Abstract
To study the resilience of distributed learning, the “Byzantine” literature considers a strong threat model where workers can report arbitrary gradients to the parameter server. Whereas this model helped obtain several fundamental results, it has sometimes been considered unrealistic, when the workers are mostly trustworthy machines. In this paper, we show a surprising equivalence between this model and data poisoning, a threat considered much more realistic. More specifically, we prove that every gradient attack can be reduced to data poisoning, in any personalized federated learning system with PAC guarantees (which we show are both desirable and realistic). This equivalence makes it possible to obtain new impossibility results on the resilience of any “robust” learning algorithm to data poisoning in highly heterogeneous applications, as corollaries of existing impossibility theorems on Byzantine machine learning. Moreover, using our equivalence, we derive a practical attack that we show (theoretically and empirically) can be very effective against classical personalized federated learning models.

1. Introduction
Learning algorithms typically leverage data generated by a large number of users (Smith et al., 2013; Wang et al., 2019a;b) to often learn a common model that fits a large population (Konecny et al., 2015), but also sometimes to construct a personalized model for each individual (Ricci et al., 2011). Autocompletion (Lehmann & Buschek, 2021), conversational (Shum et al., 2018) and recommendation (Ie et al., 2019) schemes are examples of such personalization algorithms already deployed at scale. To be effective, besides huge amounts of data (Brown et al., 2020; Fedus et al., 2021), these algorithms require customization, motivating research into the promising but challenging field of personalized federated learning (Fallah et al., 2020; Hanzely et al., 2020; Dinh et al., 2020).

Now, classical learning algorithms generally regard as desirable to fit all available data. However, this approach dangerously fails in the context of user-generated data, as goal-oriented users may provide untrustworthy data to reach their objectives. In fact, in applications such as content recommendation, activists, companies, and politicians have strong incentives to do so to promote certain views, products or ideologies (Hoang, 2020; Hoang et al., 2021). Perhaps unsurprisingly, this led to the proliferation of fabricated activities to bias algorithms (Bradshaw & Howard, 2019; Neudert et al., 2019), e.g. through “fake reviews” (Wu et al., 2020). The scale of this phenomenon is well illustrated by the case of Facebook which, in 2019 alone, reported the removal of around 6 billion fake accounts from its platform (Fung & Garcia, 2019). This is highly concerning in the era of “stochastic parrots” (Bender et al., 2021); climate denialists are incentivized to pollute textual datasets with claims like “climate change is a hoax”, rightly assuming that autocompletion, conversational and recommendation algorithms trained on such data will more likely spread these views (McGuffie & Newhouse, 2020). This raises serious concerns about the vulnerability of personalized federated learning to misleading data. Data poisoning attacks clearly constitute now a major machine learning security issue in already deployed systems (Kumar et al., 2020).

Overall, in adversarial environments like social media, and given the advent of deep fakes (Johnson & Diakopoulos, 2021), we should expect most data to be strategically crafted and labeled. In this context, the authentication of the data provider is critical. In particular, the safety of learning algorithms arguably demands that they be trained solely on cryptographically signed data, namely, data that provably come from a known source. But even signed data cannot be wholeheartedly trusted since users typically have preferences over what ought to be recommended to others. Naturally, even “authentic” users have incentives to behave strategically in order to promote certain views or products.
To study resilience, the Byzantine learning literature usually assumes that each federated learning worker may behave arbitrarily (Blanchard et al., 2017; Yin et al., 2018; Karimireddy et al., 2021; Yang & Li, 2021). To understand the implication of this assumption, recall that at each iteration of a federated learning stochastic gradient descent, every worker is given the updated model, and asked to compute the gradient of the loss function with respect to (a batch of) its local data. Byzantine learning assumes that a worker may report any gradient; without having to certify that the gradient was generated through data poisoning. Whilst very general, and widely studied in the last few years, this gradient attack threat model has been argued to be unrealistic in practical federated learning (Shejwalkar et al., 2022), especially when the workers are machines owned by trusted entities (Kairouz et al., 2021).

We prove in this paper a somewhat surprising equivalence between gradient attacks and data poisoning, in a convex setting. Essentially, we give the first practically compelling argument for the necessity to protect learning against gradient attacks. Our result enables us to carry over results on Byzantine gradient attacks to the data poisoning world. For instance, the impossibility result of El-Mhamdi et al. (2021a), combined with our equivalence result, implies that the more heterogeneous the data, the more vulnerable any “robust” learning algorithm is. Also, we derive concrete data poisoning attacks from gradient ones.

Contributions. As a preamble of our main result, we formalize local PAC* learning1 (Valiant, 1984) for personalized learning, and prove that a simple and general solution to personalized federated linear regression and classification is indeed locally PAC* learning. Our proof leverages a new concept called gradient-PAC* learning. We prove that gradient PAC* learning, which is verified by basic learning algorithms like linear and logistic regression, is sufficient to guarantee local PAC* learning. This is an important and nontrivial contribution of this paper.

Our main contribution is to then prove that local PAC* convex learning in personalized federated learning essentially implies an equivalence between data poisoning and gradient attacks. More precisely, we show how any (converging) gradient attack can be turned into a data poisoning attack, with equal harm. As a corollary, we derive new impossibility theorems on what any robust personalized learning algorithm can guarantee, given heterogeneous genuine users and under data poisoning. Given how easy it generally is to create fake accounts on web platforms and to inject poisonous data through fabricated activities, our results arguably greatly increase the concerns about the vulnerabilities of learning from user-generated data, even when “Byzantine learning algorithms” are used, especially on controversial issues like hate speech moderation, where genuine users will inevitably provide conflicting reports on which words are abusive and ought to be removed.

Finally, we present a simple but very general strategic gradient attack, called the counter-gradient attack, which any participant to federated learning can deploy to bias the global model towards any target model that better suits their interest. We prove the effectiveness of this attack under fairly general assumptions, which apply to many proposed personalized learning frameworks including Hanzely et al. (2020); Dinh et al. (2020). We then show empirically how this attack can be turned into a devastating data poisoning attack, with remarkably few data2. Our experiment also shows the effectiveness of a simple protection, which prevents attackers from arbitrarily manipulating the trained algorithm. Namely, it suffices to replace the $\ell_2^2$ regularization with a (smooth) $\ell_2$ regularization. Note that this solution is strongly related to the Byzantine resilience of the geometric median (El-Mhamdi et al., 2021b; Acharya et al., 2022).

Related work. Collaborative PAC learning was introduced by Blum et al. (2017), and then extensively studied (Chen et al., 2018a; Nguyen & Zakynthinou, 2018), sometimes assuming Byzantine collaborating users (Qiao, 2018; Jain & Orlitsky, 2020; Konstantinov et al., 2020). It was however assumed that all honest users have the same labeling function. In other words, all honest users agree on how every query should be answered. This is a very unrealistic assumption in many critical applications, like content moderation or language processing. In fact, in such applications, removing outliers can be argued to amount to ignoring minorities’ views, which would be highly unethical. The very definition of PAC learning must then be adapted, which is precisely what we do in this paper (by also adapting it to parameterized models).

A large literature has focused on data poisoning, with either a focus on backdoor (Dai et al., 2019; Zhao et al., 2020; Severi et al., 2021; Truong et al., 2020; Schwarzschild et al., 2021) or triggerless attacks (Biggio et al., 2012; Muñoz-González et al., 2017; Shafahi et al., 2018; Zhu et al., 2019; Huang et al., 2020; Barreno et al., 2006; Aghakhani et al., 2021; Geiping et al., 2021). However, most of this research analyzed data poisoning without signed data. A noteworthy exception is Mahlojifar et al. (2019), whose universal attack amplifies the probability of a (bad) property. Our work bridges the gap, for the first time, between that line of work and what has been called Byzantine resilience (Mhamdi et al., 2018; Baruch

---

1We omit complexity considerations for the sake of generality. We define PAC* to be PAC without such considerations.

2The code can be found at https://github.com/LPD-EFPL/Attack_Equivalence.
work which generalizes the models proposed by Dinh et al., 2010; Perote & Perote-Peña, 2004; Ben-Porat & Tennenholtz, 2017), classification (Cai et al., 2015), and clustering (Perote & Sevilla, 2003). While some papers provide positive results in settings where each user can only provide a single data point (Chen et al., 2018b; Perote & Perote-Peña, 2004), Suya et al. (2021) show how to arbitrarily manipulate convex learning models through multiple data injections, when a single model is learned from all data at once.

Structure of the paper. The rest of the paper is organized as follows. Section 2 presents a general model of personalized learning, formulates local PAC* learning and describes a general federated gradient descent algorithm. Section 3 proves the equivalence between data poisoning and gradient attacks, under local PAC* learning. Section 4 proves the local PAC* learning properties for federated linear regression and classification. Section 5 describes a simple and general data poisoning attack, and shows its effectiveness against $\ell_2^2$, both theoretically and empirically. Section 6 concludes. Proofs of our theoretical results and details about our experiments are given in the Appendix.

2. A General Personalized Learning Framework

We consider a set $[N] = \{1, \ldots, N\}$ of users. Each user $n \in [N]$ has a local dataset $\mathcal{D}_n$, and learns a local model $\theta_n \in \mathbb{R}^d$. Users may collaborate to improve their models. Personalized learning must then input a tuple of users’ local datasets $\mathcal{D} \equiv (\mathcal{D}_1, \ldots, \mathcal{D}_N)$, and output a tuple of local models $\tilde{\theta}^* \equiv (\tilde{\theta}_1^*, \ldots, \tilde{\theta}_N^*)$. Like many others, we assume that the users perform federated learning to do so, by leveraging the computation of a common global model $\rho \in \mathbb{R}^d$. Intuitively, the global model is an aggregate of all users’ local models, which users can leverage to improve their local models. This model typically allows users with too few data to obtain an effective local model, while it may be mostly discarded by users whose local datasets are large.

More formally, we consider a personalized learning framework which generalizes the models proposed by Dinh et al. (2020) and Hanzely et al. (2020). Namely, we consider that the personalized learning algorithm outputs a global minimum $(\rho^*, \tilde{\theta}^*)$ of a global loss given by

$$\text{Loss}(\rho, \tilde{\theta}, \mathcal{D}) \triangleq \sum_{n \in [N]} \mathcal{L}_n(\theta_n, \mathcal{D}_n) + \sum_{n \in [N]} \mathcal{R}(\rho, \theta_n),$$

where $\mathcal{R}$ is a regularization, typically with a minimum at $\theta_n = \rho$. For instance, Hanzely et al. (2020) and Dinh et al. (2020) define $\mathcal{R}(\rho, \theta_n) \triangleq \lambda \|\rho - \theta_n\|_2^2$, which we shall call the $\ell_2^2$ regularization. But other regularizations may be considered, like the $\ell_2$ regularization $\mathcal{R}(\rho, \theta_n) \triangleq \lambda \|\rho - \theta_n\|_2$, or the smooth-$\ell_2$ regularization $\mathcal{R}(\rho, \theta_n) \triangleq \lambda \sqrt{1 + \|\rho - \theta_n\|_2^2}$. Note that, for all such regularizations, the limit $\lambda \rightarrow \infty$ essentially yields the classical non-personalized federated learning framework.

2.1. Local PAC* Learning

We consider that each honest user $n$ has a preferred model $\theta_n^*$, and that they provide honest datasets $\mathcal{D}_n$ that are consistent with their preferred models. We then focus on personalized learning algorithms that provably recover a user $n$’s preferred model $\theta_n^*$, if the user provides a large enough honest dataset. Such honest datasets $\mathcal{D}_n$ could typically be obtained by repeatedly drawing random queries (or features), and by using the user’s preferred model $\theta_n^*$ to provide (potentially noisy) answers (or labels). We refer to Section 4 for examples. The model recovery condition is then formalized as follows.

Definition 1. A personalized learning algorithm is locally PAC* learning if, for any subset $\mathcal{H} \subset [N]$ of users, any preferred models $\theta_n^*$, any $\varepsilon, \delta > 0$, and any datasets $\mathcal{D}_{-\mathcal{H}}$ from other users $n \notin \mathcal{H}$, there exists $\mathcal{I}$ such that, if all users $h \in \mathcal{H}$ provide honest datasets $\mathcal{D}_h$ with at least $|\mathcal{D}_h| \geq \mathcal{I}$ data points, then, with probability at least $1 - \delta$, we have $$\|\theta^*_h(\mathcal{D}) - \theta^*_n\|_2 \leq \varepsilon$$ for all users $h \in \mathcal{H}$.

Local PAC* learning is arguably a very desirable property. Indeed, it guarantees that any honest active user will not be discouraged to participate in federated learning as they will eventually learn their preferred model by providing more and more data. Note that the required number of data points $\mathcal{I}$ also depends on the datasets provided by other users $\mathcal{D}_{-\mathcal{H}}$. This implies that a locally PAC* learning algorithm is still vulnerable to poisoning attacks as the attacker’s data set is not a priori fixed. In Section 4, we will show how local PAC* learning can be achieved in practice, by considering specific local loss functions $\mathcal{L}_n$.

2.2. Federated Gradient Descent

While the computation of $\rho^*$ and $\tilde{\theta}^*$ could be done by a single machine, which first collects the datasets $\mathcal{D}$ and
then minimizes the global loss \( \text{LOSS} \) defined in (1), modern machine learning deployments often rather rely on federated (stochastic) gradient descent (or variants), with a central trusted parameter server. In this setting, each user \( n \) keeps their data \( D_n \) locally. At each iteration \( t \), the parameter server sends the latest global model \( \rho^t \) to the users. Each user \( n \) is then expected to update its local model given the global model \( \rho^t \), either by solving \( \theta^t_n \triangleq \arg \min_{\theta_n} \mathcal{L}_n(\theta_n, D_n) + R(\rho^t, \theta_n) \) (Dinh et al., 2020) or by making a (stochastic) gradient step from the previous local model \( \theta_{n-1}^t \) (Hanzely & Richtárik, 2021). User \( n \) is then expected to report the gradient \( g^t_n = \nabla_{\rho} R(\rho^t, \theta^t_n) \) of the global model to the parameter server. The parameter server then updates the global model, using a gradient step, i.e. it computes \( \rho^{t+1} \triangleq \rho^t - \eta_t \sum_{n \in [N]} g^t_n \), where \( \eta_t \) is the learning rate at iteration \( t \). For simplicity, here, and since our goal is to show the vulnerability of personalized federated learning even in good conditions, we assume that the network is synchronous and that no node can crash. Note also that our setting could be generalized to fully decentralized collaborative learning, as was done by El-Mhamdi et al. (2021a).

Users are only allowed to send plausible gradient vectors. More precisely, we denote

\[
\text{GRAD}(\rho) \triangleq \left\{ \nabla_{\rho} R(\rho, \theta) \mid \theta \in \mathbb{R}^d \right\},
\]

the closure set of plausible (sub)gradients at \( \rho \). If user \( n \)'s gradient \( g^t_n \) is not in the set \( \text{GRAD}(\rho^t) \), the parameter server can easily detect the malicious behavior and \( g^t_n \) will be ignored at iteration \( t \). In the case of an \( \ell_2^2 \) regularization, where \( R(\rho, \theta) = \lambda \| \rho - \theta \|_2^2 \), we clearly have \( \text{GRAD}(\rho) = \mathbb{R}^d \) for all \( \rho \in \mathbb{R}^d \). It can be easily shown that, for \( \ell_2 \) and smooth-\( \ell_2 \) regularizations, \( \text{GRAD}(\rho) \) is the closed ball \( \mathcal{B}(0, \lambda) \). Nevertheless, even then, a strategic user \( s \in [N] \) can deviate from its expected behavior, to bias the global model in their favor. We identify, in particular, three sorts of attacks.

**Data poisoning:** Instead of collecting an honest dataset, \( s \) fabricates any strategically crafted dataset \( D_s \), and then performs all operations as expected.

**Model attack:** At each iteration \( t \), \( s \) fixes \( \theta^t_s \triangleq \theta^*_s \), where \( \theta^*_s \) is any strategically crafted model. All other operations would then be executed as expected.

**Gradient attack:** At each iteration \( t \), \( s \) sends any (plausible) strategically crafted gradient \( g^t_s \). The gradient attack is said to converge, if the sequence \( g^t_s \) converges.

Gradient attacks are intuitively most harmful, as the strategic user can adapt their attack based on what they observe during training. However, because of this, gradient attacks are more likely to be flagged as suspicious behaviors. At the other end, data poisoning may seem much less harmful. But it is also harder to detect, as the strategic user can report their entire dataset, and prove that they rigorously performed the expected computations. In fact, data poisoning can be executed, even if users directly provide the data to a (trusted) central authority, which then executes (stochastic) gradient descent. This is typically what is done to construct recommendation algorithms, where users’ data are their online activities (what they view, like and share). Crucially, especially in applications with no clear ground truth, such as content moderation or language processing, the strategic user can always argue that their dataset is “honest”, not strategically crafted. Ignoring the strategic user’s data on the basis that it is an “outlier” may then be regarded as unethical, as it amounts to rejecting minorities’ viewpoints.

### 3. The Equivalence Between Data Poisoning and Gradient Attacks

We now present our main result, considering “model-targeted attacks”, i.e., the attacker aims to bias the global model towards a target model \( \theta^*_s \). This attack was also previously studied by Suya et al. (2021).

**Theorem 1** (Equivalence between gradient attacks and data poisoning). Assume local \( \text{PAC}^* \) learning, and \( \ell_2^2 \), \( \ell_2 \) or smooth-\( \ell_2 \) regularization. Suppose that each loss \( \mathcal{L}_n \) is convex and that the learning rate \( \eta_t \) is constant. Consider any datasets \( \bar{D}_s \) provided by users \( n \neq s \). Then, for any target model \( \theta^*_s \in \mathbb{R}^d \), there exists a converging gradient attack of strategic user \( s \) such that \( \rho^t \to \theta^*_s \), if and only if, for any \( \varepsilon > 0 \), there exists a dataset \( \bar{D}_s \) such that

\[
\left\| \rho^t(\bar{D}) - \theta^*_s \right\|_2 \leq \varepsilon.
\]

For the sake of exposition, our results are stated for \( \ell_2^2 \) or smooth-\( \ell_2 \) regularization only. But the proof, in Appendix B, holds for all continuous regularizations \( \mathcal{R} \) with \( \mathcal{R}(\rho, \theta) \to \infty \) as \( \| \rho - \theta \|_2 \to \infty \). We now sketch our proof, which goes through model attacks.

#### 3.1. Data Poisoning and Model Attacks

To study the model attack, we define the modified loss with directly strategic user \( s \)'s reported model \( \theta^*_s \) as

\[
\text{LOSS}_s(\rho, \bar{\theta}_s, \theta^*_s, \bar{D}_s) \triangleq \text{LOSS}(\rho, (\theta^*_s, \bar{\theta}_s), (\emptyset, \bar{D}_s))
\]

(2)

where \( \bar{\theta}_s \) and \( \bar{D}_s \) are variables and datasets for users \( n \neq s \). Denote \( \rho^*(\theta^*_s, \bar{D}_s) \) and \( \bar{\theta}^*_s(\theta^*_s, \bar{D}_s) \) a minimum of the modified loss function and \( \theta^*_s(\bar{D}_s) \triangleq \theta^*_s \).

**Lemma 1** (Reduction from model attack to data poisoning). Consider any data \( \bar{D} \) and user \( s \in [N] \). Assume the global loss has a global minimum \( (\rho^*, \bar{\theta}^*_s) \). Then \( (\rho^*, \bar{\theta}^*_s) \) is also a global minimum of the modified loss with datasets \( \bar{D}_s \) and strategic reporting \( \theta^*_s(\bar{D}) \).
Now, intuitively, by virtue of local PAC* learning, strategic user \( s \) can essentially guarantee that the personalized learning framework will be learning \( \theta_s^* \approx \theta_s^* \). In the sequel, we show that this is the case.

**Lemma 2** (Reduction from data poisoning to model attack). Assume \( \ell_2^2, \ell_2 \) or smooth-\( \ell_2 \) regularization, and assume local PAC* learning. Consider any datasets \( D_{-s} \) and any attack model \( \theta_s^* \) such that the modified loss \( \mathcal{L}_{s} \) has a unique minimum \( \rho^*(\theta_s^*, D_{-s}) \). Then, for any \( \varepsilon > 0 \), there exists a dataset \( D_s \) such that we have

\[
\|\rho^*(\bar{D}) - \rho^*(\theta_s^*, D_{-s})\|_2 \leq \varepsilon \quad \text{and} \quad \forall n \neq s, \quad \|\theta_n^*(\bar{D}) - \theta_n^*(\theta_s^*, D_{-s})\|_2 \leq \varepsilon. \tag{3}
\]

**Sketch of proof.** Given local PAC*, for a large dataset \( D_s \) constructed from \( \theta_s^* \), \( s \) can guarantee \( \theta_s^*(\bar{D}) \approx \theta_s^* \). By carefully bounding the effect of the approximation on the loss using the Heine-Cantor theorem, we show that this implies \( \rho^*(\bar{D}) \approx \rho^*(\theta_s^*, D_{-s}) \) and \( \theta_n^*(\bar{D}) \approx \theta_n^*(\theta_s^*, D_{-s}) \) for all \( n \neq s \). The precise analysis is nontrivial. \( \Box \)

### 3.2. Model Attacks and Gradient Attacks

We now prove that any successful converging model-targeted gradient attack can be transformed into an equivalent locally successful model attack.

**Lemma 3** (Reduction from model attack to gradient attack). Assume that \( \mathcal{L}_n \) is convex for all users \( n \in [N] \), and that we use \( \ell_2^2 \), \( \ell_2 \) or smooth-\( \ell_2 \) regularization. Consider a converging gradient attack \( g_t^\infty \) with limit \( g_\infty^\infty \) that makes the model global \( \rho^* \) converge to \( \rho^\infty \) with a constant learning rate \( \eta \). Then, for any \( \varepsilon > 0 \), there is \( \theta_s^* \in \mathbb{R}^d \) such that

\[
\|\rho^\infty - \rho^*(\theta_s^*, D_{-s})\|_2 \leq \varepsilon.
\]

**Sketch of proof.** The proof is based on the observation that since GRAD is closed and \( g_\infty^\infty \in \text{GRAD} \), we can construct \( \theta_s^* \) which approximately yields the gradient \( g_\infty^\infty \).

Since any model attack can clearly be achieved by the corresponding honest gradient attack for a sufficiently small and constant learning rate, model attacks and gradient attacks are thus equivalent. In light of our previous results, this implies that gradient attacks are essentially equivalent to data poisoning (Theorem 1).

### 3.3. Convergence of the Global Model

Note that Theorem 1 (and Lemma 3) assumes that the global model converges. Here, we prove that this assumption is automatically satisfied for converging gradients, at least when local models \( \theta_n^t \) are fully optimized given \( \rho^t \), at each iteration \( t \), in the manner of (Dinh et al., 2020), and under smoothness assumptions.

**Proposition 1.** Assume that \( \mathcal{L}_n \) is convex and \( L \)-smooth for all users \( n \in [N] \), and that we use \( \ell_2^2 \) or smooth-\( \ell_2 \) regularization. If \( g_t^\infty \) converges and if \( \eta_t = \eta \) is a constant small enough, then \( \rho^t \) will converge too.

**Sketch of proof.** Denote \( g_\infty^\infty \) the limit of \( g_t^\infty \). Gradient descent then behaves as though it was minimizing the loss plus \( \rho^* g_\infty^\infty \) (and ignoring \( R(\rho, \theta_s) \)). Essentially, classical gradient descent theory then guarantees \( \rho^t \to \rho^\infty \), though the precise proof is nontrivial (see Appendix C). \( \Box \)

### 3.4. Impossibility Corollaries

Given our equivalence, impossibility theorems on (heterogeneous) federated learning under (converging) gradient attacks imply impossibility results under data poisoning. For instance, El-Mhamdi et al. (2021a) and He et al. (2020) proved theorems saying that the more heterogeneous the learning, the more vulnerable it is in a Byzantine context, even when “Byzantine-resilient” algorithms are used (Blanchard et al., 2017). In fact, and interestingly, El-Mhamdi et al. (2021a) and He et al. (2020) actually leverage a model attack. Before translating the corresponding result, some work is needed to formalize what Byzantine resilience may mean in our setting.

**Definition 2.** A personalized learning algorithm ALG achieves \((F, N, C)\)-Byzantine learning if, for any subset \( \mathcal{H} \subset [N] \) of honest users with \( |\mathcal{H}| = N - F \), any honest vectors \( \vec{\theta}_\mathcal{H}^\dagger \in (\mathbb{R}^d)^{|\mathcal{H}|} \), given any \( \varepsilon, \delta > 0 \), there exists \( \mathcal{I} \) such that, when each honest user \( h \in \mathcal{H} \) provides honest datasets \( D_h^\dagger \) by answering \( \mathcal{I} \) queries with model \( \vec{\theta}_\mathcal{H}^\dagger \), then, with probability at least \( 1 - \delta \), for any poisoning dataset \( D_F^\dagger \) provided by Byzantine users \( f \notin \mathcal{H} \), denoting \( \rho^{\mathrm{ALG}} \equiv \rho^{\mathrm{ALG}}(D_h^\dagger, D_F^\dagger) \) and \( \vec{\theta}^{\mathrm{ALG}} \equiv \vec{\theta}^{\mathrm{ALG}}(D_h^\dagger, D_F^\dagger) \), we have the guarantee

\[
\|\rho^{\mathrm{ALG}} - \vec{\theta}^{\mathrm{ALG}}\|_2^2 \leq C^2 \max_{h, h' \in \mathcal{H}} \|\theta_h^t - \theta_{h'}^t\|_2^2 + \varepsilon, \tag{4}
\]

where \( \vec{\theta}^{\mathrm{ALG}} \) is the average of honest users’ preferred models.

Note that \( \theta_h^t \) is what we would have learned, under local PAC* and \( \ell_2^2 \) regularization, in the absence of Byzantine users \( f \in [N] - \mathcal{H} \), in the limit where all honest users \( h \in \mathcal{H} \) provide a very large amount of data. Meanwhile, \( \max_{h, h' \in \mathcal{H}} \|\theta_h^t - \theta_{h'}^t\|_2^2 \) is a reasonable measure of the heterogeneity among honest users. Thus, our definition captures well the robustness of the algorithm ALG, for heterogeneous learning under data poisoning. Interestingly, our equivalence theorem allows to translate the model-attack-based impossibility theorems of El-Mhamdi et al. (2021a) into an impossibility theorem on data poisoning resilience.

**Corollary 1.** No algorithm achieves \((F, N, C)\)-Byzantine learning with \( F \geq N/2 \).
Corollary 2. No algorithm achieves \((F, N, C)\)-Byzantine learning with \(C < F/(N - F)\).

The proofs are given in Appendix D.

4. Examples of Locally PAC* Learning Systems

To the best of our knowledge, though similar to collaborative PAC learning (Blum et al., 2017), local PAC* learnability is a new concept in the context of personalized federated learning. It is thus important to show that it is not unrealistic. To achieve this, in this section, we provide sufficient conditions for a personalized learning model to be locally PAC* learnable. First, we construct local losses \(\mathcal{L}_n\) as sums of losses per input, i.e.

\[
\mathcal{L}_n(\theta_n, D_n) = \nu \|\theta_n\|_2^2 + \sum_{x \in D_n} \ell(\theta_n, x),
\]

for some “loss per input” function \(\ell\) and a weight \(\nu > 0\). Appendix E gives theoretical and empirical arguments are provided for using such a sum (as opposed to an expectation). Remarkably, for linear or logistic regression, given provided for using such a sum (as opposed to an expectation).

Theorem 2 (Personalized least square linear regression is locally PAC* learning). Consider \(\ell_2^2, \ell_2\) or smooth-\(\ell_2\) regularization. Assume that, to generate a data \(x_i\), a user with preferred parameter \(\theta^i \in \mathbb{R}^d\) first independently draws a random vector query \(Q_i \in \mathbb{R}^d\) from a bounded query distribution \(\mathcal{Q}\), with positive definite matrix \(\Sigma = \mathbb{E}[Q_i Q_i^T]\). Assume that the user labels \(Q_i \in \mathbb{R}^d\) with answer \(A_i = Q_i^T \theta^i + \xi_i\), where \(\xi_i\) is a zero-mean sub-Gaussian random noise with parameter \(\sigma_{\xi}\), independent from \(Q_i\), and other data points. Finally, assume that \(\ell(\theta, (Q_i, A_i)) = \frac{1}{2}(\theta^T Q_i - A_i)^2\) then the personalized learning algorithm is locally PAC* learning.

Theorem 3 (Personalized logistic regression is locally PAC*-learning). Consider \(\ell_2^2, \ell_2\) or smooth-\(\ell_2\) regularization. Assume that, to generate a data \(x_i\), a user with preferred parameter \(\theta^i \in \mathbb{R}^d\) first independently draws a random vector query \(Q_i \in \mathbb{R}^d\) from a query distribution \(\mathcal{Q}\), whose support \(\text{supp} (\mathcal{Q})\) is bounded and spans the full vector space \(\mathbb{R}^d\). Assume that the user then labels \(Q_i\) with answer \(A_i = 1\) with probability \(\sigma(Q_i^T \theta^i)\), and labels it \(A_i = -1\) otherwise, where \(\sigma(z) \triangleq (1 + e^{-z})^{-1}\). Finally, assume that \(\ell(\theta, (Q_i, A_i)) = -\ln(\sigma(A_i \theta^T Q_i))\). Then the personalized learning algorithm is locally PAC* learning.

4.1. Proof Sketch

The full proofs of theorems 2 and 3 are given in Appendix F. Here, we provide proof outlines. In both cases, we leverage the following stronger form of PAC* learning.

Definition 3 (Gradient-2-PAC*). Let \(\mathcal{E}(\mathcal{D}, \theta^i, I, A, B, \alpha)\) the event defined by

\[
\forall \theta \in \mathbb{R}^d, \quad (\theta - \theta^i)^T \nabla \mathcal{L}(\theta, \mathcal{D}) \geq A \mathcal{I} \min \left\{ \|\theta - \theta^i\|_2, \|\theta - \theta^i\|_2^2 \right\} - B \mathcal{I} \alpha \|\theta - \theta^i\|_2.
\]

The loss \(\mathcal{L}\) is gradient-PAC* if, for any \(K > 0\), there exist constants \(A_K, B_K > 0\) and \(\alpha_K < 1\), such that for any \(\theta^i \in \mathbb{R}^d\) with \(\|\theta^i\|_2 \leq K\), assuming that the dataset \(\mathcal{D}\) obtained by honestly collecting and labeling \(I\) data points according to the preferred model \(\theta^i\), the probability of the event \(\mathcal{E}(\mathcal{D}, \theta^i, I, A_K, B_K, \alpha_K)\) goes to 0 as \(I \to \infty\).

Intuitively, this definition asserts that, as we collect more data from a user, then, with high probability, the gradient of the loss at any point \(\theta\) too far from \(\theta^i\) will point away from \(\theta^i\). In particular, gradient descent is then essentially guaranteed to draw \(\theta\) closer to \(\theta^i\). The right-hand side of the equation defining \(\mathcal{E}(\mathcal{D}, \theta^i, I, A, B, \alpha)\) is subtly chosen to be strong enough to guarantee local PAC*, and weak enough to be verified by linear and logistic regression.

Lemma 4. Logistic and linear regression, defined in theorems 2 and 3, are gradient PAC* learning.

Sketch of proof. For linear regression, remarkably, the discrepancy between the empirical and the expected loss functions depends only on a few key random variables, such as \(\min \left\{ \frac{1}{I} \sum Q_i Q_i^T, \sum \xi_i Q_i \right\}\), which can be controlled by appropriate concentration bounds. Meanwhile, for logistic regression, for \(|b| \leq K\), we observe that \((a - b)\sigma(a) - \sigma(b)) \geq c_K \min \left\{|a - b|, |a - b|^2\right\}$. Essentially, this proves that gradient-PAC* would hold if the empirical loss was replaced by the expected loss. The actual proofs, however, are nontrivial, especially in the case of logistic regression, which leverages topological considerations to derive a critical uniform concentration bound.

Now, under very mild assumptions on the regularization \(R\) (not even convexity!), which are verified by the \(\ell_2^2, \ell_2\) and smooth-\(\ell_2\) regularizations, we prove that the gradient-PAC* learnability through \(\ell\) suffices to guarantee that personalized learning will be locally PAC* learning.

Lemma 5. Consider \(\ell_2^2, \ell_2\) or smooth-\(\ell_2\) regularization. If \(\ell\) is gradient-PAC* and nonnegative, then personalized learning is locally PAC*-learning.

Sketch of proof. Given other users’ datasets, \(R\) yields a fixed bias. But as the user provides more data, by gradient-PAC*, the local loss dominates, thereby guaranteeing local PAC* learning. Appendix G provides a full proof.
Combining the two lemmas clearly yields theorems 2 and 3 as special cases. Note that our result actually applies to a more general set of regularizations and losses.

4.2. The Case of Deep Neural Networks

Deep neural networks generally do not verify gradient PAC	ag{4}. After all, because of symmetries like neuron swapping, different values of the parameters might compute the same neural network function. Thus the “preferred model” \( \theta^\dagger \) is arguably ill-defined for neural networks\(^3\). Nevertheless, we may consider a strategic user who only aims to bias the last layer. In particular, assuming that all layers but the last one of a neural network are pretrained and fixed, thereby defining a “shared representation” (Collins et al., 2021), and assuming the last layer performs a linear regression or classification, then our theory essentially applies to the fine-tuning of the parameters of the last layer (sometimes known as the “head”).

Note that for our data poisoning reconstruction (see Section 5) to be applicable, the attacker would need to have the capability to generate a data point whose vector representation matches any given predefined latent vector. In certain applications, this can be achieved through generative networks (Goodfellow et al., 2020). If so, then our data poisoning attacks would apply as well to deep neural network head tuning.

5. A Practical Data Poisoning Attack

We now construct a practical data poisoning attack, by introducing a new gradient attack, and by then leveraging our equivalence to turn it into a data poisoning attack.

5.1. The Counter-Gradient Attack

We define a simple, general and practical gradient attack, which we call the counter-gradient attack (CGA). Intuitively, this attack estimates the sum \( g_{\theta_s}^{1:s} \) of the gradients of other users based on its value at the previous iteration, which can be inferred from the way the global model \( \rho^{t-1} \) was updated into \( \rho^t \). More precisely, apart from initialization \( \hat{g}_s^{1:s} = 0 \), CGA makes the estimation

\[
\hat{g}_s^{t} = \frac{\rho^{t-1} - \rho^t}{\eta_{t-1}} - g_{\theta_s}^{t} = g_{\theta_s}^{t-1}. \tag{6}
\]

Strategic user \( s \) then reports the plausible gradient that moves the global model closest to the user’s target model \( \theta_s^\dagger \), assuming others report \( \hat{g}_s^{t} \). In other words, at every iteration, CGA reports

\[
g_s^t \in \arg \min_{g \in \text{GRAD}(\rho^t)} \| \rho^t - \eta_t (\hat{g}_s^t + g) - \theta_s^\dagger \|_2. \tag{7}
\]

Note that this attack only requires user \( s \) to know the learning rates \( \eta_{t-1} \) and \( \eta_t \), the global models \( \rho^{t-1} \) and \( \rho^t \), and their target model \( \theta_s^\dagger \).

**Computation of CGA.** Define \( h_s^t \triangleq g_s^{t-1} + \rho^{t-1} - \rho^t \). For convex sets \( \text{GRAD}(\rho^t) \), it is straightforward to see that CGA boils down to computing the orthogonal projection of \( h_s^t \) on \( \text{GRAD}(\rho^t) \). This yields very simple computations for \( \ell_2^2, \ell_2 \) and smooth-\( \ell_2 \) regularizations.

**Proposition 2.** For \( \ell_2^2 \) regularization, CGA reports \( g_s^t = h_s^t \). For \( \ell_2 \) or smooth-\( \ell_2 \) regularization, CGA reports \( g_s^t = h_s^t \min \{1, \lambda/\|h_s^t\|_2\} \).

**Proof.** Equation (7) boils down to minimizing the distance between \( \rho^{t-1} - \hat{g}_s^t \) and \( \text{GRAD}(\rho^t) \), which is the ball \( B(0, \lambda) \). This minimum is the orthogonal projection.

**Theoretical analysis.** We prove that CGA is perfectly successful against \( \ell_2^2 \) regularization. To do so, we suppose that, at each iteration \( t \) and for each user \( n \neq s \), the local models \( \theta_n \) are fully optimized with respect to \( \rho^t \), and the honest gradients of \( g_{\theta_n}^t \) are used to update \( \rho \).

**Theorem 4.** Consider \( \ell_2^2 \) regularization. Assume that \( \ell_2^2 \) is convex and \( L_\ell \)-smooth, and that \( \eta_t \) is small enough. Then CGA is converging and optimal, as \( \rho^t \to \theta_s^\dagger \).

**Sketch of proof.** The main challenge is to guarantee that the other users’ gradients \( g_{\theta_n}^t \) for \( n \neq s \) remain sufficiently stable over time to guarantee convergence, which can be done by leveraging \( L_\ell \)-smoothness. The full proof, with the necessary upper-bound on \( \eta_t \), is given in Appendix H.

The analysis of the convergence against smooth-\( \ell_2 \) is unfortunately significantly more challenging. Here, we simply make a remark about CGA at convergence.

**Proposition 3.** If CGA against smooth-\( \ell_2 \) regularization converges for \( \eta_t = \eta \), then it either achieves perfect manipulation, or it is eventually partially honest, in the sense that the gradient by CGA correctly points towards \( \theta_s^\dagger \).

**Proof.** Denote \( P \) the projection onto the closed ball \( B(0, \lambda) \). If CGA converges, then, by Proposition 2, \( P \left( g_s^\infty + \frac{\rho^\infty - \theta_s^\dagger}{\eta} \right) = g_s^\infty \). Thus \( \rho^\infty - \theta_s^\dagger \) and \( g_s^\infty \) must be colinear. If perfect manipulation is not achieved (i.e. \( \rho^\infty \neq \theta_s^\dagger \)), then we must have \( g_s^\infty = \lambda \|\rho^\infty - \theta_s^\dagger\|_2 \).

It is interesting that, against smooth-\( \ell_2 \), CGA actually favors partial honesty. Overall, this condition is critical for the safety of learning algorithms, as they are usually trained to generalize their training data. However, it should be
stressed that this is evidence that CGA is suboptimal, as (El-Mhamdi et al., 2021b) instead showed that the geometric median rather (slightly) incentivizes untruthful strategic behaviors. The problem of designing general strategyproof learning algorithms is arguably still mostly open, despite recent progress (Meir et al., 2012; Chen et al., 2018b; Farhadkhani et al., 2021).

**Empirical evaluation of CGA.** We deployed CGA to bias the federated learning of MNIST. We consider a strategic user whose target model is one that labels 0’s as 1’s, 1’s as 2’s, and so on, until 9’s that are labeled as 0’s. In particular, this target model has a nil accuracy. Figure 1 shows that such a user effectively hacks the $\ell_2^2$ regularization against 10 honest users who each have 6,000 data points of MNIST, in the case where local models only undergo a single gradient step at each iteration, but fails to hack the $\ell_2$ regularization. This suggests the effectiveness of simple defense strategies like the geometric median (El-Mhamdi et al., 2021b; Acharya et al., 2022). See Appendix I for more details. We also ran a similar successful attack on the last layer of a deep neural network trained on cifar-10, which is detailed in Appendix J.

**Figure 1. Accuracy of the global model under attack by CGA.**

### 5.2. From Gradient Attack to Model Attack Against $\ell_2^2$

We now show how to turn a gradient attack into model attack, against $\ell_2^2$ regularization. It is trivial to transform any gradient $g_s^\infty$ such that $\rho^\infty = \theta_0^1$ into a model attack by setting $\theta_s^\bullet \triangleq \theta_0^1 - \frac{1}{\lambda}g_s^\infty$, as guaranteed by the following result, and as depicted by Figure 2a and Figure 2b.

**Proposition 4.** Consider the $\ell_2^2$ regularization. Suppose that $g_s^\infty \rightarrow g_s^{\infty\infty}$ and $\rho^1 \rightarrow \theta_0^1$, with a constant learning rate $\eta_t = \eta$. Then, under the model attack $\theta_s^\bullet \triangleq \theta_0^1 - \frac{1}{\lambda}g_s^{\infty\infty}$, the gradient at $\rho = \theta_0^1$ vanishes, i.e. 
\[
\nabla_{\rho} \text{Loss}(\theta_s^\bullet, \theta_s^\bullet, \theta_0^1, D_{-s}, \theta_s^\bullet, D_{-s}) = 0.
\]

**Proof.** Given a constant learning rate, the convergence $\rho^1 \rightarrow \theta_0^1$ implies that the sum of honest users’ gradients at $\rho = \theta_0^1$ equals $-g_s^{\infty\infty}$. Therefore, to achieve $\rho^* = \theta_0^1$, it suffices to send $\theta_s^\bullet$ such that the gradient of $\lambda \Vert \rho - \theta_s^\bullet \Vert_2^2$ with respect to $\rho$ at $\rho = \theta_0^1$ equals $g_s^{\infty\infty}$. Since the gradient is $\lambda(\theta_0^1 - g_s^{\infty\infty})$, $\theta_s^\bullet \triangleq \theta_0^1 - \frac{1}{\lambda}g_s^{\infty\infty}$ does the trick. □

### 5.3. From Model Attack to Data Poisoning Against $\ell_2^2$

**The case of linear regression.** In linear regression, any model attack can be turned into a single data poisoning attack, as proved by the following theorem whose proof is given in Appendix K.

**Theorem 5.** Consider the $\ell_2^2$ regularization and linear regression. For any data $D_{-s}$ and any target value $\theta_0^1$, there is a datapoint $(Q, A)$ to be injected by user $s$ such that $\rho^*((Q, A), D_{-s}) = \theta_0^1$.

**Sketch of proof.** We first identify the sum $g$ of honest users’ gradients, if the global model $\rho$ took the target value $\theta_0^1$. We then determine the value $\theta_s^\bullet$ that the strategic user’s model must take, to counteract other users’ gradients. Reporting datapoint $(Q, A) \triangleq (g, g^T \theta_s^\bullet - 1)$ then guarantees that the strategic user’s learned model will equal $\theta_s^\bullet$. □

Note that this single datapoint attack requires reporting a query $Q$ whose norm grows as $\Theta(N)$, while the answer $A$ grows as $\Theta(N^2)$. Assuming a large number of users, this query will fall out of the distribution of users’ queries, and could thus be flagged by basic outlier detection techniques. We stress, however, that our proof can be trivially transformed into an attack with $\Theta(N^2)$ data points, all of which have a query whose norm is $O(1)$.

**The case of linear classification.** We now consider linear classification, with the case of MNIST. By Lemma 2, any model attack can be turned into data poisoning, by (mis)labeling sufficiently many (random) data points. However, this may require creating too many data labelings, especially if the norm of $\theta_s^\bullet$ is large (which holds if $s$ faces many active users), as suggested by Theorem 3.

For efficient data poisoning, define the indifference affine subspace $V \subset \mathbb{R}^d$ as the set of images with equiprobable labels. Intuitively, labeling images close to $V$ is very informative, as it informs us directly about the separating hyperplanes. To generate images, we draw random images, project them orthogonally on $V$ and add a small noise. We then label the image probabilistically with model $\theta_s^\bullet$.

Figure 2d shows the effectiveness of the resulting data poisoning attack, with only 2,000 data points, as opposed to the 60,000 honestly labeled data points that the 10 other users cumulatively have. Remarkably, complete data relabeling was achieved by poisoning merely 3.3% of the total database. More details are given in Appendix L.

Note that this attack leads us to consider images not in
Data poisoning and Byzantine gradients

Figure 2. (a) Distance between $\rho^t$ and $\theta^*_s$ (target_dist), under model attack (combining CGA and Proposition 4). (b) Accuracy of $\rho^t$ according to $\theta^*_s$ (which relabels $0 \rightarrow 1 \rightarrow 2 \rightarrow \ldots \rightarrow 9 \rightarrow 0$), under model attack (combining CGA and Proposition 4). (c) Distance between the global model $\rho^t$ and the target model $\theta^*_s$ (target_dist), under our data poisoning attack. (d) Accuracy of $\rho^t$ according to $\theta^*_s$ (which relabels $0 \rightarrow 1 \rightarrow 2 \rightarrow \ldots \rightarrow 9 \rightarrow 0$), under our data poisoning attack.

$[0, 1]^d$. In Appendix L.3, we report another equivalently effective attack, which only reports images in $[0, 1]^d$, though it requires significantly more data injection.

5.4. Gradient Attack on Local Models

Note that CGA aims to merely bias the global model. However, the attacker may instead prefer to bias other users’ local models. To this end, we present here a variant of CGA, which targets the average of other users’ local models. At each iteration of this variant, the attacker reports

$$g^*_s \in \arg \min_{g \in \text{Grad}(\rho^t)} \left\| \rho^t - \eta_t (\hat{g}^t_s + g) - \theta^*_s - \frac{\hat{g}^t_s}{2\lambda(N-1)} \right\|_2.$$\

Figure 3 shows the effectiveness of this attack. This gradient attack can evidently be turned into data poisoning similar to what was achieved for CGA.

6. Conclusion

We showed that, unlike what has been argued, e.g., Shejwalkar et al. (2022), the gradient attack threat is not unrealistic. More precisely, for personalized federated learning with local PAC guarantees, effective gradient attacks can be derived from strategic data reporting, with potentially surprisingly few data. In fact, by leveraging our newly found equivalence, we derived new impossibility theorems on what any robust learning algorithm can guarantee, under data poisoning attacks, especially, in highly-heterogeneous settings. Yet such attacks are known to be ubiquitous for high-risk applications, many of which are known to feature especially high heterogeneity, like online content recommendation. Arguably, a lot more security measures are urgently needed to make large-scale learning algorithms safe.

Acknowledgement

This work has been supported in part by the Swiss National Science Foundation (SNSF) project 200021_200477. The authors are thankful to the anonymous reviewers of ICLR 2022 and ICML 2022 for their constructive comments.
References


Chen, Y., Liu, Y., and Podimata, C. Learning strategy-aware linear classifiers. In Larochelle, H., Ranzato, M.,
Data poisoning and Byzantine gradients


Fung, B. and Garcia, A. Facebook has shut down 5.4 billion fake accounts this year. *CNN Business*, 2019.


Data poisoning and Byzantine gradients


Data poisoning and Byzantine gradients


Data poisoning and Byzantine gradients


A. Convexity Lemmas

A.1. General Lemmas

**Definition 4.** We say that $f : \mathbb{R}^d \to \mathbb{R}$ is locally strongly convex if, for any convex compact set $C \subset \mathbb{R}^d$, there exists $\mu > 0$ such that $f$ is $\mu$-strongly convex on $C$, i.e., for any $x, y \in C$ and any $\lambda \in [0, 1]$, we have

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) - \frac{\mu}{2}(1 - \lambda)\|x - y\|_2^2. \quad (8)$$

It is well-known that if $f$ is differentiable, this condition amounts to saying that $\|\nabla f(x) - \nabla f(y)\|_2 \geq \mu \|x - y\|_2$ for all $x, y \in C$. And if $f$ is twice differentiable, then it amounts to saying $\nabla^2 f(x) \succeq \mu I$ for all $x \in C$.

**Lemma 6.** If $f$ is locally strongly convex and $g$ is convex, then $f + g$ is locally strongly convex.

**Proof.** Indeed, $(f + g)(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) - \frac{\mu}{2}(1 - \lambda)\|x - y\|_2^2 + \lambda g(x) + (1 - \lambda)g(y) = \lambda(f + g)(x) + (1 - \lambda)(f + g)(y) - \frac{\mu}{2}(1 - \lambda)\|x - y\|_2^2$. \hfill $\Box$

**Definition 5.** We say that $f : \mathbb{R}^d \to \mathbb{R}$ is $L$-smooth if it is differentiable and if its gradient is $L$-Lipschitz continuous, i.e., for any $x, y \in \mathbb{R}^d$,

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L \|x - y\|_2. \quad (9)$$

**Lemma 7.** If $f$ is $L_f$-smooth and $g$ is $L_g$-smooth, then $f + g$ is $(L_f + L_g)$-smooth.

**Proof.** Indeed, $\|\nabla (f + g)(x) - \nabla (f + g)(y)\|_2 \leq \|\nabla f(x) - \nabla f(y)\|_2 + \|\nabla g(x) - \nabla g(y)\|_2 \leq L_f \|x - y\|_2 + L_g \|x - y\|_2 = (L_f + L_g) \|x - y\|_2$. \hfill $\Box$

**Lemma 8.** Suppose that $f : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is locally strongly convex and $L$-smooth, and that, for any $x \in X$, where $X \subset \mathbb{R}^d$ is a convex compact subset, the map $y \mapsto f(x, y)$ has a minimum $y^*(x)$. Note that local strong convexity guarantees the uniqueness of this minimum. Then, there exists $K$ such that the function $y^*$ is $K$-Lipschitz continuous on $X$.

**Proof.** The existence and uniqueness of $y^*(x)$ hold by strong convexity. Fix $x, x'$. By optimality of $y^*$, we know that $\nabla_y f(x, y^*(x)) = \nabla_y f(x', y^*(x')) = 0$. We then have the following bounds

$$\mu \|y^*(x) - y^*(x')\|_2 \leq \|\nabla_y f(x, y^*(x)) - \nabla_y f(x, y^*(x'))\|_2 = \|\nabla_y f(x, y^*(x'))\|_2 \quad (10)$$
$$= \|\nabla_y f(x, y^*(x')) - \nabla_y f(x', y^*(x'))\|_2 \quad (11)$$
$$\leq \|\nabla f(x, y^*(x')) - \nabla f(x', y^*(x'))\|_2 \quad (12)$$
$$\leq L \|x - x', y^*(x') - y^*(x')\|_2 = L \|x - x'\|_2, \quad (13)$$

where we first used the local strong convexity assumption, then the fact that $\nabla_y f(x, y^*(x)) = 0$, then the fact that $\nabla_y f(x', y^*(x')) = 0$, and then the $L$-smooth assumption. \hfill $\Box$

**Lemma 9.** Suppose that $f : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is locally strongly convex and $L$-smooth, and that, for any $x \in X$, where $X \subset \mathbb{R}^d$ is a convex compact subset, the map $y \mapsto f(x, y)$ has a minimum $y^*(x)$. Define $g(x) = \min_{y \in X} f(x, y)$. Then $g$ is convex and differentiable on $X$ and $\nabla g(x) = \nabla_x f(x, y^*(x))$.

**Proof.** First we prove that $g$ is convex. Let $x_1, x_2 \in \mathbb{R}^d$, and $\lambda_1, \lambda_2 \in [0, 1]$ with $\lambda_1 + \lambda_2 = 1$. For any $y_1, y_2 \in \mathbb{R}^d$, we have

$$g(\lambda_1 x_1 + \lambda_2 x_2) = \min_{y \in \mathbb{R}^d} f(\lambda_1 x_1 + \lambda_2 x_2, y) \quad (14)$$
$$\leq f(\lambda_1 x_1 + \lambda_2 x_2, \lambda_1 y_1 + \lambda_2 y_2) \quad (15)$$
$$\leq \lambda_1 f(x_1, y_1) + \lambda_2 f(x_2, y_2). \quad (16)$$
Taking the infimum of the right-hand side over \( y_1 \) and \( y_2 \) yields \( g(\lambda_1 x_1 + \lambda_2 x_2) \leq \lambda_1 g(x_1) + \lambda_2 g(x_2) \), which proves the convexity of \( g \).

Now denote \( h(x) = \nabla_x f(x, y^*(x)) \). We aim to show that \( \nabla g(x) = h(x) \). Let \( \varepsilon \in \mathbb{R}^d \) small enough so that \( x + \varepsilon \in X \). Now note that we have

\[
g(x + \varepsilon) = \min_{y \in \mathbb{R}^d} f(x + \varepsilon, y) \leq f(x + \varepsilon, y^*(x))
\]

\[
= f(x, y^*(x)) + \varepsilon^T \nabla_x f(x, y^*(x)) + o(\|\varepsilon\|_2)
\]

\[
= g(x) + \varepsilon^T h(x) + o(\|\varepsilon\|_2),
\]

which shows that \( h(x) \) is also a subderivative. To do so, first note that its value at \( x + \varepsilon \) is approximately the same, i.e.

\[
\| h(x + \varepsilon) - h(x) \|_2 \leq \| \nabla_x f(x + \varepsilon, y^*(x) + \varepsilon^T (x, y^*(x)) \|_2
\]

\[
+ \| \nabla_x f(x, y^*(x) + \varepsilon^T (x, y^*(x)) - \nabla_x f(x, y^*(x)) \|_2
\]

\[
\leq L \| \varepsilon \|_2 + L \| y^*(x + \varepsilon) - y^*(x) \|_2 \leq \left( L + \frac{L^2}{\mu} \right) \| \varepsilon \|_2,
\]

where we used the \( L \)-smoothness of \( f \) and Lemma 8. Now notice that

\[
g(x) = \min_{y \in \mathbb{R}^d} f(x, y) \leq f(x, y^*(x + \varepsilon)) = f((x + \varepsilon) - \varepsilon, y^*(x + \varepsilon))
\]

\[
= f(x + \varepsilon, y^*(x + \varepsilon)) - \varepsilon^T \nabla_x f(x + \varepsilon, y^*(x + \varepsilon)) + o(\|\varepsilon\|_2)
\]

\[
= g(x + \varepsilon) - \varepsilon^T h(x) - \varepsilon^T (h(x + \varepsilon) - h(x)) + o(\|\varepsilon\|_2),
\]

But we know that \( \| h(x + \varepsilon) - h(x) \|_2 = O(\|\varepsilon\|_2) \). Rearranging the terms then yields

\[
g(x + \varepsilon) \geq g(x) + \varepsilon^T h(x) - o(\|\varepsilon\|_2),
\]

which shows that \( h(x) \) is also a subderivative. Therefore, we know that \( g(x + \varepsilon) = g(x) + \varepsilon^T h(x) + o(\|\varepsilon\|_2) \), which boils down to saying that \( g \) is differentiable in \( x \in X \), and that \( \nabla g(x) = h(x) \).

**Lemma 10.** Suppose that \( f : X \times \mathbb{R}^d \to \mathbb{R} \) is \( \mu \)-strongly convex, where \( X \subset \mathbb{R}^d \) is closed and convex. Then \( g : X \to \mathbb{R} \), defined by \( g(x) = \inf_{y \in Y} f(x, y) \), is well-defined and \( \mu \)-strongly convex too.

**Proof.** The function \( y \mapsto f(x, y) \) is still strongly convex, which means that it is at least equal to a quadratic approximation around 0, which is a function that goes to infinity in all directions as \( \|y\|_2 \to \infty \). This proves that the infimum must be reached within a compact set, which implies the existence of a minimum. Thus \( g \) is well-defined. Moreover, for any \( x_1, x_2 \in X, y_1, y_2 \in \mathbb{R}^d \), and \( \lambda_1, \lambda_2 \geq 0 \) with \( \lambda_1 + \lambda_2 = 1 \), we have

\[
g(\lambda_1 x_1 + \lambda_2 x_2) = \inf_y f(\lambda_1 x_1 + \lambda_2 x_2, y)
\]

\[
\leq f(\lambda_1 x_1 + \lambda_2 x_2, \lambda_1 y_1 + \lambda_2 y_2)
\]

\[
\leq \lambda_1 f(x_1, y_1) + \lambda_2 f(x_2, y_2) - \frac{\mu}{2} \lambda_1 \lambda_2 \| (x_1 - x_2, y_1 - y_2) \|_2^2
\]

\[
\leq \lambda_1 f(x_1, y_1) + \lambda_2 f(x_2, y_2) - \frac{\mu}{2} \lambda_1 \lambda_2 \| x_1 - x_2 \|_2^2,
\]

where we used the \( \mu \)-strong convexity of \( f \). Taking the infimum over \( y_1, y_2 \) implies the \( \mu \)-strong convexity of \( g \).

**A.2. Applications to Loss**

Now instead of proving our theorems for different cases separately, we make the following assumptions on the components of the global loss that encompasses both \( \ell_2^2 \) and smooth-\( \ell_2 \) regularization, as well as linear regression and logistic regression.

**Assumption 1.** Assume that \( \ell \) is convex and \( \ell\)-smooth, and that \( R(\rho, \theta) = R_0(\rho - \theta) \), where \( R_0 : \mathbb{R}^d \to \mathbb{R} \) is locally strongly convex (i.e. strongly convex on any convex compact set), \( L_{R_0} \)-smooth and satisfy \( R_0(z) = \Omega(\|z\|_2) \) as \( \|z\|_2 \to \infty \).
Lemma 11. Under Assumption 1, LOSS is locally strongly convex and L-smooth.

Proof. All terms of LOSS are \( L_0 \)-smooth, for an appropriate value of \( L_0 \). By Lemma 7, their sum is thus also \( L \)-smooth, for an appropriate value of \( L \). Now, given Lemma 6, to prove that LOSS is locally strongly convex, it suffices to prove that
\[
\nu \sum_{n \in [N]} \| \theta_n \|_2^2 + R_0(\rho - \theta) \text{ is locally strongly convex.}
\]
Consider any convex compact set \( C \subset \mathbb{R}^{d \times (1+N)} \). Since \( R_0 \) is locally strongly convex, we know that there exists \( \mu > 0 \) such that \( \nabla^2 R_0 \succeq \mu I \). As a result,
\[
(\rho, \bar{\theta})^T (\nabla^2 \text{LOSS}) (\rho, \bar{\theta}) \geq \nu \sum_{n \in [N]} \| \theta_n \|_2^2 + \mu \| \rho - \theta \|_2^2 \tag{30}
\]
\[
= \nu \| \theta_1 \|_2^2 + \mu \| \rho \|_2^2 + \mu \| \theta_1 \|_2^2 - 2\mu \rho^T \theta_1 + \nu \sum_{n \neq 1} \| \theta_n \|_2^2. \tag{31}
\]
Now define \( \alpha \triangleq \frac{2\mu}{\nu + \nu^2} \). Clearly, \( 0 < \alpha < 1 \). Moreover, \( 0 \leq \| \frac{1}{\alpha} \theta_1 - \alpha \rho \|_2^2 = \frac{1}{\alpha^2} \| \theta_1 \|_2^2 + \alpha^2 \| \rho \|_2^2 - 2\rho^T \theta_1 \). Therefore
\[
2\rho^T \theta_1 \leq \alpha^2 \| \rho \|_2^2 + \frac{1}{\alpha^2} \| \theta_1 \|_2^2, \tag{32}
\]
which thus implies
\[
(\rho, \bar{\theta})^T (\nabla^2 \text{LOSS}) (\rho, \bar{\theta}) \geq (\nu + \mu (1 - \alpha^{-2})) \| \theta_1 \|_2^2 + \mu (1 - \alpha^2) \| \rho \|_2^2 + \nu \sum_{n \neq 1} \| \theta_n \|_2^2 \geq \min \left\{ \frac{\nu}{2}, \frac{2\mu}{\nu + \nu^2} \right\} \| (\rho, \bar{\theta}) \|_2^2, \tag{33}
\]
which proves that \( \nabla^2 \text{LOSS} \succeq \kappa I \), with \( \kappa > 0 \). This shows that LOSS is locally strongly convex. \( \square \)

Lemma 12. Under Assumption 1, \( \rho \mapsto \hat{\theta}^*(\rho, \bar{D}) \) is Lipchitz continuous on any compact set.

Proof. Define \( f_n(\rho, \theta) \triangleq \nu \| \theta_n \|_2^2 + \sum_{x \in |D_n|} \ell(\theta_n, x) + \lambda \| \rho - \theta_n \|_2^2 \). If \( \ell \) is \( L \)-smooth, then \( f_n \) is clearly \( (|D_n| \cdot L + \nu + \lambda) \)-smooth. Moreover, if \( \ell \) is convex, then for any \( \rho \), the function \( \theta_n \mapsto f_n(\rho, \theta_n) \) is at least \( \nu \)-strongly convex. Thus Lemma 8 applies, which guarantees that \( \rho \mapsto \hat{\theta}^*(\rho, \bar{D}) \) is Lipchitz. \( \square \)

Lemma 13. Under Assumption 1, \( \rho \mapsto \text{LOSS}(\rho, \hat{\theta}^*(\rho, \bar{D}), \bar{D}) \) is L-smooth and locally strongly convex.

Proof. By Lemma 11, the global loss is known to be \( L \)-smooth, for some value of \( L \) and locally strongly convex. Denoting \( f : \rho \mapsto \text{LOSS}(\rho, \hat{\theta}^*(\rho, \bar{D}), \bar{D}) \), we then have
\[
\| \nabla f(\rho) - \nabla f(\rho') \|_2 \leq \left\| \nabla_{\rho} \text{LOSS}(\rho, \hat{\theta}^*(\rho, \bar{D}), \bar{D}) - \nabla_{\rho} \text{LOSS}(\rho', \hat{\theta}^*(\rho', \bar{D}), \bar{D}) \right\|_2 \tag{34}
\]
\[
\leq L \left\| (\rho, \hat{\theta}^*(\rho, \bar{D})) - (\rho', \hat{\theta}^*(\rho', \bar{D})) \right\|_2 \tag{35}
\]
\[
\leq L \| \rho - \rho' \|_2, \tag{36}
\]
which proves that \( f \) is \( L \)-smooth.

For strong convexity, note that since the global loss function is locally strongly convex, for any compact convex set \( C \), there exists \( \mu \) such that \( \text{LOSS}(\rho, \hat{\theta}, \bar{D}) \) is \( \mu \)-strongly convex on \( C = (C_1, C_2) \subset (\mathbb{R}^d, \mathbb{R}^{N \times d}) \), therefore, by Lemma 10, \( f(\rho) \) will also be \( \mu \)-strongly convex on \( C_1 \) which means that \( f(\rho) \) is locally strongly convex. \( \square \)

B. Proof of the Equivalence

B.1. Proof of the Reduction from Model Attack to Data Poisoning

Proof of Lemma 1. We omit making the dependence of the optima on \( \bar{D} \) explicit, and we consider any other models \( \rho \) and \( \bar{\theta} \). We have the following inequalities:
\[
\text{LOSS}_s(\rho^*, \bar{\theta}^*_s, \theta^*_{\phi}, \bar{D}) = \text{LOSS}(\rho^*, \bar{\theta}^*_s, \bar{D}) - \mathcal{L}(\theta^*_{\phi}, D_s) \quad (37)
\]
\[
\leq \text{LOSS}(\rho, \bar{\theta}^*_s, \theta^*_{\phi}, \bar{D}) - \mathcal{L}(\theta^*_{\phi}, D_s) = \text{LOSS}_s(\rho, \bar{\theta}^*_s, \theta^*_{\phi}, \bar{D}), \quad (38)
\]
where we used the optimality of \( (\rho^*, \bar{\theta}^*_s) \) in the second line, and where we repeatedly used the fact that \( \theta^*_{\phi} = \theta^*_{\phi} \). This proves that \( (\rho^*, \bar{\theta}^*_s) \) is a global minimum of the modified loss. \( \square \)
B.2. Proof of the Reduction from Data Poisoning to Model Attack

First, we define the following modified loss function:

\[ \text{LOSS}_s(\rho, \bar{\theta}_s, \theta_s^*, \bar{D}_s) \triangleq \text{LOSS}(\rho_s, (\theta_s^*, \bar{\theta}_s), (\emptyset, \bar{D}_s)) \]

where \( \bar{\theta}_s \) and \( \bar{D}_s \) are variables and datasets for users \( n \neq s \). We then define \( \rho_s^*(\theta_s^*, \bar{D}_s) \) and \( \rho_s^*(\theta_s^*, \bar{D}_s) \) as a minimum of the modified loss function, and \( \theta_s^*(\theta_s^*, \bar{D}_s) \) as a minimum of the modified loss function, and \( \theta_s^*(\theta_s^*, \bar{D}_s) \) as a minimum of the modified loss function. We now prove a slightly more general version of Lemma 2, which applies to a larger class of regularizations. It also shows how to construct the strategic’s user data poisoning attack.

**Lemma 14** (Reduction from data poisoning to model attack). Assume local PAC* learning. Suppose also that \( R \) is continuous and that \( R(\rho, \theta) \to \infty \) when \( \|\rho - \theta\|_2 \to \infty \). Consider any datasets \( D_\ldots \) and any attack model \( \theta_\ldots \) such that the modified loss \( \text{LOSS}_s \) has a unique minimum \( \rho_s^*(\theta_s^*, \bar{D}_s), \bar{\theta}_s^*(\theta_s^*, \bar{D}_s) \). Then, for any \( \varepsilon, \delta > 0 \), there exists \( I \) such that if user \( s \)’s dataset \( D_s \) contains at least \( I \) inputs drawn from model \( \theta_s^* \), then, with probability at least 1 – \( \delta \), we have

\[ \left\| \rho_s^*(\bar{D}) - \rho_s^*(\theta_s^*, \bar{D}_s) \right\|_2 \leq \varepsilon \quad \text{and} \quad \forall n \neq s, \left\| \rho_s^*(\bar{D}) - \rho_s^*(\theta_s^*, \bar{D}_s) \right\|_2 \leq \varepsilon. \]

Clearly, \( \ell_2^2, \ell_2 \) and smooth-\( \ell_2 \) are continuous regularizations, and verify \( R(\rho, \theta) \to \infty \) when \( \|\rho - \theta\|_2 \to \infty \). Moreover, setting \( \delta \equiv 1/2 \) shows that the probability that the dataset \( D_s \) satisfies the inequalities of Lemma 14 is positive. This implies in particular that there must be a dataset \( D_s \) that satisfies these inequalities. All in all, this shows that Lemma 14 implies Lemma 2.

**Proof of Lemma 14.** Let \( \varepsilon, \delta > 0 \) and \( \theta_s^* \in \mathbb{R}^d \). Denote \( \rho_s^* \equiv \rho_s^*(\theta_s^*, \bar{D}_s) \) and \( \bar{\rho}_s^* \equiv \bar{\rho}_s^*(\theta_s^*, \bar{D}_s) \) the result of strategic user \( s \)’s model attack. We define the compact set \( C \) by

\[ C \triangleq \left\{ \rho, \bar{\theta}_s : \|\rho - \rho_s^*\|_2 \leq \varepsilon \land \forall n \neq s, \|\rho_s^* - \rho_n^*\|_2 \leq \varepsilon \right\}. \]

We define \( D \equiv \mathbb{R}^{d \times N} - C \) the closure of the complement of \( C \). Clearly, \( \rho_s^*, \bar{\rho}_s^* \notin D \). We aim to show that, when strategic user \( s \) reveals a large dataset \( D_s \) whose answers are provided using the attack model \( \theta_s^* \), then the same holds for any global minimum of the global loss \( \rho^*(\bar{D}), \bar{\rho}_s^*(\theta_s^*, \bar{D}_s) \in C \). Note that, to prove this, it suffices to prove that the modified loss takes large values, even when \( \theta_s^* \) is replaced by \( \bar{\theta}_s^*(\bar{D}) \).

Let us now formalize this. Denote \( L_s^\bullet \equiv \text{LOSS}_s(\rho_s^*, \bar{\theta}_s^*, \theta_s^*, \bar{D}_s) \). We define

\[ \eta \equiv \inf_{\rho, \bar{\theta}_s \in D} \text{LOSS}_s(\rho, \bar{\theta}_s, \theta_s^*, \bar{D}_s) - L_s^\bullet. \]

By a similar argument as that of Lemma 5, using the assumption \( R \to \infty \) at infinity, we know that the infimum is actually a minimum. Moreover, given that the minimum of the modified loss \( \text{LOSS}_s \) is unique, we know that the value of the loss function at this minimum is different from its value at \( \rho_s^*, \bar{\theta}_s^* \). As a result, we must have \( \eta > 0 \).

Now, since the function \( R \) is differentiable, it must be continuous. By the Heine–Cantor theorem, it is thus uniformly continuous on all compact sets. Thus, there must exist \( \kappa > 0 \) such that, for all models \( \theta_s \) satisfying \( \|\theta_s - \theta_s^*\|_2 \leq \kappa \), we have

\[ |R(\theta_s, \rho_s^*) - R(\theta_s^*, \rho_s^*)| \leq \eta/3. \]

Now, Lemma 5 guarantees the existence of \( I \) such that, if user \( s \) provides a dataset \( D_s \) of at least \( I \) answers with the model \( \theta_s^* \), then with probability at least 1 – \( \delta \), we will have \( \|\theta_s^*(\bar{D}) - \theta_s^*\|_2 \leq \min(\kappa, \varepsilon) \). Under this event, we then have

\[ \text{LOSS}_s(\rho_s^*, \bar{\theta}_s^*, \theta_s^*(\bar{D}), \bar{D}_s) \leq L_s^\bullet + \eta/3. \]

Then

\[ \inf_{\rho, \bar{\theta}_s \in D} \text{LOSS}_s(\rho, \bar{\theta}_s, \theta_s^*(\bar{D}), \bar{D}_s) \geq \inf_{\rho, \bar{\theta}_s \in D} \text{LOSS}_s(\rho, \bar{\theta}_s, \theta_s^*, \bar{D}_s) - \eta/3 \]

\[ \geq L_s^\bullet + \eta - \eta/3 \geq L_s^\bullet + 2\eta/3 \]

\[ \geq \text{LOSS}_s(\rho_s^*, \bar{\theta}_s^*, \theta_s^*(\bar{D}), \bar{D}_s). \]
This shows that there is a high probability event under which the minimum of \( \rho, \bar{\theta}_n \mapsto \text{LOSS}_s(\rho, \bar{\theta}_n, \theta^*_s(\bar{D}), \bar{D}_n) \) cannot be reached in \( D \). This is equivalent to what the theorem we needed to prove states.

\[ \text{B.3. Proof of Reduction from Model Attack to Gradient Attack} \]

\textit{Proof of Lemma 3.}\hspace{1em} We define

\[
\text{LOSS}_s^1(\rho) \triangleq \inf_{\bar{\theta}_n} \left\{ \text{LOSS}(\rho, \bar{\theta}, \bar{D}) - \mathcal{L}_n(\theta_n, D_n) - \mathcal{R}(\rho, \theta_n) \right\} + \rho^T g_s^\infty
\]

\[
= \inf_{\bar{\theta}_n} \left\{ \sum_{n \neq s} \mathcal{L}_n(\theta_n, D_n) + \sum_{n \neq s} \mathcal{R}(\rho, \theta_n) \right\} + \rho^T g_s^\infty,
\]

By Lemma 13, we know that \( \text{LOSS}_s^1(\rho) \) is locally strongly convex and has a unique minimum. By the definition of \( \rho^\infty \), we must have \( \sum_{n \neq s} \nabla \mathcal{R}(\rho^\infty, \theta^*_s(\rho^\infty)) + g_s^\infty = 0 \), and thus \( \nabla \rho \text{LOSS}_s^1(\rho^\infty) = 0 \). Now define

\[
\text{LOSS}_s^2(\rho, \theta_s) \triangleq \inf_{\bar{\theta}_n} \left\{ \text{LOSS}(\rho, \bar{\theta}, \bar{D}) - \mathcal{L}_n(\theta_n, D_n) \right\}
\]

\[
= \text{LOSS}_s^1(\rho) + \mathcal{R}(\rho, \theta_s) - \rho^T g_s^\infty,
\]

and \( \rho^*(\theta_s) \), its minimizer. Therefore, we have

\[
\nabla \rho \text{LOSS}_s^2(\rho, \theta_s) = \nabla \rho \text{LOSS}_s^1(\rho) + \nabla \rho \mathcal{R}(\rho, \theta_s) - g_s^\infty.
\]

By Lemma 13, we know that \( \text{LOSS}_s^2(\rho, \theta_s) \) is \( \mu_1 \)-strongly convex in \( \{ \theta_s, \rho : \| \nabla \mathcal{R}(\rho^\infty, \theta^*_s(\rho^\infty)) - g_s^\infty \|_2 \leq \varepsilon_2, \| \rho - \rho^*(\theta_s) \|_2 \leq 1 \} \) for \( \varepsilon_2 \) small enough. Therefore, since \( \nabla \rho \text{LOSS}_s^2(\rho^*(\theta_s), \theta_s) = 0 \), for any \( 0 < \varepsilon < 1 \), if \( \| \rho^\infty - \rho^*(\theta_s) \|_2 \| > \varepsilon \), we then have

\[
\varepsilon \| \nabla \rho \text{LOSS}_s^2(\rho^\infty, \theta_s) \|_2 \geq (\rho^\infty - \rho^*(\theta_s))^T \nabla \rho \text{LOSS}_s^2(\rho^\infty, \theta_s)
\]

\[
\geq \mu_1 \| \rho^\infty - \rho^*(\theta_s) \|_2 \varepsilon \geq \frac{\mu_1 \varepsilon^2}{2},
\]

and thus \( \| \nabla \rho \text{LOSS}_s^2(\rho^\infty, \theta_s) \|_2 \geq \mu_1 \varepsilon \).

Now since \( g_s^\infty \in \text{GRAD}(\rho^\infty) \) there exists \( \theta_s^\bullet \in \mathbb{R}^d \) such that \( \| \nabla \rho \mathcal{R}(\rho^\infty, \theta_s^\bullet) - g_s^\infty \|_2 \leq \min \{ \varepsilon_2, \frac{\mu_1 \varepsilon}{2} \} \) which yields

\[
\| \nabla \rho \text{LOSS}_s^2(\rho^\infty, \theta_s^\bullet) \|_2 = \| \nabla \rho \text{LOSS}_s^1(\rho^\infty) + \nabla \rho \mathcal{R}(\rho^\infty, \theta_s^\bullet) - g_s^\infty \|_2
\]

\[
= \| \nabla \rho \mathcal{R}(\rho^\infty, \theta_s^\bullet) - g_s^\infty \|_2 \leq \frac{\mu_1 \varepsilon}{2},
\]

which contradicts (54) if \( \| \rho^\infty - \rho^*(\theta_s^\bullet) \|_2 > \varepsilon \). Therefore, we must have \( \| \rho^\infty - \rho^*(\theta_s^\bullet, \bar{D}_n) \|_2 \leq \varepsilon. \)

\[ \text{C. Proof of Convergence for the Global Model} \]

In this section, we prove a slightly more general result than Proposition 1. Namely, instead of working with specific regularizations, we consider a more general class of regularizations, identified by Assumption 1.

\textbf{Lemma 15.} Suppose Assumption 1 holds true. Assume that \( \mathcal{L}_n \) is convex and \( L \)-smooth for all users \( n \in [N] \). If \( g_s^l \) converges and if \( \eta_l = \eta \) is a constant small enough, then \( \rho^l \) will converge too.

Note that since \( t_2^l \) and smooth-\( \ell_2 \) regularizations satisfy Assumption 1, Lemma 15 clearly implies Proposition 1. We now introduce the key objects of the proof of Lemma 15.

\[ ^{\text{In fact, if } g_s^\infty \text{ belongs to the interior of } \text{GRAD}(\rho^\infty), \text{ we can guarantee } \nabla \rho \mathcal{R}(\rho^\infty, \theta_s^\bullet) = g_s^\infty.} \]
Denote \( g_s^\infty \) the limit of the attack gradients \( g_s^t \). We now define

\[
\text{LOSS}_s^1(\rho) \triangleq \inf_{\theta_s} \left\{ \text{LOSS}(\rho, \theta_s, D_s) - \mathcal{R}(\rho, \theta_s) \right\} + \rho^T g_s^\infty
\]

(57)

and prove that \( \rho^t \) will converge to the minimizer of \( \text{LOSS}_s^1(\rho) \). By Lemma 13, we know that \( \text{LOSS}_s^1(\rho) \) is both locally strongly convex and \( L \)-smooth.

Now define \( \zeta_s^t \triangleq g_s^t - g_s^\infty \). We then have \( \zeta_s^t \rightarrow 0 \) and \( \nabla \text{LOSS}_s^1(\rho^t) \) is the sum of all gradient vectors received from all users assuming the strategic user \( s \) sends the vector \( g_s^\infty \) in all iterations. Thus, at iteration \( t \) of the optimization algorithm, we will take one step in the direction \( \rho^t \triangleq \nabla \text{LOSS}_s^1(\rho^t) + \zeta_s^t \), i.e.,

\[
\rho^{t+1} = \rho^t - \eta_t G^t.
\]

(59)

We now prove the following lemma that bounds the difference between the function value in two successive iterations.

**Lemma 16.** If \( \text{LOSS}_s^1(\rho) \) is \( L \)-smooth and \( \eta_t \leq 1/L \), we have

\[
\text{LOSS}_s^1(\rho^{t+1}) - \text{LOSS}_s^1(\rho^t) \leq -\frac{\eta_t}{2} \|G^t\|^2 + \eta_t \zeta_s^{tT} G^t.
\]

(60)

**Proof.** Since \( \text{LOSS}_s^1 \) is \( L \)-smooth, we have

\[
\text{LOSS}_s^1(\rho^{t+1}) \leq \text{LOSS}_s^1(\rho^t) + (\rho^{t+1} - \rho^t)^T \nabla \text{LOSS}_s^1(\rho^t) + \frac{L}{2} \|\rho^{t+1} - \rho^t\|^2.
\]

(61)

Now plugging \( \rho^{t+1} - \rho^t = -\eta_t G^t \) and \( \nabla \text{LOSS}_s^1(\rho^t) = G^t - \zeta_s^t \) into the inequality implies

\[
\text{LOSS}_s^1(\rho^{t+1}) - \text{LOSS}_s^1(\rho^t) \leq \left(-\eta_t G^t\right)^T (G^t - \zeta_s^t) + \frac{L}{2} \|G^t - \zeta_s^t\|^2
\]

(62)

\[
\leq -\frac{\eta_t}{2} \|G^t\|^2 + \eta_t \zeta_s^{tT} G^t,
\]

(63)

where we used the fact \( \eta_t \leq 1/L \).

\( \blacksquare \)

**C.0.1.** **The Global Model Remains Bounded**

**Lemma 17.** There is \( M \) such that, for all \( t \), \( \text{LOSS}_s^1(\rho^t) \leq M \).

**Proof.** Consider the closed ball \( B(\rho^*, 1) \) centered on \( \rho^* \) and of radius 1. By Lemma 13, we know that \( \text{LOSS}_s^1 \) is locally strongly convex and thus there exists a \( \mu_1 > 0 \) such that \( \text{LOSS}_s^1 \) is \( \mu_1 \)-strongly convex on \( B(\rho^*, 1) \). Now consider a point \( \rho_1 \) on the boundary of \( B(\rho^*, 1) \). By strong convexity we have

\[
\|\nabla \text{LOSS}_s^1(\rho_1)\|_2 \geq (\rho_1 - \rho^*)^T \nabla \text{LOSS}_s^1(\rho_1) \geq \mu_1 \|\rho_1 - \rho^*\|^2 = \mu_1.
\]

(64)

Now similarly, by the convexity of \( \text{LOSS}_s^1 \) on \( \mathbb{R}^d \), for any \( \rho \in \mathbb{R}^d - B(\rho^*, 1) \), we have \( \|\nabla \text{LOSS}_s^1(\rho_1)\|_2 \geq \sqrt{\mu_1} \). Now since \( \zeta_s^t \rightarrow 0 \), there exists an iteration \( T_1 \) after which (\( t \geq T_1 \)), we have \( \|\zeta_s^t\|_2 \leq \frac{3}{4} \sqrt{\mu_1} \), and thus \( \|G^t\|_2 \geq \|\nabla \text{LOSS}_s^1(\rho^t)\|_2 - \|\zeta_s^t\|_2 \geq \frac{3}{4} \sqrt{\mu_1} \). Thus, Lemma 16 implies that for \( t \geq T_1 \), if \( \|\rho^t - \rho^*\|_2 \geq 1 \), then

\[
\text{LOSS}_s^1(\rho^{t+1}) - \text{LOSS}_s^1(\rho^t) \leq -\frac{\eta_t}{2} \|G^t\|^2 + \eta_t \zeta_s^{tT} G^t
\]

(65)

\[
\leq -\frac{\eta_t}{2} \|G^t\|^2 + \eta_t \zeta_s^{tT} G^t
\]

(66)

\[
\leq -\frac{\eta_t}{2} \|G^t\|^2 \left(\|G^t\|^2 - 2\|\zeta_s^t\|^2\right)
\]

(67)

\[
\leq -\frac{\eta_t}{2} \left(\frac{3}{4} \sqrt{\mu_1}\right) \left(\frac{3}{4} \sqrt{\mu_1} - \frac{2}{4} \sqrt{\mu_1}\right) \leq -\frac{3\eta}{32} \mu_1 < 0.
\]

(68)
Thus, for $\|\rho^t - \rho^*\|_2 \geq 1$, the loss cannot increase at the next iteration.

Now consider the case $\|\rho^t - \rho^*\|_2 < 1$ for $t \geq T_1$. The smoothness of $\text{LOSS}_s^1$ implies $\|\nabla \text{LOSS}_s^1(\rho^t)\|_2 < L$. Therefore,

$$
\|\rho^{t+1} - \rho^*\|_2 = \|\rho^t - \eta(\nabla \text{LOSS}_s^1(\rho^t)) - \rho^*\|_2 \\
\leq \|\rho^t - \rho^*\|_2 + \eta(L + \frac{1}{4}\|\rho^t\|_2) \leq \eta(L + \frac{1}{4}\|\rho^t\|_2). 
$$

Now we define $M_1 \triangleq \max_{\rho \in \mathcal{B}(\rho^*, 1 + \eta(L + \frac{1}{4}\|\rho^t\|_2))} \text{LOSS}_s^1(\rho)$, the maximum function value in the closed ball $\mathcal{B}(\rho^*, 1 + \eta(L + \frac{1}{4}\|\rho^t\|_2))$. Therefore, we have $\text{LOSS}_s^1(\rho^{t+1}) \leq M_1$. So far we proved that for $t \geq T_1$, in each iteration of gradient descent either the function value will not increase or it will be upper-bounded by $M_1$. This implies that for all $t$, the function value $\text{LOSS}_s^1(\rho^t)$ is upper-bounded by

$$
M \triangleq \max_{t \leq T_1} \left\{ \text{LOSS}_s^1(\rho^t) \right\}, M_1.
$$

This concludes the proof.

**Lemma 18.** There is a compact set $X$ such that, for all $t$, $\rho^t \in X$.

**Proof.** Now since $\text{LOSS}_s^1$ is $\mu_1$-strongly convex in $\mathcal{B}(\rho^*, 1)$, for any point $\rho \in \mathbb{R}^d$ such that $\|\rho - \rho^t\|_2 = 1$, we have

$$
\text{LOSS}_s^1(\rho) \geq \text{LOSS}_s^1(\rho^*) + \frac{\mu_1}{2}\|\rho - \rho^*\|_2^2 = \text{LOSS}_s^1(\rho^*) + \frac{\mu_1}{2}. 
$$

But now by the convexity of $\text{LOSS}_s^1$ in $\mathbb{R}^d$, for any $\rho$ such that $\|\rho - \rho^*\|_2 \geq 1$, we have

$$
\text{LOSS}_s^1(\rho) \geq \text{LOSS}_s^1(\rho^*) + \|\rho - \rho^*\|_2 \frac{\mu_1}{2}. 
$$

This implies that if $\|\rho^t - \rho^*\|_2 > \frac{\mu_1}{2} (M_2 - \text{LOSS}_s^1(\rho^*))$, then $\text{LOSS}_s^1(\rho^t) > M_2$. Therefore, we must have $\|\rho^t - \rho^*\|_2 \leq \frac{\mu_1}{2} (M_2 - \text{LOSS}_s^1(\rho^*))$, for all $t \geq 0$. This describes a closed ball, which is a compact set.

**C.0.2. Convergence of the Global Model under Converging Gradient Attack**

**Lemma 19.** Suppose $u_t \geq 0$ verifies $u_{t+1} \leq \alpha u_t + \delta_t$, with $\delta_t \to 0$. Then $u_t \to 0$.

**Proof.** We now show that for any $\varepsilon > 0$, there exists an iteration $T(\varepsilon)$, such that for $t \geq T(\varepsilon)$, we have $u_t \leq \varepsilon$. For this, note that by induction, we observe that, for all $t \geq 0$,

$$
u_{t+1} \leq u_0 \alpha^{t+1} + \sum_{\tau=0}^{t} \alpha^\tau \delta_{t-\tau}. 
$$

Since $\delta_t \to 0$, there exists an iteration $T_2(\varepsilon)$ such that for all $t \geq T_2(\varepsilon)$, we have $\delta_t \leq \frac{\varepsilon(1-\alpha)}{2}$. Therefore, for $t \geq T_2(\varepsilon)$, we have

$$
u_{t+1} \leq u_0 \alpha^{t+1} + \sum_{\tau=0}^{t-T_2(\varepsilon)} \alpha^\tau \delta_{t-\tau} + \sum_{\tau=T_2(\varepsilon)+1}^{t} \alpha^\tau \delta_{t-\tau} \\
\leq u_0 \alpha^{t+1} + \frac{\varepsilon(1-\alpha)}{2} \sum_{\tau=0}^{t-T_2(\varepsilon)} \alpha^\tau + \sum_{s=0}^{T_2(\varepsilon)-1} \alpha^{t-s} \delta_s \\
\leq \left( u_0 + \sum_{s=0}^{T_2(\varepsilon)-1} \alpha^{t-s-1} \delta_s \right) \alpha^{t+1} + \frac{\varepsilon(1-\alpha)}{2} \sum_{\tau=0}^{\infty} \alpha^\tau. 
$$
Data poisoning and Byzantine gradients

Denoting $M_0(\varepsilon) \triangleq \sum_{s=0}^{T_2(\varepsilon)-1} \alpha^{-s-1} \delta_s$, we then have

$$u_{t+1} \leq (u_0 + M_0(\varepsilon)) \alpha^{t+1} + \frac{\varepsilon}{2}.$$  \hfill (78)

Therefore, for $t \geq \frac{\ln \frac{\mu_2 + M_0(\varepsilon)}{\ln \alpha}}{\ln \alpha}$, we have

$$u_{t+1} \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$  \hfill (79)

This proves that $u_t \to 0$.  \hfill \Box

We now prove Lemma 15 (and hence Proposition 1).

Proof of Lemma 15. Define $X$ based on Lemma 18. Since $\text{LOSS}_s^1$ is locally strongly convex, there exists $\mu_2 > 0$ such that $\text{LOSS}_s^1$ is $\mu_2$-strongly convex in a convex compact set $X$ containing $\rho^t$ for all $t \geq 0$. By the strong convexity of $\text{LOSS}_s^1(\rho)$, we have

$$\text{LOSS}_s^1(\rho^t) - \text{LOSS}_s^1(\rho^*) \leq (\rho^t - \rho^*)^T \nabla \text{LOSS}_s^1(\rho^t) - \frac{\mu_2}{2} \|\rho^t - \rho^*\|^2_2$$  \hfill (80)

Now, using the fact

$$(\rho^t - \rho^*)^T G^t = \frac{1}{\eta} (\rho^t - \rho^*)^T (\rho^t - \rho^{t+1})$$  \hfill (82)

we have

$$\text{LOSS}_s^1(\rho^t) - \text{LOSS}_s^1(\rho^*) \leq \frac{\eta}{2} \|G^t\|^2_2 + \frac{1}{2\eta} \left( \|\rho^t - \rho^*\|^2_2 - \|\rho^{t+1} - \rho^*\|^2_2 - (\rho^t - \rho^*)^T \zeta_s - \frac{\mu_2}{2} \|\rho^t - \rho^*\|^2_2 \right).$$  \hfill (83)

But now note that $\text{LOSS}_s^1(\rho^t) - \text{LOSS}_s^1(\rho^*) \geq \text{LOSS}_s^1(\rho^t) - \text{LOSS}_s^1(\rho^{t+1})$. Thus, combining Equation (87) and Lemma 16 yields

$$-\eta \zeta_s^T G^t \leq \frac{1}{2\eta} \left( \|\rho^t - \rho^*\|^2_2 - \|\rho^{t+1} - \rho^*\|^2_2 \right).$$  \hfill (88)

By rearranging the terms, we then have

$$\|\rho^{t+1} - \rho^*\|^2_2 \leq (1 - \mu_2\eta) \|\rho^t - \rho^*\|^2_2 - \eta \left( \rho^t - \rho^* \right)^T \zeta_s - \frac{\mu_2}{2} \|\rho^t - \rho^*\|^2_2.$$  \hfill (89)

Now note that $\eta \leq 1/L < 1/\mu_2$ and thus $0 < 1 - \mu_2\eta < 1$. We now define two sequences $u_t \triangleq \|\rho^t - \rho^*\|^2_2$ and $\delta_t = \|\zeta_s\|^2_2$. We already know that $\delta_t \to 0$, and we want to show $u_t$ also converges to 0. By Equation (90), we have

$$u_{t+1}^2 \leq (1 - \eta\mu_2) u_t^2 + \delta_t u_{t+1},$$  \hfill (91)

which implies

$$\left( u_{t+1} - \frac{\delta_t}{2} \right)^2 = u_{t+1}^2 - u_{t+1} \delta_t + \frac{\delta_t^2}{4} \leq (1 - \eta\mu_2) u_t^2 + \frac{\delta_t^2}{4},$$  \hfill (92)

and thus

$$u_{t+1} \leq \sqrt{(1 - \eta\mu_2) u_t^2 + \frac{\delta_t^2}{4}} + \frac{\delta_t}{2} \leq \sqrt{(1 - \eta\mu_2) u_t^2 + \frac{\delta_t^2}{2}} \leq \left( 1 - \frac{\eta\mu_2}{2} \right) u_t + \frac{\delta_t}{2}.$$  \hfill (93)

Lemma 19 allows to conclude.  \hfill \Box
D. Proofs of the Impossibility Corollaries

D.1. Lower Bound on Byzantine Resilience

Proof of Corollary 1. Assume \( d = 1 \). Denote \( H_0 = [N/2] \), \( H_1 = [N/2] \) and \( H_2 = [N/2] \). Let us define \( \theta_n^\dagger = -1 \) for all users \( n \in [H_0] \), \( \theta_n^\dagger = 1 \) for all users \( n \in [2H_0] - [H_0] \) and \( \theta_n^\dagger = 0 \) for all users \( n \in [N] - [2H_0] \) (which is either empty or contains one element). Now fix \( \varepsilon, \delta > 0 \), with \( \varepsilon = 1/4 \) and \( \delta = 1/3 \). Consider the honest datasets \( \mathcal{D} \) of size \( I \) that may have reported, where \( I \) is chosen to guarantee high-probability \((F, N, C)\)-Byzantine learning, as guaranteed by Definition 2. Since the guarantee must hold for \( H \subseteq [H_0] \) and for \( H \subseteq [2H_0] - [H_0] \), with probability at least \( 1 - 2\delta \geq 1/3 > 0 \) (so that both guarantees hold), we must then have \( |\rho^{\text{ALG}} - 1|^2 \leq \varepsilon \) (for \( H \subseteq [H_0] \)) and \( |\rho^{\text{ALG}} - 1|^2 \leq \varepsilon \) (for \( H \subseteq [2H_0] - [H_0] \)). But then, by the triangle inequality, we must have

\[
2 = |1 - \rho^{\text{ALG}} + \rho^{\text{ALG}} - (-1)| \leq |1 - \rho^{\text{ALG}}| + |\rho^{\text{ALG}} - (-1)| \leq \sqrt{\varepsilon} + \sqrt{\varepsilon} = 1/2 + 1/2 = 1.
\]

This is a contradiction. Thus \((F, N, C)\)-Byzantine learning cannot be guaranteed for \( d = 1 \).

D.2. Lower Bound on Correctness

Proof of Corollary 2. Consider \( d = 1 \). Let us define \( \theta_n^\dagger = 0 \) for all users \( n \in |H| \), \( \theta_n^\dagger = 1 \) for all users \( n \in [N] - |H| \). Now fix \( \varepsilon, \delta > 0 \), with \( \varepsilon < F/2(N - F)^2 \) and \( \delta = 1/3 \). Consider the honest datasets \( \mathcal{D} \) of size \( I \) that may have reported, where \( I \) is chosen to guarantee high-probability \((F, N, C)\)-Byzantine learning, as guaranteed by Definition 2. Since the guarantee must hold for \( H = |H| \) and for \( H = |N| - |F| \), with probability at least \( 1 - 2\delta \geq 1/3 > 0 \) (so that both guarantees hold), we must then have \( |\rho^{\text{ALG}}|^2 \leq \varepsilon \) (for \( H = |H| \)) and

\[
|\rho^{\text{ALG}} - \frac{F}{N - F}|^2 \leq C^2 + \varepsilon,
\]

for the case \( H = |N| - |F| \). The first inequality implies \( |\rho^{\text{ALG}}| \leq F/(N - F) \), while the second can then be rewritten

\[
C^2 \geq \left( \frac{F}{N - F} - \varepsilon \right)^2 - \varepsilon.
\]

But this equation is now deterministic. Since it must hold with a strictly positive probability, it must thus hold deterministically. Moreover, it holds for any \( \varepsilon > 0 \). Taking the limit \( \varepsilon \rightarrow 0 \) yields the result.

E. Sum over Expectations

In this section, we provide both theoretical and empirical results to argue for using a sum-based local loss over an expectation-based local loss.

E.1. Theoretical Arguments

Indeed, intuitively, if one considers an expectation \( E_{x \sim D_n} [\ell(\theta_n, x)] \) rather than a sum, as is done by \( \text{(Hanzely et al., 2020), (Dinh et al., 2020) and (El-Mhamdi et al., 2021a)} \), then the weight of an honest active user’s local loss will not increase as a user provides more and more data, which will hinder the ability of \( \theta_n \) to fit the user’s local data. In fact, intuitively, using an expectation wrongly yields the same influence to any two users, even when one (honest) user provides a much larger dataset \( D_n \) than the other, and should thus intuitively be regarded as “more reliable”.

There is another theoretical argument for using the sum rather than the expectation. Namely, if the loss is regarded as a Bayesian negative log-posterior, given a prior \( \exp \left( -\sum_{n \in [N]} \nu \| \theta_n \|^2 - \sum_{n \in [N]} R(\rho, \theta_n) \right) \) on the local and global models, then the term that fits local data should equal the negative log-likelihood of the data, given the models \( (\rho, \hat{\theta}) \). Assuming that the distribution of each data point \( x \in D_n \) is independent from all other data points, and depends only on the local model \( \theta_n \), this negative log-likelihood yields a sum over data points; not an expectation.

E.2. Empirical Results

We also empirically compared the performances of sum as opposed to the expectation. To do so, we constructed a setting where 10 “idle” users draw randomly 10 data points from the FashionMNIST dataset, while one “active” user has all of the
Table 1. Accuracy of trained models, depending on the use of expectation (denoted \(E\)) or sum (\(\Sigma\)), and on the use of linear classifier (\(L\)) or a 2-layer neural net (\(NN\)). Here, all users are honest and an \(\ell^2\) regularization is used, but there is a large heterogeneity in the amount of data per user.

<table>
<thead>
<tr>
<th>Model</th>
<th>(E_L)</th>
<th>(\Sigma_L)</th>
<th>(E_{NN})</th>
<th>(\Sigma_{NN})</th>
</tr>
</thead>
<tbody>
<tr>
<td>idle user’s model</td>
<td>0.52</td>
<td>0.80</td>
<td>0.55</td>
<td>0.79</td>
</tr>
<tr>
<td>active user’s model</td>
<td>0.58</td>
<td>0.80</td>
<td>0.56</td>
<td>0.79</td>
</tr>
<tr>
<td>global model</td>
<td>0.55</td>
<td>0.80</td>
<td>0.58</td>
<td>0.79</td>
</tr>
</tbody>
</table>

Figure 4. Linear model on noisy FashionMNIST, for \(\lambda = 0.01\).

FashionMNIST dataset (60,000 data points). We then learned local and global models, with \(R(\rho, \theta) \triangleq \lambda \|\rho - \theta\|_2^2\), \(\lambda = 1\). We compared two different classifiers to which we refer as a “linear model” and “2-layers neural network”, both using CrossEntropy loss. The linear model has \((784 + 1) \times 10\) parameters. The neural network has 2 layers of 784 parameters with bias, with ReLU activation in between, adding up to \((784 + 1) \times 784 + (784 + 1) \times 10\).

Note also that, in all our experiments, we did not consider any local regularization, i.e. we set \(\nu \triangleq 0\). All our experiments are seeded with seed 999.

E.2.1. Noisy FashionMNIST

To see a strong difference between sum and average, we made the FashionMNIST dataset harder to learn, by randomly labeling 60% of the training set. Table 1 reports the accuracy of local and global models in the different settings. Our results clearly and robustly indicate that the use of sums outperforms the use of expectations.

On each of the following plots, we display the top-1 accuracy on the MNIST test dataset (10 000 images) for the active user, for the global model and for one of the idle users (in Table 1, the mean accuracy for idle users is reported), as we vary the value of \(\lambda\). Intuitively, \(\lambda\) models how much we want the local models to be similar.

In the case of learning FashionMNIST, given that the data is i.i.d., larger values of \(\lambda\) are more meaningful (though our experiments show that they may hinder convergence speed). However, in settings where users have different data distributions, e.g. because the labels depend on users’ preferences, then smaller values of \(\lambda\) may be more relevant.

Note that the use of a common value of \(\lambda\) in both cases is slightly misleading, as using the sum intuitively decreases the comparative weight of the regularization term. To reduce this effect, for this experiment only, we divide the local losses by the average of the number of data points per user for the sum version. This way, if the number of points is equal for all users, the two losses will be exactly the same. More importantly, our experiments seem to robustly show that using the sum consistently outperforms the expectation, for both a linear classifier and a 2-layer neural network, for the problem of noisy FashionMNIST classification.

E.2.2. FashionMNIST without noise

Recall that we introduced noise into FashionMNIST to make the problem harder to learn and observe a clear difference between the average and the sum. In this section, we present results of our experiments when the noise is removed.

Even without noise, the difference between using the sum and using the expectation still seems important. We acknowledge,
Data poisoning and Byzantine gradients

Figure 5. 2-layer neural network on noisy FashionMNIST, for $\lambda = 0.01$.

Figure 6. Linear model on noisy FashionMNIST, for $\lambda = 0.1$.

Figure 7. 2-layer neural network on noisy FashionMNIST, for $\lambda = 0.1$.

Figure 8. Linear model on noisy FashionMNIST, for $\lambda = 1$. 
Data poisoning and Byzantine gradients

Figure 9. 2-layer neural network on noisy FashionMNIST, for $\lambda = 1$.

Figure 10. Linear model on noisy FashionMNIST, for $\lambda = 10$.

Figure 11. 2-layer neural network on noisy FashionMNIST, for $\lambda = 10$.

Figure 12. Linear model on noisy FashionMNIST, for $\lambda = 100$. 
Figure 13. 2-layer neural network on noisy FashionMNIST, for $\lambda = 100$.

Figure 14. Linear model on FashionMNIST (without noise), for $\lambda = 1$.

Figure 15. 2-layer neural network on FashionMNIST (without noise), for $\lambda = 1$. 
Data poisoning and Byzantine gradients

however, that the plots suggest that even though we ran this experiment for 10 times more (and 5 times more for the linear model) than other experiments, we might not have reached convergence yet, and that the use of the expectation might still eventually get closer to the case of sum. We believe that the fact that the difference between sum and expectation in the absence of noise is weak is due to the fact that the FashionMNIST dataset is sufficiently linearly separable. Thus, we achieve a near-zero loss in both cases, which make the sum and the expectation close at optimum.

Even in this case, however, we observed that the sum clearly outperforms the expectation especially, in the first epochs. We argue that the reason for this is the following. By taking the average in local losses, the weights of the data of idle users are essentially blown out of proportion. As a result, the optimizer will very quickly fit these data. However, the signal from the data of the active user will then be too weak, so that the optimizer has to first almost perfectly fit the idle nodes’ data before it can catch the signal of the active user’s data and hence the average achieves weaker convergence performances than the sum.

F. Linear Regression and Classification are Gradient PAC*

Throughout this section, we use the following terminology.

**Definition 6.** Consider a parameterized event $E(\mathcal{I})$. We say that the event $E$ occurs with high probability if $\mathbb{P}[E(\mathcal{I})] \to 1$ as $\mathcal{I} \to \infty$.

**F.1. Preliminaries**

Define $\|\Sigma\|_2 \triangleq \max_{\|x\|_2 \neq 0} (\|\Sigma x\|_2 / \|x\|_2)$ the $\ell_2$ operator norm of the matrix $\Sigma$. For symmetric matrices $\Sigma$, this is also the largest eigenvalue in absolute value.

**Theorem 6** (Covariance concentration, Theorem 6.5 in (Wainwright, 2019)). Denote $\Sigma = \mathbb{E}[Q_i Q_i^T]$, where $Q_i \in \mathbb{R}^d$ is from a $\sigma_2$-sub-Gaussian random distribution $\tilde{Q}$. Then, there are universal constants $c_1, c_2$ and $c_3$ such that, for any set $\{Q_i\}_{i \in [\mathcal{I}]}$ of i.i.d. samples from $\tilde{Q}$, and any $\delta > 0$, the sample covariance $\tilde{\Sigma} = \frac{1}{\mathcal{I}} \sum Q_i Q_i^T$ satisfies the bound

$$\mathbb{P} \left[ \frac{1}{\sigma_2^2} \|\tilde{\Sigma} - \Sigma\|_2 \geq c_1 \left( \sqrt{\frac{d}{\mathcal{I}}} + \frac{d}{\mathcal{I}} \right) + \delta \right] \leq c_2 \exp \left( -c_3 \mathcal{I} \min(\delta, \delta^2) \right).$$

(97)

**Theorem 7** (Weyl’s Theorem, Theorem 4.3.1 in (Horn & Johnson, 2012)). Let $A$ and $B$ be Hermitian and let the respective eigenvalues of $A$ and $B$ and $A + B$ be $\{\lambda_i(A)\}_{i=1}^d$, $\{\lambda_i(B)\}_{i=1}^d$, and $\{\lambda_i(A + B)\}_{i=1}^d$, each increasingly ordered. Then

$$\lambda_i(A + B) \leq \lambda_{i+j}(A) + \lambda_{d-j}(B), \quad j = 0, 1, \ldots, d - i,$$

(98)

and

$$\lambda_{i+j}(A) + \lambda_{j+1}(B) \leq \lambda_i(A + B), \quad j = 0, \ldots, i - 1,$$

(99)

for each $i = 1, \ldots, d$.

**Lemma 20.** Consider two symmetric definite positive matrices $S$ and $\Sigma$. Denote $\rho_{\min}$ and $\lambda_{\min}$ their minimal eigenvalues. Then $|\rho_{\min} - \lambda_{\min}| \leq \|S - \Sigma\|_2$.

**Proof.** This is a direct consequence of Theorem 7, for $A = S$, $B = \Sigma - S$, $i = 1$, and $j = 0$.

**Corollary 3.** There are universal constants $c_1$, $c_2$ and $c_3$ such that, for any $\sigma_2$-sub-Gaussian vector distribution $\tilde{Q} \in \mathbb{R}^d$ and any $\delta > 0$, the sample covariance $\tilde{\Sigma} = \frac{1}{\mathcal{I}} \sum Q_i Q_i^T$ satisfies the bound

$$\mathbb{P} \left[ \frac{1}{\sigma_2^2} \min \text{Sp}(\tilde{\Sigma}) - \min \text{Sp}(\Sigma) \geq c_1 \left( \sqrt{\frac{d}{\mathcal{I}}} + \frac{d}{\mathcal{I}} \right) + \delta \right] \leq c_2 \exp \left( -c_3 \mathcal{I} \min(\delta, \delta^2) \right),$$

(100)

where $\min \text{Sp}(\tilde{\Sigma})$ and $\min \text{Sp}(\Sigma)$ are the minimal eigenvalues of $\tilde{\Sigma}$ and $\Sigma$.

**Proof.** This follows from Theorem 6 and Lemma 20.

---

*For real matrices, Hermitian is the same as symmetric.
Lemma 21. With high probability, \( \min \mathbb{S}^P(\tilde{\Sigma}) \geq \min \mathbb{S}^P(\Sigma)/2 \).

Proof. Denote \( \lambda_{\min} \triangleq \min \mathbb{S}^P(\Sigma) \) and \( \tilde{\lambda}_{\min} \triangleq \min \mathbb{S}^P(\tilde{\Sigma}) \). Since each \( Q_i \) is drawn i.i.d. from a \( \sigma_Q \)-sub-Gaussian, we can apply Corollary 3. Namely, there are constants \( c_1, c_2 \) and \( c_3 \), such that for any \( \delta > 0 \), we have
\[
P \left[ \left| \tilde{\lambda}_{\min} - \lambda_{\min} \right| \geq c_1 \sigma_Q^2 \left( \sqrt{\frac{d}{T}} + \frac{d}{T} \right) + \delta \sigma_Q^2 \right] \leq c_2 \exp \left( -c_3 I \min \{ \delta, \delta^2 \} \right). \tag{101} \]
We now set \( \delta = \lambda_{\min}/(4\sigma_Q^2) \) and we consider \( I \) large enough so that \( c_1 \left( \sqrt{\frac{d}{T}} + \frac{d}{T} \right) \leq \lambda_{\min}/(4\sigma_Q^2) \). With high probability, we then have \( \tilde{\lambda}_{\min} \geq \lambda_{\min}/2 \). \hfill \Box

\section*{F.2. Linear Regression is Gradient-PAC*}

In this section, we prove the first part of Lemma 4. Namely, we prove that linear regression is gradient-PAC* learning.

\subsection*{F.2.1. Lemmas for Linear Regression}

Before moving to the main proof that linear regression is gradient-PAC*, we first prove a few useful lemmas. These lemmas will rest on the following well-known theorems.

Theorem 8 (Lemma 2.7.7 in \cite{Vershynin2018}). If \( X \) and \( Y \) are sub-Gaussian, then \( XY \) is sub-exponential.

Theorem 9 (Equation 2.18 in \cite{Wainwright2019}). If \( X_1, \ldots, X_T \) are iid sub-exponential variables, then there exist constants \( c_4, c_5 \) such that, for all \( I \), we have
\[
\forall t \in [0, c_4], \ P \left[ \|X - \mathbb{E}[X]\| \geq tI \right] \leq 2 \exp \left( -c_5 I t^2 \right). \tag{102} \]

Lemma 22. For all \( j \in [d] \), the random variables \( X_i \triangleq \xi_i Q_i[j] \) are iid, sub-exponential and have zero mean.

Proof. The fact that these variables are iid follows straightforwardly from the fact that the noises \( \xi_i \) are iid, and the queries \( Q_i \) are also iid. Moreover, both are sub-Gaussian, and by Theorem 8, the product of sub-Gaussian variables is sub-exponential. Finally, we have \( \mathbb{E}[X] = \mathbb{E}[\xi_i Q_i[j]] = \mathbb{E}[\xi] \mathbb{E}[Q_i[j]] = 0 \), using the independence of the noise and the query, and the fact that noises have zero mean (\( \mathbb{E}[\xi] = 0 \)). \hfill \Box

Lemma 23. There exists \( B \) such that \( \left\| \sum_{i \in I} \xi_i Q_i[j] \right\|_2 \leq BI^{3/4} \) with high probability.

Proof. By Lemma 22, the terms \( \xi_i Q_i[j] \) are iid, sub-exponential and have zero mean. Therefore, by Theorem 9, there exist constants \( c_4 \) and \( c_5 \) such that for any coordinate \( j \in [d] \) of \( \xi_i Q_i \) and for all \( 0 \leq u \leq c_4 \), we have
\[
P \left[ \left\| \sum_{i \in I} \xi_i Q_i[j] \right\| \geq I u \right] \leq 2 \exp \left( -c_5 I u^2 \right). \tag{103} \]
Plugging \( u = vI^{(-1/4)} \) into the inequality for some small enough constant \( v \), and using union bound then yields
\[
P \left[ \left\| \sum_{i \in I} \xi_i Q_i \right\|_2 \geq I^{(3/4)} v \sqrt{d} \right] \leq P \left[ \left\| \sum_{i \in I} \xi_i Q_i \right\|_\infty \geq I^{(3/4)} v \right] \leq 2d \exp \left( -c_5 \sqrt{I} v^2 \right). \tag{104} \]
Defining \( B = v \sqrt{d} \) yields the lemma. \hfill \Box

\subsection*{F.2.2. Proof that Linear Regression is Gradient-PAC*}

We now move on to proving that least square linear regression is gradient-PAC*.
Proof of Theorem 2. Note that $\nabla_{\theta} \ell(\theta, Q, A) = (\theta^T Q - A^T) Q$. Thus, on input $i \in [I]$, we have

$$\nabla_{\theta} \ell(\theta, Q_i, A(Q_i, \theta^1)) = ((\theta - \theta^1)^T Q_i) Q_i - \xi_i Q_i.$$  

(105)

Moreover, we have

$$(\theta - \theta^1)^T \nabla_{\theta} (\nu \|\theta\|^2_2) = 2 \nu (\theta - \theta^1)^T \theta = 2 \nu \|\theta - \theta^1\|^2_2 + 2 \nu (\theta - \theta^1)^T \theta.$$  

(106)

As a result, we have

$$(\theta - \theta^1)^T \nabla_{\theta} \mathcal{L}(\theta, D) =$$

$$\mathcal{I}(\theta - \theta^1)^T \Sigma(\theta - \theta^1) - (\theta - \theta^1)^T \left( \sum_{i \in I} \xi_i Q_i \right) + 2 \nu \|\theta - \theta^1\|^2_2 + 2 \nu (\theta - \theta^1)^T \theta.$$  

(108)

But now, with high probability, we have $(\theta - \theta^1)^T \Sigma (\theta - \theta^1) \geq (\lambda_{\text{min}}/2) \|\theta - \theta^1\|^2_2$ (Lemma 21) and $\|\sum_{i \in I} \xi_i Q_i\|_2 \leq B \mathcal{T}^{3/4}$ (Lemma 23). Using the fact that $\|\theta^1\|_2 \leq K$ and the Cauchy-Schwarz inequality, we have

$$(\theta - \theta^1)^T \nabla_{\theta} \mathcal{L}(\theta, D) \geq (\lambda_{\text{min}}/2 + \nu) \|\theta - \theta^1\|^2_2 - (B \mathcal{T}^{3/4} + 2 \nu K) \|\theta - \theta^1\|_2.$$  

(109)

Denoting $A_K \triangleq \lambda_{\text{min}}/2$ and $B_K \triangleq B + 2 \nu K$ and using the fact that $\mathcal{I} \geq 1$, we then have

$$(\theta - \theta^1)^T \nabla_{\theta} \mathcal{L}(\theta, D) \geq A_K \mathcal{I} (\|\theta - \theta^1\|^2_2 - B_K \mathcal{T}^{3/4}) \|\theta - \theta^1\|_2$$

(110)

$$\geq A_K \mathcal{I} \min \left\{ \|\theta - \theta^1\|^2_2, \|\theta - \theta^1\|_2^2 \right\} - B_K \mathcal{T}^{3/4} \|\theta - \theta^1\|_2,$$

(111)

with high probability. This corresponds to saying Assumption 3 is satisfied for $\alpha = 3/4$.

\[\square\]

F.3. Logistic Regression

In this section, we now prove the second part of Lemma 4. Namely, we prove that logistic regression is gradient-PAC learning.

F.3.1. Lemmas about the Sigmoid Function

We first prove two useful lemmas about the following logistic distance function.

Definition 7. We define the logistic distance function by $\Delta(a, b) \triangleq (a - b) (\sigma(a) - \sigma(b))$.

Lemma 24. If $a, b \in \mathbb{R}$ such that for some $k > 0$, $|a| \leq k$ and $|b| \leq k$, then there exists some constant $c_k > 0$ such that

$$\Delta(a, b) \geq c_k |a - b|^2.$$  

(112)

Proof. Note that the derivative of $\sigma(z)$ is strictly positive, symmetric $\sigma'(z) = \sigma'(-z)$) and monotonically decreasing for $z \geq 0$. Therefore, for any $z \in [-k, k]$, we know $\sigma'(z) \geq c_k \triangleq \sigma'(k)$. Thus, by the mean value theorem, we have

$$\frac{\sigma(a) - \sigma(b)}{a - b} \geq c_k.$$  

(113)

Multiplying both sides by $(a - b)^2$ then yields the lemma.  

\[\square\]

Lemma 25. If $b \in \mathbb{R}$, and $|b| \leq k$, for some $k > 0$, then there exists a constant $d_k$, such that for any $a \in \mathbb{R}$, we have

$$\Delta(a, b) \geq d_k |a - b| - d_k$$  

(114)

Proof. Assume $|a - b| \geq 1$ and define $d_k \triangleq \sigma(k + 1) - \sigma(k)$. If $b \geq 0$, since $\sigma'(z)$ is decreasing for $z \geq 0$, we have $\sigma(b) - \sigma(b - 1) \geq \sigma(b + 1) - \sigma(b) \geq d_k$, and by symmetry, a similar argument holds for $b \leq 0$. Thus, we have

$$|\sigma(a) - \sigma(b)| \geq \min \{ \sigma(b) - \sigma(b - 1), \sigma(b + 1) - \sigma(b) \} \geq d_k.$$  

(115)

Therefore,

$$(a - b) (\sigma(a) - \sigma(b)) \geq d_k |a - b| \geq d_k |a - b| - d_k.$$  

(116)

For the case of $|a - b| \leq 1$, we also have $(a - b) (\sigma(a) - \sigma(b)) \geq 0 \geq d_k |a - b| - d_k$.  

\[\square\]
F.3.2. A uniform lower bound

Definition 8. Denote \( S^{d-1} \triangleq \{ u \in \mathbb{R}^d \mid \| u \|_2 = 1 \} \) the hypersphere in \( \mathbb{R}^d \).

Lemma 26. Assume \( \text{SUPP}(\tilde{Q}) \) spans \( \mathbb{R}^d \). Then, for all \( u \in S^{d-1} \), \( \mathbb{E} \left[ \| Q^T u \| \right] > 0. \)

Proof. Let \( u \in S^{d-1} \). We know that there exists \( Q_1, \ldots, Q_d \in \text{SUPP}(\tilde{Q}) \) and \( \alpha_1, \ldots, \alpha_d \in \mathbb{R} \) such that \( u \) is colinear with \( \sum \alpha_j Q_j \). In particular, we then have \( u^T \sum \alpha_j Q_j = \sum \alpha_j (Q_j^T u) \neq 0 \). Therefore, there must be a query \( Q_\ast \in \text{SUPP}(\tilde{Q}) \) such that \( Q_\ast^T u \neq 0 \), which implies \( a \triangleq \| Q_\ast^T u \| > 0 \). By continuity of the scalar product, there must then also exist \( \varepsilon > 0 \) such that, for any \( Q \in B(Q_\ast, \varepsilon) \), we have \( \| Q^T u \| \geq a/2 \), where \( B(Q_\ast, \varepsilon) \) is an Euclidean ball centered on \( Q_\ast \) and of radius \( \varepsilon \).

But now, by definition of the support, we know that \( p \triangleq \mathbb{P} \left[ Q \in B(Q_\ast, \varepsilon) \right] > 0 \). By the law of total expectation, we then have

\[
\mathbb{E} \left[ \| Q^T u \| \right] = \mathbb{E} \left[ \| Q^T u \| \mid Q \in B(Q_\ast, \varepsilon) \right] \mathbb{P} \left[ Q \in B(Q_\ast, \varepsilon) \right] + \mathbb{E} \left[ \| Q^T u \| \mid Q \notin B(Q_\ast, \varepsilon) \right] \mathbb{P} \left[ Q \notin B(Q_\ast, \varepsilon) \right]
\]

\[
\geq ap/2 + 0 > 0,
\]

which is the lemma.

\[ \square \]

Lemma 27. Assume that, for all unit vectors \( u \in S^{d-1} \), we have \( \mathbb{E} \left[ \| Q^T u \| \right] > 0 \), and that \( \text{SUPP}(\tilde{Q}) \) is bounded by \( M_\mathcal{Q} \). Then there exists \( C > 0 \) such that, with high probability,

\[
\forall u \in S^{d-1}, \sum_{i \in \mathcal{I}} |Q_i^T u| \geq CT.
\]

Proof. By continuity of the scalar product and the expectation operator, and by compactness of \( S^{d-1} \), we know that

\[
C_0 \triangleq \inf_{u \in \mathbb{R}^d} \mathbb{E} \left[ \| Q^T u \| \right] > 0.
\]

Now define \( \varepsilon \triangleq C_0/4M_\mathcal{Q} \). Note that \( S^{d-1} \subset \bigcup_{u \in S^{d-1}} B(u, \varepsilon) \). Thus we have a covering of the hypersphere by open sets. But since \( S^{d-1} \) is compact, we know that we can extract a finite covering. In other words, there exists a finite subset \( S \subset S^{d-1} \) such that \( S^{d-1} \subset \bigcup_{u \in S} B(u, \varepsilon) \). Put differently, for any \( v \in S^{d-1} \), there exists \( u \in S \) such that \( \| u - v \|_2 \leq \varepsilon \).

Now consider \( u \in S \). Given that \( \text{SUPP}(\tilde{Q}) \) is bounded, we know that \( |Q_i^T u| \in [0, M_\mathcal{Q}] \). Moreover, such variables \( |Q_i^T u| \) are iid. By Hoeffding’s inequality, for any \( t > 0 \), we have

\[
\mathbb{P} \left[ \left| \sum_{i \in \mathcal{I}} |Q_i^T u| - \mathbb{I} \mathbb{E} \left[ |Q_i^T u| \right] \right| \geq \mathbb{I}t \right] \leq 2 \exp \left( \frac{-2t^2}{M_\mathcal{Q}} \right).
\]

Choosing \( t = C_0/2 \) then yields

\[
\mathbb{P} \left[ \sum_{i \in \mathcal{I}} |Q_i^T u| \leq \frac{C_0 \mathcal{I}}{2} \right] \leq \mathbb{P} \left[ \sum_{i \in \mathcal{I}} |Q_i^T u - \mathbb{I} \mathbb{E} \left[ |Q_i^T u| \right] \geq \frac{\mathbb{I}C_0}{2} \right] \]

\[
\leq 2 \exp \left( \frac{-\mathbb{I}C_0^2}{2M_\mathcal{Q}} \right).
\]

Taking a union bound for \( u \in S \) then guarantees

\[
\mathbb{P} \left[ \forall u \in S, \sum_{i \in \mathcal{I}} |Q_i^T u| \geq \frac{C_0 \mathcal{I}}{2} \right] \geq 1 - 2 |S| \exp \left( \frac{-\mathbb{I}C_0^2}{2M_\mathcal{Q}} \right),
\]

which clearly goes to 1 as \( \mathcal{I} \rightarrow \infty \). Thus \( \forall u \in S, \sum_{i \in \mathcal{I}} |Q_i^T u| \geq \frac{C_0 \mathcal{I}}{2} \) holds with high probability.
Now consider $v \in S^{d-1}$. We know that there exists $u \in S$ such that $\|u - v\|_2 \leq \varepsilon$. Then, we have

$$
\sum_{i \in [2]} |Q_i^T v| = \sum_{i \in [2]} |Q_i^T u + Q_i^T (v - u)| \geq \sum_{i \in [2]} |Q_i^T u| - 2M_Q \|v - u\|_2 \geq \sum_{i \in [2]} \frac{C_0 I \theta_i}{2} - 2M_Q \frac{C_0 I}{4} = \frac{C_0 I}{4},
$$

which proves the lemma.

\[\square\]

F.3.3. LOWER BOUND ON THE DISCREPANCY BETWEEN PREFERRED AND REPORTED ANSWERS

**Lemma 28.** Assume that $\tilde{Q}$ has a bounded support, whose interior contains the origin. Suppose also that $\|\theta\|_2 \leq K$. Then there exists $A_K$ such that, with high probability, we have

$$
\sum_{i \in [2]} \Delta(Q_i^T \theta, Q_i^T \theta^t) \geq A_K \min \left\{ \|\theta - \theta^t\|_2, \|\theta - \theta^t\|_2^2 \right\}.
$$

**Proof.** Note that by Cauchy-Schwarz inequality we have

$$
|Q_i^T \theta^t| \leq \|Q_i\|_2 \|\theta^t\|_2 \leq M_Q K.
$$

Thus, Lemma 25 implies the existence of a positive constant $d_K$, such that for all $\theta \in \mathbb{R}^d$, we have

$$
\sum_{i \in [2]} \Delta(Q_i^T \theta, Q_i^T \theta^t) \geq \sum_{i \in [2]} \left( d_K |Q_i^T \theta^t - Q_i^T \theta^t| - d_K \right).
$$

$$
= -d_K I + d_K \|\theta - \theta^t\|_2 \sum_{i \in [2]} |Q_i^T \theta^t - \theta^t|,
$$

where $\theta_{\theta - \theta^t} \triangleq (\theta - \theta^t)/\|\theta - \theta^t\|_2$ is the unit vector in the direction of $\theta - \theta^t$.

Now, by Lemma 27, we know that, with high probability, for all unit vectors $u \in S^{d-1}$, we have $\sum_i |Q_i^T u| \geq CI$. Thus, for $I$ sufficiently large, for any $\theta \in \mathbb{R}^d$, with high probability, we have

$$
\sum_{i \in [2]} \Delta(Q_i^T \theta, Q_i^T \theta^t) \geq \frac{d_K C_{\min} I}{2} \|\theta - \theta^t\|_2 - d_K I.
$$

Defining $e_K \triangleq \frac{d_K C_{\min}}{4}$, and $f_K \triangleq \frac{4}{C_{\min}}$, for $\|\theta - \theta^t\|_2 > f_K$, we then have

$$
\sum_{i \in [2]} \Delta(Q_i^T \theta, Q_i^T \theta^t) \geq e_K I \|\theta - \theta^t\|_2.
$$

We now focus on the case of $\|\theta - \theta^t\|_2 \leq f_K$. The triangle inequality yields $\|\theta\|_2 \leq \|\theta - \theta^t\|_2 + \|\theta^t\|_2 \leq f_K + K$. By Cauchy-Schwarz inequality, we then have $|Q_i^T \theta| \leq (K + f_K) M_Q \triangleq g_K$ and $|Q_i^T \theta^t| \leq K M_Q \leq g_K$. Thus, by Lemma 24, we know there exists some constant $c_K$ such that

$$
\sum_{i \in [2]} (\sigma(Q_i^T \theta) - \sigma(Q_i^T \theta^t)) (Q_i^T \theta - Q_i^T \theta^t) \geq \sum_{i \in [2]} c_K |Q_i^T \theta - Q_i^T \theta^t|^2
$$

$$
= \sum_{i \in [2]} c_K (\theta - \theta^t)^T Q_i Q_i^T (\theta - \theta^t)
$$

$$
= c_K (\theta - \theta^t)^T \left( \sum_{i \in [2]} Q_i Q_i^T \right) (\theta - \theta^t).
$$
Proof of Theorem 3.

Note that

Now we proceed with the proof that logistic regression is gradient-PAC*.

Applying now Cauchy-Schwarz inequality, with high probability, we have

where \( \lambda_{\text{min}} \) is the smallest eigenvalue of \( \mathbb{E} \left[ Q_{i} Q_{i}^{T} \right] \). Thus, for \( \| \theta - \theta^{*} \|_{2} \leq f_{K} \), we have

Combining this with (133), and defining \( A_{K} \equiv \min \left\{ \frac{\lambda_{\text{min}} c_{K}}{2}, e_{K} \right\} \), we then obtain the lemma.

\[ \Box \]

F.3.4. Proof that Logistic Regression is Gradient-PAC*

Now we proceed with the proof that logistic regression is gradient-PAC*.

Proof of Theorem 3. Note that \( \sigma(-z) = e^{-z} \sigma(z) = 1 - \sigma(z) \) and \( \sigma'(z) = -e^{-z} \sigma^{2}(z) \). We then have

where \( \mathbb{I} \{ A = 1 \} \) is the indicator function that outputs 1 if \( A = 1 \), and 0 otherwise. As a result,

By Lemma 28, with high probability, we have

To control the second term of (145), note that the random vectors \( Z_{i} \equiv (\sigma(Q_{i}^{T} \theta)) - \mathbb{I} \{ A_{i} = 1 \} Q_{i} \) are iid with norm at most \( M_{Q} \). Moreover, since \( \mathbb{E} \mathbb{I} \{ A_{i} = 1 \} Q_{i} = \mathbb{E} \sigma(Q_{i}^{T} \theta) \), by the tower rule, we have \( \mathbb{E} Z_{i} = \mathbb{E} \mathbb{E} Z_{i} | Q_{i} \] = 0. Therefore, by applying Hoeffding’s bound to every coordinate of \( Z_{i} \), and then taking a union bound, for any \( B > 0 \), we have

Applying now Cauchy-Schwarz inequality, with high probability, we have

\[ \left\| \mathbb{I} \{ A = 1 \} \right\|_{2} \right) \leq 2d \exp \left( \frac{B^{2} \sqrt{T}}{2d M_{Q}^{2}} \right). \] (148)

Applying now Cauchy-Schwarz inequality, with high probability, we have

\[ \left\| \mathbb{I} \{ A = 1 \} \right\|_{2} \right) \leq B T^{3/4} \| \theta - \theta^{*} \|_{2}. \]
Combining this with (138) and using $\|\theta^g\|^2 \leq \mathcal{K}$, we then have
\[
(\theta - \theta^g)^T \nabla_{\theta} L(\theta, \mathcal{D}) \geq (A_K I + \nu) \left\{ \|\theta - \theta^g\|_2, \|\theta - \theta^g\|_2^2 \right\} - (B I^{(3/4)} + 2\nu K) \|\theta - \theta^g\|_2
\]
(149)
\[
\geq A_K I \left\{ \|\theta - \theta^g\|_2, \|\theta - \theta^g\|_2^2 \right\} - B_K I^{(3/4)} \|\theta - \theta^g\|_2,
\]
(150)
\[
\geq A_K I \left( \max_{\theta^g} \right) \left\{ \|\theta - \theta^g\|_2, \|\theta - \theta^g\|_2^2 \right\} - B_K I^{(3/4)} \|\theta - \theta^g\|_2,
\]
(151)
where $B_K = B + 2\nu K$. This shows that Assumption 3 is satisfied for logistic loss for $\alpha = 3/4$, and $A_K$ and $B_K$ as previously defined.

\[\square\]

G. Proofs of Local PAC*-Learnability

Let us now prove Lemma 5. To do so, consider the preferred models $\hat{\theta}^g$ and a subset $\mathcal{H} \subset [N]$ of honest users. Denote $\tilde{\mathcal{D}}_{-\mathcal{H}}$ the datasets provided by users $n \in [N] - \mathcal{H}$. Each honest user $h \in \mathcal{H}$ provides an honest dataset $\mathcal{D}_h$ of cardinality at least $I \geq 1$. Consider the bound $K_{\mathcal{H}} \triangleq \max_{h \in \mathcal{H}} \|\theta^h\|_2$ on the parameter norm of honest active users $h \in \mathcal{H}$.

G.1. Bounds on the Optima

Before proving the theorem, we prove a useful lemma that bounds the set of possible values for the global model and honest local models.

**Lemma 29.** Assume that $\mathcal{R}$ and $\ell$ are nonnegative. For $\mathcal{I}$ large enough, if all honest active users $h \in \mathcal{H}$ provide at least $\mathcal{I}$ data, then, with high probability, $\theta^*_h$ must lie in a compact subset of $\mathbb{R}^{d \times \mathcal{H}}$ that does not depend on $\mathcal{I}$.

**Proof.** Denote $L^0 \triangleq \text{LOSS}(0, (\hat{\theta}^g, 0_{-\mathcal{H}}), (0, \tilde{\mathcal{D}}_{-\mathcal{H}}))$. Essentially, we will show that, if $\theta^*_h$ is too far from $\hat{\theta}^g$, then the loss will take values strictly larger than $L^0$.

Assumption 3 implies the existence of an event $\mathcal{E}$ that occurs with probability at least $P_0 \triangleq P(\mathcal{K}_\mathcal{H}, \mathcal{I})^{\mathcal{H}}$, under which, for any $\theta_h \in \mathbb{R}^d$, we have
\[
(\theta_h - \theta^g)\nabla_{\theta} L_h(\theta_h) \geq A_K I \min \left\{ \|\theta_h - \theta^g\|_2, \|\theta_h - \theta^g\|_2^2 \right\} - B_K I^\alpha \|\theta_h - \theta^g\|_2,
\]
(152)
which implies
\[
u_{\theta_h - \theta^g}\nabla_{\theta} L_h(\theta_h) \geq A_K I \min \left\{ 1, \|\theta_h - \theta^g\|_2 \right\} - B_K I^\alpha.
\]
(153)
Note also that $P_0 \to 1$ as $\mathcal{I} \to \infty$. We now integrate both sides over the line segment from $\theta^g_h$ to $\theta_h$. The fundamental theorem of calculus for line integrals then yields
\[
\mathcal{L}_h(\theta_h) - \mathcal{L}_h(\theta^g_h) = \|\theta_h - \theta^g_h\|_2 \int_{t=0}^{1} \nu_{\theta_h - \theta^g_h} \nabla_{\theta} L_h(\theta^g_h + t(\theta^g_h - \theta_h^g)) dt
\]
(154)
\[
\geq \|\theta_h - \theta^g_h\|_2 \int_{t=0}^{1} \left( A_K I \min \left\{ 1, t \|\theta_h - \theta^g_h\|_2 \right\} - B_K I^\alpha \|\theta_h - \theta^g_h\|_2 \right) dt
\]
(155)
\[
= \|\theta_h - \theta^g_h\|_2 \int_{t=0}^{1} \left( A_K I \min \left\{ 1, t \|\theta_h - \theta^g_h\|_2 \right\} - B_K I^\alpha \|\theta_h - \theta^g_h\|_2 \right) dt
\]
(156)
Now, if $\|\theta_h - \theta^g_h\|_2 > 2$, we then have
\[
\mathcal{L}_h(\theta_h) - \mathcal{L}_h(\theta^g_h) \geq \left( \frac{A_K I}{2} - B_K I^\alpha \right) \|\theta_h - \theta^g_h\|_2
\]
(157)
\[
\geq A_K I - 2B_K I^\alpha.
\]
(158)
Now for $I > I_1 \triangleq \max \left\{ 2L^0 / A_{K_h}, (4B_{K_h} / A_{K_h})^{\frac{1}{2}} \right\}$, we have

$$\mathcal{L}_h(\theta_h) - \mathcal{L}_h(\theta_h^*) > L^0. \tag{159}$$

This implies that if $\|\theta_h - \theta_h^*\|_2 > 2$ for any $h \in \mathcal{H}$, then we have

$$\text{LOSS}(0, (\theta_h^*, 0_{-h}), \mathcal{D}) < \text{LOSS}(\rho, (\bar{\theta}_h, \bar{\theta}_{-h}), \mathcal{D}), \tag{160}$$

regardless of $\rho$ and $\theta_{-h}$. Therefore, we must have $\|\theta_h^* - \theta_h^\dagger\|_2 \leq 2$. Such inequalities describe a bounded closed subset of $\mathbb{R}^{d \times H}$, which is thus compact.

**Lemma 30.** Assume that $\mathcal{R}(\rho, \theta) \to \infty$ as $\|\rho - \theta\|_2 \to \infty$, and that $\|\theta_h^\dagger - \theta_h^*\|_2 \leq 2$ for all honest users $h \in \mathcal{H}$. Then $\rho^*$ must lie in a compact subset of $\mathbb{R}^d$ that does not depend on $I$.

**Proof.** Consider an honest user $h'$. Given our assumption on $\mathcal{R} \to \infty$, we know that there exists $D_{K_h}$ such that if $\|\rho - \theta_{h'}\|_2 \geq D_{K_h}$, then $\mathcal{R}(\rho, \theta_{h'}) \geq L^0 + 1$. Thus any global optimum $\rho^*$ must satisfy $\|\rho^* - \theta_{h'}\|_2 \leq \|\rho^* - \theta_{h'}\|_2 + \|\theta_{h'} - \theta_{h'}^\dagger\|_2 \leq D_{K_h} + 2$.

**G.2. Proof of Lemma 5**

**Proof of Lemma 5.** Fix $\varepsilon, \delta > 0$. We want to show the existence of some value of $I(\varepsilon, \delta, \bar{\mathcal{D}}_{-h}, \mathcal{D})$ that will guarantee $(\varepsilon, \delta)$-locally PAC* learning for honest users.

By lemmas 29 and 30, we know that the set $C$ of possible values for $(\rho^*, \bar{\theta}_{h'})$ is compact. Now, we define

$$E_{K_h} \triangleq \max_{(\rho, \theta) \in C} \|\nabla \mathcal{R}(\rho, \theta)\|_2 \tag{161}$$

the maximum of the norm of achievable gradients at the optimum. We know this maximum exists since $C$ is compact.

Using the optimality of $(\rho^*, \bar{\theta}_{h'})$, for all $h \in \mathcal{H}$, we have

$$0 \in (\theta_h^* - \theta_h^\dagger)^T \nabla_{\theta_h} \text{LOSS}(\rho^*, \bar{\theta}_{h'}) \tag{162}$$

$$= (\theta_h^* - \theta_h^\dagger)^T \nabla \mathcal{L}_h(\theta_h) + (\theta_h^* - \theta_h^\dagger)^T \nabla_{\theta_h} \mathcal{R}(\rho^*, \theta_h^*) \tag{163}$$

$$\geq (\theta_h^* - \theta_h^\dagger)^T \nabla \mathcal{L}_h(\theta_h) - \|\theta_h^\dagger - \theta_h^*\|_2 \|\nabla \mathcal{R}(\rho^*, \theta_h^*)\|_2 \tag{164}$$

$$\geq (\theta_h^* - \theta_h^\dagger)^T \nabla \mathcal{L}_h(\theta_h) - E_{K_h} \|\theta_h^* - \theta_h^\dagger\|_2. \tag{165}$$

We now apply assumption 3 for $\theta = \theta_h^*$ (for $h \in \mathcal{H}$). Thus, there exists some other event $\mathcal{E}'$ with probability at least $P_0$, under which, for all $h \in \mathcal{H}$, we have

$$0 \geq A_{K_h} I \min \left\{ \|\theta_h^* - \theta_h^\dagger\|_2, \|\theta_h^\dagger - \theta_h^\dagger\|_2, \|\theta_h^\dagger - \theta_h^\dagger\|_2 \right\} - B_{K_h} T^\alpha \|\theta_h^* - \theta_h^\dagger\|_2 - E_{K_h} \|\theta_h^* - \theta_h^\dagger\|_2. \tag{166}$$

Now if $I > I_2 \triangleq \max \left\{ 2E_{K_h} / A_{K_h}, (2B_{K_h} / A_{K_h})^{\frac{1}{2}} \right\}$ this inequality cannot hold for $\|\theta_h^* - \theta_h^\dagger\|_2 \geq 1$. Therefore, for $I > I_2$, we have $\|\theta_h^* - \theta_h^\dagger\|_2 < 1$, and thus,

$$0 \geq A_{K_h} I \|\theta_h^* - \theta_h^\dagger\|_2 - B_{K_h} T^\alpha \|\theta_h^* - \theta_h^\dagger\|_2 - E_{K_h} \|\theta_h^* - \theta_h^\dagger\|_2 \tag{167}$$

and thus,

$$\|\theta_h^* - \theta_h^\dagger\|_2 \leq \frac{B_{K_h} T^\alpha + E_{K_h}}{A_{K_h} I}. \tag{168}$$

Now note that $P[\mathcal{E} \land \mathcal{E}'] = 1 - P[\neg \mathcal{E} \lor \neg \mathcal{E}'] \geq 1 - P[\neg \mathcal{E}] - P[\neg \mathcal{E}'] = 2P_0 - 1$. It now suffices to consider $I$ larger than $I_2$ and large enough so that $P(K_h, I)^{|N|} \geq 1 - \delta/2$ (whose existence is guaranteed by Assumption 3, and which guarantees $2P_0 - 1 \geq 1 - \delta$) and so that $\frac{B_{K_h} T^\alpha + E_{K_h}}{A_{K_h} I} \leq \varepsilon$ to obtain the theorem. \qed
H. Convergence of CGA Against $\ell_2^2$

To write our proof, we define $\text{LOSS}_s^\rho : \mathbb{R}^d \to \mathbb{R}$ by

$$
\text{LOSS}_s^\rho (\rho) \triangleq \inf_{\theta} \left\{ \text{LOSS}(\rho, \theta, \mathcal{D}) - \mathcal{L}_s(\theta, \mathcal{D}_s) - \mathcal{R}(\rho, \theta, s) \right\}
$$

(169)

$$
= \inf_{\theta} \sum_{n \neq s} \mathcal{L}(\theta_n, \mathcal{D}_n) + \lambda \sum_{n \neq s} \| \rho - \theta_n \|^2.
$$

(170)

In other words, it is the loss when local models are optimized, and when the data of strategic user $s$ are removed.

Lemma 31. Assuming $\ell_2^2$ regularization and convex loss-per-input functions $\ell$, for any datasets $\mathcal{D}$, $\text{LOSS}$ is strongly convex. As a result, so is $\text{LOSS}_s^\rho$.

Proof. Note that the global loss can be written as a sum of convex function, and of $\nu \sum \| \theta_n \|^2 + \| \rho - \theta_1 \|^2$. Using tricks similar to the proof of Lemma 11, we see that the loss is strongly convex. The latter part of the lemma is then a straightforward application of Lemma 10.

We now move on to the proof of Theorem 4. Note that our statement of the theorem was not fully explicit, especially about the upper bound on the constant learning rate $\eta$. Here, we prove that it holds for $\eta_t = \eta \leq 1/3L$, where $L$ is a constant such that $\text{LOSS}_s^\rho$ is $L$-smooth. The existence of $L$ is guaranteed by Lemma 13.

Proof of Theorem 4. Note that by Lemma 9, $\text{LOSS}_s^\rho$ is convex, differentiable and $L$-smooth, and $\nabla \text{LOSS}_s^\rho(\rho^t) = g_{-s}^{t,t}$. For $\ell_2^2$ regularization, we have $\text{GRAD}(\rho) \subset \mathbb{R}^d$ for all $\rho \in \mathbb{R}^d$. Then the minimum of equation 7 is zero, which is obtained when $g_s^t \triangleq \frac{\rho^t - \theta_1}{\eta} - \hat{g}_{-s}^t = g_{s}^{t-1} + \frac{\rho^t - \theta_1}{\eta} + \frac{\rho^t - \rho^{t-1}}{\eta}$. Note that

$$
\rho^{t+1} = \rho^t - \eta g_{-s}^{t,t} - \eta g_s^t
$$

(171)

$$
= \rho^t - \eta (g_{-s}^{t,t} - (\rho^t - \theta_1^t)) + (\rho^t - \rho^t) - \eta g_s^{t-1}
$$

(172)

$$
= \theta_1^t - \eta (g_{-s}^{t,t} + g_{s}^{t-1}) + \eta (g_{-s}^{t,t-1} + g_{s}^{t-1})
$$

(173)

$$
= \theta_1^t - \eta (g_{-s}^{t,t} - g_{s}^{t,t-1}).
$$

(174)

Therefore, $\rho^{t+1} - \rho^t = \eta (g_{-s}^{t,t} - g_{s}^{t,t-1}) - \eta (g_{-s}^{t,t-1} - g_{s}^{t,t-2})$.

Then, using the $L$-smoothness of $\text{LOSS}_s^\rho$, and denoting $u_t \triangleq \| \rho^{t+1} - \rho^t \|^2$, we have $u_{t+1} \leq L \eta u_t + L \eta u_{t-1}$. Now assume that $\eta \leq 1/3L$. Then $u_{t+1} \leq \frac{1}{3}(u_t + u_{t-1})$. We then know that $u_{t+2} \leq \frac{1}{3}(u_{t+1} + u_t) \leq \frac{1}{3}(\frac{1}{3}(u_t + u_{t-1}) + u_t) = \frac{4}{9}u_t + \frac{1}{9}u_{t-1}$.

Now define $v_t \triangleq u_t + u_{t-1}$. We then have $v_{t+2} \leq u_{t+2} + u_{t+1} \leq \frac{7}{9}u_t + \frac{8}{9}u_{t-1} \leq \frac{7}{9}(u_t + u_{t-1}) \leq \frac{7}{9}v_t$. By induction, we know that $v_t \leq (7/9)^{(t-1)/2} \max \{v_0, v_1\} \leq (\sqrt{7}/3)^t (\sqrt{7}/3) \max \{v_0, v_1\}$. Thus, defining $\alpha \triangleq \sqrt{7}/3 < 1$, there exists $C > 0$ such that $u_t \leq v_t \leq C\alpha^t$. This implies that $\sum \| \rho^{t+1} - \rho^t \|^2 \leq \sum C\alpha^t < \infty$. Thus $\sum (\rho^{t+1} - \rho^t)$ converges, which implies the convergence of $\rho^t$ to a limit $\rho^\infty$. By $L$-smoothness, we know that $g_{s}^{t,t}$ must converge too. Taking equation 174 to the limit then implies $\rho^\infty = \theta_1^t$. This shows that the strategic user achieves precisely what they want with CGA. It is thus optimal.

I. CGA on MNIST

In this section, CGA is executed against 10 honest users, each one having 6,000 randomly and data points of MNIST, drawn randomly and independently. CGA is run by a strategic user whose target model $\theta_1^t$ labels 0’s as 1’s, 1’s as 2’s, and so on, until 9’s as 0’s. We learn $\theta_1^t$ by relabeling the MNIST training dataset and learning from the relabeled data. We use $\lambda = 1$, Adam optimizer and a decreasing learning rate.
Data poisoning and Byzantine gradients

Figure 16. Norm of global model, distance to initialisation and distance to target, under attack by CGA. In particular, we see that the attack against $\ell_2^2$ is successful, as the distance between the global model and the target model goes to zero.

J. Cifar-10 on VGG 13-BN Experiments

We considered VGG 13-BN, which was pretrained on cifar-10 by (Phan, 2021). We now assume that 10 users are given part of the cifar-10 database, while a strategic user also joins to the personalized federated gradient descent algorithm. The strategic user’s goal is to bias the global model towards a target model, which misclassifies the cifar-10 data, by reclassifying 0 into 1, 1 into 2... and 9 into 0.

J.1. Counter-Gradient Attack

We first show the result of performing counter-gradient attack on the last layer of the neural network. Essentially, images are now reduced to their vector embedding, and the last layer performs a simple linear classification akin to the case of MNIST (see Appendix I).

Figure 17. CGA on cifar-10.

J.2. Reconstructing a Model Attack

Reconstructing an attack model whose effect is equivalent to the counter-gradient attack is identical to what was done in the case of MNIST (see Section 5.2).

J.3. Reconstructing Data Poisoning

This last step is however nontrivial. On one hand, we could simply use the attack model to label a large number of random images. However, this solution would likely require a large sample complexity. For a more efficient data poisoning, we can construct vector embeddings on the indifference affine subspace $V$, as was done for MNIST in Section 5.3. This is what is shown below.
We acknowledge however that this does not quite correspond to data poisoning, as it requires reporting a vector embedding and its label, rather than an actual image and its label. The challenge is then to reconstruct an image that has a given vector embedding. We note that, while this is not a straightforward task in general, this has been shown to be at least somewhat possible for some neural networks, especially when they are designed to be interpretable (Zeiler & Fergus, 2014; Wang et al., 2019c; Mai et al., 2019).

K. Single Data Poisoning for Least Square Linear Regression

Proof of Theorem 5. We define the minimized loss with respect to $\rho$ and without strategic user $s$ by

$$\text{Loss}^*_{\rho, s}(\rho, \vec{D}_{-s}) \triangleq \min_{\vec{\theta}_L \in \mathbb{R}^{d \times (N-1)}} \left\{ \sum_{n \neq s} L_n(\theta_n, D_n) + \sum_{n \neq s} \lambda \|\theta_n - \rho\|_2^2 \right\}.$$

(175)

Now consider a subgradient $g \in \nabla_\rho \text{Loss}^*_{\rho, s}(\vec{\theta}^+_s, \vec{D}_{-s})$ of the minimized loss at $\vec{\theta}^+_s$. For $x \triangleq \frac{-g}{\lambda}$, then have $-g \in \nabla \left( \lambda \|x\|_2^2 \right)$. We then define $\vec{\theta}^\bullet_s \triangleq \vec{\theta}^+_s - x$.

$$0 = g - g \in \nabla_\rho \text{Loss}^*_{\rho, s}(\vec{\theta}^+_s, \vec{D}_{-s}) + \nabla_\rho \left( \lambda \|\vec{\theta}^\bullet_s - \vec{\theta}^+_s\|_2^2 \right).$$

(176)

$$= \nabla_\rho \text{Loss}^*_{\rho, s}(\vec{\theta}^+_s, \vec{\theta}^\bullet_s, \vec{D}_{-s}), \vec{\theta}^\bullet_s, \vec{D}_{-s}),$$

(177)

where $\text{Loss}^*_{\rho, s}$ is defined by (39). Now consider the data point $(Q, A) = (g, g^T \theta^\bullet_s - 1)$. For $D_s = \{(Q, A)\}$, we then have $\nabla L_s(\vec{\theta}^\bullet_s, D_s) = g$, which implies

$$\nabla_{\vec{\theta}_s} \text{Loss}^*_{\rho, s}(\vec{\theta}^+_s, \vec{\theta}^\bullet_s, \vec{D}_s) = 0.$$

(178)
Combining it all together with the uniqueness of the solution then yields
\[
\arg\min_{(\rho, \theta)} \left\{ \text{LOSS}(\rho, \theta, \mathbf{D}) \right\} = \left( \theta^{\star}_s, (\mathbf{a}^{\star}_s, \mathbf{b}^{\star}_s, (\mathbf{a}^{\star}_s, \mathbf{b}^{\star}_s)) \right),
\] (179)
which is what we wanted.

\section{Data Poisoning Against Linear Classification}

\subsection{Generating Efficient Poisoning Data and Initialization}

For every label \(a \in \{1, \ldots, 9\}\), we define \(y_a = \theta^{\star}_a - \theta^{\star}_0\), and \(c_a = -(\theta^{\star}_a - \theta^{\star}_0)\) (where \(\theta^{\star}_0\) is the bias of the linear classifier). The indifference subspace \(V\) is then the set of images \(Q \in \mathbb{R}^d\) such that \(Q^T y_a = c_a\) for all \(a \in \{1, \ldots, 9\}\).

To project any image \(X \in \mathbb{R}^d\) on \(V\), let us first construct an orthogonal basis of the vector space orthogonal to \(V\), using the Gram-Schmidt algorithm. Namely, we first define \(z_1 \triangleq y_1\). Then, for any answer \(a \in \{1, \ldots, 9\}\), we define
\[
z_a \triangleq y_a - \sum_{b < a} \frac{y_a^T z_b}{\|z_b\|^2} z_b.
\] (180)

It is easy to check that for \(b < a\), we have \(z_a^T z_b = 0\). Moreover, if \(Q \in V\), then
\[
z_a^T Q = y_a^T Q - \sum_{b < a} \frac{y_a^T z_b}{\|z_b\|^2} (z_a^T Q) = c_a - \sum_{b < a} \frac{y_a^T z_b}{\|z_b\|^2} (z_a^T Q).
\] (181)

By induction, we see that \(z_a^T Q\) is a constant independent from \(Q\). Indeed, for \(a = 1\), this is clear as \(z_1^T Q = y_1^T Q = c_1\). Moreover, for \(a > 1\), then, in the computation of \(z_a^T Q\), \(Q\) always appear as \(z_b^T Q\) for \(b < a\). Moreover, denoting \(c'_a\) the constant such that \(z_a^T Q = c'_a\) for all \(a \in \{1, \ldots, 9\}\), we see that these constants can be computed by
\[
c'_a = c_a - \sum_{b < a} \frac{y_a^T z_b}{\|z_b\|^2} c'_b.
\] (182)

Finally, we can simply perform repeated projection onto the hyperplanes where \(a\) is equally probable as the answer 0. To do this, we first define the orthogonal projection \(P(X, y, c)\) of \(X \in \mathbb{R}^d\) on the hyperplane \(x^T y = c\), which is given by
\[
P(X, y, c) = X - \frac{(X^T y - c) y}{\|y\|^2}.
\] (183)

It is straightforward to verify that \(P(X, y, c)^T y = c\) and that \(P(P(X, y, c), y, c) = P(X, y, c)\). We then canonically define repeated projection by induction, as
\[
P(X, (y_1, \ldots, y_{k+1}), (c_1, \ldots, c_{k+1})) \triangleq P(P(X, (y_1, \ldots, y_k), (c_1, \ldots, c_k)), y_{k+1}, c_{k+1}).
\] (184)

Now consider any image \(X \in \mathbb{R}^d\). Its projection can be obtained by setting
\[
Q \triangleq P(X, (z_1, \ldots, z_9), (c'_1, \ldots, c'_9)) + \xi.
\] (185)

Note that to avoid being exactly on the boundary, and thus retrieve information about the scales of \(\theta^{\star}\) and on which side of the boundary favors which label, we add a small noise \(\xi\), to make sure \(Q\) does not lie exactly on \(V\) (which would lead to multiple solutions for the learning), but small enough so that the probabilities of the different label remain close to 0.1 (the equiprobable probability).

We acknowledge that images obtained this way may not be in \([0, 1]^d\), like the images of the MNIST dataset. In general, one could search for points \(Q \in V \cap [0, 1]^d\). Note that in theory, by Theorem 3 (or a generalization of it), labeling random images in \([0, 1]^d\) should suffice. However, in the case where \(V \cap [0, 1]^d\) is empty, this procedure may require the labeling of significantly more images to be successful. This is discussed in more detail in Section L.3.

The convergence to the optimum is slow. But given that the problem is strictly convex, we focus here mostly on showing that the minimum is indeed a poisoned model. To boost the convergence, we initialize our learning algorithm at a point close to what we expect to be the minimum, by taking this minimum and adding a Gaussian noise, and then we observe the convergence to this minimum.
Accuracy of $\rho^t$ according to $\theta^t_\alpha$ (which relabels $0 \rightarrow 1 \rightarrow 2 \rightarrow \ldots \rightarrow 9 \rightarrow 0$), under our data poisoning attack with poisoned images in $[0, 1]^d$, with one attacker against two honest users.

L.2. A Brief Theory of Data Poisoning for Linear Classification

Using the efficient poisoning data fabrication, we thus have a set of images $(Q, p(Q))$, where $p_a(Q)$ is the probability assigned to image $Q$ and label $a$. This defines the following local loss for the strategic user:

$$L_s(\theta_s, D_s) = \sum_{(Q, p(Q)) \in D_s} \sum_{a \in \{0, 1, \ldots, 9\}} p_a(Q) \ln \sigma_a(\theta_s, Q), \quad (186)$$

where $\sigma_a(\theta_s, Q) = \frac{\exp(\theta_s^T Q + \theta_a)}{\sum_{b} \exp(\theta_s^T Q + \theta_b)}$ is the probability that image $Q$ has label $a$, according to the model $\theta_s$. We acknowledge that such labelings of queries is unusual. Evidently, in practice, an image may be labeled $N$ times, and the number of labels $N_a$ it received can be set to be approximately $N_s \approx N p_a(Q)$.

It is noteworthy that the gradient of the loss function is then given by

$$(\theta_s - \theta^\bullet)^T \nabla_{\theta_s} L_s(\theta_s, D_s) = \sum_{Q \in D_s} \sum_{a \in \{0, 1, \ldots, 9\}} (\sigma_a(\theta_s, Q) - \sigma_a(\theta^\bullet, Q)) (\theta_{sa} - \theta^\bullet_{sa})^T Q^+, \quad (187)$$

where we defined $Q^+ = (1, Q)$ (which allows to factor in the bias of the model. This shows that $\nabla_{\theta_s} L_s(\theta_s, D_s)$ points systematically away from $\theta^\bullet$, and thus that gradient descent will move towards $\theta^\bullet$. In fact, if the set of images $Q$ cover all dimensions (which occurs if there are $\Omega(d)$ images, which is the case for 2,000 images, since $d = 784$), then gradient descent will always move the model in the direction of $\theta^\bullet$, which will be the minimum. Moreover, by overweighting each data $(Q, p(Q))$ by a factor $\alpha$ (as though the image $Q$ was labeled $\alpha$ times), we can guarantee gradient-PAC* learning, which means that we will have $\theta^*_s \approx \theta^\bullet_s$, even in the personalized federated learning framework. This shows why data poisoning should work in theory, with relatively few data injections.

Note that the number of other users does make learning harder. Indeed, the gradient of the regularization $R(\rho, \theta_s)$ at $\rho = \theta_s^t$ and $\theta_s = \theta^\bullet_s$ is equal to $2\lambda \| \theta^t_s - \theta^\bullet_s \|^2_2$. As the number $N - 1$ of other users grows, we should expect this distance to grow roughly proportionally to $N$. In order to make strategic user $s$ robustly learn $\theta^\bullet_s$, the norm of the gradient of the local loss $L_s$ at $\theta_s^t$ must be vastly larger than $2\lambda \| \theta^t_s - \theta_s^\bullet \|^2_2$. This means that the value of $\alpha$ (or, equivalently, the number of data injected in $D_s$) must also grow proportionally to $N$.

L.3. Data Poisoning Against MNIST with Images in $[0, 1]^d$

Note that in the data poisoning attack depicted in Figure 2, poisoned data points are easily detectable, as they do not necessarily lie in $[0, 1]^d$ like the pristine images of the MNIST dataset. However, this can be mitigated by the attacker with the cost of providing significantly more data points ($\sim 10^5$). For this, we conduct another experiment in which the attacker divides the poisoned images by the maximum value and clips negative values to 0 (to get images located in $[0, 1]^d$). The results of this experiment are depicted in Figure 20.