Beyond Denial-of-Service: The Puppeteer's Attack for Fine-Grained Control in Ranking-Based Federated Learning

Zhihao Chen
Fujian Normal University
Fuzhou, Fujian, China
zhihao_edu@163.com

Zirui Gong
Griffith University
Gold Coast, Queensland, Australia
zirui.gong@griffithuni.edu.au

Jianting Ning*
Fujian Normal University
Fuzhou, Fujian, China
jtning88@gmail.com

Yanjun Zhang
Griffith University
Brisbane, Queensland, Australia
yanjun.zhang@griffith.edu.au

Leo Yu Zhang
Griffith University
Gold Coast, Queensland, Australia
leo.zhang@griffith.edu.au

Abstract

Federated Rank Learning (FRL) is a promising Federated Learning (FL) paradigm designed to be resilient against model poisoning attacks due to its discrete, ranking-based update mechanism. Unlike traditional FL methods that rely on model updates, FRL leverages discrete rankings as a communication parameter between clients and the server. This approach significantly reduces communication costs and limits an adversary's ability to scale or optimize malicious updates in the continuous space, thereby enhancing its robustness. This makes FRL particularly appealing for applications where system security and data privacy are crucial, such as web-based auction and bidding platforms. While FRL substantially reduces the attack surface, we demonstrate that it remains vulnerable to a new class of local model poisoning attack, i.e., fine-grained control attacks.

We introduce the Edge Control Attack (ECA), the first fine-grained control attack tailored to ranking-based FL frameworks. Unlike conventional denial-of-service (DoS) attacks that cause conspicuous disruptions, ECA enables an adversary to precisely degrade a competitor's accuracy to any target level while maintaining a normal-looking convergence trajectory, thereby avoiding detection. ECA operates in two stages: (i) identifying and manipulating Ascending and Descending Edges to align the global model with the target model, and (ii) widening the selection boundary gap to stabilize the global model at the target accuracy. Extensive experiments across seven benchmark datasets and nine Byzantine-robust aggregation rules (AGRs) show that ECA achieves fine-grained accuracy control with an average error of only 0.224%, outperforming the baseline by up to 17×. Our findings highlight the need for stronger defenses against advanced poisoning attacks.

CCS Concepts

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

*Corresponding author.

Leo's work is supported in part by the Australian Research Council under grant LP240100315.

This work is licensed under a Creative Commons Attribution 4.0 International License.

WWW '26, Dubai, United Arab Emirates.

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2307-0/2026/04
https://doi.org/10.1145/3774904.3792604

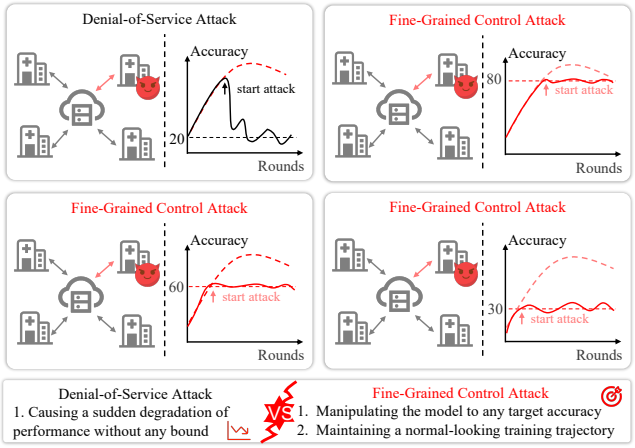


Figure 1: Illustration of the strategic advantages of a fine-grained control attack over a Denial-of-Service (DoS) attack.

Keywords

Machine Learning; Deep Learning; Trustworthy Machine Learning; Poisoning Attack; Federated Learning

ACM Reference Format:
Zhihao Chen, Zirui Gong, Jianting Ning, Yanjun Zhang, and Leo Yu Zhang. 2026. Beyond Denial-of-Service: The Puppeteer's Attack for Fine-Grained Control in Ranking-Based Federated Learning. In *Proceedings of the ACM Web Conference 2026 (WWW '26), April 13–17, 2026, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3774904.3792604

Resource Availability:
We release our code at https://doi.org/10.5281/zenodo.18297600.

1 Introduction

Web-based auction and bidding platforms [8, 11, 40] are systems where multiple advertisers compete in real time to display ads to users, with the highest bidders winning ad placement opportunities. These platforms rely heavily on machine learning models to predict click-through rates, optimize bid strategies, and detect fraudulent behavior. The models are trained on vast amounts of user interaction data, such as browsing history, past ad interactions, and contextual signals, to maximize platform revenue and user engagement. However, centralizing such sensitive data raises privacy and

security concerns, as it could expose personal user information or proprietary bidding strategies. Federated learning (FL) [24] offers a promising solution to this challenge. FL is a distributed machine learning paradigm where multiple clients collaborate to train a shared model without exposing their local data. Each client trains a local model on its dataset and sends updates to a central server, which aggregates them to form a global model, then redistributes it for further training.

However, the distributed nature of FL makes it particularly vulnerable to local model poisoning attacks [1–4, 13, 18, 21, 23, 32, 42]. One type is the Denial-of-Service (DoS) attack, where the goal is to maximally degrade the global model's accuracy. Although effective, DoS attacks have two major limitations. First, they only aim to maximize the global model's accuracy drop, so the eventual level of degradation is unpredictable. Second, they typically produce a sudden and erratic accuracy drop, making the attack conspicuous (see Fig. 1). Recently, a new attack paradigm, i.e., fine-grained control attack has been proposed [28, 42]. It aims to (i) manipulate the global model's accuracy to **any** target value and (ii) maintain a plausible training trajectory that mimics normal convergence. As illustrated in Fig. 1, this allows the attacker to steer the model to various target accuracies (e.g., 80%, 60%, or 30%) while preserving a natural-looking training curve. This affords the attacker more precise, covert, and practical control over global model performance.

To defend against poisoning attacks in FL, various methods have been proposed. One significant category is Byzantine-robust aggregation rules (AGRs) [4, 5, 12, 14, 17, 27, 34, 37, 41, 44], which aim to detect and exclude malicious updates. However, these can be bypassed by adaptive attacks, in which adversaries carefully craft updates to look similar to normal ones while causing significant damage. Recent advancements suggest redesigning the FL framework for inherent robustness, with ranking-based FL [19, 20, 25, 31, 45] being a state-of-the-art (SOTA) approach. This method replaces gradient exchange with discrete rankings and differs from standard FL in two key ways. First, it uses discrete rankings as the communication parameters between clients and the server, where each ranking indicates the relative importance of edges in the network. Second, on the server side, it employs majority voting to determine the global ranking and then applies the Lottery Ticket Hypothesis (LTH) [16] to select only the top k% most important edges to form a subnetwork, which can achieve the same performance as the full supernetwork. By avoiding gradient exchange and reducing the attack surface, ranking-based FL significantly improves robustness and makes existing gradient-based attacks ineffective.

Our work: In this work, we demonstrate that even when the victim system employs the ranking-based FL, i.e., FRL, to train its models, it is still feasible to launch effective fine-grained control attacks by introducing a small amount of malicious clients into the system. Specifically, the attacker aims to (i) manipulate the model's accuracy to any desired target value, and (ii) ensure the global model maintains a normal convergence trend throughout the training process. This dual capability of achieving precise control while maintaining stealth makes the attack highly practical in real-world scenarios. Unlike overt DoS attacks that trigger obvious alarms, our fine-grained approach allows the adversary to subtly dictate the model's performance, making it a more realistic threat.

To achieve the goal, we introduce the Edge Control Attack (ECA), the first fine-grained control attack that targets the SOTA robust FL framework, i.e., FRL. To execute the attack, we propose a two-stage algorithm. In the first stage, we identify the set of Ascending and Descending Edges that represent the edge difference between the current global model and the target model. These edges are then manipulated to positions that maximize the attack's impact. In the second stage, we aim to fix the global model's accuracy at the target value while simulating a normal convergence trend. To achieve this, we deliberately increase the gap required for the global model to change, preventing benign updates from reversing our manipulation. This stabilizes the model at the target state, ensuring that its accuracy cannot increase or decrease further.

We evaluate our attack on seven benchmark datasets under nine robust aggregation rules (AGRs), demonstrating our attack can achieve a high level of control across all combinations of datasets, target accuracy, and defenses, with an overall 0.224% control error, and outperforms another attack by 17×. Our key contributions are highlighted as follows:

- We identify a new strategic vulnerability in ranking-based FL paradigms that extends beyond simple disruption. We demonstrate that an adversary can exploit the discrete update mechanism not just to damage a model, but to precisely steer its accuracy to any specific, attacker-chosen level, creating a potent threat for competitive scenarios like commercial bidding.
- We design the Edge Control Attack (ECA), the first attack method specifically tailored to achieve fine-grained control in FRL. We propose a novel algorithm that is more practical in real-world scenarios, enabling attackers to achieve dual objectives: (i) manipulating the model to any desired accuracy, and (ii) maintaining a normal-looking training trajectory to evade detection. We provide a theoretical analysis that formally establishes the conditions under which this precise control is feasible and highly probable.
- We conduct extensive experiments demonstrating that ECA poses a significant and practical threat. Our results show that ECA can maintain control with an exceptionally low error (0.224%) across diverse settings, significantly outperforming relevant baselines.

2 Background and Related Work

2.1 Federated Rank Learning (FRL) and Poisoning Attack in FRL

FRL [25] is a SOTA FL paradigm that operates in a discrete space by leveraging the Lottery Ticket Hypothesis [16] and utilizing the edge-popup (EP) algorithm [29] to identify sparse subnetworks capable of achieving comparable performance to dense models. Specifically, the FRL has the following processes.

Server Initialization: The server uses a random seed to initialize the weights θ_w and scores θ_s for each edge within the *global supernetwork*. During training, only the scores θ_s are updated. The server then sorts $theta_s$ in ascending order to create the global ranking R_g, where each value represents the ID of the edge and a higher rank indicates greater importance. Finally, the server sends the global ranking R_g to the randomly selected U clients.

Clients Training: Each client reconstructs the score θ_s using the received global ranking R_g and the local data D_u. The scores θ_s for each edge are then updated and ranked using the EP algorithm

and local data to obtain the updated ranking. If an edge e in the *subnetwork* reduces the training loss (e.g., cross-entropy), then the score of this edge will increase. Finally, clients send the updated rankings to the server for aggregation.

Server Aggregation: Upon receiving the rankings R_u from U clients, the server performs majority voting (MV) to get the aggregated global ranking. Specifically, it has the following steps: i) the server first aligns the *importance scores* I_u of all edges across clients based on their edge IDs; ii) then it performs a dimension-wise summation across clients, and calculates the *aggregated importance scores* for each edge, i.e., $S = \sum_{u=1}^{U} I_u$; iii) finally, the edges are rearranged based on the *aggregated importance scores* to produce the new global ranking R_g^{t+1}. For example, suppose two clients upload rankings $R_1 = [0, 4, 2, 3, 5, 1]$, $R_2 = [0, 1, 2, 5, 4, 3]$, where e_0 is the least important edge (at the first index). Then the server computes the *importance score* vector as $I_1 = [0, 5, 2, 3, 1, 4]$, $I_2 = [0, 1, 2, 5, 4, 3]$. After aggregating scores, the server calculates the *aggregated importance scores*, i.e., $S = [0, 6, 4, 8, 5, 7]$. Based on the *aggregated importance scores* of each edge, the server gets the aggregated global ranking $R_g^{t+1} = [0, 2, 4, 1, 5, 3]$.

Finally, the global ranking R_g^{t+1} is transformed into a binary supermask $\mathbf{M}$, where the top k% edges are set to 1, and the remaining edges are set to 0, then reordered to align with the original edge IDs. For example, if the global ranking is $R_g^{t+1} = [0, 2, 4, 1, 5, 3]$ and k=50%, edges e_1, e_5, e_3 will be selected for *subnetwork*, therefore, the corresponding mask is $\mathbf{M}[R_g^{t+1}] = [0, 1, 0, 1, 0, 1]$. In summary, the goal of FRL is to find a global ranking R_g, such that the resulting *subnetwork* $\theta_w \odot \mathbf{M}$ minimizes the average loss of all clients. Formally, we have

$$\min_{R_g} F(\theta^w, R_g) = \min_{R_g} \sum_{u=1}^{U} \lambda_u L_u(\theta^w \odot \mathbf{M}), \quad (1)$$
$$\text{s.t. } \mathbf{M}[R_g[e]] = \begin{cases} 0 & \text{if } e < t, \\ 1 & \text{if } e \geq t, \end{cases}$$

where U denotes the number of clients selected in training, θ^w denotes the global model weights. L_u and λ_u denote the loss function and weight at aggregation for the u^{th} client, respectively, with $\lambda_u = 1/U$. The index of the selection boundary for the *subnetwork* is $t = \lfloor (1-k)\, n \rfloor$, where n is the number of edges in each layer and typically we use the *subnetwork* size $k = 50\%$.

While FRL's discrete nature and majority voting mechanism were designed to enhance robustness, recent work has demonstrated that it is not immune to poisoning. Specifically, the Vulnerable Edge Manipulation (VEM) attack [18] revealed that FRL is susceptible to DoS attacks. The core finding of VEM is that not all edges are equally robust; those near the subnetwork selection boundary are particularly "vulnerable" to manipulation. By identifying this set of vulnerable edges, VEM employs an optimization-based method (Gumbel-Sinkhorn) to craft malicious rankings that maximize performance degradation, successfully undermining the global model.

However, the threat model addressed by VEM is confined to DoS objectives, revealing several critical limitations. First, its focus on maximizing disruption conflicts with the need for precise control. VEM's optimization approach is designed for a one-way, unbounded degradation, making it incapable of reliably steering a model to a specific, predetermined accuracy level. Second, this pursuit of maximum damage inherently sacrifices stealth. A significant drop in model accuracy is a telltale sign of attack activity, making the compromised system highly detectable. For attackers seeking to maintain a low profile, such attacks are both difficult to execute and impractical.

2.2 Fine-Grained Control Attacks in FL

Beyond simple DoS, a more advanced threat is the fine-grained control attack, where an adversary aims to manipulate a model's performance to a specific level of accuracy. This type of attack is stealthier and can be more damaging in competitive commercial scenarios than a DoS attack. However, existing fine-grained control attacks are designed for traditional FL and do not apply to FRL.

For instance, two prominent methods, the Flexible Model Poisoning Attack (FMPA) [42] and the Federated Learning Sliding Attack (FedSA) [28], both operate in a continuous parameter space. FMPA's core strategy is model replacement: it first predicts the likely state of the benign global model for the next round, and then crafts a malicious update intended to overwrite it with a poisoned version that exhibits a specific low accuracy. FedSA, on the other hand, approaches the problem from a control theory perspective. It models the entire FL training process as a dynamic system and defines an "error" as the difference between the current model's accuracy and the desired target accuracy. The attacker then calculates a malicious update using a specifically designed "control law". This isn't a fixed poisoned model, but rather a dynamic adjustment calculated each round, acting to continuously correct the global model's trajectory, actively steering it towards the compromised state and theoretically guaranteeing its convergence to the attacker's precise target.

Crucially, since both methods achieve control by manipulating continuous weights or gradients, they are fundamentally incompatible with FRL, which functions within a discrete ranking space where such perturbations are meaningless. This reveals a critical research gap: no existing methods are capable of performing fine-grained accuracy control on ranking-based FL systems.

3 Edge Control Attack (ECA)

3.1 Threat Model

Attacker's objective. In this work, we move beyond the traditional DoS threat model [2, 13, 32]. Instead of aiming for maximum disruption, the adversary pursues a more sophisticated, goal-oriented objective inspired by competitive scenarios like commercial bidding. The attacker's primary goal is to control the competitor's global model precisely, steering its final accuracy to a specific, **any** desired value (τ) that is strategically advantageous (e.g., marginally lower than their own model's accuracy). This primary goal is coupled with a critical secondary objective: stealth. The attack must be executed in a way that the model's training trajectory appears normal to an outside observer. This means avoiding the conspicuous, abrupt performance drops characteristic of DoS attacks. The attacker's dual objectives are therefore: (i) to ensure the global model converges to the precise target accuracy (τ), and (ii) to maintain a convergence path that mimics a natural, albeit slower or less effective, training process to evade detection.

Attacker's capability. We assume that the attacker can control m malicious clients out of U total clients per round. In the context of

our threat model, these could be sleeper agents introduced into a competitor's team during an FL competition. Following the common assumption in previous studies [13, 32], we set the proportion of malicious clients to 20% in our main experiments.

Attacker's knowledge. We consider update-agnostic settings, where the attacker has no knowledge of benign client updates but uses the clean dataset of malicious clients and the information received from the global model to generate malicious updates.

3.2 Motivation

In FRL, model accuracy is determined by the mask $\mathbf{M}$, which controls the selection of edges that form the subnetwork. Therefore, to achieve the goal of steering the global model accuracy to any desired level while maintaining the normal training convergence trend, we focus on designing the malicious mask. Specifically, we aim to minimize the mask difference between the poisoned global ranking $\hat{R}_g$ and a target ranking R_τ. This leads to the following objective function:

$$\begin{aligned} &\min \ \left|\mathbf{M}[R_\tau] - \mathbf{M}[\hat{R}_g]\right|, \\ &\text{s.t. } \hat{R}_g = \mathrm{MV}(\hat{R}_1, \ldots, \hat{R}_m, R_{m+1}, \ldots, R_U), \end{aligned} \tag{2}$$

where R_τ is the target global ranking with a test accuracy of τ, $\hat{R}_g$ is the poisoned global ranking, $\mathbf{M}[R_\tau]$ and $\mathbf{M}[\hat{R}_g]$ are the masks of the two rankings, MV() is the majority voting process, $\hat{R}_u$ (for $u \in m$) are the malicious rankings, and R_u (for $u \in [m+1, U]$) are the benign rankings. However, achieving this objective is challenging and cannot be solved by existing methods. Existing fine-grained control attack [42] usually adds an optimized perturbation on the malicious gradient, aiming to cancel out further changes in the global gradient. But this mechanism relies on adding a small noise to gradient updates, which will introduce floating-point values into the ranking updates, violating the format constraint of FRL. Alternatively, one might consider applying a similar optimized perturbation to the client's scores θ_s to mimic gradient-level manipulation in traditional FL. However, in FRL, clients cannot obtain the global score, which is also a key factor contributing to the robustness of FRL. This inherent property makes such optimization infeasible. Although VEM proposes to use the Gumbel-Sinkhorn method to solve optimization problems in discrete spaces [18], the stochastic and uncertain nature of permutation-based optimization prevents it from precisely steering the model toward any target accuracy.

3.3 Overall of ECA

To solve the problem, we propose a deterministic two-stage algorithmic based on permutation-based methods [9, 30, 43] to directly find a feasible malicious ranking $\hat{R}$ that achieves our goal. Our method, as shown in Fig. 2, consists of two main stages: Manipulate Ascending & Descending Edges and Freeze the Mask.

In the first stage, the adversary aims to align $\mathbf{M}[\hat{R}_g]$ with $\mathbf{M}[R_\tau]$ by manipulating specific edges. To achieve this, it first identifies the mask difference between the target ranking (R_τ) and the aggregated ranking of other $U-m$ benign clients ($\overline{R}$). These differing edges are classified into ascending edges (AE) and descending edges (DE). The adversary then manipulates AE & DE to generate intermediate malicious rankings ($\hat{R}'$) that offset these discrepancies (details see Section 3.4). In the second stage, the adversary aims to freeze the mask of $\hat{R}_g$ to prevent further updates. To achieve this, it reverses the ranking on each side of the subnetwork boundary based on $\hat{R}'$, we refer to it as internal reverse in the following section. This process will reduce the importance scores of edges on the proximal left side of the boundary (outside the subnetwork) while increasing the importance scores of edges on the proximal right side (inside the subnetwork). Then during the majority voting stage, the generated malicious ranking will widen the boundary gap required for a mask change in $\hat{R}_g$, thereby ensuring continued alignment with the mask of target ranking $\mathbf{M}[R_\tau]$ (details see Section 3.5).

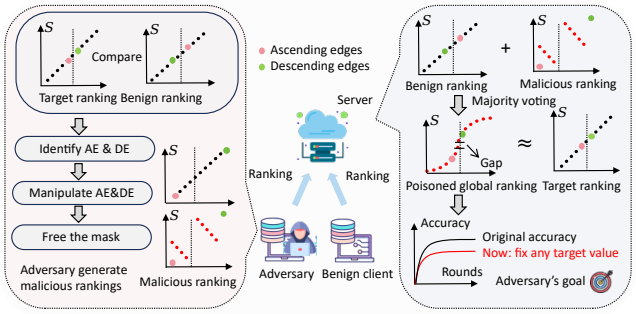


Figure 2: Overview of the ECA attack. The left shows the adversary generates malicious updates by aligning the global mask with the target and freezing it via a widened boundary gap. The right shows the attack's effect on the global model, enabling precise and stable control of its accuracy.

3.4 Manipulate Ascending & Descending Edges

Since $\hat{R}_g \approx \mathrm{MV}(\hat{R}, \overline{R})$, where $\hat{R}$ are malicious rankings and $\overline{R}$ is aggregated ranking of other $U-m$ benign clients, to achieve Eq. 2, we first need to identify the mask different between R_τ and $\overline{R}$, and then use malicious rankings to offset this difference.

Assuming we have the aggregated rankings of other $U-m$ benign clients ($\overline{R}$), to identify the mask difference between R_τ and $\overline{R}$, we first need to initialize the target ranking R_τ. Specifically, we assign $R_\tau = R_g^{t_0}$ (where t_0 is the first round when $ACC(R_g^{t_0}) \geq \tau$). And we use ε to denote the initialization error between $ACC(R_g^{t_0})$ and τ, i.e., $\varepsilon = |ACC(R_g^{t_0}) - \tau|$. During subsequent rounds, R_τ and ε are dynamically updated to minimize ε. Specifically, if $|ACC(R_g^t) - \tau| < \varepsilon$, we assign R_g^t to R_τ, and update $\varepsilon = |ACC(R_g^t) - \tau|$.

After determining R_τ, we start to identify different edges where $\mathbf{M}[R_\tau[e]] \neq \mathbf{M}[\overline{R}[e]]$. Upon observation, we find two types of edges: Ascending Edge (AE) and Descending Edge (DE). AE are those edges that are not selected by R_τ, but selected by $\overline{R}$, defined as $\mathbb{I}_{\mathrm{AE}} = \{e \mid \mathbf{M}[R_\tau[e]] = 0 \text{ and } \mathbf{M}[\overline{R}[e]] = 1\}$. Conversely, DE are those edges that are selected by R_τ, but not selected by $\overline{R}$, defined as $\mathbb{I}_{\mathrm{AE}} = \{e \mid \mathbf{M}[R_\tau[e]] = 1 \text{ and } \mathbf{M}[\overline{R}[e]] = 0\}$. Then to achieve Eq. 2 and align $\mathbf{M}[\hat{R}_g]$ with $\mathbf{M}[R_\tau]$, we reduce the importance of AE and increase the importance of DE in our malicious rankings. As illustrated in Fig. 2, we move AE to the beginning of the $\overline{R}$ and move DE to the end of $\overline{R}$ to get intermediate malicious ranking $\hat{R}'$.

Now we illustrate two methods to estimate the aggregated rankings of other $U-m$ benign clients ($\overline{R}$). **Historical estimation**,

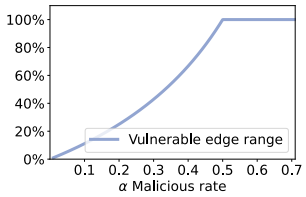


Figure 3: Vulnerable edge percentage.

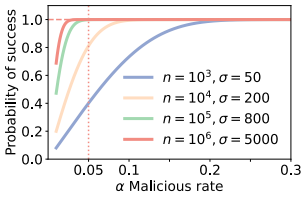


Figure 4: Success probability of edge manipulation.

which assumes that the global model changes slowly over training rounds, allowing the ranking from the previous round (R_g^{t-1}) to approximate the current $\overline{R}^t$, i.e., $\overline{R}^t \approx R_g^{t-1}$. **Alternative estimation** uses malicious clients' local rankings to approximate the $\overline{R}$ for the current round, i.e., $\overline{R} \approx \text{MV}(R_1, \ldots, R_m)$.

3.5 Freeze the Mask

After aligning the mask of poisoned global ranking $\hat{R}_g$ with target raking R_τ, we aim to freeze the mask of $\hat{R}_g$ to prevent further updates. However, In FRL, the mask near the subnetwork boundary is particularly sensitive to changes because the aggregated importance scores S of edges on either side of the boundary are shown to be close. This sensitivity is a natural feature of FRL, as it enables dynamic selection of subnetwork edges to optimize model accuracy.

To overcome this challenge and freeze the mask, we maliciously amplify the cost (i.e., the boundary gap) of edges that need to cross the subnetwork boundary. We reverse the rankings on both sides of the boundary based on $\hat{R}'$ (except AE & DE) to obtain the final malicious ranking $\hat{R}$. This approach reduces the importance score of edges on the proximal left side of the boundary (outside the subnetwork) while increases those on the proximal right side of the boundary (inside the subnetwork) as much as possible while ensuring that the edge selection results remain unchanged. Then after server-side majority voting, the global ranking forms a sigmoid-like distribution, where the boundary gap increases.

Algorithm. 1 describes our ECA process. The attack is triggered once the global accuracy $ACC(R_g^t)$ surpasses the target threshold τ. The attacker initializes the target ranking R_τ from the current global ranking and iteratively updates it to approximate the target accuracy τ (lines 4-9). The main attack then commences (lines 10-18). In each round, the attacker first estimates the aggregated ranking of benign clients, $\overline{R}^t$ (line 12). By comparing $\overline{R}^t$ with the target R_τ, it identifies the precise sets of AE/DE (line 13). To align the mask with the target, an intermediate malicious ranking $\hat{R}^{t'}$ is constructed by manipulating the importance of AE&DE (lines 14-15). Finally, the attacker applies the internal reverse to $\hat{R}^{t'}$ to generate the final malicious ranking $\hat{R}^t$ (line 17), which amplifies the boundary gap of the poisoned global ranking.

Algorithm 1 Edge Control Attack (ECA)

Input: Number of rounds T, number of malicious clients m, malicious clients dataset D_m, model weights θ^w, model scores θ^s, learning rate η, target accuracy τ, initialization error ε.
Output: Malicious rankings $\hat{R}$.

1: // Initialize: `Start` ← False, ε
2: **for** $t \in [0, T]$ **do**
3: $R_m^t \leftarrow$ LOCAL_TRAINING$(D_m, \theta^w, \theta^s, \eta)$
4: **if** $ACC(R_g^t) \geq \tau$ and not `Start` **then**
5: $R_\tau \leftarrow R_g^t$; $\varepsilon \leftarrow |ACC(R_g^t) - \tau|$; `Start` ← True
6: **end if**
7: **if** $|ACC(\hat{R}_g^t) - \tau| \leq \varepsilon$ and `Start` **then**
8: $R_\tau \leftarrow \hat{R}_g^t$; $\varepsilon \leftarrow |ACC(\hat{R}_g^t) - \tau|$
9: **end if**
10: **if** `Start` **then**
11: // Manipulate AE&DE
12: $\overline{R}^t \leftarrow$ ESTIMATION
13: $\mathbb{I}_{\text{AE}}, \mathbb{I}_{\text{DE}} \leftarrow$ IDENTIFY_EDGES$(R_\tau, \overline{R}^t)$
14: $\mathbb{I}'_{\text{AE}}, \mathbb{I}'_{DE} \leftarrow$ MANIPULATE_AE&DE$(\overline{R}, \mathbb{I}_{\text{AE}}, \mathbb{I}_{\text{DE}})$
15: $\hat{R}^{t'} \leftarrow$ GET_MALICIOUS$(\overline{R}, \mathbb{I}'_{\text{AE}}, \mathbb{I}'_{\text{DE}})$
16: // Freeze the Mask
17: $\hat{R}^t \leftarrow$ INTERNAL_REVERSE$(\mathbb{I}'_{\text{AE}}, \mathbb{I}'_{\text{DE}}, \hat{R}^{t'})$
18: **end if**
19: **end for**

4 Theoretical Analyses

In this section, we theoretically analyze the range of vulnerable edges and the probability that AE and DE fall into this range, which reflects the success rate of our attack. Intuitively, we call an edge e vulnerable if the aggregated importance score S between the edge e and the subnetwork boundary edge e_k is less than the maximum damage that the attacker can possibly inflict in a round (either by increasing the importance score of DE or decreasing the importance score of AE). Based on the range of vulnerable edges, we can calculate the success probability of manipulating AE&DE. Formally, we have the following theorems.

Lemma 1. ***Vulnerable edge range.*** *Given the subnetwork percentage k, malicious rate α, and the number of edges in each layer n, let $\pi^{-1}(e)$ denote the position (i.e., rank index) of edge e in R, Then, the position of an edge e that can be manipulated by the ECA attack is bounded by*

$$\frac{kn - \alpha(n-1)}{(1-\alpha)} \leq \pi^{-1}(e) < \frac{kn}{(1-\alpha)}. \tag{3}$$

Next, we apply Lemma 1 to quantify the likelihood of AE and DE falling within the vulnerable edge range, which theoretically verifies the effectiveness of our attack.

Theorem 1. ***Probability of Manipulating AE&DE.*** *Let e be a random edge sampled from $\mathbb{I}_{AE} \cup \mathbb{I}_{DE}$, we assume the distribution of $\pi^{-1}(e)$ approximates a discrete Gaussian distribution centered at μ with standard deviation σ, i.e., $\pi^{-1}(e) \sim D_{\mathbb{Z},\mu,\sigma}$. The probability of e falling into the vulnerable range is*

$$P \approx \Phi\left(\frac{2kn + (1-\alpha)(1-2\mu)}{2(1-\alpha)\sigma}\right) - \Phi\left(\frac{2kn - 2\alpha(n-1) - (1-\alpha)(1+2\mu)}{2(1-\alpha)\sigma}\right), \tag{4}$$

where $\Phi()$ is the cumulative distribution function of the standard normal distribution $N(0,1)$.

We will discuss the implications of these two theorems and leave their proofs to the Appendix A.

Table 1: Comparison of RRA and our attack ECA under various AGRs in a control scenario. The "no attack" row reports the global model accuracy without attack $ACC(R_g)$ (%); other rows report control error ξ (%) under various AGRs and τ.

Dataset (Model)	Attack	Target τ (%)	FRL	Multi-Krum-FRL	AFA-FRL	Fang-ERR-FRL	Fang-LFR-FRL	Fang-Union-FRL	FABA-FRL	DnC-FRL	FLTrust-FRL	Average ξ
CIFAR10 (Conv8)	no attack	/	83.85	84.43	84.47	84.64	85.16	84.83	84.53	84.44	84.62	/
	RRA	70	0.47 (±1.24)	0.64 (±1.17)	1.56 (±1.16)	9.67 (±3.70)	7.87 (±4.87)	8.49 (±3.61)	2.26 (±2.71)	0.88 (±1.27)	3.76 (±0.04)	3.96
		75	0.93 (±0.85)	1.31 (±1.03)	1.50 (±1.61)	1.56 (±0.80)	1.68 (±1.77)	1.13 (±1.33)	1.59 (±1.34)	1.05 (±1.61)	1.09 (±0.98)	1.32
		80	0.34 (±0.69)	0.53 (±0.59)	0.74 (±0.53)	2.01 (±1.37)	2.31 (±1.45)	1.39 (±1.01)	2.44 (±1.80)	0.86 (±0.74)	1.25 (±0.81)	1.32
	ECA	70	**0.11 (±0.27)**	**0.14 (±0.28)**	**0.14 (±0.25)**	**0.11 (±0.21)**	**0.06 (±0.19)**	**0.30 (±0.21)**	**0.32 (±0.25)**	**0.34 (±0.53)**	**0.16 (±0.25)**	**0.19**
		75	**0.02 (±0.07)**	**0.04 (±0.16)**	**0.01 (±0.04)**	**0.11 (±0.09)**	**0.05 (±0.16)**	**0.02 (±0.09)**	**0.05 (±0.16)**	**0.26 (±0.39)**	**0.06 (±0.14)**	**0.07**
		80	**0.00 (±0.02)**	**0.02 (±0.09)**	**0.01 (±0.03)**	**0.01 (±0.04)**	**0.01 (±0.03)**	**0.01 (±0.05)**	**0.03 (±0.08)**	**0.08 (±0.16)**	**0.02 (±0.11)**	**0.02**
EMNIST (LeNet)	no attack	/	86.69	86.64	86.67	86.81	86.91	86.69	86.69	86.91	86.84	/
	RRA	70	9.13 (±1.75)	7.73 (±2.20)	8.15 (±1.51)	11.50 (±1.83)	11.43 (±2.00)	11.23 (±1.95)	4.56 (±3.25)	10.98 (±2.06)	8.55 (±1.87)	9.25
		75	4.23 (±1.56)	3.40 (±1.61)	4.54 (±1.44)	6.43 (±1.55)	6.76 (±1.60)	6.64 (±1.42)	0.83 (±3.17)	6.61 (±1.48)	3.59 (±1.62)	4.78
		80	0.62 (±0.93)	0.63 (±1.11)	0.68 (±0.98)	2.00 (±1.00)	1.96 (±1.16)	1.75 (±1.04)	0.58 (±2.16)	1.95 (±0.97)	0.16 (±1.13)	1.15
	ECA	70	**0.47 (±0.61)**	**0.59 (±0.64)**	**0.46 (±0.52)**	**0.37 (±0.50)**	**0.44 (±0.58)**	**0.24 (±0.31)**	**0.24 (±0.46)**	**0.98 (±0.83)**	**0.23 (±0.45)**	**0.45**
		75	**0.29 (±0.31)**	**0.26 (±0.30)**	**0.43 (±0.44)**	**0.21 (±0.27)**	**0.34 (±0.37)**	**0.13 (±0.30)**	**0.21 (±0.33)**	**0.72 (±0.62)**	**0.16 (±0.26)**	**0.31**
		80	**0.08 (±0.17)**	**0.07 (±0.17)**	**0.02 (±0.13)**	**0.05 (±0.17)**	**0.06 (±0.15)**	**0.05 (±0.16)**	**0.05 (±0.18)**	**0.15 (±0.21)**	**0.10 (±0.17)**	**0.07**
FashionMNIST (LeNet)	no attack	/	91.35	91.44	91.30	91.42	91.33	91.33	91.33	91.33	91.33	/
	RRA	75	9.90 (±1.33)	8.72 (±1.63)	9.60 (±1.34)	11.24 (±1.56)	11.12 (±1.55)	11.19 (±1.57)	6.78 (±2.52)	11.58 (±1.52)	9.32 (±1.36)	8.69
		80	4.91 (±1.14)	4.12 (±1.32)	4.85 (±1.25)	6.35 (±1.28)	6.28 (±1.30)	6.57 (±1.18)	2.19 (±2.34)	6.56 (±1.25)	4.28 (±1.38)	5.12
		85	0.49 (±0.84)	0.39 (±0.85)	0.56 (±0.86)	1.61 (±0.73)	1.60 (±0.73)	1.71 (±0.74)	0.45 (±2.16)	2.17 (±0.82)	0.25 (±0.95)	1.03
	ECA	75	**0.40 (±0.70)**	**0.68 (±1.16)**	**0.76 (±1.14)**	**0.84 (±1.04)**	**0.47 (±0.77)**	**0.41 (±0.68)**	**0.47 (±0.86)**	**0.87 (±1.98)**	**0.86 (±1.39)**	**0.64**
		80	**0.33 (±0.53)**	**0.10 (±0.42)**	**0.09 (±0.25)**	**0.25 (±0.46)**	**0.05 (±0.18)**	**0.13 (±0.30)**	**0.33 (±0.30)**	**0.57 (±0.86)**	**0.21 (±0.32)**	**0.23**
		85	**0.01 (±0.09)**	**0.02 (±0.11)**	**0.00 (±0.09)**	**0.02 (±0.09)**	**0.08 (±0.13)**	**0.05 (±0.15)**	**0.07 (±0.14)**	**0.09 (±0.25)**	**0.08 (±0.18)**	**0.05**

Implications. Lemma 1 shows that the attacker's ability to manipulate edges is limited to a specific range, with only those edges within this range being controllable. As shown in Fig. 3, increasing the malicious rate α expands the vulnerable edge range, when α reaches 50%, the attacker can control 100% of the edges. Thm. 1 quantifies the probability that the AE or DE falls within the vulnerable range, which reflects the success rate of our attack. We report the probability for the worst-case edge, i.e., the one furthest from the selection boundary, to demonstrate the lower bound of success. As shown in Fig. 4, where n is the number of edges, $k = 50\%$, $\mu = kn$ and σ is the standard deviation, we observe that the success probability increases as the malicious rate α rises. Even when $\alpha = 5\%$, the success probability exceeds 40%, highlighting the threat posed by a small malicious rate. Additionally, when α is around 10%-20% (depending on n and σ), the probability nearly reaches 100%.

5 Experimental Results and Analyses

5.1 Experimental Setup

Datasets and Models. We evaluate our attacks under seven benchmark datasets and three models. The main experiment involves CIFAR10 [22] with Conv8, EMNIST [7] with LeNet, and FashionMNIST [36] also with LeNet. We also describe additional datasets such as CIFAR100 [22], Location30 [38, 39], Purchase100 [10], and Texas100 [33] in the Appendix B.

FL Settings. We consider the cross-device FL setting, where 1000 clients collaborate to train a global model, with the server randomly selects 25 clients each round. We run our attack for 500 rounds after the global model reaches the target accuracy τ. For our main experiment, we adopt a more realistic scenario by distributing the data in a non-IID manner using the Dirichlet distribution with parameter $\beta = 1$. Unless otherwise stated, we assume a 20% malicious rate.

Poisoning Attacks for Comparison. Our goal is to precisely control the global model's accuracy to a given target value τ under the FRL framework. However existing fine-grained model control attacks [42] do not apply to FRL, so we propose a possible attack called Random Ranking Attack (RRA) for comparison. Specifically, in RRA, the attacker upload randomized rankings once the model accuracy exceeds the target threshold τ, causing slight performance degradation. Conversely, when the model accuracy drops below τ, the attacker ceases the attack, allowing the model to naturally recover. This strategy causes the global model's performance to oscillate around the target accuracy τ, achieving a certain degree of control without fully disrupting the training process.

Evaluated Defenses. We consider nine SOTA AGRs integrated with FRL. Not only including distance-based (Multi-Krum [4], AFA [26], FABA [35],DnC [32]), weighted-aggregation (FLTrust [5]) and validation-based AGRs (Fang [13]). Notably, we do not consider norm-bounded [34] and dimension-wise AGRs [41] because these approaches are incompatible with the FRL. More details about AGRs are deferred to Appendix B.

Evaluation Metrics. Let τ denote the target global accuracy for the attacker, and $ACC(\hat{R}_g)$ denote the global model accuracy after the attack. We use the control error $\xi = |ACC(\hat{R}_g) - \tau|$ to evaluate the performance of the control attack. A smaller ξ indicates a more successful attack.

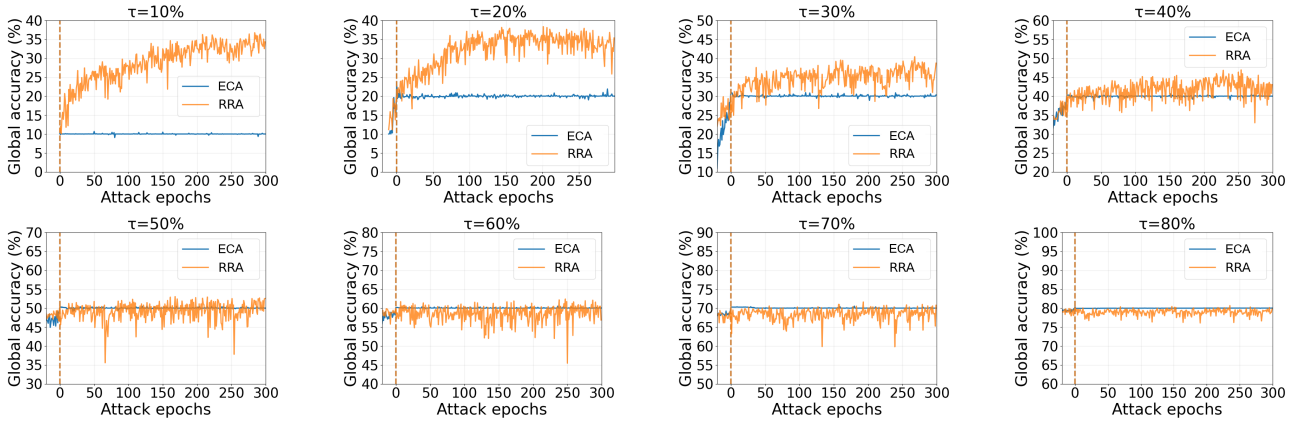


Figure 5: Global model accuracy (%) under ECA and RRA attacks with different τ on the CIFAR10 dataset. The orange dash indicates the epoch when we start the attack. Note that we have only visualized the accuracy changes after the attack began.

5.2 Evaluation Results

Our extensive experiments confirm the potent effectiveness and precision of ECA. As shown in Table. 1, ECA maintains tight control across all combinations of datasets, target accuracy, and defense methods, with an average control error of only 0.224% and is 17 × smaller than RRA. Specifically, under the FRL framework, the average control error is 0.19% (18.1 × lower than RRA), and in the most robust setting, DnC-FRL, it is 0.45% (10.5 × lower compared to RRA). More experimental results are provided in Appendix C, including: computational efficiency analysis; studies on additional datasets (CIFAR100, Location30, Purchase100, Texas100); evaluations against SOTA defenses (FLCert, FoundationFL); and performance under lower target accuracies (τ) and minimal malicious rates (1%).

We also visualize the control performance of ECA and RRA. Fig. 5 shows that RRA exhibits significant fluctuations, while ECA demonstrates a smoother and more precise control. Notably, RRA struggles to control the global accuracy at low τ values, such as τ = 10% and τ = 20%, where the global accuracy exhibits large fluctuations and deviates significantly from the target value. In contrast, our ECA consistently maintains the global accuracy close to the target with minimal variance. This stark difference highlights the superiority of ECA, particularly in complex settings where fine-grained control is critical, making it more reliable than RRA.

5.3 Ablation Studies

Impact of the malicious rate. Fig. 6 shows the ECA attack results when the malicious rate changes from 1% to 20% under the FRL framework. We notice that as the proportion of malicious clients increases, the control effect becomes better. Notably, even with only 1% of malicious clients, the attack achieves the desired control effect, with an average control error of 0.823%.

Impact of the Non-IID degree. Fig. 7 illustrates the impact of varying non-IID degrees on the control error. The hyperparameter β defines the degree of non-IID, where smaller β values indicate a higher non-IID degree. As β decreases, the data distribution becomes increasingly heterogeneous, making it more challenging to predict benign client updates. Despite this, our approach consistently achieves the desired attack performance. Specifically, our

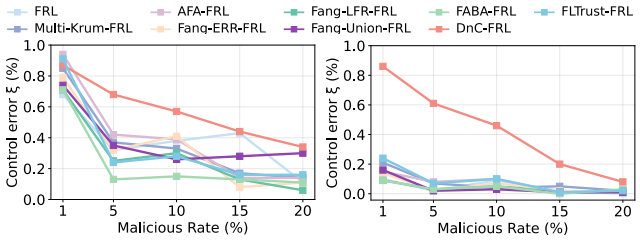


Figure 6: Control errors ξ (%) for different defenses at τ = 70% (left) and τ = 80% (right) with different malicious rates on the CIFAR10 dataset.

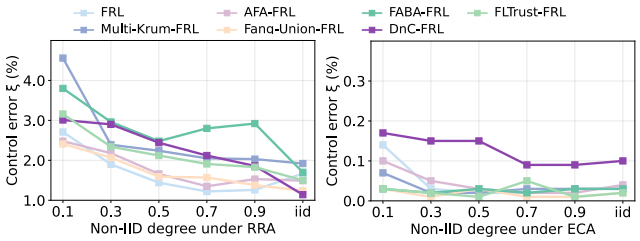


Figure 7: Control error ξ (%) for different defenses on different non-iid degree β (x-axis) when τ = 70% on the CIFAR10 dataset.

ECA attack achieves a control error of less than 0.2%, the average control error is 0.045%, significantly outperforming RRA by 47.3×.

Impact of number of clients. Fig. 8 illustrates the impact of the number of total clients on the control error for ECA and RRA attacks under FRL. The results show that the number of clients has almost no effect on ECA.

Impact of the estimation methods. We compare control errors from two estimation methods: Historical and Alternative estimation. As shown in Fig. 9, both methods achieve the desired performance, with control errors below 0.35%. However, the Historical estimation method demonstrates slightly superior performance, making it the preferred choice for our main experiments.

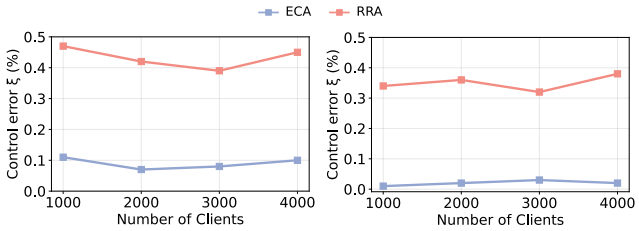


Figure 8: Control error ξ (%) for different numbers of clients under τ = 70% (left) and τ = 80% (right) on the CIFAR10 dataset.

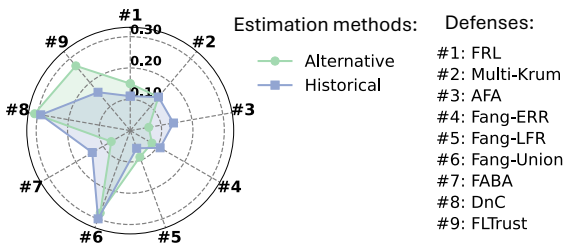


Figure 9: Control errors ξ (%) for different estimation methods under τ = 70% on the CIFAR10 dataset.

6 Discussion

Comparison with Related Work. We provide a detailed comparison with the SOTA MPA designed for FRL, namely VEM [18], from three key perspectives: attack objective, technical methodology, and empirical performance. The main difference between VEM and ECA lies in their fundamental attack objectives. VEM aims for a Denial of Service (DoS) attack, which focuses on causing a continuous degradation of global model performance without any bounds. In contrast, ECA introduces a novel fine-grained control attack. This attack is both dynamic and bidirectional, designed to maintain model performance precisely at a specified target value. Its objective is to prevent the performance from exceeding or falling below the target accuracy, making it a significantly greater challenge. As a result, existing optimization-based methods are not suitable for addressing this type of attack.

The differences in objectives require fundamentally different technical approaches. VEM employs a stochastic optimization method, specifically the Gumbel-Sinkhorn method, to iteratively search for a maximally damaging update. In contrast, ECA utilizes a deterministic two-stage algorithm that directly constructs the malicious ranking. This distinction also extends to the types of vulnerability that are exploited. In VEM, edges are static and defined globally based on their proximity to the mask boundary. On the other hand, ECA's Ascending and Descending Edges (AE/DE) are task-specific and determined dynamically. Our experiments reveal a significant **96% non-overlap** between these two sets of edges.

Empirically, Table 2 shows VEM's control error grows with target accuracy, whereas ECA maintains consistently low error. Fig. 10 further confirms ECA's stable convergence compared to VEM's volatility. This demonstrates that fine-grained control is a distinct challenge requiring our unique algorithmic solution.

Table 2: Comparison of the control error ξ (%) of our method and VEM under the FashionMNIST dataset, LeNet architecture, and different target accuracies τ (%).

Attack	τ = 50%	τ = 55%	τ = 60%	τ = 65%	τ = 70%
VEM	5.92	3.51	5.13	9.71	14.72
ECA (Ours)	**1.95**	**1.18**	**0.87**	**0.59**	**0.76**

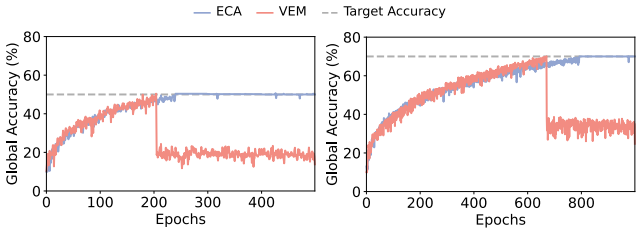


Figure 10: Convergence curves of ECA and VEM attacks on the CIFAR10 dataset when τ = 50% (left) and τ = 70% (right).

Limitations and Future Work. Our work currently focuses on untargeted attacks; extending it to targeted scenarios is a priority for future work. We will also explore its effectiveness against more robust FL frameworks and defenses. The broader impact of our work underscores the critical need to evaluate the security of promising FL frameworks like FRL, which is based on the LTH and aims for lightweight models. Ultimately, these explorations highlight the urgent need to develop more resilient FL methods, particularly for enhancing the security of discrete-space FL paradigms.

7 Conclusion

In this work, we reassess security analysis for ranking-based FL, revealing its vulnerability to sophisticated threats beyond simple disruptions. Our analysis shows that an adversary can exploit FRL's update mechanism not just to damage a model but to precisely steer its accuracy to a specific, attacker-chosen level. This capability is particularly dangerous in competitive commercial scenarios, where an attacker can stealthily degrades a competitor's model to secure a decisive strategic advantage without raising alarms. Based on the observations, we proposed the ECA, the first fine-grained control attack target FRL system. Our novel two-stage algorithm successfully achieves the attacker's dual objectives of precise control and stealth. We provided a theoretical foundation for ECA, formally characterizing the attack's feasibility and establishing the precise conditions and high probability of its success. Our extensive experiments then confirmed this theoretical promise, demonstrating that ECA can consistently maintain control with an exceptionally low average error of 0.224% across diverse settings. This confirms that the strategic threat we identified is not merely theoretical but highly practical and effective.

8 Ethical Considerations

This research reveals FRL vulnerabilities strictly to improve defense. Experiments rely exclusively on public datasets and open-source models, involving no personal data. We advocate responsible disclosure and will release code with appropriate safeguards.

References

[1] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to Backdoor Federated Learning. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, Vol. 108. 2938–2948.

[2] Gilad Baruch, Moran Baruch, and Yoav Goldberg. 2019. A Little Is Enough: Circumventing Defenses for Distributed Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 32. 8632–8642.

[3] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing Federated Learning Through an Adversarial Lens. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, Vol. 97. 634–643.

[4] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 30. 119–129.

[5] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. 2021. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping. In *28th Annual Network and Distributed System Security Symposium (NDSS)*.

[6] Xiaoyu Cao, Zaixi Zhang, Jinyuan Jia, and Neil Zhenqiang Gong. 2022. Flcert: Provably secure federated learning against poisoning attacks. *IEEE Transactions on Information Forensics and Security* 17 (2022), 3691–3705.

[7] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. 2017. EMNIST: Extending MNIST to Handwritten Letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*. 2921–2926.

[8] Yue Cui, Liuyi Yao, Yaliang Li, Ziqian Chen, Bolin Ding, and Xiaofang Zhou. 2024. An auction-based marketplace for model trading in federated learning. *arXiv preprint arXiv:2402.01802* (2024).

[9] Aryan Deshwal, Syrine Belakaria, Janardhan Rao Doppa, and Dae Hyun Kim. 2022. Bayesian optimization over permutation spaces. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 36. 6515–6523.

[10] DMDave, Todd B, and Will Cukierski. 2014. Acquire Valued Shoppers Challenge. https://kaggle.com/competitions/acquire-valued-shoppers-challenge. Kaggle.

[11] Paul Dutting, Zhe Feng, Harikrishna Narasimhan, David C. Parkes, and Sai S. Ravindranath. 2021. Optimal auctions through deep learning. *Commun. ACM* 64, 8 (2021), 109–116.

[12] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. 2018. The Hidden Vulnerability of Distributed Learning in Byzantium. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, Vol. 80. 3518–3527.

[13] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. 2020. Local model poisoning attacks to Byzantine-Robust federated learning. In *29th USENIX Security Symposium (USENIX Security 20)*. 1605–1622.

[14] Minghong Fang, Jia Liu, Neil Zhenqiang Gong, and Elizabeth S Bentley. 2022. AFLguard: Byzantine-robust asynchronous federated learning. In *Proceedings of the 38th Annual Computer Security Applications Conference (ACSAC)*. 632–646.

[15] Minghong Fang, Seyedsina Nabavirazavi, Zhuqing Liu, Wei Sun, Sundaraja Sitharama Iyengar, and Haibo Yang. 2025. Do We Really Need to Design New Byzantine-robust Aggregation Rules?. In *32nd Annual Network and Distributed System Security Symposium (NDSS)*.

[16] Jonathan Frankle and Michael Carbin. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *7th International Conference on Learning Representations (ICLR)*.

[17] Zirui Gong, Liyue Shen, Yanjun Zhang, Leo Yu Zhang, Jingwei Wang, Guangdong Bai, and Yong Xiang. 2024. AgrAmplifier: Defending Federated Learning Against Poisoning Attacks Through Local Update Amplification. *IEEE Transactions on Information Forensics and Security* 19 (2024), 1241–1250.

[18] Zirui Gong, Yanjun Zhang, Leo Yu Zhang, Zhaoxi Zhang, Yong Xiang, and Shirui Pan. 2025. Not All Edges are Equally Robust: Evaluating the Robustness of Ranking-Based Federated Learning. In *2025 IEEE Symposium on Security and Privacy (SP)*. 2527–2545.

[19] Adriano Guastella, Lorenzo Sani, Alex Iacob, Alessio Mora, Paolo Bellavista, and Nicholas Donald Lane. 2025. SparsyFed: Sparse Adaptive Federated Learning. In *The 13th International Conference on Learning Representations (ICLR)*.

[20] Pengxin Guo, Yinong Wang, Wei Li, Mengting Liu, Ming Li, Jinkai Zheng, and Liangqiong Qu. 2025. Exploring Federated Pruning for Large Language Models. *arXiv preprint arXiv:2505.13547* (2025).

[21] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. 2018. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*. 19–35.

[22] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning Multiple Layers of Features from Tiny Images.* Technical Report. University of Toronto.

[23] Minghui Li, Wei Wan, Yuxuan Ning, Shengshan Hu, Lulu Xue, Leo Yu Zhang, and Yichen Wang. 2024. DarkFed: A Data-Free Backdoor Attack in Federated Learning. In *Proceedings of the 33nd International Joint Conference on Artificial Intelligence (IJCAI)*. 4443–4451.

[24] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Vol. 54. 1273–1282.

[25] Hamid Mozaffari, Virat Shejwalkar, and Amir Houmansadr. 2023. Every Vote Counts:{Ranking-Based} Training of Federated Learning to Resist Poisoning Attacks. In *32nd USENIX Security Symposium (USENIX Security 23)*. 1721–1738.

[26] Luis Muñoz-González, Kenneth T. Co, and Emil C. Lupu. 2019. Byzantine-Robust Federated Machine Learning through Adaptive Model Averaging. *ArXiv* abs/1909.05125 (2019).

[27] Thien Duc Nguyen, Phillip Rieger, Roberta De Viti, Huili Chen, Björn B Brandenburg, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, et al. 2022. FLAME: Taming backdoors in federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*. 1415–1432.

[28] Huazi Pan, Yanjun Zhang, Leo Yu Zhang, Scott Adams, Abbas Kouzani, and Suiyang Khoo. 2025. Performance Guaranteed Poisoning Attacks in Federated Learning: A Sliding Mode Approach. 6021-6029 pages.

[29] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. 2020. What's hidden in a randomly weighted neural network?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11890–11899.

[30] Stanley Reiter and Gordon Sherman. 1965. Discrete optimizing. *J. Soc. Indust. Appl. Math.* 13, 3 (1965), 864–889.

[31] Kyeongkook Seo, Dong-Jun Han, and Jaejun Yoo. 2025. PRISM: Privacy-Preserving Improved Stochastic Masking for Federated Generative Models. In *The 13th International Conference on Learning Representations (ICLR)*.

[32] Virat Shejwalkar and Amir Houmansadr. 2021. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *28th Annual Network and Distributed System Security Symposium (NDSS)*.

[33] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*. 3–18.

[34] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. 2019. Can you really backdoor federated learning? *ArXiv* abs/1911.07963 (2019).

[35] Qi Xia, Zeyi Tao, Zijiang Hao, and Qun A. Li. 2019. FABA: An Algorithm for Fast Aggregation against Byzantine Attacks in Distributed Neural Networks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*. 4824–4830.

[36] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *CoRR* abs/1708.07747 (2017).

[37] Yichang Xu, Ming Yin, Minghong Fang, and Neil Zhenqiang Gong. 2024. Robust Federated Learning Mitigates Client-side Training Data Distribution Inference Attacks. In *Companion Proceedings of the ACM on Web Conference 2024*. 798–801.

[38] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudré-Mauroux. 2019. Revisiting User Mobility and Social Relationships in LBSNs: A Hypergraph Embedding Approach. In *The World Wide Web Conference (WWW)*. 2147–2157.

[39] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudré-Mauroux. 2022. LBSN2Vec++: Heterogeneous Hypergraph Embedding for Location-Based Social Networks. *IEEE Transactions on Knowledge and Data Engineering* 34, 4 (2022), 1843–1855.

[40] Jiaxi Yang, Zihao Guo, Sheng Cao, Cuifang Zhao, and Li-Chuan Tsai. 2023. Federated learning incentive mechanism under buyers' auction market. *arXiv preprint arXiv:2309.05063* (2023).

[41] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, Vol. 80. 5636–5645.

[42] Hangtao Zhang, Zeming Yao, Leo Yu Zhang, Shengshan Hu, Chao Chen, Alan Liew, and Zhetao Li. 2023. Denial-of-service or fine-grained control: towards flexible model poisoning attacks on federated learning. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*. 4567–4575.

[43] Jianming Zhang, Xifan Yao, Min Liu, and Yan Wang. 2019. A bayesian discrete optimization algorithm for permutation based combinatorial problems. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. 874–881.

[44] 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 (KDD)*. 2545–2555.

[45] Yaru Zhao, Jianbiao Zhang, Yihao Cao, and Tao Ye. 2024. FEFL: A Fair & Efficient Defense Against Poisoning Attacks via Discrete Update Space. In *2024 7th International Conference on Information Communication and Signal Processing (ICICSP)*. 295–300.

A Proof

A.1 Proof of Lemma 1.

In this section, we detail the proof of Lemma 1. We will prove the upper bound of the vulnerable edge position $\pi^{-1}(e_{mal})$, and the method of proving the lower bound will be similar.

Since the aggregated importance scores of the benign clients S_{ben} is an increasing sequence, we assume that the S_{ben} contribution from benign clients is $S_{ben}(e) = \sum_{u=1}^{U-m} I_u(e) \approx (U-m)\pi^{-1}(e)$.

We consider the edge ($e_{ae} \in \mathbb{I}_{\text{AE}}$) to be manipulated that has the largest benign position in $\mathbb{I}_{\text{AE}}$, and in order to place this edge below the boundaries of the final poisoned global model $\hat{R}_g$, the ECA method gives this edge the lowest score in the malicious rankings, that is, $I(e_{ae}) = 0$. So we have the following equation:

$$\begin{aligned}
&\sum_{u=1}^{U-m} I_u(e_{ae}) + \sum_{v=1}^{m} I_v(e_{ae}) < \sum_{z=1}^{U} I_z(e_k) \\
&\approx (U-m) \times \pi^{-1}(e_{ae}) + m \times 0 < U \times kn \\
&= (1-\alpha)U\pi^{-1}(e_{ae}) < knU \\
&= \pi^{-1}(e_{ae}) < \frac{kn}{(1-\alpha)},
\end{aligned} \tag{5}$$

where e_k denotes the smallest edge at the boundary, i.e., the edge at the kn^{th} position.

Similarly, we consider the edge e_{de} to be manipulated to have the least benign position in $\mathbb{I}_{\text{DE}}$, and in order to place this edge above the boundaries of the final poisoned global model $\hat{R}_g$, the ECA method gives the highest score to this edge in the malicious rankings, i.e., $I(e_{de}) = n-1$. Thus, we have the following equation:

$$\begin{aligned}
&\sum_{u=1}^{U-m} I_u(e_{de}) + \sum_{v=1}^{m} I_v(e_{de}) \geq \sum_{z=1}^{U} I_z(e_k) \\
&\approx (U-m) \times \pi^{-1}(e_{de}) + m \times (n-1) \geq U \times kn \\
&= (1-\alpha)U\pi^{-1}(e_{de}) + \alpha U(n-1) \geq knU \\
&= \pi^{-1}(e_{de}) \geq \frac{kn - \alpha(n-1)}{(1-\alpha)}.
\end{aligned} \tag{6}$$

Given all the $\mathbb{I}_{\text{AE}}$ and $\mathbb{I}_{\text{DE}}$, we obtain the range of benign positions $\pi^{-1}(e_{mal})$ that the attacker can control by considering their worst-case scenarios in combination with Eq. 5 and Eq. 6, we have:

$$\frac{kn - \alpha(n-1)}{(1-\alpha)} \leq \pi^{-1}(e_{mal}) < \frac{kn}{(1-\alpha)}. \tag{7}$$

A.2 Proof of Theorem 1.

In this section, we present a detailed proof of Theorem 1. Additionally, we provide our experimental results, which confirm the effectiveness of our attack with higher true probabilities compared to the theoretical values.

When performing an ECA attack, the attacker selects a set of target edges ($\mathbb{I}_{\text{AE}} \cup \mathbb{I}_{\text{DE}}$). Given these edges, we treat the positions of the edges as a random variable. We assume that the positional distribution approximates a discrete Gaussian distribution centered at μ with standard deviation σ, i.e., $\pi^{-1}(e) \sim D_{\mathbb{Z},\mu,\sigma}$. Fig. 11 shows that we conducted multiple rounds of experimentation, resulting in similar distributions for the position data. The red dashed line indicates the kn positions with $k = 50\%$. We found that the kernel density distribution of $\mathbb{I}_{\text{AE}} \cup \mathbb{I}_{\text{DE}}$ overlapped significantly across the rounds, with the highest density region concentrated around $\frac{n}{2}$. This is highly consistent with our intuition about the scheme, since the edges at the boundary are prone to change, so we consider the expectation of this distribution to be $\mu = kn$, that is, $\pi^{-1}(e) \sim D_{\mathbb{Z},kn,\sigma}$.

According to Lemma 1, we get the range of vulnerable edges that the attacker can control, so the probability that the attacker can successfully control the DE and AE edges can be defined as the probability that any of the target edges fall into the vulnerable range. We use the successive Gaussian probabilities within the interval as the discrete probabilities, i.e.,

$$P(X=k) = \Phi\left(\frac{k+0.5-\mu}{\sigma}\right) - \Phi\left(\frac{k-0.5-\mu}{\sigma}\right), \tag{8}$$

for interval $[k-0.5, k+0.5)$. To simplify the notation, let us define the lower and upper bounds of the vulnerable range as L and U, respectively:

$$L = \frac{kn - \alpha(n-1)}{1-\alpha}, \quad U = \frac{kn}{1-\alpha}.$$

So we have:

$$\begin{aligned}
P &= P(L \leq \pi^{-1}(e) < U), \\
&= \textstyle\sum_{z=\lfloor L \rfloor}^{\lceil U \rceil} P(\pi^{-1}(e) = z), \\
&\approx \underline{\Phi((L+0.5-\mu)/\sigma)} - \Phi((L-0.5-\mu)/\sigma) \\
&\quad + \Phi((L+1+0.5-\mu)/\sigma) - \underline{\Phi((L+1-0.5-\mu)/\sigma)} \\
&\quad + \cdots \\
&\quad + \Phi((U+0.5-\mu)/\sigma) - \underline{\Phi((U-0.5-\mu)/\sigma)}, \\
&\approx \Phi((U+0.5-\mu)/\sigma) - \Phi((L-0.5-\mu)/\sigma), \\
&\approx \Phi\left(\frac{2kn + (1-\alpha)(1-2\mu)}{2(1-\alpha)\sigma}\right) \\
&\quad - \Phi\left(\frac{2kn - 2\alpha(n-1) - (1-\alpha)(1+2\mu)}{2(1-\alpha)\sigma}\right).
\end{aligned} \tag{9}$$

As displayed in Fig. 12, we conducted experiments at various edge scales ranging from 10^3 to 10^6 to assess the likelihood of a real attacker successfully manipulating AE&DE. In each of these experiments, we observed the success rates over 20 trials. Our findings revealed that the majority of experimental success probabilities exceeded those predicted by the theoretical model. Additionally, as the size of n increased, our experimental probabilities often reached 100%, which further validates the effectiveness of our attacks.

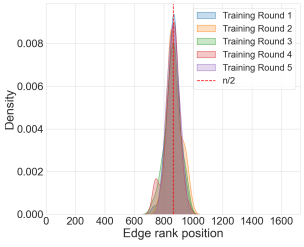


Figure 11: Kernel density distribution of $\mathbb{I}_{\text{AE}} \cup \mathbb{I}_{\text{DE}}$ position data.

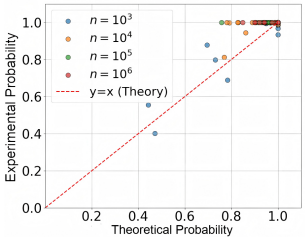


Figure 12: Comparison of theoretical and experimental probabilities.

B Details of Experiment Settings

In this section, we provide the details of the experimental setup that are missing from Section 5.

B.1 Datasets

For each dataset experiment, we train the model using a default sparsity rate of 0.5. In local training, we follow the configuration outlined in [25]: each client trains for 5 epochs using the SGD optimizer with the following parameters: learning rate 0.4, momentum 0.9, weight decay 1e-4, and batch size 32.

CIFAR10 [22] is a 10-class classification task consisting of 60000 RGB images (50,000 for training and 10,000 for testing), each sized by 32×32. We experiment with Conv8 architecture.

EMNIST [7] consists of 814255 gray-scale images of numeric, upper, and lower case handwritten letters, with a total of 62 categories. We experiment with the LeNet architecture.

Fashion-MNIST [36] is a 10-class fashion image for classification tasks, using a training set of 60000 images and a testing set of 10000 images with a resolution of 28×28 pixels. We experiment with the LeNet architecture.

CIFAR100 [22] is a standard 100-class image classification benchmark, consisting of 60,000 32×32 RGB images (50,000 for training and 10,000 for testing), with 500 training and 100 test samples per class. We experiment with Conv8 architecture.

Location30 [38, 39] is a 30-class geosocial classification dataset derived from the Foursquare check-in records, containing 5,010 samples with 446 binary features. Each feature indicates whether a user has visited a specific region or location type. We obtain a simplified and preprocessed Location dataset provided by Shokri et al. [33]. We experiment with FC architecture.

Purchase100 [10] is a 100-class classification task comprising 197,324 data samples with 600 binary features, derived from Kaggle's Acquire Valued Shoppers Challenge and preprocessed by Shokri et al. [33]. Each feature indicates the purchase status of a specific product, and samples are categorized into distinct consumer behavior patterns. We experiment with FC architecture.

Texas100 [33] is a 100-class medical procedure prediction dataset derived from public patient records released by the Texas Department of State Health Services. The preprocessed dataset [33] contains 67,330 samples with 6,170 binary characteristics, encoding patient demographics (e.g., age, sex), external causes of injury (e.g., suicide, drug misuse) and diagnoses (e.g., schizophrenia). The classification task predicts the most frequent procedures (e.g., surgery) from patient profiles. We experiment with FC architecture.

B.2 Evaluated Defenses and Settings

Krum [4] selects a single local update as the global model by identifying the update closest (in Euclidean distance) to $U - m - 2$ neighboring updates, where m is the upper bound on the number of malicious clients, and U is the total number of selected clients in the current training round.

Multi-Krum [4] extends Krum by selecting c local updates such that $U - c > 2 * m + 2$. These c updates are then averaged to form the global model.

Adaptive federated average (AFA) [26] computes cosine similarity between each local update and a benign reference gradient. Gradients with similarities outside a predefined range—determined by the mean, median, and standard deviation of all similarities—are discarded.

Fang [13] assumes that the server has access to a clean validation dataset and uses it to evaluate the uploaded model updates for detecting malicious clients. This method consists of three variants: Fang-ERR, Fang-LFR, and Fang-Union. Fang-ERR identifies malicious updates by assessing the classification error rate on the validation dataset, where updates leading to significantly higher error rates are flagged as malicious. Fang-LFR, on the other hand, evaluates the loss values of the updated models on the validation dataset, marking updates with substantially higher losses as potential outliers. Fang-Union combines the two approaches by simultaneously considering both error rates and loss values; an update is deemed malicious only if it exhibits abnormal behavior in both metrics, thereby enhancing the robustness and accuracy of the detection process. This methodology effectively leverages the server's clean validation dataset to distinguish between benign and malicious updates in federated learning frameworks.

FABA [35] iteratively removes the local update that is furthest from the average of all updates until the number of excluded updates equals the estimated number of attackers.

Divide-and-conquer (DnC) [32] uses singular value decomposition (SVD) to detect and filter outliers. It first creates a gradient subsample by randomly selecting indices from the input gradient dimensions, then centers the subsample by subtracting its mean. The centered gradients are projected onto the top singular eigenvector, capturing the dominant variation direction. Gradients with the highest projection scores are removed as outliers.

FLTrust [5] incorporates a trusted validation dataset on the server to compute reference gradients. Each client is assigned a trust score based on the similarity between its local update and the reference gradient. These trust scores are used as weights when aggregating to form the global model.

C More Experiment Results

Computational efficiency analysis. Our proposed attack is computationally efficient as its core manipulation is based on reordering operations rather than costly optimization. The process involves identifying differing edges (AE/DE), adjusting malicious rankings, and applying an internal reverse strategy. The time complexity is primarily bounded by sorting, at $O(n \log n)$, which is necessary for generating or aggregating rankings. This contrasts sharply with optimization-based attacks that require extensive resources for iterative gradient-based solutions, often involving numerous model propagations. Our method circumvents these complex procedures by oper ating directly in the discrete ranking space, allowing for the direct construction of malicious updates.

The attack construction phase imposes a minimal computational burden on the adversary. While malicious clients still perform standard local training (Algorithm 1, line 3), the subsequent manipulation is lightweight. This is especially true when using the "historical estimation" method, which leverages the previous round's global ranking and requires negligible additional computation. This efficiency allows the attack to be executed effectively even in resource-constrained environments, enhancing its practical threat and stealth.

Table 3: Comparison of RRA and our attack ECA under various AGRs in a control scenario. The "no attack" row reports the global model accuracy without attack $ACC(R_g)$ (%); other rows report control error ξ (%) under various AGRs and τ.

Dataset (Model)	Attack	Target τ (%)	FRL	Multi-Krum-FRL	AFA-FRL	Fang-ERR-FRL	Fang-LFR-FRL	Fang-Union-FRL	FABA-FRL	DnC-FRL	FLTrust-FRL
CIFAR10 (Conv8)	no attack	/	83.85	84.43	84.47	84.64	85.16	84.83	84.53	84.44	84.62
	RRA	40	2.41 (±1.58)	1.65 (±1.52)	2.01 (±1.41)	3.00 (±1.96)	1.90 (±2.11)	1.99 (±1.48)	2.07 (±2.10)	3.98 (±2.66)	1.83 (±1.41)
		50	1.72 (±1.70)	1.99 (±1.79)	1.62 (±1.88)	1.73 (±1.26)	1.84 (±1.56)	1.70 (±1.25)	2.10 (±2.20)	2.39 (±1.62)	1.75 (±1.73)
		60	1.68 (±1.32)	2.23 (±2.38)	1.63 (±1.73)	1.39 (±1.24)	1.65 (±1.52)	1.77 (±1.99)	2.25 (±2.18)	1.57 (±1.27)	1.77 (±1.68)
	ECA	40	**0.18 (±0.27)**	**0.14 (±0.40)**	**0.11 (±0.22)**	**0.08 (±0.18)**	**0.05 (±0.10)**	**0.08 (±0.19)**	**0.17 (±0.28)**	**0.34 (±0.54)**	**0.09 (±0.23)**
		50	**0.09 (±0.17)**	**0.05 (±0.13)**	**0.05 (±0.10)**	**0.05 (±0.10)**	**0.04 (±0.08)**	**0.09 (±0.20)**	**0.06 (±0.12)**	**0.38 (±0.58)**	**0.08 (±0.14)**
		60	**0.10 (±0.27)**	**0.04 (±0.08)**	**0.03 (±0.09)**	**0.06 (±0.13)**	**0.04 (±0.14)**	**0.04 (±0.09)**	**0.03 (±0.08)**	**0.18 (±0.41)**	**0.06 (±0.12)**

Table 4: Control error ξ (%) of our attack on other AGRs with different malicious rates under FRL framework.

AGRs	Malicious Rate	$\tau = 70\%$	$\tau = 75\%$	$\tau = 80\%$	Average ξ
FLCert	10%	0.43 (±0.35)	0.15 (±0.11)	0.11 (±0.13)	0.36
	20%	0.15 (±0.27)	0.05 (±0.10)	0.04 (±0.08)	0.11
FoundationFL	10%	0.55 (±0.49)	0.31 (±0.37)	0.23 (±0.15)	0.40
	20%	0.35 (±0.29)	0.11 (±0.25)	0.06 (±0.19)	0.20

Table 5: Average control error ξ (%) of our attack on other datasets with different malicious rates under FRL framework.

Malicious rate	CIFAR100 (Conv8)	Location30 (FC)	Purchase100 (FC)	Texas100 (FC)
10%	0.33 (±0.35)	0.21 (±0.22)	0.48 (±0.41)	0.27 (±0.31)
20%	0.03 (±0.05)	0.07 (±0.12)	0.15 (±0.20)	0.06 (±0.11)

Impact of lower τ values. We further compared ECA with RRA using lower target τ values on the CIFAR-10, EMNIST, and FashionMNIST datasets. As detailed in Table 3 for CIFAR-10, ECA consistently achieves superior precision with control errors below 0.4%, significantly outperforming RRA, which exhibits much higher errors (1.57% to 3.98%). This quantitative advantage is visually confirmed across all three datasets in Fig. 13. These figures collectively demonstrate that under ECA, the global model accuracy rapidly converges to and stabilizes at the target τ shortly after the attack begins. In stark contrast, RRA's performance shows substantial fluctuations and an inability to maintain stable control, a weakness that is particularly pronounced at lower τ values. These results underscore ECA's robustness and precision in achieving fine-grained model control, a capability where RRA is demonstrably deficient.

Experiments on additional defense mechanisms. We evaluated our attack against SOTA defenses: FoundationFL [15], an augmented defense, and FLCert [6], a leading authentication-based defense. As shown in Table 4, our attack maintains precise control with low average error ξ (%) against both. For instance, with a 20% malicious rate, the average error was merely 0.11% against FLCert and 0.20% against FoundationFL. These results demonstrate that our attack effectively bypasses these advanced defenses, achieving its objective with consistently low error rates regardless of the defense mechanism or malicious participation level.

Experiments on additional datasets. Table 5 presents the average error control of our attack on larger datasets, as well as three tabular datasets with 10% and 20% malicious rates under the FRL framework. From the table, we can see that our attack achieves desirable attack control in all cases.

Table 6: Control error ξ (%) of our ECA under various AGRs.

Target τ	Malicious Rate	FRL	FLTrust-FRL	FABA-FRL	DnC-FRL	Fang-Union-FRL
40(%)	1(%)	1.27 (±1.90)	1.40 (±1.42)	1.64 (±1.92)	1.90 (±2.27)	1.32 (±1.22)
70(%)	1(%)	0.68 (±0.48)	0.91 (±0.83)	0.71 (±0.55)	0.87 (±0.60)	0.74 (±0.49)
40(%)	2(%)	0.97 (±1.52)	1.21 (±1.45)	1.35 (±1.88)	1.53 (±1.39)	1.21 (±1.75)
70(%)	2(%)	0.61 (±0.48)	0.47 (±0.36)	0.54 (±0.35)	0.75 (±0.43)	0.41 (±0.50)

Impact of lower malicious rates. We conduct additional experiments to verify the effectiveness of our method under lower malicious user rates. As shown in Table 6, our ECA attack demonstrates potent effectiveness against a wide array of AGRs even with only 1% malicious clients. For a control target of $\tau = 40\%$, the control error (ξ) remains consistently low across all defenses, such as 1.40 ± 1.42 against FLTrust-FRL and 1.32 ± 1.22 against Fang-Union-FRL. Furthermore, when we increase the control target to $\tau = 70\%$, the attack's precision improves and the control error drops even further; for instance, the error against Fang-Union-FRL decreases from 1.32 ± 1.22 to 0.74 ± 0.49. This trend is amplified when the malicious client rate is increased to 2%, where the control error for $\tau = 70\%$ against Fang-Union-FRL is reduced to just 0.41 ± 0.50.

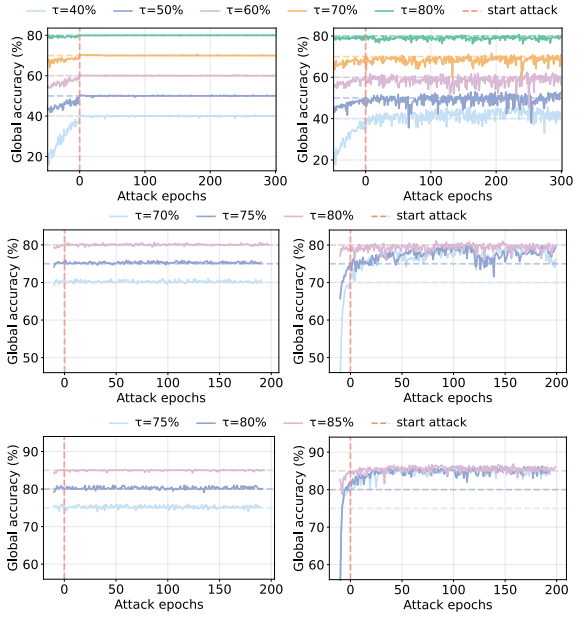


Figure 13: Global model accuracy (%) of ECA (left) and RRA (right) on CIFAR10 (first row), EMNIST (second row), and Fashion-MNIST (third row) datasets with different τ.