Learning to Attack Federated Learning: A Model-based Reinforcement Learning Attack Framework

Anonymous Author(s)
Affiliation
Address
`email`

Appendix

A Appendix to Section 1: Introduction

A.1 Broader Impact

To study the vulnerabilities of federated learning, we propose a model-based reinforcement learning attack framework. Our work shows that non-myopic attacks can break federated learning systems even when they are equipped with sophisticated defense rules. This reveals the urgent need of developing more advanced defense mechanisms for federated learning systems. While we have focused on adversarial attacks against federated learning in our work, we note that one possible solution to defending RL-based attacks would be to dynamically adjust FL parameters such as the subsampling rate or the aggregation rule. Future work is needed to identify how best to do so.

B Appendix to Section 3: RL-based Attacks Against Federated Learning

B.1 Algorithms

In this subsection, we present the detailed algorithms for federated learning (Algorithm 1), and distribution learning (Algorithm 2). We did not make it clear in the main text that a batch of images are reconstructed in each FL epoch during distribution learning. Algorithm 2 gives the full details of distribution learning. The algorithm first initializes $D_{reconstructed}$ with attackers' local data. A synthetic noisy dataset is then built by adding Gaussian noise to $D_{reconstructed}$. A denoising autoencoder is then learned using paired clean data and noisy data. In each FL epoch, a batch of dummay data samples are first generated randomly, which are then updated iteratively by matching their average gradient with the aggregated gradient estimated from received model parameters. When no attacker is sampled in an FL epoch, the same process is applied by reusing the most recent model parameters received from the server. Due to the randomness of the algorithm, new data samples are generated and added (after denoising) to $D_{reconstructed}$ in each FL epoch during distribution learning.

C Proof of Theorem 1

C.1 Preliminaries

Our theoretic analysis relies on the following definitions and results. First, we formally define the Wasserstein distance [22], which will be used to measure the distance between the estimated and true

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Algorithm 1 Federated Learning

```
Input: Initial weight $\theta^0$, $K$ workers indexed by $k$, size of subsampling $w$, local minibatch size $B$, step size $\eta$, number of global training steps $\mathcal{T}$
Output: $\theta^{\mathcal{T}}$
Server executes:
  for $t = 0$ to $\mathcal{T} - 1$ do
    $\mathcal{S}^t \leftarrow$ randomly select $w$ workers from $K$ workers
    for each worker $j \in \mathcal{S}^t$ in parallel do
      $g_j^{t+1} \leftarrow$ WorkerUpdate$(j, \theta^t)$
    end for
    $g^{t+1} \leftarrow Aggr(g_{k_1}^{t+1}, ..., g_{k_w}^{t+1}), k_i \in \mathcal{S}^t$
    $\theta^{t+1} \leftarrow \theta^t - \eta g^{t+1}$
  end for
WorkerUpdate$(j, \theta)$:
  Sample a minibatch $b$ of size $B$
  $g \leftarrow \frac{1}{B} \sum_{z \in b} \nabla_\theta \ell(\theta, z)$
  return $g$ to server
```

Algorithm 2 Distribution Learning

```
Input: number of steps for distribution learning $\tau_E$, number of iterations for each step $max\_iter$, learning rate for FL $\eta$ learning rate for inverting gradients $\eta'$, number of reconstructed data per epoch $B'$, and model parameters $\{\theta^{t(\tau)}\}$
Output: $D_{reconstructed}$
$D_{Reconstructed} \leftarrow M$ attackers' local data
$D_{Noisy} \leftarrow$ Add Gaussian noise to $D_{reconstructed}$ and clip data to the valid range
Train a denoising autoencoder $A_{denoise}$ using $D_{reconstructed}$ and $D_{noisy}$
for $\tau = 0$ to $\tau_E$ do
  Generate $D_{dummy}$ with $B'$ random data and label pairs
  Compute aggregated gradient $\bar{g}^\tau \leftarrow (\theta^{t(\tau-1)} - \theta^{t(\tau)})/(\eta(t(\tau) - t(\tau-1)))$
  for $i = 0$ to $max\_iter - 1$ do
    $F_{dummy}(\theta) \leftarrow \frac{1}{B'} \sum_{(x_j, y_j) \in D_{dummy}} \ell(\theta; (x_j, y_j))$
    $\mathcal{L} \leftarrow 1 - \frac{\langle \nabla_\theta F_{dummy}(\theta^{t(\tau)}), \bar{g}^\tau \rangle}{||\nabla_\theta F_{dummy}(\theta^{t(\tau)})|| \cdot ||\bar{g}^\tau||} + \frac{\beta}{B'} \sum_{(x_j, y_j) \in D_{dummy}} \text{TV}\,(x_j)$
    $x_j \leftarrow x_j - \eta' \nabla_{x_j} \mathcal{L}, y_j \leftarrow y_j - \eta' \nabla_{y_j} \mathcal{L}, \;\forall (x_j, y_j) \in D_{dummy}$
  end for
  Denoise the dummy batch $D_{dummy}$ using $A_{denoise}$ and add it to $D_{reconstructed}$
end for
```

data distributions as well as the distance between the corresponding transition dynamics introduced by different data distributions.

Definition 1. *(Wasserstein distance) Let $(\mathbf{M}, d)$ be a metric space and $\mathcal{P}_p(\mathbf{M})$ be the set of all probability measures on $\mathbf{M}$ with p^{th} moment, then the p^{th} Wasserstein distance between two probability distributions μ_1 and μ_2 in $\mathcal{P}_p(\mathbf{M})$ is defined as:*

$$W_p(\mu_1, \mu_2) := \left(\inf_{j \in \mathcal{J}} \int \int d(s_1, s_2)^p j(s_1, s_2) ds_1 ds_2 \right)^{1/p}$$

where $\mathcal{J}$ is the collection of all joint distributions j on $\mathbf{M} \times \mathbf{M}$ with marginals μ_1 and μ_2.

In the following, we focus on 1-Wasserstein distance and denote $W(\mu_1, \mu_2) := W_1(\mu_1, \mu_2)$. Wasserstein distance is also known as "Earth Mover's distance" that measures the minimum expected distance between two pairs of points where the joint distribution is constrained to match their corresponding marginals . Compared with Kullback-Leibler (KL) divergence and Total Variation (TV) distance, Wasserstein distance is more sensitive to how far the points are from each other [2].

We will also need the following special form of Lipschitz continuity from [2].

Definition 2. *(Lipschitz Continuity) Given two metric spaces* $(\mathbf{M}_1, d_1)$ *and* $(\mathbf{M}_2, d_2)$*, a function* $f : \mathbf{M}_1 \to \mathbf{M}_2$ *is Lipschiz continuous if the Lipschiz constant, defined as*

$$K_{d_1,d_2}(f) := \sup_{s_1 \in \mathbf{M}_1, s_2 \in \mathbf{M}_2} \frac{d_2(f(s_1), f(s_2))}{d_1(s_1, s_2)}$$

is finite. Similarly, a function $f : \mathbf{M}_1 \times A \to \mathbf{M}_2$ *is uniformly Lipschitz continuous in* A *if:*

$$K^A_{d_1,d_2}(f) := \sup_{a \in A} \sup_{s_1, s_2} \frac{d_2(f(s_1, a), f(s_2, a))}{d_1(s_1, s_2)}$$

is finite.

Let $\mathcal{M} = (S, A, T, r)$ be a generic MDP, where S and A denote the state space and the action space respectively, $T(s'|s, a)$ denotes the probability of reaching a state s' from the current state s and action a, and $r(s, a, s')$ denotes the reward given the current state s, action a, and the next state s'. We then introduce the concept of Lipschiz model class from [2], which allows us to represent the stochastic transition dynamics of an MDP as a distribution over a set of deterministic transitions.

Definition 3. *(Lipschitz model class) Given a metric state space* (S, d_S) *and an action space* A*, we define* F_g *as a collection of functions:* $F_g = \{f : S \to S\}$ *distributed according to* $g(f|a)$ *where* $a \in A$*. We say that* F_g *is a Lipschitz model class if*

$$K_F := \sup_{f \in F_g} K_{d_S,d_S}(f),$$

is finite. We say that a transition function T *is induced by a Lipschitz model class* F_g *if* $T(s'|s, a) = \sum_f \mathbb{1}(f(s) = s')g(f|a)$ *for any* $s, s' \in S$ *and* $a \in A$*.*

We will show later that the transition dynamics of our MDP model for attackers is induced by a Lipschitz model class.

Finally we give a formal definition of finite-horizon value functions [21].

Definition 4. *Given an MDP* $\mathcal{M}$ *and a stationary policy* π*, the value function of* π *at time* l *is defined as* $V^{\pi}_{\mathcal{M},l}(s) := \mathbb{E}_{\pi,T}[\sum_{t=l}^{H-1} r(s^t, a^t)|s^l = s]$*, where* $r(s, a) = \mathbb{E}_{s' \sim T(\cdot|s,a)}[r(s, a, s')]$*.* $V^{\pi}_{\mathcal{M},l}(\cdot)$ *satisfies the following backward recursion form:*

$$V^{\pi}_{\mathcal{M},l}(s) = \mathbb{E}_{a \sim \pi(s)}[r(s, a) + \sum_{s' \in S} T(s'|s, a)V^{\pi}_{\mathcal{M},l+1}(s')]$$

where $V^{\pi}_{\mathcal{M},H-1}(s) = \mathbb{E}_{a \sim \pi(s)}[r(s, a)]$*. The optimal value function is defined as* $V^{*}_{\mathcal{M},l}(s) := \max_{\pi} V^{\pi}_{\mathcal{M},l}(s)$ *for any* s*.*

To analyze the impact of inaccurate transition on the value function, we also make use of the following lemmas [2].

Lemma 1. *Given two distributions over states* μ_1 *and* μ_2*, a transition function* T *induced by a Lipschitz model class* F_g *is uniformly Lipschitz continuous in action space* A *with a constant:*

$$K^A_{W,W}(T) := \sup_{a \in A} \sup_{\mu_1, \mu_2} \frac{W(T(.|\mu_1, a), T(.|\mu_2, a))}{W(\mu_1, \mu_2)} \leqslant K_F$$

Lemma 2. *Given a Lipschiz function* $f : S \to \mathbb{R}$ *with constant* $K_{d_S,d_{\mathbb{R}}}(f)$*:*

$$K^A_{d_S,d_{\mathbb{R}}}(\int f(s')T(s'|s, a)ds') \leqslant K_{d_S,d_{\mathbb{R}}}(f)K^A_{d_S,W}(T)$$

Below we state the assumptions needed for establishing Theorem 1. The first assumption models the inaccuracy of distribution learning as well as the heterogeneity of benign worker's local data.

Assumption 1. $W_1(\widetilde{P}, \widehat{P}_k) \leqslant \delta$ *for any benign worker* k*.*

We further need the following standard assumptions on the loss function.

Assumption 2. *Let* Z *denote the domain of data samples across all the workers. For any* $s_1, s_2 \in S$ *and* $z_1, z_2 \in Z$*, the loss function* $\ell : S \times Z \to \mathbb{R}$ *satisfies:*

1. $|\ell(s_1, z_1) - \ell(s_2, z_2)| \leqslant L\|(s_1, z_1) - (s_2, z_2)\|_2$ *(Lipschitz continuity w.r.t. s and z);*
2. $\|\nabla_s \ell(s_1, z_1) - \nabla_s \ell(s_1, z_2)\|_2 \leqslant L_z\|z_1 - z_2\|_2$ *(Lipschitz smoothness w.r.t. z);*
3. $\ell(s_2, z_1) \geqslant \ell(s_1, z_1) + \langle \nabla_s \ell(s_1, z_1), s_2 - s_1 \rangle + \frac{\alpha}{2}\|s_2 - s_1\|_2^2$ *(strongly convex w.r.t. s);*
4. $\ell(s_2, z_1) \leqslant \ell(s_1, z_1) + \langle \nabla_s \ell(s_1, z_1), s_2 - s_1 \rangle + \frac{\beta}{2}\|s_2 - s_1\|_2^2$ *(strongly smooth w.r.t. s);*
5. $\ell(\cdot, \cdot)$ *is twice continuously differentiable with respect to s.*

where $\|(s_1, z_1) - (s_2, z_2)\|_2^2 = \|s_1 - s_2\|_2^2 + \|z_1 - z_2\|_2^2$.

For simplicity, we further make the following assumption on the FL environment, although our analysis can be readily applied to more general settings.

Assumption 3. *The server adopts FedAvg without subsampling ($w = K$). All workers have same amount of data ($p_k = \frac{1}{K}$) and the local minibatch size $B = 1$. In each epoch of federated learning, each normal worker's local minibatch is sampled independently from the local empirical data distribution $\widehat{P}_k$.*

C.2 Measuring the Uncertainty: From Data Distributions to Total Returns

Let $\mathcal{M} = (S, \boldsymbol{A}, T, r, H)$ denote the true MDP for attacking the federated learning system, and $\widetilde{\mathcal{M}} = (S, \boldsymbol{A}, T', r', H)$ the estimated MDP used in the policy learning stage, where T' and r' are derived from the estimated joint data distribution $\{\widetilde{P}_k\}$ where $\widetilde{P}_k = \widehat{P}_k$ when k is an attacker and $\widetilde{P}_k = \widetilde{P}$ otherwise. Our main goal is to compare the optimal attack performance that can be obtained from the true MDP $\mathcal{M}$ and that derived from the simulated MDP $\widetilde{\mathcal{M}}$. We will focus on understanding the impact of inaccurate data distributions (obtained from distribution learning) and assume that other system parameters are known to the attackers.

Without loss of generality, we assume the M attackers' indexes are from $K - M + 1$ to K. Let $\epsilon = \frac{K-M}{M}$ denote the fraction of benign nodes. We consider the idealized setting where the M attackers are perfectly coordinated by a single leading attacker. Because of these simplifications, the state s^t in each epoch t is completely defined by the current model parameters θ^t. In the following, we abuse the notation a bit and assume $S = \Theta$.

Let $\mathcal{J}_{\mathcal{M}}(\pi) := \mathbb{E}_{\pi, T, \mu_0}[\sum_{t=0}^{H-1} r(s^t, a^t, s^{t+1})]$ denote the expected return over H attack steps under the MDP $\mathcal{M}$, policy π and initial state distribution μ_0. Let π^* be an optimal policy of $\mathcal{M}$ that maximizes $\mathcal{J}_{\mathcal{M}}(\pi)$. Define $\mathcal{J}_{\widetilde{\mathcal{M}}}(\pi)$ similarly and let $\widetilde{\pi}^*$ be an optimal policy for $\widetilde{\mathcal{M}}$, with the same initial state distribution μ_0.

Our analysis is built upon the following lemma that compares the performance of π^* and that of $\widetilde{\pi}^*$ with respect to the true MDP $\mathcal{M}$. It extends a similar result in [27] to a finite-horizon MDP where the reward in each step depends on not only the current state and action but also the next state. Note that the lemma relies on the key assumption that both $V^*_{\mathcal{M},l}(\cdot)$ and $V^*_{\widetilde{\mathcal{M}},l}(\cdot)$ are L_v-Lipschitz continuous (with respect to the l_2 norm of states) for all l. That is, $|V^*_{\mathcal{M},l}(s_1) - V^*_{\mathcal{M},l}(s_2)| \leqslant L_v\|s_1 - s_2\|_2$ for any $s_1, s_2 \in S$ where L_v is a constant independent of l. A similar requirement holds for $V^*_{\widetilde{\mathcal{M}},l}(\cdot)$. Let $W(T, T') := \sup_{a \in \boldsymbol{A}} \sup_{s \in S} W(T(\cdot|s, a), T'(\cdot|s, a))$.

Lemma 3. *Assume Assumptions 2.1 holds and both $V^*_{\mathcal{M},l}(\cdot)$ and $V^*_{\widetilde{\mathcal{M}},l}(\cdot)$ are L_v-Lipschitz continuous for all l. Then,*

$$|\mathcal{J}_{\mathcal{M}}(\pi^*) - \mathcal{J}_{\mathcal{M}}(\widetilde{\pi}^*)| \leqslant 2H[(L + L_v)W(T, T') + 2L\epsilon\delta]$$

Proof. Let F_l be the expected return when π^* is applied to $\widetilde{\mathcal{M}}$ for the first l steps, then changing to $\mathcal{M}$ for l to $H - 1$. That is,

$$F_l = \mathop{\mathbb{E}}_{\substack{a^t \sim \pi^*(s^t) \\ t<l: s^{t+1} \sim T'(s^t, a^t), r^t = r' \\ t \geqslant l: s^{t+1} \sim T(s^t, a^t), r^t = r}} \left[\sum_{t=0}^{H-1} r^t(s^t, a^t, s^{t+1})\right]$$

By the definition of F_l, we have $\mathcal{J}_M(\pi^*) = F_0$ and $\mathcal{J}_{\widetilde{\mathcal{M}}} = F_H$, which implies that $\mathcal{J}_{\mathcal{M}}(\pi^*) - \mathcal{J}_{\widetilde{\mathcal{M}}}(\pi^*) = \sum_{l=0}^{H-1}(F_l - F_{l+1})$. Note that

$$F_l = R_{l-1} + \mathbb{E}_{s^{l+1} \sim T(s^l, a^l)}[r(s^l, a^l, s^{l+1})] + \mathbb{E}_{s^l, a^l \sim T', \pi*}[\mathbb{E}_{s^{l+1} \sim T(s^l, a^l)}[V^*_{\mathcal{M},l+1}(s^{l+1})]]$$

$$F_{l+1} = R_{l-1} + \mathbb{E}_{s^{l+1}\sim T'(s^l,a^l)}[r'(s^l,a^l,s^{l+1})] + \mathbb{E}_{s^l,a^l\sim T',\pi*}[\mathbb{E}_{s^{l+1}\sim T'(s^l,a^l)}[V^*_{\mathcal{M},l+1}(s^{l+1})]]$$

where R_{l-1} is the expected return of the first $l-1$ steps, which are taken with respect to $\mathcal{M}$. Thus,

$$\begin{aligned}F_l - F_{l+1} &= \mathbb{E}_{s^{l+1}\sim T(s^l,a^l)}[r(s^l,a^l,s^{l+1})] - \mathbb{E}_{s^{l+1}\sim T'(s^l,a^l)}[r'(s^l,a^l,s^{l+1})] \\ &\quad + \mathbb{E}_{s^l,a^l\sim T',\pi*}[\mathbb{E}_{s^{l+1}\sim T(s^l,a^l)}[V^*_{\mathcal{M},l+1}(s^{l+1})] - \mathbb{E}_{s^{l+1}\sim T'(s^l,a^l)}[V^*_{\mathcal{M},l+1}(s^{l+1})]]\end{aligned}$$

Define $G^*_{\widetilde{\mathcal{M}},l}(s^l,a^l) := \mathbb{E}_{s^{l+1}\sim T(s^l,a^l)}[V^*_{\mathcal{M},l}(s^{l+1})] - \mathbb{E}_{s^{l+1}\sim T'(s^l,a^l)}[V^*_{\mathcal{M},l}(s^{l+1})]$. We have

$$\begin{aligned}\mathcal{J}_{\mathcal{M}}(\pi^*) - \mathcal{J}_{\widetilde{\mathcal{M}}}(\pi^*) &= \sum_{l=0}^{H-1}(F_l - F_{l+1}) \\ &= \sum_{l=0}^{H-1}\Big(\mathbb{E}_{s^{l+1}\sim T(s^l,a^l)}[r(s^l,a^l,s^{l+1})] - \mathbb{E}_{s^{l+1}\sim T'(s^l,a^l)}[r'(s^l,a^l,s^{l+1})]\Big) \\ &\quad + \sum_{l=0}^{H-2}\mathbb{E}_{s^l,a^l\sim T',\pi*}[G^*_{\widetilde{\mathcal{M}},l}(s^l,a^l)] \\ &= \sum_{l=0}^{H-1}\left(\mathbb{E}_{s^{l+1}\sim T(s^l,a^l)}[\frac{1}{K}\sum_{k=1}^{K}(\ell_k(s^{l+1}) - \ell_k(s^l))] - \mathbb{E}_{s^{l+1}\sim T'(s^l,a^l)}[\frac{1}{K}\sum_{k=1}^{K}\ell'_k(s^{l+1}) - \ell'_k(s^l))]\right) \\ &\quad + \sum_{l=0}^{H-2}\mathbb{E}_{s^l,a^l\sim T',\pi*}[G^*_{\widetilde{\mathcal{M}},l}(s^l,a^l)] \\ &= \sum_{l=0}^{H-1}\left(\mathbb{E}_{s^{l+1}\sim T(s^l,a^l)}[\frac{1}{K}\sum_{k=1}^{K}\ell_k(s^{l+1})] - \mathbb{E}_{s^{l+1}\sim T'(s^l,a^l)}[\frac{1}{K}\sum_{k=1}^{K}\ell'_k(s^{l+1})]\right) \\ &\quad + \sum_{l=0}^{H-1}\left(\frac{1}{K}\sum_{k=1}^{K}\ell'_k(s^l) - \frac{1}{K}\sum_{k=1}^{K}\ell_k(s^l)\right) \\ &\quad + \sum_{l=0}^{H-2}\mathbb{E}_{s^l,a^l\sim T',\pi*}[G^*_{\widetilde{\mathcal{M}},l}(s^l,a^l)]\end{aligned}$$

where $\ell_k(s) := \mathbb{E}_{z_k\sim\widehat{P}_k}[\ell(s,z_k)]$, $\ell'_k(s) := \mathbb{E}_{z_k\sim\widetilde{P}_k}[\ell(s,z_k)]$ and the last equality follows from the definition of reward function $r(s,a,s') = \frac{1}{K}\sum_{k=1}^{K}\ell_k(s') - \frac{1}{K}\sum_{k=1}^{K}\ell_k(s)$, and $r'(s,a,s') = \frac{1}{K}\sum_{k=1}^{K}\ell'_k(s') - \frac{1}{K}\sum_{k=1}^{K}\ell'_k(s)$.

Since $V^*_{\mathcal{M},l}$ is L_v-Lipschitz, we have $|G^*_{\widetilde{\mathcal{M}},l}(s,a)| \leqslant L_v W(T(s,a),T'(s,a))$ from the definition of 1-Wasserstein distance. We further have

$$\begin{aligned}\left|\frac{1}{K}\sum_{k=1}^{K}\ell'_k(s^l) - \frac{1}{K}\sum_{k=1}^{K}\ell_k(s^l)\right| &\leqslant \frac{1}{K}\sum_{k=1}^{K}|\ell'_k(s^l) - \ell_k(s^l)| \\ &\leqslant \frac{1}{K}\sum_{k=1}^{K}LW(\widetilde{P}_k,\widehat{P}_k) \\ &\leqslant L\epsilon\delta,\end{aligned}$$

where the second inequality follows from the definition of 1-Wasserstein distance and Assumption 3.1, and the last inequality follows from Assumption 1 and the fact that $\widetilde{P}_k = \widehat{P}_k$ for any attacker k. Similarly, We have

$$\begin{aligned}&\left|\mathbb{E}_{s'\sim T(s,a)}[\frac{1}{K}\sum_{k=1}^{K}\ell_k(s')] - \mathbb{E}_{s'\sim T'(s,a)}[\frac{1}{K}\sum_{k=1}^{K}\ell'_k(s')]\right| \\ \leqslant &\frac{1}{K}\sum_{k=1}^{K}\left|\mathbb{E}_{s'\sim T(s,a)}[\ell_k(s')] - \mathbb{E}_{s'\sim T'(s,a)}[\ell'_k(s')]\right|\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{K} \sum_{k=1}^{K} \left| \mathbb{E}_{s' \sim T(s,a), z_k \sim \widehat{P}_k} [\ell_k(s', z_k)] - \mathbb{E}_{s' \sim T'(s,a), z_k \sim \widetilde{P}_k} [\ell'_k(s', z_k)] \right| \\
&\leqslant L(W(T,T') + \epsilon\delta),
\end{aligned}
$$

where the last inequality follows Assumption 1, Assumption 3.1, and the property of 1-Wasserstein distance with respect to product measures. Thus,

$$
\mathcal{J}_{\mathcal{M}}(\pi^*) - \mathcal{J}_{\widetilde{\mathcal{M}}}(\pi^*) \leqslant H(L_v + L)W(T,T') + 2HL\epsilon\delta.
$$

A similar argument shows that

$$
\mathcal{J}_{\widetilde{\mathcal{M}}}(\widetilde{\pi}^*) - \mathcal{J}_{\mathcal{M}}(\widetilde{\pi}^*) \leqslant H(L_v + L)W(T,T') + 2HL\epsilon\delta.
$$

Let $U = H(L_v + L)W(T,T') + 2HL\epsilon\delta$. Thus,

$$
\mathcal{J}_{\mathcal{M}}(\pi^*) \leqslant \mathcal{J}_{\widetilde{\mathcal{M}}}(\pi^*) + U \leqslant \mathcal{J}_{\widetilde{\mathcal{M}}}(\widetilde{\pi}^*) + U \leqslant \mathcal{J}_{\mathcal{M}}(\widetilde{\pi}^*) + 2U.
$$

□

As indicated in [27], an important obstacle to applying Lemma 3 to real reinforcement learning problems is to bound the Lipschitz constant L_v for optimal value functions. Further, we need to bound $W(T,T')$, the 1-Wasserstein distance between two transition functions. We study these two problems in the following two subsections, respectively.

C.3 Lipschitz Constant of Value Functions

In this section, we show that the Lipschitz constant L_v can be upper bounded for any optimal value function in our setting. We first rewrite the update of model parameters in each epoch of FedAvg as follows:

$$
f_z(s, \{\tilde{g}_i\}_{i \in [M]}) := s - \eta \frac{1}{K} \left[\sum_{k=1}^{K-M} \nabla_s \ell(s, z_k) + \sum_{k=M+1}^{K} \tilde{g}_k \right] \tag{1}
$$

where $z = \{z_k\}$ denotes the set of data points sampled by each worker. That is, the above equation gives the one-step *deterministic* transition when the data samples are fixed. An important observation is that the transition function T is induced by a Lipschitz model class $F_g = \{f_z : z \in Z^K\}$with $g(f_z|a)$ equal to the probability that z is sampled according to the joint distribution $\prod_{k \in [K]} \widehat{P}_k$. Similarly, T' is induced by $F_{g'} = \{f_z : z \in Z^K\}$ with $g'(f_z|a)$ equal to the probability that z is sampled according to the joint distribution $\prod_{k \in [M]} \widehat{P}_k \widetilde{P}^{K-M}$. This observation allows us to apply the techniques in [2] to bound the Lipschitz constant L_v of an optimal value function once we bound the Lipschitz continuity of individual f_z.

We first show that for any joint action $a = \{\tilde{g}_i\}_{i \in [M]}$, the deterministic transition $f_z(\cdot, a)$ is Lipschitz continuous with a Lipschitz constant $K_{d_S,d_S}(f_z(\cdot, a))$ that can be upper bounded independent of z.

Lemma 4. *Assume Assumptions 2.3, 2.4, and 2.5 hold. For any Lipschitz model class $F_g = \{f_z : z \in Z^K\}$, we have $K_F \leqslant \max\{\epsilon|1 - \eta\alpha|, \epsilon|1 - \eta\beta|\}$.*

Proof. It suffices to show that for any action a, $K_{d_S,d_S}(f_z(\cdot, a)) \leqslant \max\{\epsilon|1 - \eta\alpha|, \epsilon|1 - \eta\beta|\}$. By (1), we have for any $s_1, s_2 \in S$,

$$
\begin{aligned}
\|f_z(s_1, a) - f_z(s_2, a)\|_2 &= \left\| s_1 - \eta \frac{1}{K} \sum_{k=1}^{K-M} \nabla_s \ell(s_1, z_k) - (s_2 - \eta \frac{1}{K} \sum_{k=1}^{K-M} \nabla_s \ell(s_2, z_k)) \right\|_2 \\
&\overset{(a)}{\leqslant} \frac{1}{K} \sum_{k=1}^{K-M} \|s_1 - \eta \nabla_s \ell(s_1, z_k) - (s_2 - \eta \nabla_s \ell(s_2, z_k))\|_2 \\
&\overset{(b)}{=} \frac{1}{K} \sum_{k=1}^{K-M} \left\| (I - \eta \frac{\partial^2 \ell(\bar{s}, z_k)}{\partial s^2})(s_1 - s_2) \right\|_2
\end{aligned}
$$

$$\overset{(c)}{\leqslant} \frac{1}{K}\sum_{k=1}^{K-M} \left\| I - \eta \frac{\partial^2 \ell(\bar{s}, z_k)}{\partial s^2}\right\|_2 \|s_1 - s_2\|_2$$

where (a) follows from the triangle inequality, (b) follows from the fact that $\ell(s, z)$ is twice continuously differentiable with respect to s and the mean value theorem, where $\bar{s}$ is a point on the line segment connecting s_1 and s_2, and I is the identity matrix with its dimension equal to the dimension of the model parameters, and (c) is due to the Cauchy–Schwarz inequality.

By the strong convexity and smoothness of $\ell(s, z)$ with respect to s, the eigenvalues of $\frac{\partial^2 \ell(\bar{s}, z_k)}{\partial s^2}$ are between α and β [15]. It follows that

$$\left\| I - \eta \frac{\partial^2 \ell(\bar{s}, z_k)}{\partial s^2}\right\|_2 \leqslant \max\{|1-\eta\alpha|, |1-\eta\beta|\}, \quad \forall k$$

Therefore, for any s_1, s_2,

$$\frac{\|f_z(s_1, a) - f_z(s_2, a)\|_2}{\|s_1 - s_2\|_2} \leqslant \max\{\epsilon|1-\eta\alpha|, \epsilon|1-\eta\beta|\}$$

By Definition 2, we then have

$$\begin{aligned} K_{d_S, d_S}(f_z(\cdot, a)) := \sup_{s_1, s_2} & \frac{\|f_z(s_1, a) - f_z(s_2, a)\|_2}{\|s_1 - s_2\|_2} \\ & \leqslant \max\{\epsilon|1-\eta\alpha|, \epsilon|1-\eta\beta|\} \end{aligned}$$

□

Note that by using a small enough learning rate η, K_F can be made less than 1 so that the one-step deterministic transition becomes a contraction. We next show that the optimal value function $V^*_{\mathcal{M},l}(\cdot)$ has a bounded Lipschitz constant. Note that the bound is independent of $\mathcal{M}$; hence it also applies to $V^*_{\widetilde{\mathcal{M}},l}(\cdot)$

Lemma 5. *Assume Assumptions 2.1, 2.3, 2.4, and 2.5 hold. The optimal value function $V^*_{\mathcal{M},l}(\cdot)$ is Lipschitz continuous with a Lipschitz constant bounded by $\sum_{t=0}^{H-l-1}(K_F)^t(L + LK_F)$.*

Proof. The proof is adapted from the proof of Theorem 3 in [2]. Let $Q^\pi_{\mathcal{M},l}(s,a) = r(s,a) + \sum_{s'\in S} T(s'|s,a)V_{\mathcal{M},l+1}(s')$ denote the state-action value function, where $r(s,a) = \mathbb{E}_{s'\sim T(s'|s,a)}[r(s,a,s')]$. We have for the optimal state-action value function

$$Q^*_{\mathcal{M},l}(s,a) = r(s,a) + \sum_{s'\in S} T(s'|s,a) \max_{a'\in A} Q^*_{\mathcal{M},l+1}(s',a')$$

with $Q^*_{\mathcal{M},H-1}(s,a) = r(s,a)$. The Lipschitz constant of $Q^*_{\mathcal{M},l}$ is bounded by:

$$\begin{aligned} K^A_{d_S,d_{\mathbb{R}}}(Q^*_{\mathcal{M},l}) &\leqslant K^A_{d_S,d_{\mathbb{R}}}(r) + K^A_{d_S,d_{\mathbb{R}}}\left(\sum_{s'\in S} T(s'|s,a) \max_{a'\in A} Q^*_{\mathcal{M},l+1}(s',a')\right) \\ &\overset{(a)}{\leqslant} K^A_{d_S,d_{\mathbb{R}}}(r) + K^A_{W,W}(T) K^A_{d_S,d_{\mathbb{R}}}(\max_{a'\in A} Q^*_{\mathcal{M},l+1}) \\ &\overset{(b)}{\leqslant} K^A_{d_S,d_{\mathbb{R}}}(r) + K^A_{W,W}(T) K^A_{d_S,d_{\mathbb{R}}}(Q^*_{\mathcal{M},l+1}) \\ &\leqslant K^A_{d_S,d_{\mathbb{R}}}(r) + K^A_{W,W}(T)[K^A_{d_S,d_{\mathbb{R}}}(r) + K^A_{W,W}(T) K^A_{d_S,d_{\mathbb{R}}}(Q^*_{\mathcal{M},l+2})] \\ &\leqslant K^A_{d_S,d_{\mathbb{R}}}(r) + \sum_{t=1}^{H-l-2} (K^A_{W,W}(T))^t K^A_{d_S,d_{\mathbb{R}}}(r) + K^A_{W,W}(T)^{H-l-1} K^A_{d_S,d_{\mathbb{R}}}(Q^*_{\mathcal{M},H-1}) \\ &= \sum_{t=0}^{H-l-1} (K^A_{W,W}(T))^t K^A_{d_S,d_{\mathbb{R}}}(r) \end{aligned}$$

where (a) follows Lemma 2 and (b) is due to the fact that the max operator is 1-Lipschitz, that is, $K_{\|\|_\infty, d_{\mathbb{R}}}(\max(x)) = 1$ [1]. From the definition of $r(s,a)$, we further have

$$\begin{aligned}|r(s_1,a) - r(s_2,a)| &\leqslant \frac{1}{K}\sum_{k=1}^{K}|\ell_k(s_1) - \ell_k(s_2)| + \frac{1}{K}\sum_{k=1}^{K}|\mathbb{E}_{s_1' \sim T(s_1,a)}[\ell_k(s_1')] - \mathbb{E}_{s_2' \sim T(s_2,a)}[\ell_k(s_2')]| \\ &\leqslant (L + LK^A_{W,W}(T))\|s_1 - s_2\|_2\end{aligned}$$

where $\ell_k(s) := \mathbb{E}_{z_k \sim \widehat{P}_k}[\ell(s, z_k)]$. The first term of the second inequality comes from the Lipschitz continuity of the loss function ℓ, which gives $|\ell_k(s_1) - \ell_k(s_2)| \leqslant L\|s_1 - s_2\|_2$ for any k, and the second term follows from Lemma 2 by letting $f(s) = \ell_k(s)$, which gives $K^A_{d_S, d_{\mathbb{R}}}(\mathbb{E}_{s' \sim T}[\ell_k(s')]) \leqslant LK^A_{W,W}(T)$ for all k.

Since the above inequality holds for any $a \in A$, $r(s,a)$ is uniformly Lipschitz continuous in action space A with a Lipschitz constant $K^A_{d_S,d_{\mathbb{R}}}(r) = L + LK^A_{W,W}(T)$. Thus, $K^A_{d_S,d_{\mathbb{R}}}(Q^*_{\mathcal{M},l}) \leqslant \sum_{t=0}^{H-l}(K^A_{W,W}(T))^t(L + LK^A_{W,W}(T))$. Since the optimal value function $V^*_{\mathcal{M},l}(s) = \max_{a \in A} Q^*_{\mathcal{M},l}(s,a)$ and the max operator is 1-Lipschitz [1], we have $K_{d_S,d_{\mathbb{R}}}(V^*_{\mathcal{M},l}) \leqslant K^A_{d_S,d_{\mathbb{R}}}(Q^*_{\mathcal{M},l}) \leqslant \sum_{t=0}^{H-l-1}(K^A_{W,W}(T))^t(L + LK^A_{W,W}(T))$.

By Lemma 1, we have $K^A_{W,W}(T) \leqslant K_F$. The desired result then follows by applying Lemma 4.

□

The lemma immediately implies that $V^*_{\mathcal{M},l}(\cdot)$ is L_v-Lipschitz for any l where $L_v \leqslant \sum_{t=0}^{H-1}(K_F)^t(L + LK_F)$.

C.4 Wasserstein Distance between Transitions

In this section, we bound the 1-Wasserstein distance of transition functions. Recall that the true transition dynamics $T(\cdot|s,a)$ depends on the joint distribution $\prod_{k=1}^{K-M}\widehat{P}_k$, while $T'(\cdot|s,a)$ depends on $\widetilde{P}^{K-M}$. We have the following lemma.

Lemma 6. *Assume Assumptions 1-3 hold. For any state-action pair (s,a), the 1-Wasserstein distance between transition dynamics $T(\cdot|s,a)$ and $T'(\cdot|s,a)$ generated from the real FL environment and the estimated environment, respectively, is bounded by $\eta L_z \epsilon \delta$, that is,*

$$W(T(\cdot|s,a), T'(\cdot|s,a)) \leqslant \eta L_z \epsilon \delta$$

Proof. Let $z_1 = \{z_{1k}\}_{k=1,\ldots,K-M}$ and $z_2 = \{z_{2k}\}_{k=1,\ldots,K-M}$ denote two data sets of normal workers sampled from $\prod_{k=1}^{K-M}\widehat{P}_k$ and $\widetilde{P}^{K-M}$ respectively. Let $j = \prod_{k=1}^{K-M} j_k$ denote an arbitrary coupling between the two joint distributions that is independent across workers, and $\mathcal{J}$ the set of all such couplings. Let $\mathcal{J}_s$ denote the collection of couplings between $T(\cdot|s,a)$ and $T'(\cdot|s,a)$ generated from the couplings of joint distributions in $\mathcal{J}$. To simplify the notation, let $s(z) := f_z(s,a)$ denote the successive state given the current state-action pair (s,a) and the sampled data z of normal workers. From the definition of 1-Wasserstein distance, we have

$$\begin{aligned}W(T(\cdot|s,a), T'(\cdot|s,a)) &\overset{(a)}{\leqslant} \inf_{j_s \in \mathcal{J}_s}\sum_{(s_1',s_2')}\|s_1' - s_2'\|_2 j_s(s_1', s_2') \\ &\overset{(b)}{\leqslant} \inf_{j \in \mathcal{J}}\sum_{(z_1,z_2)}\|s(z_1) - s(z_2)\|_2 j(z_1,z_2) \\ &= \inf_{j \in \mathcal{J}}\sum_{(z_1,z_2)}\Big\|s - \frac{1}{K}(\sum_{k=1}^{K-M}\nabla_s \ell(s, z_{1k}) + a) \\ &\qquad - [s - \frac{1}{K}(\sum_{k=1}^{K-M}\nabla_s \ell(s, z_{2k}) + a)]\Big\|_2 \prod_{k=1}^{K-M} j_k(z_{1k}, z_{2k})\end{aligned}$$

$$
\begin{aligned}
&= \inf_{j\in\mathcal{J}} \sum_{(z_1,z_2)} \left\| \frac{1}{K}\sum_{k=1}^{K-M} \nabla_s \ell(s,z_{1k}) - \frac{1}{K}\sum_{k=1}^{K-M} \nabla_s \ell(s,z_{2k}) \right\|_2 \prod_{k=1}^{K-M} j_k(z_{1k},z_{2k}) \\
&\overset{(c)}{\leqslant} \frac{\eta L_z}{K} \inf_{j\in\mathcal{J}} \sum_{(z_1,z_2)} \sum_{k=1}^{K-M} \|z_{1k}-z_{2k}\|_2 \prod_{k=1}^{K-M} j_k(z_{1k},z_{2k}) \\
&\overset{(d)}{\leqslant} \frac{\eta L_z}{K} \inf_{j\in\mathcal{J}} \sum_{(z_1,z_2)} \sum_{k=1}^{K-M} \|z_{1k}-z_{2k}\|_2 j_k(z_{1k},z_{2k}) \\
&\leqslant \frac{\eta L_z}{K} \sum_{k=1}^{K-M} \inf_{j_k} \sum_{(z_{1k},z_{2k})} \|z_{1k}-z_{2k}\|_2 j_k(z_{1k},z_{2k}) \\
&= \frac{\eta L_z}{K} \sum_{k=1}^{K-M} W(\widehat{P}_k, \widetilde{P}) \overset{(e)}{\leqslant} \frac{\eta L_z}{K}(K-M)\delta
\end{aligned}
$$

where (a) is due to the fact that we consider a restrictive collection of couplings, (b) is due to the fact that $\mathcal{J}_s$ is generated from $\mathcal{J}$, (c) follows from the smoothness of $\ell(s,z)$ with respect to z, (d) is due to $j_k(z_{1k},z_{2k}) \leqslant 1, \forall k$, and (e) follows from Assumption 1. □

C.5 Difference between Expected Returns

Combining the results from the previous three sections, we have the following main result.

Theorem 1. *Assume Assumptions 1-3 hold. Let* $\mathcal{J}_{\mathcal{M}}(\pi) := \mathbb{E}_{\pi,T,\mu_0}[\sum_{t=0}^{H-1} r(s^t,a^t,s^{t+1})]$ *denote the expected return over H attack steps under MDP $\mathcal{M}$, policy π and initial state distribution μ_0. Let π^* and $\widetilde{\pi}^*$ be optimal policies for $\mathcal{M}$ and $\widetilde{\mathcal{M}}$ respectively, with the same initial state distribution μ_0. Then,*

$$
|\mathcal{J}_{\mathcal{M}}(\pi^*) - \mathcal{J}_{\mathcal{M}}(\widetilde{\pi}^*)| \leqslant 2H\epsilon\delta[(L+L_v)\eta L_z + 2L]
$$

where $L_v \leqslant \sum_{t=0}^{H-1}(K_F)^t(L+LK_F)$ *and* $K_F \leqslant \epsilon \max\{|1-\eta\alpha|, |1-\eta\beta|\}$.

Proof. By Lemma 3, $|\mathcal{J}_{\mathcal{M}}(\pi^*) - \mathcal{J}_{\mathcal{M}}(\widetilde{\pi}^*)| \leqslant 2(H(L+L_v)W(T(\cdot|s,a),T'(\cdot|s,a)) + 2HL\epsilon\delta)$. From Lemma 6, we have $W(T(\cdot|s,a),T'(\cdot|s,a)) \leqslant \eta L_z \epsilon\delta$. Thus, $|\mathcal{J}_{\mathcal{M}}(\pi^*) - \mathcal{J}_{\mathcal{M}}(\widetilde{\pi}^*)| \leqslant 2H[(L+L_v)\eta L_z\epsilon\delta + 2L\epsilon\delta]$. By Lemma 5 and the comment below it, $L_v \leqslant \sum_{t=0}^{H-1}(K_F)^t(L+LK_F)$ where $K_F \leqslant \epsilon\max\{|1-\eta\alpha|, |1-\eta\beta|\}$. □

D Appendix to Section 5: Experiments

D.1 Experiment Setup

Datasets. We consider three real world datasets: MNIST [13], Fashion-MNIST [23] and Balanced EMNIST [9]. Both MNIST and Fashion-MNIST include $60,000$ training examples and $10,000$ testing examples, where each example is a 28×28 grayscale image, associated with a label from 10 classes. Balanced EMNIST includes $112,800$ training examples and $18,800$ testing examples, where each example is a 28×28 grayscale image, associated with a label from 47 classes. For the *i.i.d.* setting, we randomly split the dataset into K groups, each of which consists of the same number of training samples. For the *non-i.i.d.* setting, we follow the method of [10] to quantify the heterogeneity of local data distribution across clients. Suppose there are C classes in the dataset, e.g., $C=10$ for the MNIST and Fashion-MNIST datasets. We evenly split the worker devices into C groups, where each group is assigned $1/C$ of training samples as follows. A training instance with label c is assigned to the c-th group with probability $q \geqslant 1/C$ and to every other group with probability $(1-q)/(C-1)$. Within each group, instances are evenly distributed. A higher q indicates a higher degree of *non-i.i.d.*. We set $q=0.5$ as the default *non-i.i.d.* degree. To demonstrate the power of distribution learning, we assume that the set of attackers share m true data points sampled from the training instances assigned to them. We set $m=200$ for MNIST and Fashion-MNIST, and $m=500$ for EMNIST.

Federated learning setting. We adopt the following parameters for the federated learning models: learning rate $\eta = 0.01$ (0.05 for EMNIST and the synthetic data), total number of workers $= 100$, number of attackers $= 20$ (0 for NA), subsampling rate $= 10\%$, and number of total epochs $= 1,000$. For the three real datasets, we train a neural network classifier consisting of 8×8, 6×6, and 5×5 convolutional filter layers with ReLU activations followed by a fully connected layer and softmax output. The cross-entropy loss is used to optimize the model. We set the local batch size $B = 128$. We implement the FL model with PyTorch [14] and run all the experiments on the same 2.30GHz Linux machine with 16GB NVIDIA Tesla P100 GPU. We simulate subsampling and local data sampling with different random seeds in each test run. Error bars are reported in Figure 4(c) in the main paper. We set cross-entropy as our default loss function, and stochastic gradient descent (SGD) as our default optimizer.

Baselines. We compare our RL-based attack (RL) with no attack (NA), and the state-of-the-art model poisoning FL attack methods: explicit boosting (EB) [3], inner product manipulation (IPM) [24], and local model poisoning attack (LMP) [10]. The EB attack [3] is originally proposed for the targeted setting. We adapt it to the untargeted setting by using empirical loss as the objective, which is optimized through multi-step gradient ascent using attackers' local data, where the number of steps is 5 and the step size equals to the FL learning rate η. The model update is then boosted by a factor of $\frac{K}{M}$. We compare our RL-based attack with the full knowledge LMP [10], where the attackers require not only the knowledge of the aggregation rule but also the information of all normal workers' updates. We use the LMP attack tailored to Krum when the Krum defense is used, and the LMP attack tailored to coordinate-wise Median when the Clipping Median defense is used. Further, we implement the adaptive version of LMP introduced in [7], which requires the attackers to know the server's updates derived from its root data, as a baseline against the FLTrust defense [7]. In our implementation of IPM [24], we set the default boosting factor (i.e., ϵ in [24]) as 5.

We consider three representative robust aggregation rules of different types [18]: Krum [4], which applies a vector-wise filtering to model updates, coordinate-wise median [26], which adopts a dimension-wise filtering, and FLTrust [7], which requires the server to collect a small training dataset D_0 (called root dataset). In the experiments, we actually consider an extension of the vanilla coordinate-wise median where a norm-bound clipping [20] is first applied before aggregation. This gives a more powerful defense as we observed in experiments. We set the default clipping threshold to 2. In FLTrust, the root data is used to calculate a server model update $g_0 = \frac{1}{|D_0|} \sum_{z \in D_0} [\nabla_\theta \ell(\theta; z)]$ in each epoch. The aggregation weight of each received client' update is then determined through its ReLU-clipped cosine similarity with g_0. Given that the server has no access to the true training data distribution, the root dataset is often biased in practice. We adopt the approach in [7] to model such bias. Among the $|D_0|$ root data samples, a fraction q_0 of them are sampled from a certain class c in the training data, and the rest are sampled from other classes with equal probabilities. For a dataset with C classes, D_0 is unbiased only when $q_0 = 1/C$. We set the size of root dataset $|D_0| = 100$ following [7].

Distribution learning setting. In distribution learning, we set the step size for inverting gradients $\eta' = 0.05$, the total variation parameter $\beta = 0.02$, optimizer as Adam, the number of iterations for inverting gradients $max\_iter = 10,000$, and learn the data distribution from scratch. The number of steps for distribution learning is set to $\tau_E = 100$. 32 images are reconstructed (i.e., $B' = 32$) and denoised in each FL epoch. If no attacker is selected in the current epoch, the aggregate gradient estimated from previous model updates is reused for reconstructing data. To build the denoising autoencoder, a Gaussian noise sampled from $0.3\mathcal{N}(0, 1)$ is added to each dimension of images in $D_{reconstructed}$, which are then clipped to the range of [0,1].

Policy learning setting. In policy learning, we implement our simulated environment with OpenAI Gym [6] and adopt OpenAI Stable Baseline3 [16] to implement Twin Delayed DDPG (TD3) [11] and Proximal Policy Optimization (PPO) [17] algorithms. The default parameters are described as follows : the length of simulating environment = $1,000$ epochs, policy learning rate = $1e - 7$, the policy model is $MultiInputPolicy$, batch size = 256 and gamma = 1 for updating the target networks. Note that the length of each simulating epoch is typically much shorter than the length of each real FL training epoch. In practice, the server usually needs to wait for some time (typically a few minutes) to receive the gradients from the clients before conducting model aggregation [25] [5] [12]. In addition,

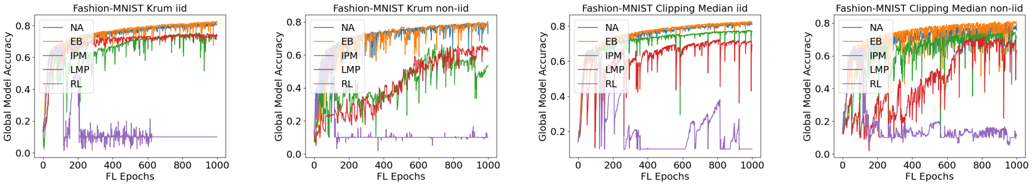

Figure 1: A comparison of global model accuracy on Fashion-MNIST under Krum and Clipping Median for both *i.i.d.* data and *non-i.i.d.* data. All parameters are set as default.

if the leader agent has access to GPUs or other parallel computing facilities, it can run multiple training episodes in parallel [8].

As described in Section 3.2, we compress the MDP state to include the the parameters of the last hidden layer of $\theta^{t(\tau)}$ and the number of attackers sampled, $m^{t(\tau)}$. We set the bound of each last hidden layer parameter to $[-\infty, +\infty]$ and the bound of $m^{t(\tau)}$ to $\{0, \dots, 10\}$. In our experiment, we restrict all attackers to take the same action in each epoch.

For the Krum defense and the Clipping Median defense, the local search objective is $F(\theta) = \mathbb{E}_{z \sim \widetilde{P}}[\ell(\theta; z)]$ (i.e., $\lambda = 0$). In this case, the action space becomes (γ, E), where $\gamma \in [0, 10]$ and $E \in \{0, \dots, 20\}$ for the Krum defense, and $\gamma \in [0, 10]$ and $E \in \{0, \dots, 50\}$ for the Clipping Median defense.

For FLTrust, we consider two cases, when the attackers have access to the server's root data D_0 or equivalently, the model updates g_0 in each epoch, and when they only know how D_0 is sampled from the true training data distribution. Note that even the former setting is more realistic than the adaptive LMP setting, which also requires access to normal workers' updates. In the former case, we fix $\gamma(\theta^{t(\tau)}) = \|g_0(\theta^{t(\tau)})\|_2$ and set the local search objective as $L(\theta) := (1-\lambda)F(\theta) + \lambda \cos(\theta^{t(\tau)} - \theta, g_0(\theta^{t(\tau)}))$ with the constraint that $\|\theta^{t(\tau)} - \theta\|_2 \leqslant \|g_0(\theta^{t(\tau)})\|_2$. In the latter case, we use the same objective but approximate $g_0(\theta^{t(\tau)})$ with $\mathbb{E}_{z \overset{q_0}{\sim} \widetilde{P}}[\nabla_\theta \ell(\theta^{t(\tau)}; z)]$, where q_0 models the bias of root data, which is assumed to be known to the attackers. In both cases, the action space is then (E, λ) with $E \in \{0, \dots, 20\}$ and $\lambda \in [0, 1]$. We further find that when the root data D_0 is known (or can be well approximated), the RL-based attack can be made more efficient by considering an alternate local search objective $L(\theta) := (1-\lambda)F(\theta) - \lambda F_0(\theta)$, where $F_0 = \frac{1}{|D_0|}\sum_{z \in D_0}[\ell(\theta; z)]$ is the empirical loss associated with the root data. Intuitively, the attackers aim to push the model parameters towards the region that can overfit the root data.

In our experiments, the initial model for all training episodes is set as the first model the attackers received from the actual FL environment. We assume that the server waits for 72 seconds to receive the updates from the workers before performing a model aggregation, which allows $80,000$ total time steps (i.e., 80 episodes) of policy learning for Krum, $40,000$ total time steps (i.e., 40 episodes) of policy learning for Clipping Median, and $40,000$ total time steps (i.e., 40 episodes) of policy learning for FLTrust within 400 FL epochs. It is more time consuming to train an RL policy for Clipping Median and FLTrust because large attack bounds need to be considered.

Attack execution setting. We observe that both EB and RL can occasionally produce NaNs in model updates, which when incorporated by the server, can lead to bad models in all future steps. This produces unrealistic attack scenarios as NaNs can be easily detected by the server. To have a fair comparison with other attacks, we use the built-in VecCheckNan Wrapper in OpenAI Stable Baseline3 [16] to detect abnormal values. We assume that attackers take less ambitious actions (i.e., $(0.5\gamma, E-1)$) in that epoch once they detect an NaN value. When $E = 0$ or $\gamma = 0$, the attackers will send $\widetilde{g}^{t(\tau)} = \mathbf{0}$ to the server. For our RL-based attack, both the distribution learning and policy learning phase start at the first FL epoch. The former ends at the 100th FL epoch when RL-based attack starts (all other attacks start at epoch 0). For fair comparisons, we fix all the random seeds for generating the initial model and the root data (for FLTrust), subsampling, and local data sampling when evaluating different attacks.

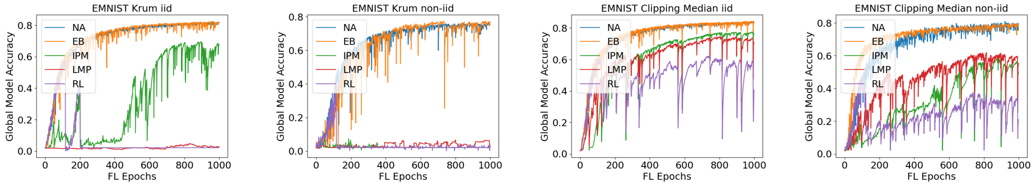

Figure 2: A comparison of global model accuracy on EMNIST under Krum and Clipping Median for both *i.i.d.* data and *non-i.i.d.* data. All parameters are set as default.

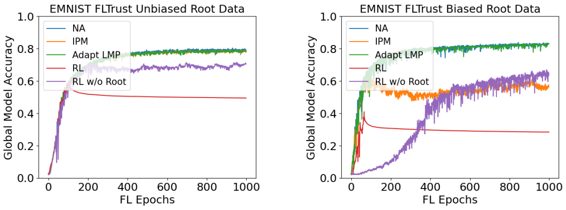

Figure 3: A comparison of global model accuracy on EMNIST under FLTrust defense with unbiased and biased root data. All parameters are set as default.

D.2 More Experiment Results

Attack performance under Fashion-MNIST and EMNIST. Figures 1 and 2 compare the test accuracy under different attacks when the server uses Krum or Clipping Median as the defense for both *i.i.d.* data and *non-i.i.d.* data ($q = 0.5$), on the Fashion-MNIST dataset and the EMNIST dataset, respectively. Our RL-based attack constantly outperforms other baselines by a large margin in all the settings. We observe that in most cases, all attacks are more effective in the *non-i.i.d.* setting. This is mainly because a higher degree of local data heterogeneity increases the variance across normal workers' updates, making it more difficult to filter out adversarial updates. Further, Clipping Median, which adopts both dimension-wise filtering and vector-wise norm clipping to model updates, provides a stronger level of defense than Krum, which only applies vector-wise filtering to model updates. In particular, our attack can reduce the model accuracy to an extremely low level ($\sim 10\%$ for Fashion-MNIST and $\sim 2\%$ for EMNIST) under the Krum defense, depending on the number of classes of the datasets.

Attack performance under FLTrust. We compare the attack performance of our RL-based attacks (details are given in D.1 policy learning setting) with and without access to server's root data and other baselines (i.e., NA, IPM, and adaptive LMP) against the FLTrust defense on the EMNIST dataset. For RL-based attacks, the attackers use all their local data to simulate the environment and skip the distribution learning phase. Thus, all attacks start from the beginning of FL. We consider both the cases when the root data are unbiased ($q_0 = 1/47$) and when they are biased against a single class ($q_0 = 0.3$). In the former case, our attack with access to root data leads to a significantly low test accuracy ($\sim 50\%$) as shown in Figure 3(left), while other attacks, including RL-based attack without access to root data, have limited effect against FLTrust. This is due to the fact that when the root data are unbiased and representative of the true training dataset, the server's update g_0 provides a good estimate of the right direction for model updates, making it difficult to reverse the trend. On the other hand, when the root data is biased, which is likely to happen in practice, the server's update g_0 is less representative or even misleading. Consequently, all attacks become more effective as shown in Figure 3(right). Further, both variants of our RL-based attack outperform other baselines.

Results for the synthetic data. In addition to the three real datasets discussed above, we also consider a two-dimensional synthetic dataset and a small network with 28 model parameters to demonstrate the full potential of our RL-based attack framework (i.e., without state and action compression). We generate the synthetic data based on the method described in [19]. In particular, we generate $55,000$ data instances (including $50,000$ training instances and $5,000$ testing instances), where for each instance $z = (x, y)$, the data $x \in \mathbb{R}^2 \sim \mathcal{N}(\mathbf{0}, I)$ and its label $y = \text{sign}(\|x\|_2) - 2$. Each worker has 500 data instances. We train a multilayer perceptron (MLP) with two hidden layers of size four and two, respectively, and use ReLU as the activation function. For our RL-based attack,

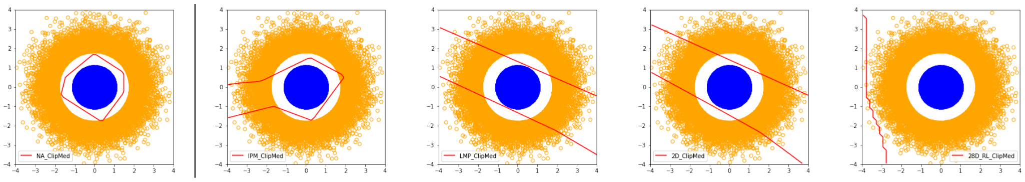

Figure 4: Classification boundaries of the final model on the synthetic data under various attacks and the Clipping Median defense. The classification accuracy of the final model: 100% (NA), 96.70% (IPM), 89.04% (LMP), 88.04% (RL with 2d actions), and 68.90% (RL with 28-dimensional actions). All parameters are set as default.

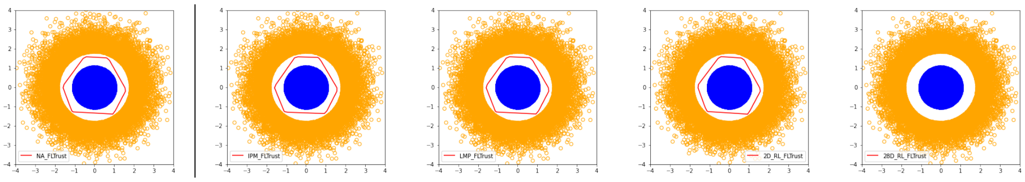

Figure 5: Classification boundaries of the final model on the synthetic data under various attacks and the FLTrust defense. The classification accuracy of the final model: 100% (NA), 100% (IPM), 100% (LMP), 100% (RL with 2d actions), and 68.90% (RL with 28-dimensional actions). All parameters are set as default.

we consider both the 2-dimensional action space (γ, E) discussed above as well as the general 28 dimensional action space where the attackers directly decide $\widetilde{g}_i(t(\tau))$ to be sent to the server in each epoch. In both cases, the state space includes the full 28 model parameters and the number of attackers in each epoch. Policy learning takes $8,000$ total time steps (i.e., 8 episodes) to learn the policy, within 10 FL epoch. The attackers use their local data $(10,000$ samples) to build simulated environment and no distribution learning is applied. Thus, the attack will immediately start once an attacker is selected. We fix all random seeds for fair comparisons across different attacks.

Figure 4 and Figure 5 illustrate the classification boundaries at the end of a federated learning episode for all the attacks when the Clipping Median defense and the FLTrust defense are applied respectively. The root dataset D_0 for FLTrust is assumed to be known for RL-based attacks. We observe that all baseline methods and our RL attack with 2d actions have a slight effect under Clipping Median or completely fail to compromise the system under FLTrust. On the other hand, the RL attack with the full 28-dimensional action space reduces the classification accuracy to 68.90% (worst-case accuracy for the given environment) under both defenses. These results indicate the potential of considering large state and action spaces in our RL-based attack when equipped with more computational power and longer training time.

References

[1] Kavosh Asadi and Michael L Littman. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning*, pages 243–252. PMLR, 2017.

[2] Kavosh Asadi, Dipendra Misra, and Michael Littman. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 264–273. PMLR, 2018.

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

[4] Peva Blanchard, Rachid Guerraoui, Julien Stainer, et al. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, pages 119–129, 2017.

[5] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

[6] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[7] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. FLTrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv preprint arXiv:2012.13995*, 2020.

[8] Alfredo V Clemente, Humberto N Castejón, and Arjun Chandra. Efficient parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1705.04862*, 2017.

[9] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[10] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to byzantine-robust federated learning. In *29th USENIX Security Symposium*, pages 1605–1622, 2020.

[11] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[12] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[13] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[14] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[15] Boris T. Polyak. *Introduction to optimization*. Optimization Software, 1987.

[16] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[18] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *IEEE Symposium on Security and Privacy*, 2022.

[19] Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

[20] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019.

[21] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[22] Leonid Nisonovich Vaserstein. Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii*, 5(3):64–72, 1969.

[23] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[24] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *Uncertainty in Artificial Intelligence*, pages 261–270. PMLR, 2020.

[25] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*, 2018.

[26] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. *arXiv preprint arXiv:1803.01498*, 2018.

[27] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *arXiv preprint arXiv:2005.13239*, 2020.