LoRA-FL: A Low-Rank Adversarial Attack for Compromising Group Fairness in Federated Learning

Sankarshan Damle [1] Ljubomir Rokvic [1] Venugopal Bhamidi [2] Manisha Padala [2] Boi Faltings [1]

Abstract

Federated Learning (FL) enables collaborative model training without sharing raw data, but agent distributions can induce unfair outcomes across sensitive groups. Existing fairness attacks often degrade accuracy or are blocked by robust aggregators like KRUM. We propose LoRA-FL: a stealthy adversarial attack that uses low-rank adapters to inject bias while closely mimicking benign updates. By operating in a compact parameter subspace, LoRA-FL evades standard defenses without harming accuracy. On standard fairness benchmarks (Adult, Bank, Dutch), LoRA-FL reduces fairness metrics (DP, EO) by over 40% with only 10–20% adversarial agents, revealing a critical vulnerability in FL's fairness-security landscape. Our code base is available at: https://github.com/sankarshandamle/LoRA-FL.

1. Introduction

Federated Learning (FL) (McMahan et al., 2017) trains machine learning models across decentralized agents without directly sharing raw data. It enables collaboration by coordinating local model updates, preserving both privacy and data ownership. FL has shown strong performance across domains: for instance, next-word prediction (Hard et al., 2018), healthcare (Sheller et al., 2020; Xu et al., 2021), and finance (Yang et al., 2019), balancing data-driven insights with regulatory and confidentiality demands.

Fairness in FL. While achieving high predictive accuracy is a central objective in FL, it does not guarantee equitable performance across *demographic groups* (e.g., gender, age, race). Prior work highlights that group-level disparities can persist even in models with strong overall accuracy (Angwin et al., 2016; Fabris et al., 2025). Several notions of *group fairness* have been proposed to formalize such concerns. We focus on three widely studied criteria: (i) *Demographic Parity (DP)* (Chouldechova, 2017), requiring equal positive prediction rates across groups; (ii) *Equalized Odds (EO)* (Hardt et al., 2016a), which demands parity in false positive and false negative rates; and (iii) *Equal Opportunity (EOpp)* (Hardt et al., 2016a), a relaxed version that considers only true positive rates. Enforcing these criteria in FL is especially challenging due to the decentralized aggregation of locally-trained models, where even benign disparities at the agent level can accumulate and amplify global unfairness (Chang & Shokri, 2023; Wang et al., 2024). This vulnerability is further amplified in the presence of *adversarial* agents, who may deliberately manipulate local updates to induce or worsen disparities in the global model's treatment of sensitive groups.

Adversarial Attacks in FL. FL remains highly vulnerable to *model poisoning attacks*, where adversarial agents inject carefully crafted updates to disrupt training or degrade global model performance (Cao et al., 2020; So et al., 2020). To counter such threats, researchers propose Byzantine-resilient aggregators such as KRUM (Blanchard et al., 2017a) and *trimmed-mean* (TM) (Yin et al., 2018). For instance, KRUM selects the update closest (in Euclidean distance) to the majority, effectively filtering out large deviations and defending against high-magnitude attacks.

Yet, not all attacks seek to degrade accuracy. Malicious agents can execute stealthy bias attacks that preserve predictive performance while systematically worsening group fairness. While prior works have explored such fairness-compromising attacks (Meerza & Liu, 2024), they fail to evade detection by robust aggregators like KRUM (Blanchard et al., 2017a). This paper proposes a stealthy attack, namely LoRA-FL, that successfully degrades group fairness even in the presence of robust aggregators, thereby exposing a critical blind spot in the current adversarial threat landscape.

Our Approach & Contributions

LoRA-FL. We introduce LoRA-FL: a novel *stealthy adversarial attack* that systematically degrades group fairness in federated learning (FL) while preserving accuracy and

[1] LIA, EPFL [2] Dept. of CSE, IIT Gandhinagar. Correspondence to: Sankarshan Damle <sankarshan.damle@epfl.ch>.

Accepted at the ICML 2025 Workshop on Collaborative and Federated Agentic Workflows (CFAgentic@ICML'25), Vancouver, Canada. July 19, 2025. Copyright 2025 by the author(s).

evading detection by robust aggregators. LoRA-FL exploits *biased subspaces* within the model parameter space using *low-rank adapters* to inject targeted unfairness. By constraining adversarial perturbations to a carefully chosen low-dimensional subspace, the attack ensures that updates mimic the natural variation of benign agents. This design allows adversaries to embed unfair behavior while maintaining proximity to legitimate updates, thus bypassing standard proximity-based defenses.

Our Contributions. **First**, we propose LoRA-FL, an attack strategy that leverages low-rank adapters to inject fairness-oriented bias into federated models without compromising predictive performance. Operating within a low-dimensional subspace, LoRA-FL allows adversarial clients to craft updates that remain indistinguishable from benign ones under distance-based defenses such as KRUM (Blanchard et al., 2017a).

Second, we demonstrate the effectiveness of LoRA-FL on standard fair classification benchmarks – Adult (Dua & Graff, 2017), Bank (Moro et al., 2014), and Dutch (Žliobaite et al., 2011). With only 10–20% of adversarial clients, LoRA-FL reduces fairness metrics (DP, EO, and EOpp) by over 40% while preserving high accuracy. An ablation study further confirms LoRA-FL's robustness to varying numbers of agents. Additionally, increasing the adapter rank makes adversarial updates more detectable, highlighting the importance of low-rank constraints.

Third, we conduct a detailed interpretability analysis to understand why low-rank adapters are effective. Lower-rank perturbations closely align with benign update distributions, allowing them to evade detection. Moreover, these adapters disrupt internal neuron-level representations that systematically skew predictions across demographic groups. Finally, we show that even higher-rank adapters concentrate changes along a few principal directions – explaining why low-rank updates suffice to achieve the attack's effect while remaining covert.

2. Related Work

Poisoning Attacks on Fairness in Centralized Learning. In a centralized setup, several studies have investigated how adversaries can compromise fairness through data poisoning. Solans et al. (Solans et al., 2020) introduced a gradient-based poisoning attack that induces classification disparities among demographic groups, effectively degrading fairness metrics such as DP and EO. Van et al. (Van et al., 2022) proposed a framework that generates poisoning samples targeting both model accuracy and algorithmic fairness. Mehrabi et al. (Mehrabi et al., 2021) presented anchoring and influence attacks that manipulate the training data to exacerbate algorithmic bias. While these works focus on instance-level poisoning in centralized training, adversaries in FL have significantly greater power: they can control entire clients and directly poison the global model through malicious updates.

Fairness-aware Aggregators in FL. To counter fairness concerns in decentralized learning, recent research has proposed fairness-aware aggregation schemes. FairFed (Ezzeldin et al., 2023) introduces a server-side mechanism that improves group fairness without requiring access to sensitive attributes, and is compatible with client-side debiasing techniques. Similarly, GIFAIR-FL (Yue et al., 2023) proposes a joint optimization framework that incorporates fairness regularizers into the federated objective, addressing both group and individual fairness. While such defenses improve fairness in benign settings, they remain vulnerable to adversarial manipulations.

Adversarial Attacks on Fairness in FL. To the best of our knowledge, only two prior works study adversarial attacks on fairness in FL. PFAttack (Gao et al., 2024) demonstrates that adversaries can subvert fairness-aware aggregation mechanisms in FL without significantly compromising overall model accuracy. Our work differs from PFAttack in several key aspects. First, PFAttack is tailored for demographic parity (DP) and assumes the presence of fairness-aware aggregators, whereas our attack is model-agnostic and applies in the presence of robust aggregators. Second, it achieves stealthiness empirically – by tuning the fairness-accuracy trade-off such that standard fairness detectors fail to detect the attack. It operates in the whole parameter space without structural constraints on the update vectors. In contrast, LoRA-FL constrains malicious updates to a low-dimensional subspace via low-rank adapters, inducing bias through representational shifts at the neuron level, making the attack inherently stealthy by design. EAB-FL (Meerza & Liu, 2024) introduces a model poisoning attack aimed at increasing bias while preserving overall utility. However, this method is ineffective against robust aggregators (Meerza & Liu, 2024, Section 5). Moreover, we compare LoRA-FL against EAB-FL and show that our attack is more effective: we achieve significantly higher accuracy while inducing comparative fairness violations across multiple metrics.

3. Background

We consider a standard supervised classification setting where each data point (x, y, a) is drawn i.i.d. from a distribution $\mathcal{D}$ over input features $x \in \mathcal{X}$, labels $y \in [C]$, and sensitive attributes $a \in \mathcal{A}$ (e.g., gender or age). The objective is to learn a classifier $f : \mathcal{X} \to [C]$ that is both accurate and fair across the groups defined by a.

Federated Learning (FL). In the FL setting (McMahan et al., 2017), a central aggregator coordinates training across K agents, each with local data (typically) sampled i.i.d.

from $\mathcal{D}$. The global model is updated iteratively through the aggregation of agent updates. The de facto method, `FedAvg` (McMahan et al., 2017), computes a weighted average of local models based on dataset size. However, in the presence of adversarial agents, `FedAvg` is vulnerable to outlier updates. To address this, researchers introduce robust aggregators: m-`KRUM` (Blanchard et al., 2017a), which filters outlier updates by minimizing ℓ-distances among agents, and `Trimmed-Mean` (`TM`) (Yin et al., 2018), which discards extreme parameter values dimension-wise to mitigate adversarial behavior.

Group Fairness. Fairness in our context requires that a classifier's performance does not disproportionately vary across groups. We focus on three widely used group fairness criteria. **(i)** *Demographic Parity (DP)* (Dwork et al., 2012) requires that the predicted label be statistically independent of the sensitive attribute: $P(\hat{Y} = y|A = a) = P(\hat{Y} = y)$. **(ii)** *Equal Opportunity (EOpp)* (Hardt et al., 2016a) demands equal true positive rates across groups: $P(\hat{Y} = 1|Y = 1, A = a) = P(\hat{Y} = 1|Y = 1)$. **(iii)** *Equalized Odds (EO)* (Chouldechova, 2017) generalizes this by requiring equal true and false positive rates across groups: $P(\hat{Y} = y'|Y = y, A = a)$ is the same across a, for each label y. As exact satisfaction of these notions is impossible (Chouldechova, 2017), they are instead relaxed and evaluated approximately. With $\mathcal{A} = \{a_1, \ldots, a_K\}$ as the sensitive attribute, we quantify group fairness violations via the maximum gap across groups:

$$\Delta_{\text{DP}} = \max_{k,k'} \left| P(\hat{Y} = 1 \mid A = a_k) - P(\hat{Y} = 1 \mid A = a_{k'}) \right|$$

$$\Delta_{\text{EOpp}} = \max_{k,k'} \left| \begin{aligned} &P(\hat{Y} = 1 \mid Y = 1, A = a_k) \\ &\quad - P(\hat{Y} = 1 \mid Y = 1, A = a_{k'}) \end{aligned} \right|$$

$$\Delta_{\text{EO}} = \max_{k,k'} \max_{y \in \{0,1\}} \left| \begin{aligned} &P(\hat{Y} = 1 \mid Y = y, A = a_k) \\ &\quad - P(\hat{Y} = 1 \mid Y = y, A = a_{k'}) \end{aligned} \right|$$

These metrics measure the extent to which the model violates each fairness criterion. A perfectly fair model will have all $\Delta = 0$, while higher values indicate greater disparity.

Due to space constraints, we present the formal FL setup, m-`KRUM` (Blanchard et al., 2017a), `TM` (Yin et al., 2018), and the formal definitions of the fairness notions in Appendix A.

4. Methodology

In this section, we formalize the overall optimization objective in the FL setting for classification. We then describe the optimization problem solved by a strategic adversary aiming to amplify group-level bias in the resulting global model. Specifically, the adversary seeks to perform a model poisoning attack that circumvents state-of-the-art robust aggregation mechanisms.

4.1. FL Optimization

Recall that our goal is to train a global classifier $f_\theta : \mathcal{X} \to \mathcal{Y}$ that optimizes for global accuracy across a set of $\mathcal{K}$ agents each having a private dataset $\mathcal{D}^{(k)} = \left\{ (x_i^{(k)}, y_i^{(k)}, a_i^{(k)}) \right\}_{i=1}^{n_k}$. Each agent $k \in \mathcal{K}$ defines a local objective $F_k(\theta)$ given by,

$$F_k(\theta) = \mathbb{E}_{x^{(k)}, y^{(k)} \sim \mathcal{D}^{(k)}} \left[\ell_{CE}(\theta; x^{(k)}, y^{(k)}) \right] \quad (1)$$

where $\ell_{CE}(\cdot)$ is the standard cross-entropy loss. The global objective is to minimize the **weighted aggregation** of local losses.

$$\min_{\Theta} F(\Theta) = \sum_{k=1}^{K} w_k F_k(\Theta)$$

Here Θ denotes the model parameters to be optimized, $w_k = \frac{n_k}{n}$ is the weight for agent k, with $n_k = |\mathcal{D}_k|$ being the size of agent k's dataset, and $n = \sum_{k=1}^{K} n_k$ representing the total number of data points across all agents.

4.1.1. FL Optimization: Adversarial Agent

When a subset of agents is malicious, they aim to compromise the fairness of the global model by launching model poisoning attacks. Let $K_A \subset K$ denote the set of *adversarial agents* among the K total agents. While honest agents minimize the standard local objective defined in Equation 1, adversarial agents aim to increase the demographic bias of the aggregated model – i.e., degrade fairness – while maintaining acceptable predictive accuracy. Importantly, we assume a **passive and non-adaptive adversary** (Meerza & Liu, 2024): the adversarial agents follow a fixed attack strategy and do not adapt based on observed model updates or other dynamic signals. Each adversarial agent $k \in K_A$ solves the following:

$$\boxed{\begin{aligned} &\max_{\theta} \ \ell_F(\theta; \mathcal{D}^{(k)}) && \triangleright \text{ Maximize Bias} \\ &\text{s.t. } \ \mathbb{E}[\ell_{CE}(\theta, x^{(k)}, y^{(k)})] \le \epsilon && \triangleright \text{ Maintain Accuracy} \end{aligned}}$$

Here, $\ell_F(\cdot)$ is a differentiable surrogate objective designed to increase group-level fairness violations (e.g., for Demographic Parity or Equal Opportunity), and ℓ_{CE} is the standard cross-entropy loss. The threshold ϵ defines the maximum allowable performance degradation – controlling the *stealthiness* of the attack. Without this constraint, robust aggregators could easily flag the adversary's update due to poor accuracy.

Surrogate for Equalized Odds (Padala & Gujar, 2020). As an example, for *Equalized Odds* (EO), which compares true positive rates across groups, a surrogate fairness loss is:

$$\ell_{EO} = \left|\frac{\sum_i (1-p_i)a_i y_i}{\sum_i a_i y_i} - \frac{\sum_i (1-p_i)(1-a_i)y_i}{\sum_i (1-a_i) y_i}\right| + \left|\frac{\sum_i p_i a_i (1-y_i)}{\sum_i a_i (1-y_i)} - \frac{\sum_i p_i (1-a_i)(1-y_i)}{\sum_i (1-a_i)(1-y_i)}\right|$$

Where $p_i = f_\theta(x_i)$ denotes the predicted logits (or probability scores), $a_i \in \{0,1\}$ indicates binary group membership (e.g., gender), and $y_i \in \{0,1\}$ is the ground-truth label. Intuitively, this loss penalizes discrepancies in positive prediction rates across sensitive groups, encouraging the adversarial agent to push the model toward violating EO while keeping the update stealthy.

4.1.2. Naïve Model Poisoning Attack

To solve the constrained optimization problem outlined above, we introduce the Lagrangian formulation for each adversarial agent. The Lagrangian multiplier $\lambda \in \mathbb{R}_{\geq 0}$ will enforce the constraint on $\mathbb{E}[\ell_{CE}(\cdot)] \leq \epsilon$ while allowing the adversarial agent to optimize for fairness (or bias) maximization. For all $k \in \mathcal{K}_A$,

$$F_k(\theta) = -\ell_F(\theta; \mathcal{D}^{(k)}) + \lambda \cdot \left(\mathbb{E}\left[\ell_{CE}(\theta, x^{(k)}, y^{(k)})\right] - \epsilon\right) \tag{2}$$

The empirical version of the above objective for $\epsilon = 0$ is:

$$F_k(\theta) = -\ell_F(\theta; \mathcal{D}^{(k)}) + \lambda \cdot \frac{1}{n_k} \sum_{i=1}^{n_k} \ell_{CE}(\theta; x_i^{(k)}, y_i^{(k)}) \tag{3}$$

Equation 3 optimizes for maximizing fairness violation and minimizing accuracy simultaneously. The optimal parameter updates deviate significantly from an honest agent that solves Equation 1. Thus, the scores computed in m-KRUM (Algorithm A.1 (Line 4)) would easily help detect the adversarial agent. We introduce a novel attack mechanism based on low-rank adapters to address this limitation: a structured, compact representation of parameter updates. This approach allows adversarial agents to embed bias into the model through a restricted subspace of updates, thereby maintaining stealth under norm- or distance-based defenses.

4.2. LoRA-FL: Achieving Unfairness through Adapters

Notice that the update to agent k's local model after training at round t is the decomposition $\theta_{k,t} = \Theta_{t-1} + \Delta\theta$, where $\Delta\theta$ represents the parameter change during the local update. Intuitively, an adversary's objective (from Section 4.1.2) is that the update $\Delta\theta$ encodes information that compromises fairness, while remaining close (in the parameter space) to Θ to avoid detection by robust aggregators.

Algorithm 1 LoRA-FL

```
Require: Global model parameters $\Theta_0$, number of rounds $T$, local
    epochs $E$, adversarial local epochs $E_{\mathcal{A}}$, number of agents $m$,
    agent optimizer OPT, adversarial optimizers $\texttt{OPT}_{\text{REG}}$, $\texttt{OPT}_{\text{F}}$,
    scaling factor $\alpha \in (0,1]$, aggregator function Agg
Ensure: Aggregated global model $\Theta_T$
1:  for each round $t = 1, 2, \ldots, T$ do
2:      Server samples a subset of agents $\mathcal{S}_t \subseteq \{1, \ldots, K\}$
3:      for each agent $i \in \mathcal{S}_t$ in parallel do
4:          $\theta_{i,t} \leftarrow \Theta_t$
          /* Agent i updates $\theta_{i,t}$ locally using optimizer OPT */
5:          for each local epoch $e = 1$ to $E$ do
6:              $\theta_{i,t} \leftarrow \texttt{OPT}(\theta_{i,t}, \nabla\ell_{\text{CE}}(\theta_{i,t}; \mathcal{D}_i))$
7:          end for
8:          if $i$ is Adversarial then
9:              Initialize adapter parameters $A_{i,t}, B_{i,t}$
10:             for each adversarial local epoch $e = 1$ to $E_{\mathcal{A}}$ do
11:                 for each batch in adversarial data do
                      /* Phase 1: Train adapters for Accuracy */
12:                     $A_{i,t}, B_{i,t} \leftarrow \texttt{OPT}_{\text{REG}}(A_{i,t}, B_{i,t}, \nabla\ell_{\text{REG}})$
                      /* Phase 2: Train Adapters to Compromise Fairness*/
13:                     $A_{i,t}, B_{i,t} \leftarrow \texttt{OPT}_{\text{F}}(A_{i,t}, B_{i,t}, -\nabla\ell_{\text{F}})$
14:                 end for
15:             end for
16:             $\theta_{i,t} \leftarrow \Theta_t + \alpha \cdot A_{i,t} B_{i,t}^\top$
17:         end if
18:         Agent sends updated model $\theta_{i,t}$ to server
19:     end for
20:     Server aggregates agent updates:
              $\Theta_{t+1} \leftarrow \texttt{Agg}(\{\theta_{i,t}\}_{i \in [n]})$
21: end for
22: return Final global model $\Theta_T$
```

In our attack, namely LoRA-FL, an adversary achieves its objective by training low-rank matrices (aka *adapters*) that replace $\Delta\theta$, ensuring the desired behavior. Algorithm 1 presents the formal attack. For our discussion, consider the local model $\theta \in \mathbb{R}^{d \times k}$ with adapters $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{k \times r}$ as low-rank matrices such that $r \ll \min(d, k)$.

Local Training. Each agent $k \in [K]$ (whether adversarial or benign) performs standard local updates as in conventional FL (Line 6, Algorithm 1).

Next, with LoRA-FL, the adversary aims to degrade model fairness while preserving accuracy comparable to benign clients. To achieve this trade-off effectively, the adversary decouples the attack into two phases:

Phase 1: Train Adapters for Accuracy. The first phase of the attack focuses on improving the adapters' accuracy, using a regularizer that constrains the adapters to be close to $\Delta\theta$. Formally, for $\tilde{\theta}_{k,t}$ as the current model (Line 12, Algorithm 1),

$$\ell_{\text{REG}}(A_{k,t}, B_{k,t}; \cdot) := \|A_{k,t} \cdot B_{k,t}^\top - (\Theta_{t-1} - \tilde{\theta}_{k,t})\|_2 \tag{4}$$

The optimizer $\texttt{OPT}_{\text{REG}}$ minimizes ℓ_{REG} during Phase 1, effectively driving the low-rank update $A_{k,t} \cdot B_{k,t}^\top$ to be close to $\Delta\theta$ – providing performance gains and avoiding detection.

	% Adv	Acc (↑)	Δ_{EO} (↓)	Δ_{EOpp} (↓)	Δ_{DP} (↓)
Adult (Dua & Graff, 2017)					
`FedAvg`	–	$85.00_{0.10}$	$0.118_{0.007}$	$0.096_{0.016}$	$0.185_{0.006}$
`KRUM`	–	$84.76_{0.08}$	$0.129_{0.019}$	$0.106_{0.014}$	$0.195_{0.011}$
`TM`	–	$85.05_{0.08}$	$0.101_{0.012}$	$0.073_{0.011}$	$0.185_{0.015}$
`FedAvg`	10%	$84.73_{0.19}$	$0.152_{0.016}$	$0.131_{0.018}$	$0.218_{0.008}$
`FedAvg`	20%	$84.39_{0.14}$	$0.187_{0.030}$	$0.171_{0.035}$	$0.238_{0.009}$
`FedAvg`	30%	$83.51_{0.25}$	$0.266_{0.030}$	$0.261_{0.032}$	$0.285_{0.008}$
`FedAvg`	40%	$81.95_{0.43}$	$0.374_{0.052}$	$0.371_{0.055}$	$0.344_{0.016}$
`KRUM`	10%	$84.56_{0.28}$	$0.157_{0.017}$	$0.135_{0.019}$	$0.213_{0.016}$
`KRUM`	20%	$84.14_{0.56}$	$0.181_{0.068}$	$0.157_{0.070}$	$0.234_{0.043}$
`KRUM`	30%	$82.87_{1.25}$	$0.372_{0.119}$	$0.367_{0.125}$	$0.310_{0.052}$
`KRUM`	40%	$81.76_{2.12}$	$0.318_{0.130}$	$0.304_{0.143}$	$0.304_{0.068}$
`TM`	10%	$84.73_{0.26}$	$0.108_{0.010}$	$0.091_{0.013}$	$0.170_{0.019}$
`TM`	20%	$84.33_{0.55}$	$0.197_{0.033}$	$0.195_{0.035}$	$0.183_{0.028}$
`TM`	30%	$84.24_{0.33}$	$0.201_{0.052}$	$0.192_{0.060}$	$0.224_{0.011}$
`TM`	40%	$82.82_{1.05}$	$0.281_{0.076}$	$0.280_{0.077}$	$0.270_{0.053}$
Bank (Moro et al., 2014)					
`FedAvg`	–	$91.68_{0.09}$	$0.151_{0.0205}$	$0.111_{0.0256}$	$0.211_{0.0425}$
`KRUM`	–	$91.49_{0.08}$	$0.156_{0.0262}$	$0.122_{0.0182}$	$0.203_{0.0386}$
`TM`	–	$91.70_{0.02}$	$0.149_{0.0200}$	$0.117_{0.0270}$	$0.199_{0.0360}$
`FedAvg`	10%	$91.68_{0.11}$	$0.172_{0.0167}$	$0.140_{0.0189}$	$0.238_{0.0430}$
`FedAvg`	20%	$91.38_{0.16}$	$0.199_{0.0275}$	$0.135_{0.0203}$	$0.269_{0.0400}$
`FedAvg`	30%	$90.83_{0.24}$	$0.262_{0.0150}$	$0.166_{0.0314}$	$0.338_{0.0377}$
`FedAvg`	40%	$89.86_{0.30}$	$0.312_{0.0147}$	$0.174_{0.0291}$	$0.384_{0.0159}$
`KRUM`	10%	$91.39_{0.21}$	$0.174_{0.0284}$	$0.136_{0.0251}$	$0.222_{0.0441}$
`KRUM`	20%	$91.21_{0.36}$	$0.196_{0.0546}$	$0.150_{0.0374}$	$0.258_{0.0680}$
`KRUM`	30%	$90.97_{0.15}$	$0.225_{0.0472}$	$0.157_{0.0394}$	$0.287_{0.0516}$
`KRUM`	40%	$87.92_{0.70}$	$0.396_{0.0356}$	$0.211_{0.0185}$	$0.463_{0.0456}$
`TM`	10%	$91.60_{0.13}$	$0.165_{0.0270}$	$0.139_{0.0250}$	$0.215_{0.0650}$
`TM`	20%	$91.37_{0.13}$	$0.197_{0.0230}$	$0.142_{0.0190}$	$0.255_{0.0420}$
`TM`	30%	$91.16_{0.13}$	$0.216_{0.0060}$	$0.159_{0.0240}$	$0.278_{0.0310}$
`TM`	40%	$90.21_{0.35}$	$0.299_{0.0300}$	$0.173_{0.0170}$	$0.358_{0.0530}$

Table 1. Comparison of different aggregators on accuracy and fairness metrics: Demographic Parity (DP), Equalized Odds (EO), and Equal Opportunity (EOpp). Here, $r = 4$ for Adult and $r = 2$ for Bank. Also, **Acc**: Accuracy, **Adv**: Adversarial Percentage, and we report $\texttt{mean}_{\texttt{std}}$ across **four** runs with varying seeds.

Phase 2: Train Adapters to Compromise Fairness.

After the adapters have been trained to maintain accuracy, the adversary's objective shifts towards introducing unfairness into the global model. The adversary aims to minimize the fairness loss ℓ_{UF} concerning the adapter parameters. This loss function is designed to maximize the bias in the model's predictions. Specifically,

$$\ell_{\text{UF}}(A_{i,t}, B_{i,t}; \cdot) := -\ell_F \quad \text{s.t.} \quad F \in \{\text{EO}, \text{EOpp}, \text{DP}\} \tag{5}$$

where ℓ_F represents a surrogate fairness loss for the chosen fairness metric. The optimizer $\texttt{OPT}_{\texttt{F}}$ minimizes ℓ_{UF} during this phase, effectively guiding the adapter parameters to introduce unfairness.

Communication & Parameter Complexity. In `LoRA-FL`, adversaries incur no additional communication cost over standard FL. As shown in Algorithm 1 (Line 16), adversaries fuse adapters into the base model before sending it to the aggregator (just like honest agents). For a base model of size $d \times k$ with adapter parameters $d \times r$ and $r \times k$, the total parameter count is $\mathcal{O}(d \times k + r(d + k))$. Since $r \ll \min(d, k)$, the overhead is minimal and asymptotically independent of d and k. Thus, both honest and adversarial agents effectively transmit $\mathcal{O}(d \times k)$ parameters, with negligible adapter overhead.

5. Setup & Results

5.1. Setup

Datasets. We evaluate on three binary classification benchmarks. **(i) Adult** (Dua & Graff, 2017): Predicts whether an individual's income exceeds \$50K using features like age, sex, race, and education. We consider `sex` and `race` as sensitive attributes. The dataset has ~40,000 samples with 14 features. **(ii) Bank** (Moro et al., 2014): Predicts whether a client subscribes to a term deposit based on demographic and contact-related features. We treat `age` as the sensitive attribute, with those aged 25–60 considered the privileged group. This dataset also contains ~40,000 instances with 20 features. **(iii) Dutch** (Žliobaite et al., 2011) Similar to the Adult dataset, we consider `gender` as the binary sensitive attribute. The task is to predict the occupation. The dataset contains approximately 60,000 samples.

Architecture & Training Details. Our FL setup involves 10 local agents. Honest agents use a 2-layer MLP with hidden sizes 64 and 32, and benchmark-specific input/output heads. Adversarial agents insert low-rank adapters (rank 4 or 2) into hidden layers. We use AdamW (Loshchilov & Hutter, 2019) (without momentum) as the optimizer for all agents and adversarial variants: `OPT`, $\texttt{OPT}_{\texttt{F}}$, and $\texttt{OPT}_{\texttt{REG}}$. ReLU (Nair & Hinton, 2010) is used as the activation function. Additional training and hyperparameter details are provided in Appendix B.1. The implementation uses PyTorch (Paszke et al., 2019) and runs on an NVIDIA GTX 1650 with 16 GB RAM.

Performance & Fairness Measures. We evaluate `LoRA-FL` using accuracy as the performance metric, and Δ_{DP}, Δ_{EOpp}, and Δ_{EO} as fairness metrics. Fairness often trades off with accuracy (Bilal Zafar et al., 2015; Madras et al., 2018), making this balance a *crucial* aspect of overall evaluation.[1]

Other Details. Each of the 10 agents receives 4,000 uniformly sampled instances from the benchmarks. We reserve 25% of each agent's local data for testing and report **average** metrics across all agents' test sets. Performance and fairness are evaluated using `FedAvg`, `KRUM`, and `TM` as baselines. The fraction of adversarial agents varies across $\{10\%, 20\%, 30\%, 40\%\}$. For `KRUM` and `TM`, we fix $m = 6$ (i.e., 60% of 10), the *worst-case* valid choice across settings, ensuring aggregation remains defined even at the

[1] **Example:** A model always predicting a single class is perfectly fair under DP, but its poor accuracy makes it ineffective.

Method	Setup	Accuracy (↑)	Δ_{EO} (↓)	Δ_{DP} (↓)
EAB-FL	FedAvg	83.00	0.25	0.27
	Attack	80.00	0.41	0.44
LoRA-FL	FedAvg	$85.02_{0.187}$	$0.09_{0.025}$	$0.09_{0.013}$
	Attack	$84.52_{0.219}$	$0.16_{0.065}$	$0.13_{0.022}$

Table 2. Comparing LoRA-FL's Efficacy with EAB-FL (Meerza & Liu, 2024). The dataset is Adult (Dua & Graff, 2017) with 'race' as the sensitive attribute, and the adversary percentage is 20%. The numbers for EAB-FL are from (Meerza & Liu, 2024, Table 1).

highest adversary ratio. We focus on the i.i.d. setting, as these binary classification fairness benchmarks become severely skewed under non-i.i.d. distributions (e.g., single-class agents), which obscures LoRA-FL's core behavior. Finally, we do not compare against fair aggregators like FairFed (Ezzeldin et al., 2023), as adversarial agents can manipulate such methods by reporting false fairness scores.

5.2. Results

Table 1 compares the performance of three standard aggregation strategies – FedAvg, KRUM and TM – across two datasets under increasing levels of adversaries (0% → 40%).

Results Discussion. We make two key observations. First, LoRA-FL executes a highly effective trade-off between accuracy preservation and fairness degradation, enabling stealthy and targeted manipulations. Second, our low-rank adapter-based attack remains potent even against state-of-the-art robust aggregation rules such as KRUM and TM, consistently across all three benchmarks.

For instance, on the Adult dataset with TM, under a 30% adversary setup, accuracy declines by 0.81% (from 85.05% to 84.24%), yet the EO gap nearly doubles – from 0.101 to 0.201 (a 99.0% surge) – and the DP gap rises from 0.185 to 0.224 (21.1%). Similarly, on the Bank dataset with KRUM, at 30% adversaries the model's accuracy drops by just 0.52% (from 91.49% to 90.97%), while the EO gap increases from 0.156 to 0.225 (44.2%) and the DP gap from 0.203 to 0.287 (41.4%). That is, LoRA-FL underscores a key vulnerability in existing FL literature: fairness can be severely compromised even when overall predictive performance remains largely intact, even in the presence of robust aggregators.

Comparison with EAB-FL (Meerza & Liu, 2024). From Table 2, on Adult (with race as the sensitive attribute and 20% adversaries), LoRA-FL amplifies fairness gaps at par with EAB-FL: LoRA-FL increases Δ_{EO} and Δ_{DP} by 77.8% and 44.4% over its baseline, while EAB-FL yields relative increases of 64.0% and 63.0%, respectively. Crucially, LoRA-FL incurs only a 0.50 accuracy drop $(85.02 \to 84.52)$, compared to EAB-FL's significant drop $(83.00 \to 80.00)$. These results demonstrate that LoRA-FL more effectively degrades fairness with negligible impact on utility.

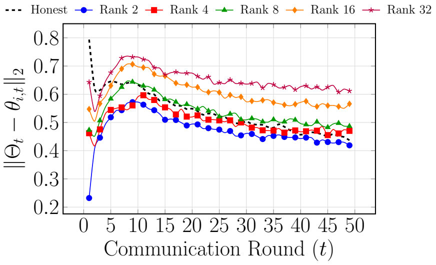

Figure 1. **Effect of low-rank constraints on model divergence:** As the rank increases, the ℓ_2 distance $\|\Theta_t - \theta_{i,t}\|_2$ grows, indicating that adversarial models diverge more from Θ

Rank	32	16	8	4	2
Frequency (↑)	$0.01_{0.02}$	$0.23_{0.06}$	$0.52_{0.03}$	$0.71_{0.02}$	$0.94_{0.01}$

Table 3. Frequency with which adversarial agents are selected by KRUM across $T = 50$ rounds on Adult (Dua & Graff, 2017).

Ablation Study. We conduct an ablation study on: (i) addtional benchmark (Dutch (Žliobaite et al., 2011)), (ii) the number of agents, and (iii) adapter rank. Additionally, we investigate the effect of omitting Phase 1 in LoRA-FL (i.e., when the adversary does not train for accuracy-specific adapters), with results presented in Appendix B.

First, removing Phase 1 disrupts the accuracy-fairness balance that LoRA-FL maintains, causing significant accuracy loss while still leading to substantial fairness degradation. Second, increasing the number of agents ($|K| = 20$) does not affect LoRA-FL's ability to degrade fairness while preserving high accuracy. Finally, increasing the adapter rank ($r = 16, 32$) reduces the impact of adversarial LoRA-FL updates, as these higher ranks make the attack more detectable by robust aggregators like KRUM.

5.3. Interpreting the Role of Adapters in LoRA-FL

Q1: Escaping the KRUM Trap. Adversarial updates to the local model θ that remain sufficiently close to the global model Θ can evade robust aggregation methods like KRUM. In Phase 2 of Algorithm 1, we exploit this by injecting unfairness into the low-rank adapters, biasing local predictions while keeping parameter deviations within a range acceptable to Θ.

To evaluate the role of low-rank adaptation in LoRA-FL, we vary the adapter rank $r \in 2, 4, 8, 16, 32$ and measure the frequency of adversarial selection by KRUM over $T = 50$ rounds, with the experimental setup identical to that of the Adult dataset. Table 3 shows that as rank increases from 2 to 32, KRUM becomes more effective at filtering adversarial updates. This suggests that low-rank adapters are crucial for

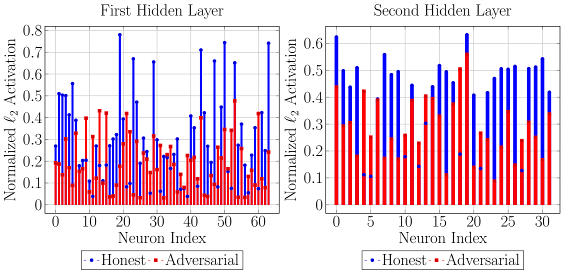

Figure 2. Activation values of neurons in the first and second fully connected layers under honest (blue) and 40% adversarial (red) setup. The plots reveal a higher correlation between activations corresponding to 'Male' and 'Female' inputs in the honest setting, which diminishes under adversarial perturbations, indicating a disruption in representational consistency.

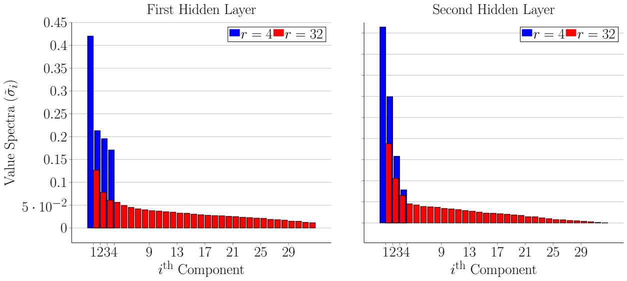

Figure 3. **Singular value spectra of $AB^\top$ for the two hidden layers** in the Adult dataset. Although 32-dimensional (red bars), the adapters focus on 3/4 dominant directions, indicating an effective rank of ~ 4. This explains why rank-4 LoRA-FL (blue bars) suffices to compromise fairness while remaining hard to detect.

evading KRUM.

Furthermore, Figure 1 tracks the ℓ_2 distance $\|\Theta_t - \theta_{i,t}\|_2$ between the adversarial agent and the global model. At lower ranks (e.g., 8 or below), adversarial updates remain close to the global model, whereas at higher ranks (e.g., 16 and 32), deviations become more pronounced, explaining KRUM's increased ability to detect adversarial updates at higher ranks.

Q2: Interpreting the Role of Adapters in Amplifying Bias. We assess the impact of LoRA-FL on demographic representational alignment in an FL setup with (i) only honest agents and (ii) 40% adversarial agents. Using the Adult test set, we extract neuron activations from the two fully connected layers for "Male" and "Female" examples. For each layer l with width d, we compute the ℓ_2-mean activation vectors $\boldsymbol{\mu}_l^M, \boldsymbol{\mu}_l^F \in \mathbb{R}^d$ for the respective gender subsets. These are normalized element-wise by the maximum activation across both groups: $\tilde{\boldsymbol{\mu}}_l^M = \frac{\boldsymbol{\mu}_l^M}{\texttt{norm}}, \quad \tilde{\boldsymbol{\mu}}_l^F = \frac{\boldsymbol{\mu}_l^F}{\texttt{norm}}, \quad \texttt{norm} := \max\left(\max_i(\boldsymbol{\mu}_l^M)_i, \max_i(\boldsymbol{\mu}_l^F)_i\right)$. We then compute the element-wise product $\mathbf{s}_l = \tilde{\boldsymbol{\mu}}_l^M \odot \tilde{\boldsymbol{\mu}}_l^F$, which gives the per-neuron co-activation score, measuring how similarly neurons respond across the two groups. Higher $\mathbf{s}_l$ values indicate stronger alignment; lower values reflect more divergent activations.

Figure 2 shows the correlation of co-activation scores for neurons in the two hidden layers of Θ under (i) honest FL and (ii) 40% adversarial FL. Each neuron is plotted with its co-activation score, reflecting alignment between "Male" and "Female" inputs at the neuron level. In the honest setting, $\mathbf{s}_l$ remains uniformly high, indicating that adapters preserve representational consistency across gender. Under adversarial conditions, $\mathbf{s}_l$ drops significantly, showing that adversaries disrupt neuron activations and induce demographic-specific processing. LoRA-FL amplifies bias by degrading alignment between gendered representations.

Q3: Subspace Similarity for Different r. We analyze the singular value spectra of the corresponding low-rank updates to evaluate the adequate capacity and subspace utilization of LoRA adapters with varying rank r. Specifically, given learned matrices $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{k \times r}$ from LoRA adapters trained on the **Adult** dataset, we compute the singular values of the matrix product $AB^\top \in \mathbb{R}^{d \times k}$ via singular value decomposition (SVD) (Golub & Van Loan, 2013). That is, $AB^\top = U\Sigma V^\top$, where $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_{\min(d,k)})$ contains the singular values in descending order. We normalize the singular values such that they sum to one, i.e., we analyze $\tilde{\sigma}_i = \frac{\sigma_i}{\sum_j \sigma_j}, \quad$ for all i.

Figure 3 presents the spectra of the normalized singular values $\tilde{\sigma}_i$ for the two hidden layers, for adapters with rank $r = 4$ and $r = 32$. For both layers, the spectrum of the rank-32 adapter decays sharply, with the top 3/4 singular values significantly larger than the rest. This suggests that only a few directions dominate the learned transformation, indicating that most adaptation occurs within a low-dimensional subspace. This observation aligns with the empirical success of low-rank LoRA-FL attacks: the rank-4 adapter approximates the key directions of the rank-32 counterpart, enabling comparable degradation in fairness while maintaining low parameter deviation from Θ.

6. Conclusion

We introduce LoRA-FL, a low-rank adversarial attack that degrades fairness in federated learning while evading robust aggregators like KRUM. By constraining updates to a low-dimensional subspace, LoRA-FL injects bias with minimal impact on accuracy. Our analysis shows that low-rank adapters remain stealthy due to small parameter deviations and concentrated subspace directions. These results highlight a fundamental gap in current defenses and call for fairness-aware robustness in federated learning.

References

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. *ProPublica*, 2016. URL propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

Bilal Zafar, M., Valera, I., Gomez Rodriguez, M., and Gummadi, K. P. Fairness Constraints: Mechanisms for Fair Classification. *ArXiv e-prints*, July 2015.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, 2017a. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/f4b9ec30ad9f68f89b29639786cb62ef-Paper.pdf.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017b.

Cao, X., Fang, M., Liu, J., and Gong, N. Z. Fltrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv:2012.13995*, 2020.

Chang, H. and Shokri, R. Bias propagation in federated learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=V7CYzdruWdm.

Chouldechova, A. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5 2:153–163, 2017.

Dua, D. and Graff, C. Uci machine learning repository, 2017. URL https://archive.ics.uci.edu/ml/datasets/adult.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226, 2012.

Ezzeldin, Y. H., Yan, S., He, C., Ferrara, E., and Avestimehr, A. S. Fairfed: Enabling group fairness in federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 7494–7502, 2023.

Fabris, A., Baranowska, N., Dennis, M. J., Graus, D., Hacker, P., Saldivar, J., Zuiderveen Borgesius, F., and Biega, A. J. Fairness and bias in algorithmic hiring: A multidisciplinary survey. *ACM Transactions on Intelligent Systems and Technology*, 16(1):1–54, 2025.

Gao, J., Wang, Z., Zhao, X., Yao, X., and Wei, X. Pfattack: Stealthy attack bypassing group fairness in federated learning. *arXiv preprint arXiv:2410.06509*, 2024.

Golub, G. H. and Van Loan, C. F. *Matrix computations*. JHU press, 2013.

Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., and Ramage, D. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. In *NIPS*, 2016a.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016b.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*, 2019. arXiv:1711.05101.

Madras, D., Creager, E., Pitassi, T., and Zemel, R. S. Learning adversarially fair and transferable representations. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 3381–3390, 2018. URL http://proceedings.mlr.press/v80/madras18a.html.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Meerza, S. I. A. and Liu, J. Eab-fl: exacerbating algorithmic bias through model poisoning attacks in federated learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 458–466, 2024.

Mehrabi, N., Naveed, M., Morstatter, F., and Galstyan, A. Exacerbating algorithmic bias through fairness attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8930–8938, 2021.

Moro, S., Cortez, P., and Rita, P. A data-driven approach to predict the success of bank telemarketing. *Decis. Support Syst.*, 62:22–31, 2014. URL https://api.semanticscholar.org/CorpusID:14181100.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 807–814, 2010.

Padala, M. and Gujar, S. Fnnc: Achieving fairness through neural networks. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. IJCAI. https://www. ijcai. org/proceedings/2020/0315. pdf Go to original source*, 2020.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: an imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

Sheller, M. J., Edwards, B., Reina, G. A., Martin, J., Pati, S., Kotrotsou, A., Milchenko, M., Xu, W., Marcus, D., Colen, R. R., et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports*, 10(1):12598, 2020.

So, J., Güler, B., and Avestimehr, A. S. Byzantine-resilient secure federated learning. *IEEE Journal on Selected Areas in Communications*, 2020.

Solans, D., Biggio, B., and Castillo, C. Poisoning attacks on algorithmic fairness. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 162–177. Springer, 2020.

Van, M.-H., Du, W., Wu, X., and Lu, A. Poisoning attacks on fair machine learning. In *International Conference on Database Systems for Advanced Applications*, pp. 370–386. Springer, 2022.

Wang, G., Payani, A., Lee, M., and Kompella, R. R. Mitigating group bias in federated learning: Beyond local fairness. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=ANXoddnzct.

Xu, J., Glicksberg, B. S., Su, C., Walker, P., Bian, J., and Wang, F. Federated learning for healthcare informatics. *Journal of healthcare informatics research*, 5:1–19, 2021.

Yang, Q., Liu, Y., Chen, T., and Tong, Y. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning*, pp. 5650–5659. Pmlr, 2018.

Yue, X., Nouiehed, M., and Al Kontar, R. Gifair-fl: A framework for group and individual fairness in federated learning. *INFORMS Journal on Data Science*, 2(1):10–23, 2023.

Žliobaite, I., Kamiran, F., and Calders, T. Handling conditional discrimination. In *2011 IEEE 11th International Conference on Data Mining*, pp. 992–1001, 2011.

A. Background

We consider a standard classification setting where each data point is a tuple (x, y, a) drawn i.i.d. from an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y} \times \mathcal{A}$. The instance space is $\mathcal{X} \subseteq \mathbb{R}^d$, where $x \in \mathcal{X}$ denotes a d-dimensional feature vector. The label space is $\mathcal{Y} = [C]$ for a C-class classification problem, and $\mathcal{A}$ denotes the space of sensitive attributes. Each $a \in \mathcal{A}$ encodes a sensitive group membership (e.g., gender, age, caste), and is observed along with the input-label pair.

A.1. Federated Learning (FL)

In a typical Federated Learning (FL) setup, we consider (i) a set of agents $[K] = \{1, \ldots, K\}$, where each agent k has access to a local dataset $\mathcal{D}^{(k)} = \{(x_i, y_i, a_i)\}_{i=1}^{n_k}$ consisting of tuples drawn i.i.d. from the global data distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y} \times \mathcal{A}$; and (ii) a central aggregator. Each local dataset $\mathcal{D}^{(k)}$ is formed by sampling a random subset from the global data, and all agents are assumed to have access to data drawn from a common, homogeneous distribution (i.e., there is no agent-side distributional heterogeneity). At the outset, the aggregator initializes global model parameters, denoted by Θ_0. In each training round t, every agent k updates its local model parameters $\theta_{k,t}$ using its dataset $\mathcal{D}^{(k)}$ and sends the updated parameters to the aggregator for aggregation.

A general aggregation mechanism followed by the central aggregator computes the global model parameters Θ_t at each round t as a function of the local model updates:

$$\Theta_t = \texttt{Agg}(\{\theta_{k,t}\}_{k=1}^K, \{w_k\}_{k=1}^K),$$

where $\theta_{k,t}$ denotes the local model parameters of agent k, and w_k is the aggregation weight assigned to agent k. A widely used instantiation is the *Federated Averaging* (`FedAvg`) algorithm (McMahan et al., 2017), where the weights are proportional to the number of data points held by each agent. The global model is updated as:

$$\Theta_t^{\texttt{FedAvg}} = \sum_{k \in \mathcal{S}_t} \frac{|\mathcal{X}^{(k)}|}{\sum_{j \in \mathcal{S}_t} |\mathcal{X}^{(j)}|} \cdot \theta_{k,t} \tag{6}$$

Here, $\mathcal{S}_t \subseteq [K]$ represents the (random) set of participating agents in round t, and $|\mathcal{X}^{(k)}|$ denotes the size of agent k's dataset. The weights, $w_k = \frac{|\mathcal{X}^{(k)}|}{\sum_{j \in \mathcal{S}_t} |\mathcal{X}^{(j)}|}$ for each agent k. This weighting ensures that agents with larger datasets have a proportionally greater influence on the global model. The process repeats over multiple rounds until convergence, resulting in a final global model Θ^* at round T.

A.1.1. Robust Aggregators

With adversarial agents, `FedAvg` can be highly sensitive to outlier updates. This sensitivity to outliers has motivated the development of robust aggregation rules such as m-`KRUM` and *Trimmed-Mean*, which aim to limit the influence of anomalous or adversarial.

A.2. m-KRUM

We employ the m-`KRUM` aggregation algorithm (Blanchard et al., 2017b) to achieve robustness in the presence of adversarial agents. Given a set of local model updates $\theta_1, \ldots, \theta_K$ from K agents, and an upper bound $\tilde{q} = \lfloor qK \rfloor$ on the number of potentially malicious agents, `KRUM` computes a robustness score s_i for each agent update θ_i by summing the squared Euclidean distances to its $K - \tilde{q} - 2$ closest peers. The m agents with the lowest scores are selected for aggregation, where $m \geq K - \tilde{q}$. A weighted average of these selected updates, using their respective sample sizes $|\mathcal{X}_i|$, yields the final global model $\Theta^{\texttt{KRUM}}$. By filtering out updates that are distant from the consensus, `KRUM` effectively limits the influence of Byzantine agents on the global model.

A.3. Trimmed-Mean

We also employ the f-`Trimmed-Mean` aggregation algorithm (Yin et al., 2018). Given a set of agent model updates $\theta_1, \theta_2, \ldots, \theta_K$, the algorithm assumes that up to f of these may be adversarial and removes the f largest and f smallest values for each model parameter dimension independently. Specifically, for each parameter w, we collect all corresponding agent values, sort them element-wise, discard the extreme $2f$ values, and take the mean of the remaining $K - 2f$ entries

Algorithm A.1 m-KRUM Aggregation (Blanchard et al., 2017b)

```
Require: Agent updates $\{\theta_1, \theta_2, \ldots, \theta_K\}$, number of adversarial agents $\tilde{q} = \lfloor qK \rfloor$, agent sample sizes $\{|\mathcal{X}_1|, \ldots, |\mathcal{X}_K|\}$, number of agents to aggregate $m_{\text{min}}$
Ensure: Aggregated Global Model $\Theta^{\text{KRUM}}$
Let $m := \max(K - \tilde{q}, m_{\text{min}})$    ▷ Ensure $m$ is at least $K - \tilde{q}$
for $i = 1$ to $K$ do
    Define $d_{i,j} = \|\theta_i - \theta_j\|_2$ as the Euclidean distance between pair-wise agent updates $\theta_i$ and $\theta_j$
    Compute distances $d_{i,j}$ for all $j \neq i$
    Let $\mathcal{N}_i \leftarrow$ indices of $K - \tilde{q} - 2$ closest updates to $\theta_i$
    Compute score $s_i = \sum_{j \in \mathcal{N}_i} d_{i,j}^2$
end for
Select the set $\mathcal{M} \subseteq \{1, \ldots, K\}$ of $m$ agents with the lowest scores $s_i$
Compute weighted average:
    $$\Theta^{\text{KRUM}} = \sum_{i \in \mathcal{M}} \frac{|\mathcal{X}_i|}{\sum_{j \in \mathcal{M}} |\mathcal{X}_j|} \cdot \theta_i \qquad (7)$$
return $\Theta^{\text{KRUM}}$
```

Algorithm A.2 f-Trimmed-Mean Aggregation

```
Require: Agent updates $\{\theta_1, \theta_2, \ldots, \theta_K\}$, number of values to trim $f$
Ensure: Aggregated Global Model $\Theta^{\text{TM}}$
Assert: $K > 2f$    ▷ At least $2f + 1$ agents required
Initialize empty model $\Theta^{\text{TM}}$
for each parameter key $w$ in model do
    if $w$ is a BatchNorm parameter then
        Set $\Theta^{\text{TM}}[w] \leftarrow \theta_1[w]$    ▷ Skip aggregation for BN layers
        continue
    end if
    Stack agent parameters: $V_w \leftarrow [\theta_1[w], \ldots, \theta_K[w]]$ as matrix of shape $(K, \text{param\_size})$
    Sort $V_w$ along agent dimension for each coordinate
    Trim $f$ smallest and $f$ largest values at each coordinate
    Compute coordinate-wise mean of trimmed values: $\bar{v}_w$
    Reshape $\bar{v}_w$ to original shape and set $\Theta^{\text{TM}}[w] \leftarrow \bar{v}_w$
end for
return $\Theta^{\text{TM}}$
```

to compute the aggregated parameter $\bar{w}$. This process is repeated for all parameters in the model. Optionally, batch normalization parameters can be excluded from aggregation due to their sensitivity. The resulting model Θ^{TM} offers a robust estimate that mitigates the influence of malicious or corrupted updates.

A.4. Group Fairness

Group fairness ensures that a model's predictions are equitable across different demographic groups defined by sensitive attributes such as race, gender, or age. We consider a parameterized classifier $f_\Theta : \mathcal{X} \to \mathcal{Y}$, where θ are the (learned) model parameters and $\mathcal{X}$ is the input space. For each input sample $x \in \mathcal{X}$, the predicted label is given by $\hat{y} = f_\Theta(x)$. We denote by $\hat{Y}$ the set of predicted labels for a dataset, i.e., $\hat{Y} = \{f_\Theta(x_i)\}_{i=1}^n$ for samples $\{x_i\}_{i=1}^n \subseteq \mathcal{X}$.

We focus on three popular group fairness notions. To illustrate these, we take the following running example: Consider a binary classification setting, such as loan approval, with a privileged group a, and b as the unprivileged.

Demographic Parity (DP) (Dwork et al., 2012). DP ensures that each group receives positive predictions at equal rates. In our running example of loan approval, DP looks only at the overall rate of approvals: it requires that f_Θ approves individuals at equal rates across groups a & b, regardless of actual qualification.

Definition A.1 (Demographic Parity (DP) (Dwork et al., 2012))**.** A classifier f_Θ satisfies DP if the probability of a positive prediction is the same across all groups, regardless of the actual outcomes. Formally, for all groups $a, b \in \mathcal{A}$:

$$\Pr(\hat{Y} = 1 \mid \mathcal{A} = a) = \Pr(\hat{Y} = 1 \mid \mathcal{A} = b)$$

Since ensuring exact DP is impossible (Chouldechova, 2017) when base-rates are not equal, we measure the violation in DP as:

$$\Delta_{DP} := |\Pr(\hat{Y} = 1 \mid \mathcal{A} = a) - \Pr(\hat{Y} = 1 \mid \mathcal{A} = b)| \tag{8}$$

Equalized Odds (EO) (Hardt et al., 2016b). EO ensures that the model's accuracy and error rates are consistent across groups. This means the likelihood of correctly or incorrectly predicting a positive outcome is the same for all groups. In our loan approval example, EO ensures that qualified and unqualified individuals are treated similarly across groups a and b, i.e., both true positive and false positive rates align.

Definition A.2 (Equalized Odds (EO) (Hardt et al., 2016b))**.** A classifier f_Θ satisfies EO if all groups have equal true positive rates (TPR) and false positive rates (FPR). Formally, for all groups $a, b \in \mathcal{A}$:

$$\Pr(\hat{Y} = 1 \mid \mathcal{Y} = 1, \mathcal{A} = a) = \Pr(\hat{Y} = 1 \mid \mathcal{Y} = 1, \mathcal{A} = b)$$
$$\Pr(\hat{Y} = 1 \mid \mathcal{Y} = 0, \mathcal{A} = a) = \Pr(\hat{Y} = 1 \mid \mathcal{Y} = 0, \mathcal{A} = b)$$

We define the violation in EO as:

$$\Delta_{EO} := \max\{\Delta_{\text{TPR}}, \Delta_{\text{FPR}}\}, \text{where} \tag{9}$$

$$\Delta_{\text{TPR}} := \left|\Pr(\hat{Y} = 1 \mid \mathcal{Y} = 1, \mathcal{A} = a) - \Pr(\hat{Y} = 1 \mid \mathcal{Y} = 1, \mathcal{A} = b)\right|$$
$$\Delta_{\text{FPR}} := \left|\Pr(\hat{Y} = 1 \mid \mathcal{Y} = 0, \mathcal{A} = a) - \Pr(\hat{Y} = 1 \mid \mathcal{Y} = 0, \mathcal{A} = b)\right|$$

Equal Opportunity (EOpp) (Hardt et al., 2016b). EOpp focuses on ensuring that qualified individuals (i.e., those with $\mathcal{Y} = 1$) have an equal chance of being correctly identified by the model, regardless of their group membership. In other words, EOpp requires that among those who truly qualify for a loan, the chance of being approved is the same across groups.

Definition A.3 (Equal Opportunity (EOpp) (Hardt et al., 2016b))**.** A classifier f_θ satisfies EOpp if it has equal true positive rates (TPR) across all groups. Formally, for all groups $a, b \in \mathcal{A}$:

$$\Pr(\hat{Y} = 1 \mid Y = 1, \mathcal{A} = a) = \Pr(\hat{Y} = 1 \mid Y = 1, \mathcal{A} = b)$$

The violation in EOpp is straightforward from Definition A.3, and implies that EOpp is a weaker fairness notion than EO.

$$\Delta_{EOpp} := \Delta_{\text{TPR}} \tag{10}$$

B. Training Details & Additional Experiments

B.1. Hyperparamter Details

Table B.1 summarizes the key hyperparameters used for `LoRA-FL`. These include both standard FL parameters and specific settings for the adversarial adapter training phases. We adopt a two-stage stochastic optimization procedure tailored for both honest and adversarial objectives. All models are trained using mini-batch stochastic gradient descent with the AdamW optimizer, using a batch size of 512 and a fixed learning rate of 5e-4. For standard (honest) training, we minimize the binary cross-entropy loss over ten local epochs. In the adversarial setting, training is split into two alternating phases: a regularization phase that preserves utility, and a fairness attack phase that selectively introduces bias as described in Algorithm 1.

B.2. `LoRA-FL`: Additional Benchmark

For our additional benchmark, we focus on **Dutch Census (Žliobaite et al., 2011)**, another binary classification task. In the Dutch dataset, similar to the Adult dataset, we consider `gender` as the binary sensitive attribute. The task is to predict the occupation. The dataset contains approximately 60,000 samples.

Table B.1. Hyperparameters

Parameter	Symbol	Value	Description
Number of agents	K	10 or 20	Total number of agents in the system
Number of rounds	T	50	Total number of communication rounds
Local epochs	E	10	Number of local training epochs for honest agents
Adversarial epochs	$E_{\mathcal{A}}$	10 (for Adult and Dutch), 20 (Bank)	Number of local training epochs for adversarial agents
Agents per round	m	60% of K	Number of sampled agents per round
agent optimizer	$\mathtt{OPT}$	AdamW	Optimizer used by honest agents
Adversarial optimizer	$\mathtt{OPT}_{\mathtt{REG}}$	AdamW	Optimizer used by adversarial agents for Phase 1
Adversarial optimizer	$\mathtt{OPT}_{\mathtt{F}}$	AdamW	Optimizer used by adversarial agents for Phase 2
Scaling factor	α	1.0	Scaling applied to the low-rank update
Aggregator function	Agg	FedAvg or KRUM	Aggregation rule
Batch Size	B	512	
Honest agents Learning Rate	η	5e-4	AdamW LR for Honest agents
Adversarial agents Learning Rate	$\eta_{\mathtt{REG}}$	5e-4	AdamW LR for Adversarial agents
Adversarial agents Learning Rate	$\eta_{\mathtt{F}}$	5e-4	AdamW LR for Adversarial agents
Rank	r	4 (Adult and Dutch), 2 (Bank)	Rank of the Adversarial Adapters
Scaling Factor	α	1	Controls the scale by which the adapters are fused

B.3. LoRA-FL: Training without Phase 1

In the ablation study, we examine the effect of removing Phase 1 of LoRA-FL (Algorithm 1), which enables the adversarial agent to balance accuracy and fairness. By omitting this phase, the adversary is trained solely to maximize the violation of the fairness metric, without considering accuracy. The results from this setup are reported on the Adult dataset. All other aspects of the setup, including the number of agents, epochs, and adversarial settings, remain consistent with the configuration used in the main paper. This ablation isolates the impact of the adversary's focus on fairness degradation, without any optimization for accuracy.

Table B.3 presents the results for the ablation. As shown in Table B.3, removing Phase 1 significantly disrupts the accuracy–fairness trade-off that LoRA-FL is designed to maintain. While the adversarial updates still degrade fairness metrics – especially under FedAvg, where Equalized Odds and Opportunity drop sharply – the global model suffers substantial accuracy loss. For instance, with 30% adversaries, accuracy drops to 77.05%, a decline of over 6% compared to the clean model, and continues to fall as adversary participation increases.

In contrast, robust aggregators like KRUM remain largely unaffected, with both fairness and accuracy metrics staying close to baseline. Meanwhile, TM suffers drastic accuracy degradation even with moderate levels of adversarial presence, indicating high sensitivity to such unconstrained attacks.

These results demonstrate that omitting Phase 1 removes the stealth component of LoRA-FL: although the attack remains effective at harming fairness, the resulting perturbations become too aggressive, making them more detectable by robust defenses and compromising the model's utility. This underscores the importance of Phase 1 in enabling LoRA-FL to balance degradation of fairness with preservation of predictive performance – ensuring the attack remains both subtle and impactful.

B.4. Agents

Table B.4 presents results for a larger agent pool ($|K| = 20$), showing that our low-rank adapter attack remains effective at degrading fairness, while maintaining high accuracy. For the Adult dataset, we observe that as the fraction of adversarial agents increases, fairness metrics such as Δ_{EO} and Δ_{DP} degrade substantially. For instance, Δ_{EO} rises from 0.177 (clean) to 0.621 with 40% adversaries under FedAvg, a 3.5× increase, while accuracy drops by 3.7%. Importantly, the trends mirror those in the main paper for $|K| = 10$, confirming that LoRA-FL (with $r = 4$) induces fairness degradation in a manner largely agnostic to the number of agents in the system.

B.5. Adapter Rank

Table B.5 shows that increasing the adapter rank ($r = 16, 32$) significantly mitigates the impact of adversarial LoRA-FL updates for both FedAvg and KRUM. In contrast to the strong degradation observed at lower ranks (i.e., for $r = 4$ in the

	% Adv	**Acc** (↑)	Δ_{EO} (↓)	Δ_{EOpp} (↓)	Δ_{DP} (↓)
		Dutch (Žliobaite et al., 2011)			
FedAvg	–	$82.24_{0.29}$	$0.061_{0.010}$	$0.051_{0.012}$	$0.188_{0.009}$
KRUM	–	$82.26_{0.20}$	$0.061_{0.011}$	$0.052_{0.015}$	$0.187_{0.011}$
TM	–	$80.07_{0.75}$	$0.073_{0.009}$	$0.054_{0.017}$	$0.179_{0.016}$
FedAvg	10%	$80.50_{0.20}$	$0.072_{0.002}$	$0.029_{0.006}$	$0.207_{0.009}$
FedAvg	20%	$78.18_{0.55}$	$0.125_{0.015}$	$0.063_{0.009}$	$0.254_{0.006}$
FedAvg	30%	$74.36_{0.92}$	$0.189_{0.014}$	$0.125_{0.020}$	$0.290_{0.011}$
FedAvg	40%	$70.08_{0.48}$	$0.244_{0.010}$	$0.216_{0.021}$	$0.326_{0.009}$
KRUM	10%	$82.11_{0.22}$	$0.065_{0.011}$	$0.053_{0.010}$	$0.188_{0.009}$
KRUM	20%	$78.85_{0.41}$	$0.091_{0.008}$	$0.029_{0.007}$	$0.220_{0.006}$
KRUM	30%	$77.84_{0.13}$	$0.115_{0.007}$	$0.055_{0.015}$	$0.242_{0.010}$
KRUM	40%	$76.85_{0.29}$	$0.122_{0.008}$	$0.076_{0.007}$	$0.250_{0.006}$
TM	10%	$79.12_{1.39}$	$0.081_{0.028}$	$0.027_{0.007}$	$0.207_{0.025}$
TM	20%	$79.79_{0.37}$	$0.087_{0.008}$	$0.025_{0.006}$	$0.220_{0.009}$
TM	30%	$79.43_{0.32}$	$0.087_{0.011}$	$0.033_{0.004}$	$0.227_{0.009}$
TM	40%	$79.18_{0.64}$	$0.100_{0.013}$	$0.036_{0.013}$	$0.235_{0.010}$

Table B.2. **Ablation Study: Additional Benchmark.** Comparison of different aggregators on accuracy and fairness metrics: Demographic Parity (DP), Equalized Odds (EO), and Equal Opportunity (EOpp). Here, $r = 4$, and **Acc**: Accuracy, **Adv**: Adversary, and we report $\texttt{mean}_{\texttt{std}}$ across **four** independent runs.

main paper), higher-rank adapters result in only marginal drops in accuracy and fairness, even under 40% adversarial agents. Notably, with $r = 32$, KRUM retains near-baseline fairness levels across all metrics, highlighting that low-rank constraints are a key enabler of the attack's potency by making adversarial updates more easily obscured or entangled in the parameter space.

Moreover, we observe that for KRUM, the **standard deviations of the fairness metrics across adversary proportions** (0%–40%) are substantially lower at $r = 32$ than at $r = 16$, indicating more stable behavior due to KRUM's ability to effectively filter out high-rank adversarial updates. Specifically, the standard deviation drops from 0.036 to 0.005 for Δ_{EO}, 0.041 to 0.009 for Δ_{EOpp}, and 0.031 to 0.004 for Δ_{DP}. These findings suggest that higher adapter rank amplifies parameter deviation, enabling distance-based defenses like KRUM to more reliably identify and discard malicious updates.

Table B.3. **Ablation Study: LoRA-FL without Phase 1.** The adversarial agents omit training for accuracy. We compare the performance on different aggregators on accuracy and fairness metrics: Demographic Parity (DP), Equalized Odds (EO), and Equal Opportunity (EOpp). Here, $|K| = 10$ and $r = 4$.

Aggregator	**% Adversary**	**Accuracy** (↑)	Δ_{EO} (↓)	Δ_{EOpp} (↓)	Δ_{DP} (↓)
		Adult (Dua & Graff, 2017)			
FedAvg	–	$83.33_{0.38}$	$0.122_{0.008}$	$0.095_{0.010}$	$0.186_{0.007}$
KRUM	–	$82.56_{0.20}$	$0.115_{0.016}$	$0.082_{0.018}$	$0.190_{0.008}$
TM	–	$83.10_{0.92}$	$0.115_{0.023}$	$0.088_{0.017}$	$0.183_{0.034}$
FedAvg	10%	$81.73_{0.53}$	$0.448_{0.034}$	$0.446_{0.036}$	$0.330_{0.023}$
FedAvg	20%	$78.41_{1.51}$	$0.594_{0.097}$	$0.594_{0.097}$	$0.449_{0.028}$
FedAvg	30%	$77.05_{1.58}$	$0.693_{0.059}$	$0.693_{0.059}$	$0.475_{0.038}$
FedAvg	40%	$76.26_{1.18}$	$0.693_{0.033}$	$0.693_{0.033}$	$0.476_{0.030}$
KRUM	10%	$82.35_{0.32}$	$0.117_{0.002}$	$0.076_{0.012}$	$0.193_{0.004}$
KRUM	20%	$82.38_{0.46}$	$0.123_{0.023}$	$0.096_{0.026}$	$0.188_{0.017}$
KRUM	30%	$82.38_{0.31}$	$0.121_{0.012}$	$0.088_{0.011}$	$0.188_{0.010}$
KRUM	40%	$82.49_{0.26}$	$0.115_{0.008}$	$0.082_{0.017}$	$0.182_{0.012}$
TM	10%	$35.12_{11.17}$	$0.094_{0.091}$	$0.032_{0.025}$	$0.103_{0.109}$
TM	20%	$35.86_{18.87}$	$0.121_{0.203}$	$0.097_{0.167}$	$0.138_{0.233}$
TM	30%	$34.20_{14.34}$	$0.142_{0.235}$	$0.089_{0.152}$	$0.149_{0.245}$
TM	40%	$52.21_{15.56}$	$0.475_{0.251}$	$0.431_{0.296}$	$0.365_{0.139}$

Table B.4. **Ablation Study: Number of Clients.** Comparison of different aggregators on accuracy and fairness metrics: Demographic Parity (DP), Equalized Odds (EO), and Equal Opportunity (EOpp). Here, $r = 4$ and $|K| = 10$, with results averaged over four independent runs. We observe that our attack remains effective regardless of the number of participating clients, indicating its robustness to varying client pool sizes.

Aggregator	**% Adversary**	**Accuracy** (↑)	Δ_{EO} (↓)	Δ_{EOpp} (↓)	Δ_{DP} (↓)
		Adult (Dua & Graff, 2017)			
FedAvg	–	85.848 ± 0.304	0.177 ± 0.0107	0.0842 ± 0.0408	0.200 ± 0.002
KRUM	–	85.698 ± 0.208	0.159 ± 0.009	0.0875 ± 0.002	0.199 ± 0.003
FedAvg	10%	85.649 ± 0.168	0.236 ± 0.0104	0.089 ± 0.008	0.232 ± 0.007
FedAvg	20%	85.026 ± 0.0462	0.317 ± 0.0376	0.100 ± 0.0027	0.278 ± 0.015
FedAvg	30%	83.926 ± 0.0724	0.488 ± 0.0532	0.117 ± 0.009	0.393 ± 0.101
FedAvg	40%	82.673 ± 0.266	0.621 ± 0.109	0.129 ± 0.005	0.377 ± 0.018
KRUM	10%	85.358 ± 0.185	0.176 ± 0.005	0.117 ± 0.0106	0.224 ± 0.005
KRUM	20%	84.700 ± 0.272	0.269 ± 0.0132	0.127 ± 0.0113	0.287 ± 0.030
KRUM	30%	83.982 ± 0.538	0.434 ± 0.118	0.174 ± 0.0192	0.304 ± 0.056
KRUM	40%	83.278 ± 0.758	0.529 ± 0.146	0.234 ± 0.0076	0.316 ± 0.0411

Table B.5. **Ablation Study: Adapter Rank.** Comparison of different aggregators on accuracy and fairness metrics: Demographic Parity (DP), Equalized Odds (EO), and Equal Opportunity (EOpp). Here, the dataset used is **Adult** and $|K| = 10$. Higher adapter rank increases parameter deviation, enabling KRUM to more effectively filter adversarial updates and stabilize fairness metrics.

Aggregator	**% Adversary**	**Accuracy** $(\uparrow)$	Δ_{EO} $(\downarrow)$	Δ_{EOpp} $(\downarrow)$	Δ_{DP} $(\downarrow)$
		$r = 16$			
FedAvg	–	84.910	0.118	0.096	0.175
KRUM	–	84.900	0.131	0.109	0.189
FedAvg	10%	84.830	0.141	0.113	0.207
FedAvg	20%	84.100	0.203	0.181	0.257
FedAvg	30%	82.850	0.305	0.297	0.312
FedAvg	40%	78.010	0.401	0.401	0.402
KRUM	10%	84.730	0.170	0.161	0.214
KRUM	20%	84.540	0.183	0.162	0.228
KRUM	30%	84.510	0.214	0.195	0.255
KRUM	40%	83.100	0.236	0.233	0.278
		$r = 32$			
FedAvg	–	84.910	0.118	0.096	0.175
KRUM	–	84.900	0.131	0.109	0.189
FedAvg	10%	85.010	0.136	0.102	0.202
FedAvg	20%	83.810	0.211	0.184	0.261
FedAvg	30%	82.340	0.297	0.287	0.319
FedAvg	40%	78.640	0.160	0.160	0.067
KRUM	10%	84.840	0.132	0.090	0.188
KRUM	20%	84.910	0.141	0.114	0.197
KRUM	30%	84.730	0.128	0.100	0.193
KRUM	40%	84.660	0.128	0.091	0.196