A Interpretation of Objective in Equation 5

We note that the ultimate goal of the defender is to obtain a clean model while the attacker wants the global model to be poisoned. In other words, the server (or attacker) wishes the attack success rate of the global model to be small (large). This objective is very challenging to directly achieve because 1) the server does not know the exact trigger and target class used by the attacker, and 2) the attacker needs to adapt its attack strategy based on the server's strategy. To address the challenge, we consider an alternative goal (i.e., our objective in Equation 5, where we wish to assign small weights to compromised clients when performing a weighted average to update the global model. Our idea is that when the weights for compromised clients are very small, our global model is less likely to be affected by the attack. As a result, the attacker's strategy is to maximize the genuine scores for compromised clients while ensuring their local models are backdoored to maximize the attack effectiveness for the global model. Thus, our objective in Equation 5 could translate to the ultimate goal. Our empirical results show the effectiveness of our defense.

B Complete Proofs

B.1 Proof of Lemma 5.3

We first present some preliminary lemmas that will be invoked for proving Lemma 5.3.

Lemma B.1. *Suppose $\mathcal{D}_i$ is the clean local training dataset of the client i. An attacker can inject the backdoor trigger to r_i^t fraction of training examples in $\mathcal{D}_i$ and relabel them as the target class. We use $\mathcal{D}_i'$ to denote the set of backdoored training examples where $r_i^t = \frac{|\mathcal{D}_i'|}{|\mathcal{D}_i|}$. Given two arbitrary Θ and Θ_c, we let $g_i = \frac{1}{|\mathcal{D}_i \cup \mathcal{D}_i'|} \nabla_\Theta \sum_{\mathbf{z} \in \mathcal{D}_i \cup \mathcal{D}_i'} \ell(\mathbf{z}; \Theta)$ and $h_i = \frac{1}{|\mathcal{D}_i|} \nabla_{\Theta_c} \sum_{\mathbf{z} \in \mathcal{D}_i} \ell(\mathbf{z}; \Theta_c)$. We then have that*

$$(\Theta - \Theta_c)^T (g_i - h_i) \geq (0.5\mu - r_i^t M) \|\Theta - \Theta_c\|_2^2 - r_i^t M, \tag{12}$$

$$\|g_i - h_i\|_2 \leq L \|\Theta - \Theta_c\|_2 + 2 r_i^t M. \tag{13}$$

Proof. We first prove Equation 12. We have the following relations:

$$\begin{aligned}
&(\Theta - \Theta_c)^T (g_i - h_i) \\
=&(\Theta - \Theta_c)^T \Big(\frac{1}{|\mathcal{D}_i \cup \mathcal{D}_i'|} \sum_{\mathbf{z}' \in \mathcal{D}_i \cup \mathcal{D}_i'} \nabla_\Theta \ell(\mathbf{z}'; \Theta) - \frac{1}{|\mathcal{D}_i|} \sum_{\mathbf{z} \in \mathcal{D}_i} \nabla_{\Theta_c} \ell(\mathbf{z}; \Theta_c)\Big) \quad \triangleright \text{definition of } g_i \text{ and } h_i
\end{aligned} \tag{14}$$

$$=(\Theta - \Theta_c)^T \Big(\frac{1}{(1 + r_i^t)|\mathcal{D}_i|} \sum_{\mathbf{z}' \in \mathcal{D}_i \cup \mathcal{D}_i'} \nabla_\Theta \ell(\mathbf{z}'; \Theta) - \frac{1}{|\mathcal{D}_i|} \sum_{\mathbf{z} \in \mathcal{D}_i} \nabla_{\Theta_c} \ell(\mathbf{z}; \Theta_c)\Big) \quad \triangleright r_i^t = \frac{|\mathcal{D}_i'|}{|\mathcal{D}_i|} \tag{15}$$

$$=\frac{1}{|\mathcal{D}_i|(1 + r_i^t)} (\Theta - \Theta_c)^T \Big(\sum_{\mathbf{z}' \in \mathcal{D}_i \cup \mathcal{D}_i'} \nabla_\Theta \ell(\mathbf{z}'; \Theta) - (1 + r_i^t) \sum_{\mathbf{z} \in \mathcal{D}_i} \nabla_{\Theta_c} \ell(\mathbf{z}; \Theta_c)\Big) \tag{16}$$

$$\begin{aligned}
=&\frac{1}{|\mathcal{D}_i|(1 + r_i^t)} (\Theta - \Theta_c)^T \Big(\sum_{\mathbf{z}' \in \mathcal{D}_i} \nabla_\Theta \ell(\mathbf{z}'; \Theta) - \sum_{\mathbf{z} \in \mathcal{D}_i} \nabla_{\Theta_c} \ell(\mathbf{z}; \Theta_c) \\
&+ \sum_{\mathbf{z}' \in \mathcal{D}_i'} \nabla_\Theta \ell(\mathbf{z}'; \Theta) - r_i^t \sum_{\mathbf{z} \in \mathcal{D}_i} \nabla_{\Theta_c} \ell(\mathbf{z}; \Theta_c)\Big)
\end{aligned} \tag{17}$$

$$\begin{aligned}
=&\frac{1}{|\mathcal{D}_i|(1 + r_i^t)} \Big(\sum_{\mathbf{z} \in \mathcal{D}_i} (\Theta - \Theta_c)^T (\nabla_\Theta \ell(\mathbf{z}; \Theta) - \nabla_{\Theta_c} \ell(\mathbf{z}; \Theta_c)) \\
&+ (\Theta - \Theta_c)^T \Big(\sum_{\mathbf{z}' \in \mathcal{D}_i'} \nabla_\Theta \ell(\mathbf{z}'; \Theta) - r_i^t \sum_{\mathbf{z} \in \mathcal{D}_i} \nabla_{\Theta_c} \ell(\mathbf{z}; \Theta_c)\Big)\Big)
\end{aligned} \tag{18}$$

$$\begin{aligned}
\geq&\frac{1}{|\mathcal{D}_i|(1 + r_i^t)} \Big(\sum_{\mathbf{z} \in \mathcal{D}_i} (\Theta - \Theta_c)^T (\nabla_\Theta \ell(\mathbf{z}; \Theta) - \nabla_{\Theta_c} \ell(\mathbf{z}; \Theta_c)) \\
&- \|(\Theta - \Theta_c)^T \Big(\sum_{\mathbf{z}' \in \mathcal{D}_i'} \nabla_\Theta \ell(\mathbf{z}'; \Theta) - r_i^t \sum_{\mathbf{z} \in \mathcal{D}_i} \nabla_{\Theta_c} \ell(\mathbf{z}; \Theta_c)\Big)\|_1\Big) \quad \triangleright \forall x, x \geq -\|x\|_1
\end{aligned} \tag{19}$$

$$\geq \frac{1}{|\mathcal{D}_i|(1+r_i^t)}(\sum_{\mathbf{z}\in\mathcal{D}_i}(\Theta-\Theta_c)^T(\nabla_\Theta\ell(\mathbf{z};\Theta)-\nabla_{\Theta_c}\ell(\mathbf{z};\Theta_c))$$

$$-\|\Theta-\Theta_c\|_2\cdot\|\sum_{\mathbf{z}'\in\mathcal{D}_i'}\nabla_\Theta\ell(\mathbf{z}';\Theta)-r_i^t\sum_{\mathbf{z}\in\mathcal{D}_i}\nabla_{\Theta_c}\ell(\mathbf{z};\Theta_c)\|_2) \quad \triangleright \text{ Cauchy–Schwarz inequality}$$

$$\geq \frac{1}{|\mathcal{D}_i|(1+r_i^t)}(\sum_{\mathbf{z}\in\mathcal{D}_i}(\Theta-\Theta_c)^T(\nabla_\Theta\ell(\mathbf{z};\Theta)-\nabla_{\Theta_c}\ell(\mathbf{z};\Theta_c))$$

$$-\|\Theta-\Theta_c\|_2\cdot(\sum_{\mathbf{z}'\in\mathcal{D}_i'}\|\nabla_\Theta\ell(\mathbf{z}';\Theta)\|_2+r_i^t\sum_{\mathbf{z}\in\mathcal{D}_i}\|\nabla_{\Theta_c}\ell(\mathbf{z};\Theta_c)\|_2) \quad \triangleright \text{ triangle inequality}$$

$$\geq \frac{1}{|\mathcal{D}_i|(1+r_i^t)}(\mu|\mathcal{D}_i|\,\|\Theta-\Theta_c\|_2^2-2r_i^t|\mathcal{D}_i|M\,\|\Theta-\Theta_c\|_2) \quad \triangleright \text{ Assumption 5.1} \tag{20}$$

$$=\frac{\mu}{1+r_i^t}\|\Theta-\Theta_c\|_2^2-\frac{1}{1+r_i^t}2r_i^tM\,\|\Theta-\Theta_c\|_2) \tag{21}$$

$$\geq 0.5\mu\,\|\Theta-\Theta_c\|_2^2-2r_i^tM\,\|\Theta-\Theta_c\|_2 \quad \triangleright r_i^t\in[0,1] \tag{22}$$

$$\geq 0.5\mu\,\|\Theta-\Theta_c\|_2^2-r_i^tM\,\|\Theta-\Theta_c\|_2^2-r_i^tM) \tag{23}$$

$$=(0.5\mu-r_i^tM)\,\|\Theta-\Theta_c\|_2^2-r_i^tM, \tag{24}$$

where Equation 23 holds based on the fact that $-2r_i^tM\,\|\Theta-\Theta_c\|_2\geq -r_i^tM\,\|\Theta-\Theta_c\|_2^2-r_i^tM$ for $\forall r_i^t\geq 0$ and $\forall M\geq 0$.

In the following, we prove inequality 13. We have that

$$\|g_i-h_i\|_2$$

$$=\frac{1}{|\mathcal{D}_i|(1+r_i^t)}\|\sum_{\mathbf{z}'\in\mathcal{D}_i\cup\mathcal{D}_i'}\nabla_\Theta\ell(\mathbf{z}';\Theta)-(1+r_i^t)\sum_{\mathbf{z}\in\mathcal{D}_i}\nabla_{\Theta_c}\ell(\mathbf{z};\Theta_c)\|_2 \quad \triangleright \text{ definition of } g_i \text{ and } h_i \tag{25}$$

$$=\frac{1}{|\mathcal{D}_i|(1+r_i^t)}\|\sum_{\mathbf{z}'\in\mathcal{D}_i'}\nabla_\Theta\ell(\mathbf{z}';\Theta)+\sum_{\mathbf{z}'\in\mathcal{D}_i}\nabla_\Theta\ell(\mathbf{z}';\Theta)-(1+r_i^t)\sum_{\mathbf{z}\in\mathcal{D}_i}\nabla_{\Theta_c}\ell(\mathbf{z};\Theta_c)\|_2 \tag{26}$$

$$\leq\frac{1}{|\mathcal{D}_i|(1+r_i^t)}\|\sum_{\mathbf{z}'\in\mathcal{D}_i'}\nabla_\Theta\ell(\mathbf{z}';\Theta)-r_i^t\sum_{\mathbf{z}\in\mathcal{D}_i}\nabla_{\Theta_c}\ell(\mathbf{z};\Theta_c)\|_2$$

$$+\frac{1}{|\mathcal{D}_i|(1+r_i^t)}\|\sum_{\mathbf{z}'\in\mathcal{D}_i}\nabla_\Theta\ell(\mathbf{z}';\Theta)-\sum_{\mathbf{z}\in\mathcal{D}_i}\nabla_{\Theta_c}\ell(\mathbf{z};\Theta_c)\|_2 \quad \triangleright \text{ triangle inequality} \tag{27}$$

$$\leq\frac{1}{1+r_i^t}(2r_i^tM+L\|\Theta-\Theta_c\|_2) \tag{28}$$

$$\leq 2r_i^tM+L\|\Theta-\Theta_c\|_2 \quad \triangleright r_i^t\in[0,1] \tag{29}$$

where Equation 28 is due to Assumption 5.1 and 5.2. □

Given Lemma B.1, we prove Lemma 5.3 as follows. Recall that we have $\alpha_i^t=\frac{p_i^t}{\sum_{i\in S}p_i^t}$ and $\beta_i^t=\frac{q_i^t}{\sum_{i\in S}q_i^t}$.

$$\|\Theta^{t+1}-\Theta_c^{t+1}\|_2 \tag{30}$$

$$=\|\Theta^t-\eta\sum_{i\in\mathcal{S}}\alpha_i^tg_i^t-(\Theta_c^t-\eta\sum_{i\in\mathcal{S}}\beta_i^th_i^t)\|_2 \quad \triangleright \text{ gradient descent for } \Theta^{t+1} \text{ and } \Theta_c^{t+1} \tag{31}$$

$$=\|\Theta^t-\eta\sum_{i\in\mathcal{S}}\alpha_i^tg_i^t-(\Theta_c^t-\eta\sum_{i\in\mathcal{S}}(\alpha_i^t+\beta_i^t-\alpha_i^t)h_i^t)\|_2 \tag{32}$$

$$=\|\Theta^t-\Theta_c^t-\eta\sum_{i\in\mathcal{S}}\alpha_i^t(g_i^t-h_i^t)+(\eta\sum_{i\in\mathcal{S}}(\beta_i^t-\alpha_i^t)h_i^t)\|_2 \quad \triangleright \text{ rearranging Equation 32} \tag{33}$$

$$\le\|\Theta^t-\Theta_c^t-\eta\sum_{i\in\mathcal{S}}\alpha_i^t(g_i^t-h_i^t)\|_2+\|\eta\sum_{i\in\mathcal{S}}(\beta_i^t-\alpha_i^t)h_i^t\|_2. \quad \triangleright \text{triangle inequality} \tag{34}$$

Next, we respectively derive an upper bound for the first and second terms in Equation 34. To derive the upper bound for the first term, we have that

$$\begin{aligned}
&\|\Theta^t-\Theta_c^t-\eta\sum_{i\in\mathcal{S}}\alpha_i^t(g_i^t-h_i^t)\|_2^2\\
=&\|\Theta^t-\Theta_c^t\|_2^2-2\eta(\Theta^t-\Theta_c^t)^T(\sum_{i\in\mathcal{S}}\alpha_i^t(g_i^t-h_i^t))+\eta^2\|\sum_{i\in\mathcal{S}}\alpha_i^t(g_i^t-h_i^t)\|_2^2 &(35)\\
=&S_1+S_2+S_3, &(36)
\end{aligned}$$

where $S_1=\left\|\Theta^t-\Theta_c^t\right\|_2^2$, $S_2=-2\eta(\Theta^t-\Theta_c^t)^T(\sum_{i\in\mathcal{S}}\alpha_i^t(g_i^t-h_i^t))$, and $S_3=\eta^2\left\|\sum_{i\in\mathcal{S}}\alpha_i^t(g_i^t-h_i^t)\right\|_2^2$. Next, we will bound S_2 and S_3. We denote $\gamma^t=\sum_{i\in\mathcal{S}_a}\alpha_i^t r_i^t M$. Note that we have $\gamma^t=\sum_{i\in\mathcal{S}}\alpha_i^t r_i^t M$ since $r_i^t=0$ for $\forall i\in\mathcal{S}\setminus\mathcal{S}_a$. We bound S_2 as follows.

$$\begin{aligned}
&S_2\\
=&-2\eta(\Theta^t-\Theta_c^t)^T(\sum_{i\in\mathcal{S}}\alpha_i^t(g_i^t-h_i^t)) &(37)\\
=&-2\eta\sum_{i\in\mathcal{S}}\alpha_i^t(\Theta^t-\Theta_c^t)^T(g_i^t-h_i^t) &(38)\\
\le&-2\eta\sum_{i\in\mathcal{S}}\alpha_i^t((0.5\mu-r_i^tM)\left\|\Theta^t-\Theta_c^t\right\|_2^2-r_i^tM) &(39)\\
=&-2\eta((0.5\mu-\sum_{i\in\mathcal{S}}\alpha_i^t r_i^tM)\left\|\Theta^t-\Theta_c^t\right\|_2^2-\sum_{i\in\mathcal{S}_a}\alpha_i^t r_i^tM) &(40)\\
=&(-\eta\mu+2\eta\gamma^t)\left\|\Theta^t-\Theta_c^t\right\|_2^2+2\eta\gamma^t, \quad \triangleright\text{definition of } \gamma^t &(41)
\end{aligned}$$

where inequality 39 holds by Lemma B.1 and the fact that $\eta,\alpha_i^t\ge 0$. We bound S_3 as follows.

$$\begin{aligned}
&S_3\\
=&\eta^2\|\sum_{i\in\mathcal{S}}\alpha_i^t(g_i^t-h_i^t)\|_2^2 &(42)\\
\le&\eta^2(\sum_{i\in\mathcal{S}}\alpha_i^t\left\|(g_i^t-h_i^t)\right\|_2)^2 &(43)\\
\le&\eta^2(\sum_{i\in\mathcal{S}}\alpha_i^t(2r_i^tM+L\|\Theta-\Theta_c\|_2)^2 \quad \triangleright \text{Lemma B.1} &(44)\\
=&\eta^2(2\gamma^t+L\|\Theta-\Theta_c\|_2)^2 &(45)\\
=&\eta^2(L^2\left\|\Theta-\Theta_c\right\|_2^2+4\gamma^tL\left\|\Theta-\Theta_c\right\|_2+4[\gamma^t]^2) &(46)\\
\le&\eta^2(L^2\left\|\Theta-\Theta_c\right\|_2^2+2\gamma^tL\left\|\Theta-\Theta_c\right\|_2^2+2L\gamma^t+4[\gamma^t]^2) &(47)\\
=&\eta^2\cdot((L^2+2L\gamma^t)\cdot\left\|\Theta-\Theta_c\right\|_2^2+2L\gamma^t+4[\gamma^t]^2) &(48)
\end{aligned}$$

where Equation 47 is based on the fact that $4\gamma^tL\left\|\Theta-\Theta_c\right\|_2\le 2\gamma^tL\left\|\Theta-\Theta_c\right\|_2^2+2\gamma^tL$ when $\gamma^tL\ge 0$.

Given the upper bounds of S_2 and S_3, we can bound $\left\|\Theta^t-\Theta_c^t-\eta\sum_{i\in\mathcal{S}}\alpha_i^t(g_i^t-h_i^t)\right\|_2^2$ as follows.

$$\begin{aligned}
&\|\Theta^t-\Theta_c^t-\eta\sum_{i\in\mathcal{S}}\alpha_i^t(g_i^t-h_i^t)\|_2^2 &(49)\\
=&S_1+S_2+S_3 &(50)\\
\le&\left\|\Theta-\Theta_c\right\|_2^2+(-\eta\mu+2\eta\gamma^t)\left\|\Theta^t-\Theta_c^t\right\|_2^2+2\eta\gamma^t
\end{aligned}$$

$$+ (\eta^2 L^2 + \eta^2 2L\gamma^t)\left\|\Theta^t - \Theta_c^t\right\|_2^2 + \eta^2 2L\gamma^t + \eta^2 4[\gamma^t]^2 \tag{51}$$

$$=(1-\eta\mu+2\eta\gamma^t+\eta^2L^2+2\eta^2L\gamma^t)\left\|\Theta^t-\Theta_c^t\right\|_2^2+2\eta\gamma^t+2\eta^2L\gamma^t+4\eta^2[\gamma^t]^2 \tag{52}$$

Next, we will derive an upper bound for $\left\|\eta\sum_{i\in\mathcal{S}}(\beta_i^t-\alpha_i^t)h_i^t\right\|_2$. We denote $r^t=\sum_{i\in\mathcal{S}_a} r_i^t$. Note that we have that $r^t=\sum_{i\in\mathcal{S}} r_i^t$ also holds since $r_i^t=0$ for $\forall i\in\mathcal{S}\setminus\mathcal{S}_a$. Given the assumption that $(1-r^t)\alpha_i^t\le\beta_i^t\le(1+r^t)\alpha_i^t$, we have

$$\|\eta\sum_{i\in\mathcal{S}}(\beta_i^t-\alpha_i^t)h_i^t\|_2 \le \eta\sum_{i\in\mathcal{S}}|\beta_i^t-\alpha_i^t|\left\|h_i^t\right\|_2\le 2\eta r^t M, \tag{53}$$

where the first inequality is due to triangle inequality and the second inequality is based on the assumption that $\|h_i^t\|_2 \le M$. Therefore, we have:

$$\|\Theta^{(t+1)}-\Theta_c^{(t+1)}\|_2$$

$$\le\|\Theta^t-\Theta_c^t-\eta\sum_{i\in\mathcal{S}}\alpha_i^t(g_i^t-h_i^t)\|_2^2+\|\eta\sum_{i\in\mathcal{S}}(\beta_i^t-\alpha_i^t)h_i^t\|_2 \quad \triangleright \text{Equation } 30,\ 34 \tag{54}$$

$$\le\sqrt{(1-\eta\mu+2\eta\gamma^t+\eta^2L^2+2\eta^2L\gamma^t)\left\|\Theta^t-\Theta_c^t\right\|_2^2+2\eta\gamma^t(1+\eta L+2\eta\gamma^t)} \tag{55}$$

$$+\,2\eta r^t M \quad \triangleright \text{Equation } 49,\ 52,\ 53 \tag{56}$$

$$\le\sqrt{1-\eta\mu+2\eta\gamma^t+\eta^2L^2+2\eta^2L\gamma^t}\left\|\Theta^t-\Theta_c^t\right\|_2+\sqrt{2\eta\gamma^t(1+\eta L+2\eta\gamma^t)}+2\eta r^t M, \tag{57}$$

where the last inequality holds due to the fact that $\sqrt{a+b}\le\sqrt{a}+\sqrt{b}$ for $\forall a\ge 0$ and $\forall b\ge 0$, which completes our proof for Lemma 5.3.

B.2 Proof of Theorem 5.4

We denote $A_t=\sqrt{1-\eta\mu+2\eta\gamma^t+\eta^2L^2+2\eta^2L\gamma^t}$, $A=\sqrt{1-\eta\mu+2\eta\gamma+\eta^2L^2+2\eta^2L\gamma}$, $B_t=\sqrt{2\eta\gamma^t(1+\eta L+2\eta\gamma^t)}+2\eta r^t M$, and $B=\sqrt{2\eta\gamma(1+\eta L+2\eta\gamma)}+2\eta rM$. Since $\gamma^t\le\gamma$ and $r^t\le r$, we have $A_t\le A$ and $B_t\le B$. Thus, based on Lemma 5.3, we have:

$$\left\|\Theta^t-\Theta_c^t\right\|_2\le A\left\|\Theta^{t-1}-\Theta_c^{t-1}\right\|_2+B. \tag{58}$$

Then, we can iteratively apply the above equation to prove our theorem. In particular, we have:

$$\left\|\Theta^t-\Theta_c^t\right\|_2$$

$$\le A\left\|\Theta^{t-1}-\Theta_c^{t-1}\right\|_2+B \tag{59}$$

$$\le A(A\left\|\Theta^{t-2}-\Theta_c^{t-2}\right\|_2+B)+B \tag{60}$$

$$=A^2\left\|\Theta^{t-2}-\Theta_c^{t-2}\right\|_2+(A^1+A^0)B \tag{61}$$

$$\le A^t\left\|\Theta^0-\Theta_c^0\right\|_2+(A^{t-1}+A^{t-2}+\cdots+A^0)B \tag{62}$$

$$=A^t\left\|\Theta^0-\Theta_c^0\right\|_2+\frac{1-A^t}{1-A}B \tag{63}$$

$$=(\sqrt{1-\eta\mu+2\eta\gamma+\eta^2L^2+2\eta^2L\gamma})^t\left\|\Theta^0-\Theta_c^0\right\|_2$$

$$+\frac{1-(\sqrt{1-\eta\mu+2\eta\gamma+\eta^2L^2+2\eta^2L\gamma})^t}{1-\sqrt{1-\eta\mu+2\eta\gamma+\eta^2L^2+2\eta^2L\gamma}}(\sqrt{2\eta\gamma(1+\eta L+2\eta\gamma)}+2\eta rM), \tag{64}$$

When the learning rate satisfies $0<\eta<\frac{\mu-2\gamma}{L^2+2L\gamma}$, we have that $0<1-\eta\mu+2\eta\gamma+\eta^2L^2+2\eta^2L\gamma<1$. Therefore, the upper bound becomes $\frac{\sqrt{2\eta\gamma(1+\eta L+2\eta\gamma)}+2\eta rM}{1-\sqrt{1-\eta\mu+2\eta\gamma+\eta^2L^2+2\eta^2L\gamma}}$ as $t\to\infty$. Hence, we prove our Theorem 5.4.

C Complete Algorithms

C.1 Complete Algorithm of FedGame

Algorithm 1 shows the complete algorithm of FedGame. In Line 3, we construct an auxiliary global model. In Line 4, the function ReverseEngineer is used to reverse engineer the backdoor trigger and target class. In Line 6, we compute the local model of client i based on its local model update. In Line 7, we compute a genuine score for client i. In Line 9, we update the global model based on genuine scores and local model updates of clients.

Algorithm 1 FLGAME

```
Input: $\Theta^t$ (global model in the $t^{\text{th}}$ communication round), $g_i^t, i \in \mathcal{S}$ (local model updates of clients), $\mathcal{D}_s$ (clean training dataset of server), $\eta$ (learning rate of global model).
Output: $\Theta^{t+1}$ (global model for the $(t+1)^{\text{th}}$ communication round)
$\Theta_a^t = \Theta^t + \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} g_i^t$
$\delta_{re}, y_{re}^{tc} = \textsc{ReverseEngineer}(\Theta_a^t)$
for $i \in \mathcal{S}$ do
    $\Theta_i^t = \Theta^t + g_i^t$
    $p_i^t = 1 - \frac{1}{|\mathcal{D}_s|} \sum_{\mathbf{x} \in \mathcal{D}_s} \mathbb{I}(G(\mathbf{x} \oplus \delta_{re}; \Theta_i^t) = y_{re}^{tc})$
end for
$\Theta^{t+1} = \Theta^t + \eta \frac{1}{\sum_{i \in \mathcal{S}} p_i^t} \sum_{i \in \mathcal{S}} p_i^t g_i^t$
return $\Theta^{t+1}$
```

C.2 Complete Algorithm for a Compromised Client

Algorithm 2 shows the complete algorithm for a compromised client. In Line 3, we randomly subsample ρ_i fraction of training data from $\mathcal{D}_i$. In Line 7, the function CREATEBACKDOOREDDATA is used to generate backdoored training examples by embedding the backdoor trigger δ to $\lfloor \min(j * \zeta, 1) |\mathcal{D}_i \setminus \mathcal{D}_i^{rev}| \rfloor$ training examples in $\mathcal{D}_i \setminus \mathcal{D}_i^{rev}$ and relabel them as y^{tc}, where $| \cdot |$ measures the number of elements in a set. In Line 8, the function TRAININGLOCALMODEL is used to train the local model on the training dataset $\mathcal{D}'_i \cup \mathcal{D}_i \setminus \mathcal{D}_i^{rev}$. In Line 9, we estimate a genuine score. In Line 15, we use the function CREATEBACKDOOREDDATA to generate backdoored training examples by embedding the backdoor trigger δ to $\lfloor \min(o * \zeta, 1) |\mathcal{D}_i| \rfloor$ training examples in $\mathcal{D}_i$ and relabel them as y^{tc}. In Line 16, we use the function TRAININGLOCALMODEL to train a local model and utilize existing state-of-the-art attacks to inject the backdoor based on the training dataset $\mathcal{D}'_i \cup \mathcal{D}_i$.

Algorithm 2 ALGORITHM FOR A COMPROMISED CLIENT

```
1: Input: $\Theta^t$ (global model in the $t^{\text{th}}$ communication round), $\mathcal{D}_i$ (local training dataset of client $i$), $\rho_i$ (fraction of reserved data to find optimal $r_i^t$), $\zeta$ (granularity of searching for $r_i^t$), $\delta$ (backdoor trigger), $y^{tc}$ (target class), and $\lambda$ (hyperparameter)
2: Output: $g_i^t$ (local model update)
3: $\mathcal{D}_i^{rev} = \textsc{RandomSampling}(\mathcal{D}_i, \rho_i)$
4: $count = \lceil \frac{1}{\zeta} \rceil$
5: $max\_value, o \leftarrow 0, 0$
6: for $j \leftarrow 0$ to count do
7:     $\mathcal{D}'_i = \textsc{CreateBackdooredData}(\mathcal{D}_i \setminus \mathcal{D}_i^{rev}, \delta, y^{tc}, \min(j * \zeta, 1))$
8:     $\Theta_{ij} = \textsc{TrainingLocalModel}(\Theta^t, \mathcal{D}'_i \cup \mathcal{D}_i \setminus \mathcal{D}_i^{rev})$
9:     $p_{ij} = 1 - \frac{1}{|\mathcal{D}_i^{rev}|} \sum_{\mathbf{x} \in \mathcal{D}_i^{rev}} \mathbb{I}(G(\mathbf{x} \oplus \delta; \Theta_{ij}) = y^{tc})$
10:    if $p_{ij} + \lambda \min(j * \zeta, 1) > max\_value$ then
11:        $o = j$
12:        $max\_value = p_{ij} + \lambda \min(j * \zeta, 1)$
13:    end if
14: end for
15: $\mathcal{D}'_i = \textsc{CreateBackdooredData}(\mathcal{D}_i, \delta, y^{tc}, \min(o * \zeta, 1))$
16: $\Theta_i^t$ = $\textsc{TrainingLocalModel}(\Theta^t, \mathcal{D}'_i \cup \mathcal{D}_i)$
17: return $\Theta_i^t - \Theta^t$
```

D Additional Experimental Setup and Results

D.1 Architecture of Global Model

Table 2 shows the global model architecture on MNIST dataset.

D.2 Parameter Setting for Compared Baselines

Recall that we compare our defense with the following methods: FedAvg [24], Krum [5], Median [51], Norm-Clipping [35], Differential Privacy (DP) [35], DeepSight [32], and FLTrust [6]. FedAvg is non-robust while Krum and Median are two Byzantine-robust baselines. Norm-Clipping clips the L_2-norm of local model updates to a given threshold $\mathcal{T}_N$. We set $\mathcal{T}_N = 0.01$ for MNIST and $\mathcal{T}_N = 0.1$ for CIFAR10. DP first clips the L_2-norm of a local model update to a threshold $\mathcal{T}_D$ and then adds Gaussian noise. We set $\mathcal{T}_D = 0.05$ for MNIST and $\mathcal{T}_D = 0.5$ for CIFAR10. We set the standard deviation of noise to be 0.01 for both datasets. In FLTrust, the server uses its clean dataset to compute a server model update and assigns a trust score to each client by leveraging the similarity between the server model update and the local model update. We set the clean training dataset of the server to be the same as FedGame in our comparison. Note that FLTrust is not applicable when the clean training dataset of the server is from a different domain from those of clients.

Table 2: Architecture of the convolutional neural network for MNIST.

Type	Parameters
Convolution	3×3, stride=1, 16 kernels
Activation	ReLU
Max Pooling	2×2
Convolution	4×4, stride=2, 32 kernels
Activation	ReLU
Max Pooling	2×2
Fully Connected	800×500
Activation	ReLU
Fully Connected	500×10

D.3 Performance of FedGame against Neurotoxin

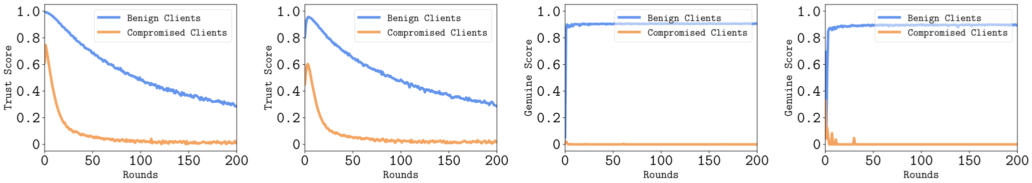
In Table 3, we compare our FedGame with other defense baselines against Neurotoxin [54]. We can observe that our FedGame is consistently more effective than existing defenses. Our observation is consistent with the experimental results for Scaling attack and DBA attack in Table 1.

Table 3: Comparison of FedGame with existing defenses against Neurotoxin on MNIST under IID setting. The total number of clients is 10 with 60% compromised. The best results when respectively comparing FedGame in each setting with existing defenses are bold.

Metrics	FedAvg (No attacks)	Defenses (Under attacks)								
									FedGame	
		FedAvg	Krum	Median	Norm-Clipping	DP	Deep-Sight	FLTrust	In-domain	Out-of-domain
TA (%)	99.04	99.02	99.32	99.08	90.75	95.28	96.36	95.73	97.27	97.33
ASR (%)	9.69	99.97	99.98	99.99	99.36	99.27	89.02	13.02	**9.93**	**10.03**

D.4 Visualization of genuine score of FedGame and trust score of FLTrust [6]

Our FedGame computes a genuine score for each client which quantifies the extent to which a client is benign in each communication round. Intuitively, our FedGame would be effective if the genuine score is small for a compromised client but is large for a benign one. FLTrust [6] computes a trust score for each client in each communication round. Similarly, FLTrust would be effective if the trust score is small for a compromised client but is large for a benign one. Figure 2 visualizes the average genuine or trust scores for compromised and benign clients of FedGame and FLTrust on MNIST dataset. We have the following observations from the figures. First, the average genuine score computed by FedGame drops to 0 quickly for compromised clients. In contrast, the average trust score computed by FLTrust drops slowly. Second, the average genuine score computed by FedGame for benign clients first increases and then becomes stable. In contrast, the average genuine score computed by FLTrust for benign clients decreases as the number of iterations increases. As a result, our FedGame outperforms FLTrust.



(a) FLTrust (IID) (b) FLTrust (non-IID) (c) FedGame (IID) (d) FedGame (non-IID)

Figure 2: (a)(b): Server-computed average trust scores for benign and compromised clients of FLTrust on MNIST under Scaling attack. (c)(d): Average genuine scores computed by the server for benign and compromised clients of FedGame on MNIST under Scaling attack. The clean sets of the server are the same for FLTrust and FedGame.

D.5 FedGame Performance in FL Consisting of 30 Clients

In Table 4, we report the performance of FedGame and baselines when the total number of clients is 30. The results also indicate that our FedGame outperforms all baselines in terms of ASR and achieves comparable TA with existing methods.

Table 4: Comparison of FedGame with existing defenses under Scaling attack. The total number of clients is 30 with 60% compromised. The best results when respectively comparing FedGame in each setting with existing defenses are bold.

Datasets	Metrics	FedAvg (No attacks)	Defenses (Under attacks)							
			FedAvg	Krum	Median	Norm-Clipping	DP	FLTrust	FedGame	
									In-domain	Out-of-domain
MNIST	TA (%)	99.02	99.09	98.16	99.01	92.77	89.77	95.27	97.81	97.64
	ASR (%)	9.74	99.98	99.98	99.98	98.20	98.83	11.04	**9.95**	**9.95**
CIFAR10	TA (%)	80.08	79.73	72.23	79.58	79.20	50.86	67.84	73.29	74.42
	ASR (%)	9.14	99.82	99.97	99.85	99.87	96.53	99.28	**10.44**	**9.15**

D.6 Additional Ablation Studies

Impact of the total number of clients. We study the impact of the total number of clients for our FedGame under the default setting. In particular, we consider the total number of clients to be 10, 30, 50, 70, and 100, where the fraction of malicious clients is 60%. We show the experimental results in Table 5. Our experimental results show that our FedGame is effective for different number of clients on different datasets.

Impact of the size of the clean data of the server. By default, we set the ratio between the number of clean examples of the server and the total number of examples of clients to be 0.1. We conduct experiments with different ratios: 0.01, 0.02, and 0.05 under the default setting. The corresponding ASRs are 9.71%, 12.38%, and 9.75%, indicating that FedGame is effective even when the server only has 1% clean data.

Analysis of computation cost for the server. Our FedGame computes a genuine score for each client in each communication round. Here we demonstrate its computational efficiency. On average, it takes 0.148s to compute a genuine score for each client in each communication round on a single

Table 5: ASRs of FedGame under different total number of clients on MNIST and CIFAR10. The fraction of compromised clients is 60%.

Dataset	Total Number of Clients				
	10	30	50	70	100
MNIST	9.72	9.95	10.03	10.01	9.89
CIFAR10	8.92	10.44	10.62	9.79	10.82

NVIDIA 2080 Ti GPU. We note that the server could from a resourceful tech company (e.g., Google, Meta, Apple), which would have enough computation resources to compute it for millions of clients. Moreover, those local models can be evaluated in parallel.

Results on Tiny-ImageNet dataset. Here we report the performance of our FedGame on the Tiny-ImageNet dataset that has 200 classes and 500 training images per class. We use ResNet-50 [15] as the global model architecture. We compare our FedGame with FLTrust against the scaling attack in the IID setting. The total number of clients is 10. We assume the server has in-domain clean data for fair comparison with FLTrust. The other hyperparameters are kept the same as the default setting. The ASRs for FLTrust and FedGame are 6.38% and 1.35%, respectively. We have two observations from the experimental results. First, our FedGame is effective for large dataset with more classes. Second, the results indicate that our FedGame consistently outperforms FLTrust under different datasets.

Performance under static attacks. In our evaluation, we consider an attacker optimizing the fraction of backdoored training examples. We also evaluate FedGame under existing attacks where the attacker does not optimize it. Under the default setting, our FedGame can achieve an ASR of 9.75%, indicating that our defense is effective under static attack.

Trigger optimization. We consider an attacker optimizes trigger pattern such that a backdoored input is more likely to be predicted as the target class. We perform experiments under the default setting. The ASR is 12.43%, which indicates that our FedGame is consistently effective for trigger optimization.

D.7 Other Adaptive Attacks

We note that an attacker can slightly manipulate the parameters of local models of compromised clients to inject a backdoor such that they are more similar to those of benign clients. However, FLGame does not rely on model parameters for detection. Instead, our FLGame leverages the model behaviors, i.e., whether the model predicts inputs with our reverse engineered trigger as the target class. As a result, our defense would be still effective even if the change in the model parameters is small as long as the model has backdoor behavior (this is required to make the attack effective). This is also the reason why our defense is better than existing methods such as FLTrust which leverages model parameters for defense.