhang. Infighting in the dark: Multi-labels backdoor attack in federated learning. *arXiv preprint arXiv:2409.19601*, 2024. 2

[20] Xiaoting Lyu, Yufei Han, Wei Wang, Jingkai Liu, Bin Wang, Jiqiang Liu, and Xiangliang Zhang. Poisoning with Cerberus: Stealthy and Colluded Backdoor Attack against Federated Learning. In *AAAI*, 2023. 2, 3, 6

[21] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*, 2017. 1

[22] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The Hidden Vulnerability of Distributed Learning in Byzantium. In *ICML*, 2018. 2

[23] Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, Farinaz Koushanfar, Ahmad-Reza Sadeghi, and Thomas Schneider. FLAME: Taming Backdoors in Federated Learning. In *USENIX Security 22*, 2022. 1, 2, 3, 5

[24] Thuy Dung Nguyen, Tuan Nguyen, Anh Tran, Khoa D. Doan, and Kok-Seng Wong. IBA: Towards Irreversible Backdoor Attacks in Federated Learning. In *NeurIPS*, 2023. 2, 6

[25] Mustafa Safa Ozdayi, Murat Kantarcioglu, and Yulia R Gel. Defending against backdoors in federated learning with robust learning rate. In *AAAI*, 2021. 1, 3, 5

[26] Ashwinee Panda, Saeed Mahloujifar, Arjun Nitin Bhagoji, Supriyo Chakraborty, and Prateek Mittal. SparseFed: Mitigating Model Poisoning Attacks in Federated Learning with Sparsification. In *AISTATS*, 2022. 3

[27] Matthias Paulik, Matt Seigel, Henry Mason, Dominic Telaar, Joris Kluivers, Rogier Van Dalen, Chi Wai Lau, Luke Carlson, Filip Granqvist, Chris Vandevelde, et al. Federated Evaluation and Tuning for On-Device Personalization: System Design & Applications. *arXiv preprint arXiv:2102.08503*, 2021. 1

[28] Phillip Rieger, Thien Duc Nguyen, Markus Miettinen, and Ahmad-Reza Sadeghi. DeepSight: Mitigating Backdoor Attacks in Federated Learning Through Deep Model Inspection. In *NDSS*, 2022. 2, 3, 5

[29] Virat Shejwalkar and Amir Houmansadr. Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for Federated Learning. In *NDSS*, 2021. 1, 2, 3, 5

[30] Karen Simonyan. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[31] Peng Sun, Xinyang Liu, Zhibo Wang, and Bo Liu. Byzantine-robust Decentralized Federated Learning via Dual-domain Clustering and Trust Bootstrapping. In *CVPR*, 2024. 2

[32] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can You Really Backdoor Federated Learning? *arXiv preprint arXiv:1911.07963*, 2019. 1, 3

[33] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris S. Papailiopoulos. Attack of the Tails: Yes, You Really Can Backdoor Federated Learning. In *NeurIPS*, 2020. 1, 2

[34] Ning Wang, Yang Xiao, Yimin Chen, Yang Hu, Wenjing Lou, and Y. Thomas Hou. FLARE: Defending Federated Learning against Model Poisoning Attacks via Latent Space Representations. In *AsiaCCS*, 2022. 2, 5

[35] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. DBA: Distributed Backdoor Attacks against Federated Learning. In *ICLR*, 2020. 1, 2, 3, 5

[36] Yueqi Xie, Minghong Fang, and Neil Zhenqiang Gong. FedREDefense: Defending against Model Poisoning Attacks for Federated Learning using Model Update Reconstruction Error. In *ICML*, 2024. 3

[37] Jiahao Xu, Zikai Zhang, and Rui Hu. Identify Backdoored Model in Federated Learning via Individual Unlearning. *arXiv preprint arXiv:2411.01040*, 2024. 3

[38] Peishen Yan, Hao Wang, Tao Song, Yang Hua, Ruhui Ma, Ningxin Hu, Mohammad Reza Haghighat, and Haibing Guan. SkyMask: Attack-Agnostic Robust Federated Learning With Fine-Grained Learnable Masks. In *ECCV*, 2025. 3

[39] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. Applied Federated Learning: Improving Google Keyboard Query Suggestions. *arXiv preprint arXiv:1812.02903*, 2018. 1

[40] Hangfan Zhang, Jinyuan Jia, Jinghui Chen, Lu Lin, and Dinghao Wu. A3FL: Adversarially Adaptive Backdoor Attacks to Federated Learning. In *NeurIPS*, 2023. 2, 6

[41] Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael W. Mahoney, Prateek Mittal, Kannan Ramchandran, and Joseph Gonzalez. Neurotoxin: Durable Backdoors in Federated Learning. In *ICML*, 2022. 1, 2, 3, 4, 5

[42] Haomin Zhuang, Mingxian Yu, Hao Wang, Yang Hua, Jian Li, and Xu Yuan. Backdoor Federated Learning by Poisoning Backdoor-Critical Layers. In *ICLR*, 2024. 1, 2, 3, 4, 5