Breaking State-of-the-Art Poisoning Defenses to Federated Learning: An Optimization-Based Attack Framework

Yuxin Yang
yuxiny22@mails.jlu.edu.cn
College of Computer Science and Technology, Jilin University
Changchun, Jilin, China
Department of Computer Science, Illinois Institute of Technology
Chicago, Illinois, USA

Qiang Li
li_qiang@jlu.edu.cn
College of Computer Science and Technology, Jilin University
Changchun, Jilin, China

Chenfei Nie
niecf21@mails.jlu.edu.cn
College of Computer Science and Technology, Jilin University
Changchun, Jilin, China

Yuan Hong
yuan.hong@uconn.edu
School of Computing, University of Connecticut
Storrs, Connecticut, USA

Binghui Wang
bwang70@iit.edu
Department of Computer Science, Illinois Institute of Technology
Chicago, Illinois, USA

Abstract

Federated Learning (FL) is a novel client-server distributed learning framework that can protect data privacy. However, recent works show that FL is vulnerable to poisoning attacks. Many defenses with robust aggregators (AGRs) are proposed to mitigate the issue, but they are all broken by advanced attacks. Very recently, some renewed robust AGRs are designed, typically with novel clipping or/and filtering strategies, and they show promising defense performance against the advanced poisoning attacks. In this paper, we show that these novel robust AGRs are also vulnerable to carefully designed poisoning attacks. Specifically, we observe that breaking these robust AGRs reduces to bypassing the clipping or/and filtering of malicious clients, and propose an optimization-based attack framework to leverage this observation. Under the framework, we then design the customized attack against each robust AGR. Extensive experiments on multiple datasets and threat models verify our proposed optimization-based attack can break the SOTA AGRs. We hence call for novel defenses against poisoning attacks to FL. Code is available at: https://github.com/Yuxin104/BreakSTOAPoisoningDefenses.

CCS Concepts

• **Security and privacy → Distributed systems security**.

Keywords

Federated Learning, Poisoning Attacks, Robust Aggregation

Binghui Wang is the corresponding author.

CIKM '24, October 21–25, 2024, Boise, ID, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679566

ACM Reference Format:
Yuxin Yang, Qiang Li, Chenfei Nie, Yuan Hong, and Binghui Wang. 2024. Breaking State-of-the-Art Poisoning Defenses to Federated Learning: An Optimization-Based Attack Framework. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3627673.3679566

1 Introduction

Federated Learning (FL) [19, 23, 33, 43], a novel client-server distributed learning paradigm, allows participating clients keep and train their data locally and only share the trained local models (e.g., gradients), instead of the raw data, with a center server for aggregation. The aggregated local models forms a shared global model, which is used by clients for their main task. FL thus has been a great potential to protect data privacy and is widely applied to medical [34], financial [20], and other privacy-sensitive applications such as on-client item ranking [23], content suggestions for on-device keyboards [6], and next word prediction [22]. However, recent works show that the invisibility of client data also renders FL vulnerable to *poisoning attacks* [2–4, 13, 31, 32, 38, 40, 46], which aim to manipulate the training phase (and testing phase) of FL to disrupt the global model behavior or/and degrade the FL performance.

To defend against the poisoning attacks to FL, numerous robust aggregation algorithms (AGRs) [5, 9, 11, 12, 15, 21, 25, 27, 29, 30, 39, 41, 42, 44] have been proposed, where the key idea is to design a robust aggregator that aims to filter malicious gradients, i.e., those largely deviate from others. For instance, the Krum AGR [5] selects the gradient that is closest to its $n - f - 2$ neighboring gradients measured by Euclidean distance, in order to filter f malicious gradients and n is the total number of clients. However, these AGRs are all broken by an advanced attack [32]. To further address the issue, some renewed AGRs equipped with an enhanced defense ability are proposed, which include FLAME [26], MDAM [14], FLDetector [45] and Centered Clipping (CC) [17] (or its variant Bucketing [18] to handle non-IID data). These new robust AGRs have shown effective defense performance against the advanced poisoning attack.

In this paper, however, we demonstrate that all these SOTA AGRs are still vulnerable to *carefully designed* (untargeted and targeted) poisoning attacks. Specifically, we first scrutinize these AGRs and find that they adopt either a novel *clipping* (e.g., in CC) or *filtering* (e.g., in MDAM and FLDetector), or the both (e.g., in FLAME) to remove the effect caused by malicious clients. Breaking these robust AGRs hence reduces to making malicious clients bypassing the clipping or/and filtering (See Figure 1). We then formalize this observation and design an optimization-based attack framework. To be specific, we first analyze the inherent attack objective and constraint within different AGRs and threat models. Then we instantiate this framework by: 1) adjusting the initial malicious gradients for targeted poisoning attacks to adhere to the constraint, and 2) constructing malicious gradients from benign ones for untargeted poisoning attacks to satisfy this constraint, thereby evading the filtering and clipping mechanisms. Finally, we propose an efficient solution to solve the optimization problem[1].

We extensively investigate the effectiveness of our attack framework against the SOTA robust AGRs under multiple experimental settings and datasets. Our empirical evaluations show that our proposed poisoning attacks vigorously disrupt these robust AGRs. We summarize the main contributions as follows:

- We show SOTA AGRs can still be broken by advanced attacks.
- We propose an optimization-based attack framework that explores both targeted and untargeted poisoning attacks against SOTA defenses under diverse scenarios.
- We validate the effectiveness of our attacks on multiple experimental settings and datasets.

2 Preliminaries

Federated Learning (FL): FL links a set of (e.g., n) clients and a server to collaboratively and iteratively train a shared global model over private client data, where the data across clients may be non independently and identically distributed (non-IID). Specifically, in a t-th round, the server selects a subset of clients $S^t \subset [n]$ and broadcasts the current global model parameters, denoted as $\mathbf{w}^t$, to the chosen clients S^t. Each chosen client $i \in S^t$ then calculates the gradient $\mathbf{g}_i^t = \partial_{\mathbf{w}^t} L(D_i; \mathbf{w}^t)$ using its local data D_i and sends $\mathbf{g}_i^t$ to the server. Here L is the loss function, e.g., cross-entropy loss. The server aggregates the collected clients' gradients using some aggregation algorithm $\text{AGR}(\mathbf{g}_{i\in[n]}^t)$, e.g., dimension-wise average in the well-known FedAvg [23] where $\text{AGR}(\mathbf{g}_{i\in[n]}^t) = \frac{1}{|S^t|}\sum_{i\in S^t} \mathbf{g}_i^t$. Finally, the server updates the global model for the next round $\mathbf{w}^{t+1}$ using the aggregator and SGD, e.g., $\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \text{AGR}(\mathbf{g}_{i\in[n]}^t)$ with a learning rate η, and broadcasts it to a new subset of randomly chosen clients S^{t+1}. This process is repeated until the global model converges or reaches the maximal round.

SOTA Robust Aggregators for FL: We review four SOTA robust AGRs FLAME-algorithmfor FL against poisoning attacks. The detailed implementations of these AGRs are shown in Algorithms 1-5.
FLAME [26]. It is a defense against targeted poisoning attacks, particularly backdoor attacks. FLAME is based on the intuition that malicious gradients tend to deviate from the benign ones in length and/or angle. To limit the impact of backdoored models, FLAME proposes a two-stage solution in each t-th round. First, it clips the client gradient with length larger than a clipping threshold q^t, i.e., $\mathbf{g}_i^t \leftarrow \frac{q^t}{\|\mathbf{g}_i^t\|_2}\mathbf{g}_i^t$, if $\|\mathbf{g}_i^t\|_2 > q^t, \forall i$. FLAME chooses q^t as $q^t = \text{MEDIAN}(\|\mathbf{g}_1\|_2^t, ..., \|\mathbf{g}_n\|_2^t)$. Second, it inputs the clipped gradients and utilizes a dynamic clustering technique HDBSCAN [8] to further filter out gradients with high angular deviations from the majority gradients. Specifically, HDBSCAN clusters the clients based on the density of cosine distance distribution on gradient pairs and dynamically determines the number of clusters. By assuming an upper bounded ($< n/2$) number of malicious clients, FLAME sets the minimum size of the largest cluster to be $n/2 + 1$ to ensure that the resulting cluster only contains benign clients. With the clipping and filtering, FLAME finally only averages the gradients of clients in the benign cluster to update the global model.
MDAM [14]. It proposes to use distributed momentum into existing aggregators (e.g., Krum, Median, TM, MDA) to enhance the robustness against poisoning attacks, and provably shows that MDA under distributed momentum (MDAM) obtains the best defense performance. Specifically, at a t-th round, each client i sends to the server the momentum $\mathbf{m}_i^t = \beta\mathbf{m}_i^{t-1} + (1-\beta)\mathbf{g}_i^t$, instead of the gradient $\mathbf{g}_i^t$, where the initial momentum is $\mathbf{m}_i^0 = 0$ and $\beta \in [0, 1)$ is the momentum coefficient. After the server receives n momentums $\{\mathbf{m}_i^t\}$, it first decides a set S^t of $n - f$ clients with the smallest diameter, i.e., $S^t \in \arg\min_{S\subset[n],|S|=n-f} \left\{\max_{i,j\in S} \left\|\mathbf{m}_i^t - \mathbf{m}_j^t\right\|_2\right\}$, where at most f clients are assumed malicious; and then updates the servers' global model as $\mathbf{w}^{t+1} = \mathbf{w}^t + \frac{1}{n-f}\sum_{i\in S^t} \mathbf{m}_i^t$.
FLDetector [45]. It aims at detecting and removing malicious clients by assessing the (in)consistency of clients' model updates in each round. Specifically, in a t-th round, the server predicts a client i's model update $\hat{g}_i^t$ using the historical global model updates and the estimated Hessian $\hat{H}^t$, i.e., $\hat{g}_i^t = g_i^{t-1} + \hat{H}^t(\mathbf{w}_t - \mathbf{w}_{t-1})$. The suspicious score s_i^t for client i is the client's average normalized Euclidean distance in the past N iterations, i.e., $s_i^t = \frac{1}{N}\sum_{r=0}^{N-1} d_i^{t-r}/\left\|d^{t-r}\right\|_1$, where $d^t = [\left\|\hat{g}_1^t - g_1^t\right\|_2, \left\|\hat{g}_2^t - g_2^t\right\|_2, ..., \left\|\hat{g}_n^t - g_n^t\right\|_2]$. Finally, FLDetector utilizes k-means with Gap statistics [35] on the clients' suspicious scores to detect and filter malicious clients.
Centered Clipping (CC) [17]. It is a simple and efficient clipping based robust AGR which provides a standardized specification for "robust" robust aggregators. To circumvent poisoning attacks, CC clips each (benign or malicious) client gradient $\mathbf{g}_i^t$ in each round t by $\mathbf{g}_i^t = \mathbf{g}_i^t \cdot \min(1, \frac{\tau}{\|\mathbf{g}_i^t\|_2})$, where τ is a predefined clipping threshold. Then the global model is updated as $\mathbf{w}^{t+1} = \mathbf{w}^t + \eta/n\sum_{i=1}^n \mathbf{g}_i^t$. Unlike the majority of other AGRs, CC is very scalable and requires only $O(n)$ computation and communication cost per round.
Centered Clipping with Bucketing (CC-B). To further adapt the robust CC AGR to heterogeneous (non-IID) datasets, [18] proposes a bucketing scheme to "mix" the data from all clients, which reduces

[1] We emphasize that our attack framework differs with [32] and extends it in two key aspects. First, the AGRs under consideration are more recent and robust than those considered in [32]. That is, all AGRs successfully broken by [32] can be also broken by our framework, while the opposite is not true. Second, our framework encompasses both targeted and untargeted attacks on robust AGRs, while [32] only focuses on untargeted attacks. Note that the generalization on targeted attacks is non-trivial. For instance, we test that the untargeted poisoning attack based on [32] completely fails to break FLAME, MDAM, and FLDetector, while our designed targeted poisoning attacks can be successful (see Table 2, Table 3, and Table 5).

the chance of any subset of the client data being consistently ignored. To be specific, in a t-th round, it first generates a random permutation π of $[n]$, and computes $\bar{\mathbf{g}}_i^t = \frac{1}{s}\sum_{k=(i-1)\cdot s+1}^{\min(n,i\cdot s)} \mathbf{g}_{\pi(k)}^t$ for $i = \{1, ..., \lceil n/s \rceil\}$, where s is the number of buckets. Then it combines with the CC AGR to update the global model, i.e., $\mathbf{w}^{t+1} = \mathbf{w}^t + \mathrm{CC}(\bar{\mathbf{g}}_1^t, \cdots, \bar{\mathbf{g}}_{\lceil n/s \rceil}^t)$.

Algorithm 1 AGR - FLAME

Input: n, $\mathbf{w}^0$, T ▷n is the number of clients, $\mathbf{w}^0$ is the initial global model parameters, and T is the number of training iterations **Output:** $\mathbf{w}^T$ ▷$\mathbf{w}^T$ is the global parameter after T iterations

```
for each training iteration $t$ in $[1, T]$ do
    for each client $i$ in $[1, n]$ do
        $\mathbf{g}_i^t \leftarrow$CLIENTUPDATE$(\mathbf{w}^{t-1}, D_i)$ ▷ The aggregator sends $\mathbf{w}^{t-1}$ to Client $i$ who trains $\mathbf{w}^{t-1}$ using its data $D_i$ locally to achieve local gradient $\mathbf{g}_i^t$ and sends $\mathbf{g}_i^t$ back to the aggregator
    end for
    $(c_{11}^t, ..., c_{nn}^t) \leftarrow \cos(\mathbf{g}_1^t, ..., \mathbf{g}_n^t)$ ▷ $\forall i, j \in (1, ..., n)$, $c_{ij}^t$ is the cosine distance between $\mathbf{g}_i^t$ and $\mathbf{g}_j^t$
    $(b_1^t, ..., b_L^t) \leftarrow$ HDBSCAN$(c_{11}^t, ..., c_{nn}^t)$ ▷$L$ is the number of admitted models, $b_l^t$ is the index of the $l$-th model
    $(e_1^t, ..., e_n^t) \leftarrow \left\|(\mathbf{w}^{t-1}, (\mathbf{g}_1^t, ..., \mathbf{g}_n^t))\right\|_2$ ▷$e_i^t$ is the Euclidean distance between $\mathbf{w}^{t-1}$ and $\mathbf{g}_i^t$ $q^t \leftarrow$ MEDIAN $(e_1^t, ..., e_n^t)$ ▷$q^t$ is the adaptive clipping bound at round $t$
    for each client $l$ in $[1, L]$ do
        $\mathbf{g}_i^t \leftarrow \mathbf{g}_i^t \cdot \min(1, (q^t/e_{b_l}^t))$ ▷$(q^t/e_{b_l}^t)$ is the clipping parameter, and $\mathbf{g}_i^t$ is clipped by the adaptive clipping bound
    end for
    $\mathbf{w}^t \leftarrow \mathbf{w}^{t-1} - \eta \sum_{l=1}^{L} \mathbf{g}_i^t / L + N(0, \sigma^2)$ ▷ Sever aggregates parameters and adds noise, and then updates the global parameter as $\mathbf{w}^t$
end for
```

Algorithm 2 AGR - FLDetector

Input: n, $\mathbf{w}^0$, N, T ▷ N is the number of past iterations, and T is the number of training iterations
Output: $\mathbf{w}^T$ ▷$\mathbf{w}^T$ is the global parameter after T iterations

```
for each training iteration $t$ in $[1, T]$ do
    for each client $i$ in $[1, n]$ do
        $\mathbf{g}_i^t \leftarrow$CLIENTUPDATE$(\mathbf{w}^{t-1}, D_i)$
        $\hat{\mathbf{g}}_i^t \leftarrow \mathbf{g}_i^{t-1} + \hat{H}^t(\mathbf{w}_t - \mathbf{w}_{t-1})$
    end for
    $d^t \leftarrow [\|\hat{\mathbf{g}}_1^t - \mathbf{g}_1^t\|_2, \|\hat{\mathbf{g}}_2^t - \mathbf{g}_2^t\|_2, ..., \|\hat{\mathbf{g}}_n^t - \mathbf{g}_n^t\|_2]$
    $s_i^t \leftarrow \frac{1}{N}\sum_{r=0}^{N-1} d_i^{t-r} / \|d^{t-r}\|_1$
    Determine the number of clusters $k$ by Gap statistics.
    if k>1 then
        Perform $k$-means clustering based on the suspicious scores $s_i^t$ with $k \leftarrow 2$. ▷ The clients in the cluster with smaller average suspicious score is benign.
    end if
    $\mathbf{g}^t \leftarrow 0$
    for each client $i$ in the benign cluster do
        $\mathbf{g}^t \leftarrow \mathbf{g}^t + \mathbf{g}_i^t$
    end for
    $\mathbf{w}^t \leftarrow \mathbf{w}^{t-1} - \eta\mathbf{g}^t$ ▷ Server updates the global parameter as $\mathbf{w}^t$
end for
```

Algorithm 3 AGR - MDAM

Input: n, $\mathbf{w}^0$, β, $\mathbf{m}^0$, T ▷ n is the number of clients, $\mathbf{w}^0$ is the initial global model parameters, $\beta \in [0, 1)$ is the momentum coefficient of all the clients, $\mathbf{m}^0 = 0$ is the initial momentum of each honest client, and T is the number of training iterations
Output: $\mathbf{w}^T$ ▷$\mathbf{w}^T$ is the global parameter after T iterations

```
for each training iteration $t$ in $[1, T]$ do
    for each client $i$ in $[1, n]$ do
        $\mathbf{g}_i^t \leftarrow$CLIENTUPDATE$(\mathbf{w}^{t-1}, D_i)$ ▷ The aggregator sends $\mathbf{w}^{t-1}$ to Client $i$ who trains $\mathbf{w}^{t-1}$ using its data $D_i$ locally to achieve local gradient $\mathbf{g}_i^t$
        $\mathbf{m}_i^t \leftarrow \beta\mathbf{m}_i^{t-1} + (1-\beta)\mathbf{g}_i^t$ ▷ Each honest client sends to the server the momentum $\mathbf{m}_i^t$
    end for
    $S^t \in \underset{S \subset [n], |S|=n-f}{\arg\min} \left\{ \max_{i,j \in S} \left\|\mathbf{m}_i^t - \mathbf{m}_j^t\right\|_2 \right\}$ ▷ Server first chooses a set $S^t$ of cardinality $n - f$ with the smallest diameter
    $\mathbf{w}^t \leftarrow \mathbf{w}^{t-1} - \frac{\eta}{n-f}\sum_{i \in S^t} \mathbf{m}_i^t$ ▷ Server then updates the global parameter as $\mathbf{w}^t$
end for
```

Algorithm 4 AGR - CC

Input: n, $\mathbf{w}^0$, τ, T ▷ τ is a predefined clipping threshold
Output: $\mathbf{w}^T$ ▷$\mathbf{w}^T$ is the global parameter after T iterations

```
for each training iteration $t$ in $[1, T]$ do
    for each client $i$ in $[1, n]$ do
        $\mathbf{g}_i^t \leftarrow$CLIENTUPDATE$(\mathbf{w}^{t-1}, D_i)$
        $\mathbf{g}_i^t \leftarrow \mathbf{g}_i^t \cdot \min(1, \frac{\tau}{\|\mathbf{g}_i^t\|_2})$ ▷$\tau$ is the clipping parameter
    end for
    $\mathbf{w}^t \leftarrow \mathbf{w}^{t-1} - \eta\sum_{i \in [n]} \mathbf{g}_i^t$
end for
```

Algorithm 5 AGR - CC-B

Input: n, $\mathbf{w}^0$, τ, s, T ▷ τ is a predefined clipping threshold, and s is the number of buckets
Output: $\mathbf{w}^T$ ▷$\mathbf{w}^T$ is the global parameter after T iterations

```
for each training iteration $t$ in $[1, T]$ do
    for each client $i$ in $[1, n]$ do
        $\mathbf{g}_i^t \leftarrow$CLIENTUPDATE$(\mathbf{w}^{t-1}, D_i)$
    end for
    Pick random permutation $\pi$ of $[n]$
    for each parameter $i$ in $[1, \lceil n/s \rceil]$ do
        $\bar{\mathbf{g}}_i^t = \frac{1}{s}\sum_{k=(i-1)\cdot s+1}^{\min(n,i\cdot s)} \mathbf{g}_{\pi(k)}^t$ ▷ Bucketing mixes the data from all clients
    end for
    $\bar{\mathbf{g}}_i^t \leftarrow \bar{\mathbf{g}}_i^t \cdot \min(1, \frac{\tau}{\|\bar{\mathbf{g}}_i^t\|_2})$ ▷$\tau$ is the clipping parameter
    $\mathbf{w}^t \leftarrow \mathbf{w}^{t-1} - \eta\sum_{i \in [n]} \bar{\mathbf{g}}_i^t$
end for
```

Table 1: The knowledge of adversary. Note that there exists no tailored attack on FLDetector. Our results in Section 5 show our AGR-agnostic attacks are already highly effective.

The SOTA AGRs	Adversary's Knowledge	
-our attack	AGR tailored	Gradients known
FLAME/MDAM	✓	✓
-targeted	✗	✓
CC/CC-B	✓	✗
-untargeted	✗	✗
FLDetector-*targeted*	✗	✗

3 Threat Model

In this section, we discuss the threat model of attacking the SOTA robust AGRs, i.e., targeted poisoning attacks to FLAME, MDAM, and FLDetector, and untargeted poisoning attacks to CC and CC-B. Note here that the targeted attack mainly refers backdoor attacks.

Adversary's Objective. *For targeted poisoning attacks*, an adversary aims to optimize the existing malicious gradients obtained by backdoor poisoning to evade filtering or/and clipping of the robust AGR (e.g., FLAME, MDAM, and FLDetector), so that the resulting global model can achieve a high level on both the main task accuracy and backdoor accuracy. *For untargeted poisoning attacks*, the adversary's goal is to craft malicious gradients based on benign gradients to disrupt the server's robust aggregation (e.g., via CC and CC with Bucketing), consequently diminishing the overall main task accuracy of the global model.

Adversary's Capability. We assume the total number of malicious clients is $f < n/2$. All malicious clients can collude with each other and have indexes within $[f]$, i.e., from 1 to f, without loss of generality. For untargeted poisoning attacks, a malicious client can carefully modify its normally trained gradient to be a malicious one such that it can fool the robust AGR. For targeted backdoor attacks, we assume the adversary uses the strong model replacement attack [2], where it aims to replace the true global model $\mathbf{w}^{t+1} = \mathbf{w}^t + \frac{\eta}{n}\sum_{i=1}^{n} \mathbf{g}_i^t$ with any model $\mathbf{x}$ by poisoning the gradients $\mathbf{g}^t_{i\in[f]}$. In our scenario, we extend the model replacement attack from 1 to f malicious clients. Specifically, the f poisoned client gradients, denoted as $\mathbf{g}^p_{i\in[f]}$[2] to differentiate with the benign gradients $\mathbf{g}_{j\in[f+1,n]}$, has the relationship: $\sum_{i=1}^{f} \mathbf{g}_i^p = \frac{n}{\eta}(\mathbf{x} - \mathbf{w}^t) - \sum_{j=f+1}^{n} \mathbf{g}_j$. As the global model converges, each local model may be close enough to the global model such that the benign gradients start to cancel out, i.e., $\sum_{j=f+1}^{n} \mathbf{g}_j = 0$ [2]. Hence, we have $\sum_{i=1}^{f} \mathbf{g}_i^p \approx \frac{n}{\eta}(\mathbf{x} - \mathbf{w}^t)$. Then an adversary can simply solve for the poisoned gradients of the malicious clients as: $\mathbf{g}_i^p \approx \frac{n}{f\eta}(\mathbf{x} - \mathbf{w}^t), \forall i \in [f]$.

Adversary's Knowledge. We consider two dimensions: knowledge of the AGR aggregator and knowledge of the gradients shared by benign clients (see Table 1). According to whether the adversary knows the AGR aggregator, we classify the malicious attacks into *AGR-tailored* and *AGR-agnostic*. Similarly, we also divide the attacks into *gradients-known* and *gradients-unknown* based on whether the adversary is aware of the benign clients' gradients. Note that the adversary performing the gradients-unknown attack can utilize the clean data of the malicious clients to obtain benign gradients.

4 Optimization-Based Poisoning Attacks to SOTA Robust AGRs in FL

Recall that SOTA defenses against untargeted and targeted poisoning attacks all design robust AGRs that involve clipping or/and filtering malicious gradients based on their statistic differences with benign gradients. Hence, the main insight of our attack framework is to carefully create malicious gradients $\{\mathbf{g}_i^c\}_{i\in[f]}$ to evade these operations in SOTA AGRs. Formally, we introduce a general optimization formula as follows:

$$\begin{aligned} \gamma = & \mathop{\arg\max}_{\gamma, i\in[f], j\in[f+1,n]} \quad dist(\mathbf{g}_i^c, \mathbf{g}_j), \\ & s.t.\ \ dist(\mathbf{g}_i^c, \mathbf{g}_j) \le th,\ \forall i \in [f], j \in [f+1, n], \\ & \quad \mathbf{g}_i^c = \mathcal{F}(\mathbf{g}_i^p, \mathbf{g}^b, \gamma),\ \forall i \in [f]. \end{aligned} \tag{1}$$

At a high-level, our objective function aims to create the malicious gradients $\{\mathbf{g}_i^c\}_{i\in[f]}$(For ease of description, we also use $\mathbf{g}_i^c$ to indicate malicious momentum $\mathbf{m}_i^c$) such that their maximum distance from the benign gradients $\{\mathbf{g}_j\}_{j\in[f+1,n]}$ is within a predefined/calculated threshold th used by the filtering or/and clipping operation in SOTA AGRs. Note that directly computing $\{\mathbf{g}_i^c\}_{i\in[f]}$ is challenging. To address it, we observe malicious gradients can be built (characterized by a function $\mathcal{F}$) from known poisoned gradients $\mathbf{g}_i^p$ and a reference benign gradient $\mathbf{g}^b$, which are coupled with a scaling hyperparameter γ. Hence, the final optimization problem of learning $\{\mathbf{g}_i^c\}_{i\in[f]}$ can be reduced to learning the γ.

Next, we investigate the vulnerabilities of the SOTA robust AGRs and then instantiate our attack framework in Equation (1) by designing optimization-based attacks against these AGRs one-by-one.

4.1 Targeted Poisoning Attacks to FLAME, MDAM, and FLDetector

AGR-Tailored Attack to FLAME. As stated in Section 2, FLAME first limits the gradients with large length to not exceed a clipping threshold q^t in each t-th round. To attack this clipping, we deploy projected gradient descent (PGD) with the adversary periodically projecting their client gradients on a small ball centered around the global model $\mathbf{w}^t$. To be specific, the malicious client calculates q^t so that the malicious gradients $\mathbf{g}^p_{i\in[f]}$ respect the constraint $\|\mathbf{g}_i^p\|_2 \le \left|q^t - \|\mathbf{w}^t\|_2\right|, \forall i \in [f]$, i.e., malicious gradients distribute over a ball with $\mathbf{w}^t$ as the center and $\left|q^t - \|\mathbf{w}^t\|_2\right|$ as the radius.

Additionally, FLAME uses pairwise cosine distances to measure the angular differences between all pairs of model gradients, and applies the HDBSCAN clustering algorithm to further filter the malicious gradients with large angular deviations. Hence, a successful attack also requires crafting the malicious gradients (denoted as $\mathbf{g}^c_{i\in[f]}$) to evade the filtering. Here, we propose to rotate $\mathbf{g}^p_{i\in[f]}$ to be the corresponding $\mathbf{g}^c_{i\in[f]}$ (with a reference benign gradient $\mathbf{g}^b$), so that its maximum cosine distance with all benign gradients is not greater than the maximum cosine distance among benign gradients. Formally, we can express the optimization problem as:

$$\begin{aligned} \gamma = & \mathop{\arg\max}_{\gamma, i\in[f], j\in[f+1,n]} \quad cos(\mathbf{g}_i^c, \mathbf{g}_j), \\ & s.t.\ \ cos(\mathbf{g}_i^c, \mathbf{g}_j) \le \max_{k,j\in[f+1,n]} cos(\mathbf{g}_k, \mathbf{g}_j), \\ & \quad \mathbf{g}_i^c = \|\mathbf{g}_i^p\|_2 / \|\mathbf{g}_i^u\|_2 \cdot \mathbf{g}_i^u;\ \ \mathbf{g}_i^u = \mathbf{g}_i^p + \gamma(\mathbf{g}^b - \mathbf{g}_i^p). \end{aligned} \tag{2}$$

[2] For notation simplicity, we will omit the round t in the gradients $\mathbf{g}_i^t$.

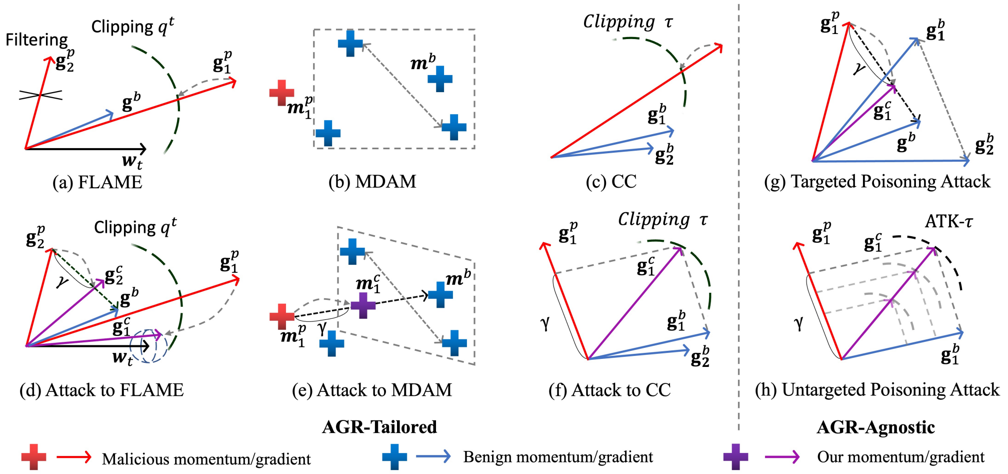

Figure 1: Illustration of the SOTA robust aggregation algorithms (a)-(c) in FL, our AGR-tailored attacks (d)-(f) and AGR-agnostic attacks (g)-(h) on them. (a) FLAME: it defends against the malicious gradients via clipping and filtering gradients that with high length and angular deviations, respectively. (b) MDAM: it chooses a subset of $n - f$ momentums with the smallest diameter for aggregation, i.e., filter out a bounded number of f malicious gradients. (c) CC: it corrects malicious gradients via a centered clipping with a parameter τ. (d) Our attack to FLAME: we project the length-deviating malicious gradients and rotate the angle-deviating malicious gradients to evade FLAME. (e) Our attack to MDAM: we optimize the original malicious momentums to new ones such that MDAM selects (part of) the new malicious ones into the subset for aggregation. (f) Our attack to CC: we construct malicious gradients from any benign one to avoid the center clipping. (g) Our AGR-agnostic targeted poisoning attack (on FLAME, MDAM, and FLDetector): we adjust malicious gradients to approach benign gradients, based on the Euclidean distance metric, to evade SOTA defenses. (h) Our AGR-agnostic untargeted poisoning attack (to CC): we generate malicious gradients of length ATK-τ by leveraging any of benign gradients to evade clipping for agnostic parameters.

where we set the reference $\mathbf{g}^b$ as averaging certain benign gradients. Specifically, if the benign clients' gradients are known to the adversary (i.e., gradients-known), it can average benign clients' benign gradients, i.e., $\mathbf{g}^b = \mathrm{Avg}(\mathbf{g}_{\{i\in[f+1,n]\}})$. On the other hand, when the benign clients' gradients are unknown (i.e., gradients-unknown), the adversary can average the "benign" gradients obtained by malicious clients on their clean data (i.e., without poisoning), i.e., $\mathbf{g}^b = \mathrm{Avg}(\mathbf{g}_{\{i\in[1,f]\}})$. γ is the scaling hyperparameter that we aim to learn and $\mathbf{g}^p_{i\in[f]}$ are defined in Section 3.

AGR-Tailored Attack to MDAM. MDAM chooses a set of $n - f$ momentums with the smallest diameter aiming to filter out the possible f malicious clients. To break this robust ARG, we need to force it to choose our tailored malicious momentums and mix them with benign ones during training. Here, we would like to replace (any) f benign clients' momentums that would have been chosen by MDAM with f *identical* tailored malicious momentums. To be specific, we learn to optimize the malicious momentums $\mathbf{m}^p_{i\in[f]}$ to linearly approach a benign $\mathbf{m}^b$ until there exists a malicious momentum from $\mathbf{m}^c_{i\in[f]}$ whose largest distance with respect to the benign momentums in any chosen benign set $S \subset [f+1, n]$ with $|S| = n - 2f$ is less than the maximum distance between any two benign momentums. The optimization problem is defined as:

$$\begin{aligned}\gamma = &\mathop{\arg\max}_{\gamma, i\in[f], j\in S\subset[f+1,n], |S|=n-2f} \left\|\mathbf{m}_i^c - \mathbf{m}_j\right\|_2, \\ s.t.\ &\left\|\mathbf{m}_i^c - \mathbf{m}_j\right\|_2 \le \max_{k,j\in[f+1,n]} \left\|\mathbf{m}_k - \mathbf{m}_j\right\|_2, \mathbf{m}_i^c = \mathbf{m}_i^p + \gamma(\mathbf{m}^b - \mathbf{m}_i^p).\end{aligned} \tag{3}$$

where $\mathbf{m}^b$ is the average of certain benign momentum. Similarly, for gradients-known and -unknown, we set $\mathbf{m}^b = \mathrm{Avg}(\mathbf{m}_{\{i\in[f+1,n]\}})$ and $\mathbf{m}^b = \mathrm{Avg}(\mathbf{m}_{\{i\in[1,f]\}})$, respectively. $\mathbf{m}^p_{i\in[f]}$ are momentum of malicious gradients defined as $\mathbf{m}_i^p = \beta\mathbf{m}_i^p + (1-\beta)\mathbf{g}_i^p, \forall i \in [f]$.

AGR-Agnostic Attacks (to FLAME, MDAM, and FLDetector). *It is challenging to design tailored attacks on FLDetector, due to it uses the history information in each round that is not easy to incorporate into the attack optimization. To relax it, we propose an AGR-agnostic attack formulation; and if the AGR-agnostic attack is already effective, then AGR-tailored attack can performance better.* Specifically, in this setting, the adversary does not know which AGR the defense uses. To design effective AGR-agnostic attacks, we need to uncover the shared property in the existing robust AGRs. Particularly, we note that, though with different techniques, existing robust AGRs mainly perform statistical analysis on client models and identify malicious clients as those largely deviate from others based on some similarity metric such as Euclidean distance and cosine similarity. Based on this intuition, we hence propose to adjust the malicious gradients to be close to the benign ones such that these malicious gradients can be selected by (any) robust AGRs. W.l.o.g, we use Euclidean distance as the similarity metric and extend Equation (3) to a more general case. Specifically, we optimize $\mathbf{g}^p_{i\in[f]}$ to approach $\mathbf{g}^b$ so that its maximum distance with respect to any benign gradient is upper bounded by the maximum distance between any two benign gradients. Formally, the equation can be expressed as follows:

$$\begin{aligned}\gamma = &\mathop{\arg\max}_{\gamma, i\in[f], j\in[f+1,n]} \left\|\mathbf{g}_i^c - \mathbf{g}_j\right\|_2, \\ s.t.\ &\left\|\mathbf{g}_i^c - \mathbf{g}_j\right\|_2 \le \max_{k,j\in[f+1,n]} \left\|\mathbf{g}_k - \mathbf{g}_j\right\|_2, \mathbf{g}_i^c = \mathbf{g}_i^p + \gamma(\mathbf{g}^b - \mathbf{g}_i^p).\end{aligned} \tag{4}$$

4.2 Untargeted Poisoning Attacks to CC and CC with Bucketing

AGR-Tailored Attack to CC. CC clips each client gradient $\mathbf{g}_i$ by $\mathbf{g}_i \min(1, \frac{\tau}{\|\mathbf{g}_i\|_2})$ to reduce the negative effect caused by malicious gradients with a large length. However, we observe that clipping is ineffective if we can always maintain $\frac{\tau}{\|\mathbf{g}_i\|_2} \geq 1$, i.e., $\|\mathbf{g}_i\|_2 \leq \tau$. Under this, we can create the malicious gradients $\mathbf{g}^c_{i\in[f]}$ such that:

$$\begin{aligned} &\gamma = \underset{\gamma, i\in[f]}{\arg\max} \left\|\mathbf{g}_i^c\right\|_2, \\ &s.t. \left\|\mathbf{g}_i^c\right\|_2 \leq \tau, \;\; \mathbf{g}_i^c = \mathbf{g}^b + \gamma \mathbf{g}_i^p, \;\; \mathbf{g}_i^p = -\mathrm{sign}(\mathbf{g}^b). \end{aligned} \tag{5}$$

There are many ways to set malicious gradients $\mathbf{g}^p_{i\in[f]}$. Motivated by [32], one simple yet effective way is setting $\mathbf{g}^p_{i\in[f]}$ as the *inverse* sign of the reference benign gradient $\mathbf{g}^b$, which can be a randomly chosen benign gradient.

AGR-Tailored Attack to CC-B. Our attack on CC with bucketing is similar to that on CC. As seen in Section 2, bucketing is a post-processing step on client gradients mainly mitigating the non-IID data across clients. In principle, a successful attack on CC will render the bucketing aggregation ineffective as well. Here, we directly craft malicious gradients using Equation (5) and then perform CC aggregation with bucketing on the crafted malicious gradients.

AGR-Agnostic Attack (to CC and CC-B). τ is the only hyperparameter in CC. In this attack setting, τ is unknown to the adversary. To differentiate between τ used in CC and that in the attack, we name it as CC-τ and ATK-τ, respectively. In practice, the used CC-τ's often have a format of 10^x. Hence the adversary can exploit these potential values as ATK-τ to execute the attack in Equation (5). Note that there is no such limitation and our experimental results in Section 5 show a larger value of ATK-τ always ensures the attack to be effective, whatever the true CC-τ is.

4.3 Solving for The Scaling Hyperparameter γ

Both the targeted and untargeted poisoning attacks to FL involve optimizing the scaling hyperparameter γ. For the untargeted attacks, the malicious gradients $\mathbf{g}^c_{i\in[f]}$ are obtained by *maximizing* γ to render the attack $\mathbf{g}^c_{i\in[f]} = \mathbf{g}^b + \gamma \mathbf{g}^p_{i\in[f]}$ be effective. For the targeted attacks, in contrast, an adversary generates $\mathbf{g}^c_{i\in[f]}$ by exploring the *minimum* γ to maintain the attack effectiveness, e.g., $\mathbf{g}^c_{i\in[f]} = \mathbf{g}^p_{i\in[f]} + \gamma(\mathbf{g}^b - \mathbf{g}^p_{i\in[f]})$. Algorithm 6 shows the details of solving γ in the two scenarios. Specifically, we start with a small (or large) γ for untargeted (or targeted) attacks and increase (or decrease) γ with a step size *step* until O returns "True", where O takes as input the union of the malicious and benign gradients, i.e., $\mathbf{g}_{i\in[n]} = \mathbf{g}^c_{i\in[f]} \cup \mathbf{g}_{\{i\in[f+1,n]\}}$, and γ, and outputs "True" if the obtained $\mathbf{g}^c_{i\in[f]}$ in Equations (2)-(5) satisfies the adversarial objective. We halve the step size for each γ update to make the search finer.

Complexity and Convergence Analysis. Algorithm 6 iteratively optimizes γ to satisfy the condition $O(\mathbf{g}_{i\in[n]}, \gamma) ==$ True. In each iteration, it verifies the adversarial objective in Equations (2)-(5), e.g., computes the pairwise gradient/momentum distance among benign clients, and that between malicious clients and benign clients and checks whether the inequality satisfies. The complexity per iteration is O($(n^2 - f^2) *$ #model parameters), where n and f are the total number of clients and malicious clients, respectively. On the other hand, Algorithm 6 guarantees to converge and stops when step $\leq \varepsilon$. Note that step is initialized as $\gamma_{init}/2$ and halved in each iteration. So the convergence iteration m is obtained when $\gamma_{init}/2^{m+1} \leq \varepsilon$, which means $m = \log_2(\gamma_{init}/\varepsilon) - 1$.

Algorithm 6 Learning the Scaling Hyperparameter γ

```
Input: $\gamma_{init}, \varepsilon, O, \mathbf{g}_{i\in[n]}$
Output: $\gamma_{succ}$
step ← $\gamma_{init}/2, \gamma \leftarrow \gamma_{init}$
while $|\gamma_{succ} - \gamma| > \varepsilon$ do
    if $O(\mathbf{g}_{i\in[n]}, \gamma)$ == True then
        $\gamma_{succ} \leftarrow \gamma$
        if attacking FLAME, MDAM, or FLDetector then
            $\gamma \leftarrow (\gamma - \text{step}/2)$
        else
            $\gamma \leftarrow (\gamma + \text{step}/2)$
        end if
    else
        if attacking FLAME, MDAM, or FLDetector then
            $\gamma \leftarrow (\gamma + \text{step}/2)$
        else
            $\gamma \leftarrow (\gamma - \text{step}/2)$
        end if
    end if
    step=step/2
end while
```

5 Experiments

In this section, we evaluate our attack framework on the SOTA robust AGRs. We first set up the experiments and then show the attack results on each robust AGR.

5.1 Experimental Setup

Datasets and Architectures. Following existing works [14, 17, 28, 40, 45], we use three benchmark image datasets, i.e., FMNIST, CIFAR10, and FEMNIST [7], where the first two datasets are IID distributed, while the third one is non-IID distributed. FMNIST has 60K training images and 10K testing images from 10 classes with image size 28x28. CIFAR10 has 50K training images and 10K testing images. FEMNIST includes 3,383 clients, 62 classes, and a total of 805,263 grayscale images. For FL training, we selected 1,000 out of the 3,383 clients. For FMNIST and FEMNIST, we consider a convolutional neural network (CNN) with 2 convolutional (Conv) layers followed by 2 fully-connected (FC) layers. While for CIFAR10, we use a CNN with 3 Conv layers and 3 FC layers on the targeted attack, and a ResNet-20 [16] on the untargeted attack. We follow FLAME, MDAM, FLDetector, CC(-B) to set the number of total clients and fraction of clients selected in each round. For instance, the total number of clients in CC is 50, and all clients are selected for training. We set the backdoor classes to 4 and 7 for targeted poisoning attacks, and use the cross entropy loss for federated training. We set the learning rate for server's global aggregation as 1.0 and client training as 0.1 and total number of rounds is 200.

Parameter Setting. We set the ratio of malicious clients f/n to be {2%, 5%, 10%, 20%} for FLAME, CC, and CC-B, and {5%, 10%, 20%,

Table 2: Results of our attack and the SOTA DBA against FLAME under various threat models. Our attacks show an BA improvement ranging from 31% to 94% over the SOTA DBA with comparable or better MA under the same settings.

Dataset	No attack (MA)	f/n (%)	DBA (MA / BA)	Gradients known		Gradients unknown	
				AGR tailored	AGR agnostic	AGR tailored	AGR agnostic
FMNIST	0.92	2	0.85 / 0.02	0.93 / 0.61	0.91 / 0.53	0.90 / 0.57	0.90 / 0.43
		5	0.88 / 0.01	0.91 / 0.84	0.90 / 0.71	0.91 / 0.77	0.89 / 0.66
		10	0.85 / 0.01	0.92 / 0.95	0.92 / 0.83	0.92 / 0.90	0.92 / 0.77
		20	0.86 / 0.19	0.92 / 0.96	0.90 / 0.84	0.92 / 0.93	0.92 / 0.74
CIFAR10	0.71	2	0.70 / 0.03	0.72 / 0.65	0.71 / 0.60	0.69 / 0.53	0.70 / 0.48
		5	0.70 / 0.13	0.70 / 0.75	0.72 / 0.74	0.70 / 0.74	0.72 / 0.50
		10	0.67 / 0.18	0.72 / 0.79	0.70 / 0.78	0.67 / 0.79	0.68 / 0.54
		20	0.69 / 0.27	0.72 / 0.79	0.72 / 0.79	0.72 / 0.80	0.72 / 0.63
FEMNIST	0.94	2	0.90 / 0.20	0.91 / 0.64	0.91 / 0.58	0.92 / 0.60	0.91 / 0.51
		5	0.89 / 0.19	0.90 / 0.89	0.92 / 0.82	0.90 / 0.82	0.90 / 0.74
		10	0.90 / 0.31	0.91 / 0.92	0.90 / 0.83	0.91 / 0.86	0.92 / 0.82
		20	0.91 / 0.43	0.91 / 0.93	0.92 / 0.91	0.92 / 0.92	0.91 / 0.86

Table 3: Results of our attack and DBA against MDAM under various threat models with the momentum coefficient $\beta = 0.9$. Our attacks significantly outperforms DBA under the same setting. For instance, when $f/n \geq 20\%$, our attacks show an BA improvement ranging from 6% to 98% over the SOTA DBA with comparable MA.

Dataset	No attack (MA)	f/n (%)	DBA (MA / BA)	Gradients known		Gradients unknown	
				AGR tailored	AGR agnostic	AGR tailored	AGR agnostic
FMNIST	0.93	5	0.93 / 0.01	0.93 / 0.01	0.93 / 0.01	0.92 / 0.01	0.92 / 0.01
		10	0.93 / 0.01	0.93 / 0.01	0.92 / 0.01	0.93 / 0.01	0.93 / 0.01
		20	0.93 / 0.01	0.93 / 0.99	0.92 / 0.61	0.92 / 0.68	0.93 / 0.45
		30	0.93 / 0.01	0.93 / 1.00	0.92 / 0.74	0.91 / 0.99	0.92 / 0.67
CIFAR10	0.73	5	0.73 / 0.03	0.73 / 0.14	0.73 / 0.11	0.73 / 0.11	0.73 / 0.08
		10	0.73 / 0.03	0.73 / 0.67	0.73 / 0.56	0.73 / 0.46	0.72 / 0.43
		20	0.73 / 0.05	0.72 / 0.81	0.73 / 0.81	0.74 / 0.78	0.73 / 0.80
		30	0.73 / 0.47	0.73 / 0.92	0.72 / 0.89	0.73 / 0.91	0.72 / 0.88
FEMNIST	0.96	5	0.96 / 0.24	0.96 / 0.30	0.96 / 0.27	0.96 / 0.29	0.96 / 0.24
		10	0.95 / 0.43	0.96 / 0.83	0.96 / 0.64	0.95 / 0.44	0.95 / 0.47
		20	0.96 / 0.68	0.96 / 0.89	0.96 / 0.88	0.96 / 0.74	0.95 / 0.78
		30	0.94 / 0.76	0.95 / 0.96	0.95 / 0.95	0.94 / 0.87	0.95 / 0.85

30%} for MDAM and FLDetector, consider their different defense performance. In each malicious client, we set the data poisoning rate as 50%, meaning 50% of the client data are poisoned. In MDAM, we consider momentum coefficients $\beta = \{0, 0.6, 0.9, 0.99\}$, where $\beta = 0$ means we do not use the momentum and it reduces to the standard MDA. In CC and CC-B, both the defense CC-τ and attack ATK-τ are set within $\{0.1,1,10,100,1000\}$. We also investigate the effect of Bucketing and set the number of buckets s in $\{0,2,5,10\}$, where $s = 0$ means we do not use buckets.

Evaluation Metric. For targeted backdoor poisoning attacks, we use both main task accuracy (MA) and backdoor accuracy (BA) as the evaluation metrics. An attack obtaining a *larger* MA and a larger BA indicates it is more effective. For untargeted poisoning attacks, we aim to reduce the main task accuracy. Hence, a *smaller* MA indicates better attack effectiveness.

5.2 Results of Our Attacks on FLAME, MDAM, and FLDetector

Comparing with a SOTA Targeted Attack. We choose the SOTA Distributed Backdoor Attack (DBA) to FL [40] as a baseline targeted poisoning attack for comparison. DBA decomposes a global trigger into several *local* triggers and embeds these local triggers separately into the training data of different malicious clients. Compared with the classic *centralized* backdoor that injects the *global* trigger, DBA is shown to be more persistent, stealthy, and effective (more details about DBA can be referred to [40]). To better show the attack effectiveness, our attack just uses the centralized backdoor, where we set the global trigger size to be the same as that in DBA. Following DBA, we use the rectangle pattern as the global trigger and DBA separates the global trigger into four local triggers. We also show the DBA performance with different number of local triggers and the results are very close (see Table 6).

Results on Attacking FLAME. The experimental results on different threat models are shown in Table 2. we have the following key observations: 1) *DBA is ineffective against FLAME, while our attack is effective.* DBA obtains BAs that are low. This shows FLAME can defend against DBA, which is also verified in [26]. In contrast, our attack achieves very high BAs, validating that it can break FLAME and our optimization-based attack is promising. 2) *Our attack can better maintain the FL performance than DBA.* Our attack obtains identical or better MAs than DBA and has closer MAs to those under no attack. This implies our attack does not affect the main task

Table 4: Results of attacking CC on IID FMNIST and CIFAR10, and attacking CC-B on non-IID FEMNIST with CC-τ = 10. Our attacks significantly outperform AGR-agnostic LIE and AGR-tailored Fang, especially when f/n is large and dataset is non-IID.

Dataset	No attack (MA)	f/n (%)	LIE	Gradients known							Gradients unknown						
				AGR-tailored		AGR-agnostic (ATK-τ)					AGR-tailored		AGR-agnostic (ATK-τ)				
				Fang	Ours	0.1	1	10	100	1000	Fang	Ours	0.1	1	10	100	1000
FMNIST	0.84	2	0.84	0.76	0.74	0.84	0.82	0.74	0.75	0.76	0.76	0.75	0.84	0.83	0.75	0.74	0.74
		5	0.84	0.76	0.72	0.84	0.82	0.72	0.73	0.72	0.77	0.69	0.84	0.81	0.69	0.71	0.70
		10	0.84	0.65	0.58	0.84	0.78	0.58	0.59	0.65	0.69	0.62	0.84	0.77	0.62	0.67	0.48
		20	0.84	0.49	0.40	0.83	0.76	0.40	0.11	0.10	0.51	0.47	0.83	0.76	0.47	0.15	0.09
CIFAR10	0.66	2	0.66	0.66	0.64	0.65	0.65	0.64	0.10	0.11	0.69	0.60	0.65	0.65	0.60	0.12	0.10
		5	0.60	0.66	0.45	0.65	0.63	0.45	0.32	0.19	0.66	0.48	0.65	0.65	0.48	0.47	0.21
		10	0.49	0.65	0.11	0.65	0.64	0.11	0.08	0.09	0.66	0.14	0.62	0.64	0.14	0.11	0.10
		20	0.43	0.21	0.11	0.61	0.60	0.11	0.09	0.13	0.25	0.10	0.62	0.57	0.10	0.09	0.11
FEMNIST	0.92	2	0.92	0.89	0.83	0.92	0.91	0.83	0.11	0.11	0.90	0.87	0.92	0.90	0.87	0.12	0.09
		5	0.89	0.80	0.10	0.92	0.90	0.10	0.09	0.11	0.88	0.09	0.92	0.89	0.09	0.12	0.09
		10	0.79	0.65	0.09	0.92	0.88	0.09	0.10	0.09	0.80	0.12	0.92	0.87	0.12	0.09	0.11
		20	0.57	0.32	0.09	0.91	0.86	0.09	0.07	0.12	0.52	0.13	0.92	0.79	0.13	0.11	0.09

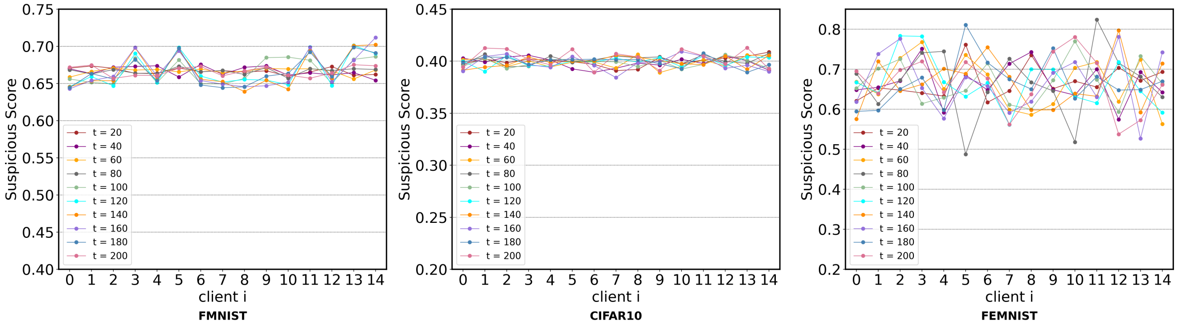

Figure 2: Suspicious scores per client and per FL round computed by FLDetector under our AGR-agnostic and gradient-unknown attack. Here, clients $0-4$ are malicious and the remaining ones are benign, and t is the FL training round. We observe the suspicious scores are similar in all clients and FL rounds, hence making k-means clustering hard to detect the malicious scores.

Table 5: Results of our AGR-agnostic and gradient-unknown targeted poisoning attack against FLDetector. The "No attack (MA)" are 0.93, 0.73, and 0.96 on the FMNIST, CIFAR10, and FEMNIST datasets, respectively. Note that FLDetector is customized for defending against the centralized BA.

f/n (%) / Dataset	5	10	20	30
	MA / BA	MA / BA	MA / BA	MA / BA
FMNIST	0.93 / 0.86	0.92 / 0.99	0.93 / 1.00	0.93 / 0.99
CIFAR10	0.75 / 0.25	0.74 / 0.72	0.75 / 0.84	0.74 / 0.89
FEMNIST	0.90 / 0.48	0.90 / 0.63	0.90 / 0.73	0.90 / 0.90

performance. 3) *In general, our attack has better performance under a strong threat model than that under a weak threat model.* Specifically, comparing AGR-tailored vs. AGR-agnostic and gradients-known vs. gradients-unknown, all the BAs obtained by our attack are larger, while with similar MAs. 4) *Defending against attacks on non-IID datasets is more challenging than on IID datasets.* We notice that the BAs on the non-IID FEMNIST are much larger than those on the IID FMNIST and CIFAR10. This is because the client models can be more diverse when trained on non-IID data, thus making it more difficult to differentiate between benign models and malicious models via similarity metrics. 5) *More malicious clients yield better attack performance.* Our attack can obtain larger BAs, with an increasing number of malicious clients. This is obvious, since more malicious clients have a larger space to perform the attack. We notice that 5%-10% colluding malicious clients are sufficient to obtain a promising attack performance.

Results on Attacking MDAM. The results are in Table 3 where the momentum coefficient $\beta = 0.9$, as suggested in [14]. We have similar observations as those on attacking FLMAE. For instance, our attack achieves larger MAs and BAs than DBA, showing our attack is more effective than DBA. MDAM can completely defend against DBA on IID datasets when the fraction of malicious clients is small. Also, defending against attacks on non-IID dataset is more challenging. Similarly, all these results show that our optimization-based attack can evade the filtering strategy in MDAM.

Results on Attacking FLDetector. Table 5 shows the results of our (AGR-agnostic and gradient-unknown) attack on FLDetector. The results reveal FLDetector almost fails to defend against our attack under the least adversary knowledge. To understand the underlying reason, we show in Figure 2 the computed suspicious scores by FLDetector for each client and in each FL round. We observe the suspicious scores are similar across all (malicious and benign) clients and FL rounds. Hence, it is challenging to use k-means to detect malicious clients based on these suspicious scores.

Table 6: DBA results on FLAME and MDAM ($\beta = 0.9$) with different number of local triggers. Here, we choose $f/n = 10\%$ on FMNIST for simplicity.

#local triggers	2 (MA / BA)	3 (MA / BA)	4 (MA / BA)
FLAME	0.86 / 0.02	0.86 / 0.02	0.85 / 0.01
MDAM	0.93 / 0.01	0.93 / 0.00	0.93 / 0.01

5.3 Results of Our Attacks to CC and CC-B

Comparing with SOTA Untargeted Attacks. We choose two SOTA untargeted poisoning attack methods, namely LIE [3] (AGR-agnostic) and Fang [13] (AGR-tailored), for comparison. Roughly speaking, LIE computes the average μ and standard deviation δ of the benign gradients, and computes a coefficient z based on the total number of benign and malicious clients, and finally computes the malicious update as $\mu + z\delta$. Fang calculates the average μ of benign gradients and introduces a perturbation $\mathbf{g}^p = -\text{sign}(\mu)$. Ultimately, it computes a malicious update as $\mathbf{g}^c = \mu + \gamma \mathbf{g}^p$. The attack initiates a relatively large γ and iteratively halves it until γ yields promising attack performance. Note that, unlike targeted poisoning attacks, our goal in this attack setting is to achieve as small MAs as possible.

Results on Attacking CC. As stated in Section 4.2, CC has the only important parameter τ. An AGR-tailored attack implies ATK-τ set by the adversary equals to the true CC-τ, while an AGR-agnostic attack means ATK-τ and CC-τ are different. Table 4 shows our attack results on IID FMNIST and CIFAR10 where we set CC-τ=10 and try different ATK-τ's. We observe that: 1) Our attack, even in the gradients-unknown setting, is more effective than LIE and Fang with known gradients; 2) Our attack obtains small MAs when the adversary knows the true CC-τ; 3) By setting ATK-τ to be a larger value, e.g., 1000, our attack is always effective. This is because a larger ATK-τ can always make the adversary easier to satisfy the adversary objective in Equation 5. Such results guide the adversary to set a relatively large τ in practice.

Results on Attacking CC with Bucketing. Since the bucketing strategy is mainly to address the heterogeneous data issue across clients, we only evaluate our attack against CC-B on the non-IID FEMNIST. As shown in Table 4, our attack can significantly reduce the MAs especially when ATK-τ is larger. This suggests our attack on non-IID datasets can also evade the aggregation of CC-B.

6 Related Work

Poisoning Attacks to FL. Poisoning attacks can be classified as targeted and untargeted attacks based on the adversary's goal. Targeted poisoning attacks to FL [2, 31, 38, 40, 46] aim to misclassify the targeted inputs as the attacker desires, while maintaining the performance on clients' clean inputs. For instance, backdoor attacks [2, 38, 40] are a subset of targeted attacks, where an adversary (e.g., malicious clients) injects a backdoor into the target inputs and tampers with their labels as the adversary desired label during training. During testing, the trained backdoored global model misclassifies any test input with the backdoor as the desired attack label, while correctly classifying the clean testing inputs. In contrast, untargeted poisoning attacks [3, 4, 13, 32] aim to minimize the accuracy of the global model on test inputs, which can be implemented by data poisoning [36] or model poisoning [4, 13, 32]. For instance, [32] proposes an untargeted poisoning attack framework to mount optimal model poisoning attacks. Different from the existing attacks, we propose an attack framework that unifies both targeted and untargeted attacks to FL.

Defenses against Poisoning Attacks to FL. Existing (empirical) defenses focus on designing robust aggregators (AGRs) and they can be roughly classified as two categories: the ones that limit the attack effectiveness via clipping malicious gradients [2, 17, 24, 26, 27, 44]; and the ones that weaken the contributions of malicious models by detecting and filtering them [1, 5, 14, 25, 30, 41]. Well-known AGRs include (Multi-) Krum [5], Bulyan [15], Trimmed-Mean (TM) [41, 44], Median [41, 44], Minimum Diameter Averaging (MDA) [29], Adaptive Federated Averaging (AFA) [25], and FLTrust [9]. However, all these defenses are broken by the optimization-based adaptive (untargeted poisoning) attack proposed by [32].

To mitigate this attack, SOTA poisoning defenses incorporating novel robust AGRs have been proposed, where the representatives are CC [17], MDAM [14], FLAME [26], and FLDetector [45][3]. For instance, CC clips malicious gradients by a tunable threshold τ (hyper-parameter), while FLAME sets the clipping threshold through computing the median value of the Euclidean distance between the global model and local models. In contrast, MDAM chooses a subset of $n - f$ clients (where f is the total number of malicious clients) with the smallest diameter to filter out the malicious gradients. Similar to MDAM, FLDetector utilizes historical information to predict gradient updates and identifies malicious clients by assessing discrepancies from the actual values.

In the paper, we design an optimization-based attack framework to break all these SOTA poisoning defenses.

7 Conclusion

We study poisoning attacks to federated learning and aim to break state-of-the-art poisoning defenses that use *robust* robust aggregators. Particularly, we propose an optimization-based attack framework, under which we design customized attacks by uncovering the vulnerabilities of these robust aggregators. Our attacks are extensively evaluated on various threat models and datasets. The experimental results validate our attacks can break all these robust aggregators and deliver significantly stronger attack performance that the SOTA attacks. Potential future works include designing effective (provable) defenses and generalizing the proposed attack framework on federated learning for, e.g., graph data [37].

Acknowledgments

We thank all anonymous reviewers for the constructive comments. Li is partially supported by the National Natural Science Foundation of China under Grant No. 62072208, Key Research and Development Projects of Jilin Province under Grant No. 20240302090GX. Hong is partially supported by the National Science Foundation under grant Nos. 2302689, 2308730, 2319277 and 2326341. Wang is partially supported by the National Science Foundation under grant Nos. 2216926, 2241713, 2331302 and 2339686.

[3] FedRecover[10] bases on FLDetector to first detect malicious clients and then recover the global model.

References

[1] Sebastien Andreina, Giorgia Azzurra Marson, Helen Möllering, and Ghassan Karame. 2021. Baffle: Backdoor detection via feedback-based federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 852–863.

[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2938–2948.

[3] Gilad Baruch, Moran Baruch, and Yoav Goldberg. 2019. A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems* 32 (2019).

[4] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*. PMLR, 634–643.

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

[6] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. 2019. Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems* (2019).

[7] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2018. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097* (2018).

[8] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining: 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II 17*. Springer, 160–172.

[9] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. 2021. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In *NDSS*.

[10] Xiaoyu Cao, Jinyuan Jia, Zaixi Zhang, and Neil Zhenqiang Gong. 2023. Fedrecover: Recovering from poisoning attacks in federated learning using historical information. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1366–1383.

[11] Lingjiao Chen, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos. 2018. Draco: Byzantine-resilient distributed training via redundant gradients. In *International Conference on Machine Learning*.

[12] Yudong Chen, Lili Su, and Jiaming Xu. 2017. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 1, 2 (2017), 1–25.

[13] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2020. Local model poisoning attacks to byzantine-robust federated learning. In *Proceedings of the 29th USENIX Conference on Security Symposium*. 1623–1640.

[14] Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafael Pinot, and John Stephan. 2022. Byzantine machine learning made easy by resilient averaging of momentums. In *International Conference on Machine Learning*. PMLR, 6246–6283.

[15] Rachid Guerraoui, Sébastien Rouault, et al. 2018. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*. PMLR, 3521–3530.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[17] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. 2021. Learning from history for byzantine robust optimization. In *International Conference on Machine Learning*. PMLR, 5311–5319.

[18] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. 2022. Byzantine-robust learning on heterogeneous datasets via bucketing. In *International Conference on Learning Representations*.

[19] Ang Li, Jingwei Sun, Binghui Wang, Lin Duan, Sicheng Li, Yiran Chen, and Hai Li. 2020. Lotteryfl: Personalized and communication-efficient federated learning with lottery ticket hypothesis on non-iid datasets. *arXiv* (2020).

[20] Denghao Li, Jianzong Wang, Lingwei Kong, Shijing Si, Zhangcheng Huang, Chenyu Huang, and Jing Xiao. 2022. A Nearest Neighbor Under-sampling Strategy for Vertical Federated Learning in Financial Domain. In *Proceedings of the 2022 ACM Workshop on Information Hiding and Multimedia Security*. 123–128.

[21] Liping Li, Wei Xu, Tianyi Chen, Georgios B Giannakis, and Qing Ling. 2019. RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[22] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60.

[23] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*.

[24] Mark Huasong Meng, Sin G Teo, Guangdong Bai, Kailong Wang, and Jin Song Dong. 2023. Enhancing Federated Learning Robustness Using Data-Agnostic Model Pruning. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 441–453.

[25] Luis Muñoz-González, Kenneth T Co, and Emil C Lupu. 2019. Byzantine-robust federated machine learning through adaptive model averaging. *arXiv preprint arXiv:1909.05125* (2019).

[26] Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, et al. 2022. FLAME: Taming Backdoors in Federated Learning. In *31st USENIX Security Symposium*.

[27] Thien Duc Nguyen, Phillip Rieger, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Ahmad-Reza Sadeghi, Thomas Schneider, et al. 2021. Flguard: Secure and private federated learning. *arXiv preprint arXiv:2101.02281* (2021).

[28] Mustafa Safa Ozdayi, Murat Kantarcioglu, and Yulia R Gel. 2021. Defending against backdoors in federated learning with robust learning rate. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 9268–9276.

[29] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. 2022. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing* 70 (2022), 1142–1154.

[30] Phillip Rieger, Thien Duc Nguyen, Markus Miettinen, and Ahmad-Reza Sadeghi. [n. d.]. DeepSight: Mitigating Backdoor Attacks in Federated Learning Through Deep Model Inspection. In *29th Annual Network and Distributed System Security Symposium, NDSS 2022*.

[31] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. 2020. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 11957–11965.

[32] Virat Shejwalkar and Amir Houmansadr. 2021. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*.

[33] Micah J Sheller, G Anthony Reina, Brandon Edwards, Jason Martin, and Spyridon Bakas. 2019. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part I 4*. Springer, 92–104.

[34] Md Fahimuzzman Sohan and Anas Basalamah. 2023. A Systematic Review on Federated Learning in Medical Image Analysis. *IEEE Access* (2023).

[35] Robert Tibshirani, Guenther Walther, and Trevor Hastie. 2001. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63, 2 (2001), 411–423.

[36] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. 2020. Data poisoning attacks against federated learning systems. In *ESORICS 2020*. 480–501.

[37] Binghui Wang, Ang Li, Meng Pang, Hai Li, and Yiran Chen. 2022. Graphfl: A federated learning framework for semi-supervised node classification on graphs. In *IEEE International Conference on Data Mining*.

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

[39] Zhaoxian Wu, Qing Ling, Tianyi Chen, and Georgios B Giannakis. 2020. Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks. *IEEE Transactions on Signal Processing* (2020).

[40] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. 2020. Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*.

[41] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. 2018. Generalized byzantine-tolerant sgd. *arXiv preprint arXiv:1802.10116* (2018).

[42] Cong Xie, Sanmi Koyejo, and Indranil Gupta. 2019. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *International Conference on Machine Learning*. PMLR, 6893–6901.

[43] Yuxin Yang, Qiang Li, Jinyuan Jia, Yuan Hong, and Binghui Wang. 2024. Distributed Backdoor Attacks on Federated Graph Learning and Certified Defenses. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*.

[44] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*. PMLR, 5650–5659.

[45] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2022. FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2545–2555.

[46] Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael Mahoney, Prateek Mittal, Ramchandran Kannan, and Joseph Gonzalez. 2022. Neurotoxin: Durable backdoors in federated learning. In *International Conference on Machine Learning*. PMLR, 26429–26446.