# Global and Local Learning for Distributed Personalization

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

We study a model for distributed personalization where the clients train personalized local models and make predictions jointly with a server-side shared model. Using this framework, which we call Global and Local Learning, the complexity of the central shared model can be minimized while still enjoying all the performance benefits that joint training provides. Our framework is robust to data heterogeneity, addressing the slow convergence problem that typical federated learning methods (often applied in these settings) face when the data is non-i.i.d. across clients. We test the theory empirically and find substantial performance gains over baselines.

## 1 Introduction

As machine learning models have scaled up to provide predictions to billions of users in tasks ranging from web search and content recommendation to predictive typing and personalized assistants, there is a natural desire to enable personalization of these models to the individual users. A natural obstacle is the typically small quantity of data available for any individual user, which seems to prohibit the kinds of complex models which are possible when pooling data across the entire population. Nevertheless, it remains desirable to perform as much learning locally, *on device*, as possible to avoid costly communication, and produce personalized predictions for individual users while harnessing population-wide trends and commonalities. In this paper, we term this question as one of Distributed Personalization, and develop a Global and Local Learning framework for learning such predictors in a distributed manner, robust to communication delays.

The research area with most similarities to our questions is federated learning (FL) [5, 13, 19, 22], where a number of distributed clients seek to learn a predictive model, with each client having access to a stream of data. The goal is to learn a significantly more accurate model than each client can learn with just the locally available data, while typically avoiding explicit communication of local data to a central entity. Though these properties align with our desiderata, most prior FL work considers learning a single centralized model using samples from all the clients. This approach effectively integrates preferences across clients, but it fails to provide significant personalization to the clients when they have different data distributions [12, 18]. Hence a direct application of prior FL works is not feasible. Indeed, several of our motivating questions are posed as interesting research directions in a recent monograph on FL (see Section 3.3.4 of [13]). We distinguish our problem setting from FL as the privacy considerations often highlighted there are not the primary concern of this paper, though an interesting direction for future research.

Our approach is very basic so it has general applicability to many kinds of models—handwriting recognition, reinforcement learning, and machine translation are all possibilities, for example. To illustrate the key challenges of our setting, we adopt the problem of wellness nudging as a main motivating example throughout the paper. In this setting, each client is typically a computer or a mobile device, associated with a user. The goal of learning is to improve the user's exercise habits,

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

measured via activity level metrics. Depending on the approach, the learning task might involve predicting the value of different interventions, and using them to make a recommendation decision. Some salient aspects of this setting are:

1. Different users have different preferences, so a personalized model is highly desirable.
2. The samples collected from each user are not enough to train a powerful personalized model.
3. Incorporating all personalization in a centralized model can result in a huge model size, making it intractable from storage and computational perspectives.

**Our contributions.** We address the above issues by proposing a *model separation* approach. Specifically, we consider the scenario where the server of the system maintains a *global model* that is shared across all clients, and each client maintains its own personalized *local model*. Each client makes predictions by jointly using the global model and the local model. As an example, the final prediction value can be the sum of the predictions given by the global and local models.

For this setting, we develop novel distributed learning algorithms. Since the key hallmark of our approach is the joint training of global and local models that learn to be accurate relative to each other, we name our framework and algorithms Global and Local Learning, or simply Glocal.

This new framework has several desirable properties that make it suitable for large-scale deployment. First, the clients have freedom to adjust the complexity of their local model relative to their anticipated data rate. For instance, in linear prediction, a client might consider only a subset of features in its local model, or might even partition the features between globally and locally relevant ones to use in the respective models. This flexibility allows devices of different hardware complexity to join the system with low cost. Second, in a version of our algorithm (the SGD-variant introduced in Section 3.2), all information about the local model and the local features that the client uses to train the local model can be summarized as gradients with respect to the server parameters for the server. The raw examples are never communicated. In this paper, we consider a server which maintains and updates the global model. If such a server is unavailable, we hope to leverage approaches for fully decentralized learning [11, 23] in future work.

To model real-world scenarios, we allow *communication delay* between the server and the clients into our algorithm design and analysis, incorporating robustness to such delays. This is inspired by prior works on delayed feedback stochastic optimization [1, 7, 8, 26], but requires new insights because our problem is complicated by model splitting. We derive regret bounds for our algorithms, exhibiting improvements over purely global and local learning schemes, and showing their robustness to delays. Our algorithms and analysis nicely work with mini-batches, which we show in Section D.

Empirically, we evaluate the algorithm across a number of datasets from multitask and multiclass learning problems to capture heterogenous data distributions across clients while still having some overlap to leverage the global component. We demonstrate the efficacy of our algorithm over natural baselines, showing that it generally improves upon both purely local and global learning methods in the absence of communication delays as the number of clients is varied. We further investigate the robustness to delays and find extremely mild performance degradation on most datasets.

The rest of the paper is organized as follows. After a survey of related work, we present the formal setting. We then describe the two main algorithms and their theoretical properties. We conclude with the empirical evaluation.

**Related work** As mentioned, Federated learning has relatively similar concerns as our goal of Distributed Personalization. As proposed, the main focus is on communication efficiency [19], with a global shared model in the federated learning system. There are also works dealing with the heterogeneity of the data distribution in federated systems [5, 12, 14, 17, 20, 22]. However, a fundamental difference between our work and theirs is that their global and local models still operate in the same parameter space, while our framework provides more flexibility in the design of local models, as we will see in Section 2. A concurrent work of [21] adopts a similar idea of global and local model splitting, and demonstrates the efficacy of this scheme through some primary empirical results. Compared to their work, our work provides a more general framework, and investigates wider aspects such as communication delay and mini-batching along with theoretical justifications.

Several papers have addressed stochastic optimization with delayed feedback [1, 7, 8, 26] with different approaches. They all concluded that the asymptotic performance of stochastic optimization is not affected by the delay in feedback, provided that the amount of delay is bounded, and the objective

function is smooth. Inspired by them, we extend their results to a more challenging distributed setting solving multiple problems distributions and split models, and draw similar conclusions. These techniques have also been extended to the fully decentralized setting [11, 23], but do not consider the additional challenges of model separation and statistical issues that we investigate.

For reducing the complexity of a centralized model, the work of Weinberger et al. [24] proposed to use the feature hashing approach for personalized spam filtering. Though they demonstrated dramatic compressions, there are many applications where feature hashing may harm performance. We note that the global model in our system might further leverage task clustering approaches from the multitask learning literature (see e.g. [15, 25]) if additional modeling flexibility is desired.

## 2 Problem Setting

We consider an online learning scenario in a distributed learning system which consists of one server and $P$ clients.[1] At any time $t$, the server keeps a *global model*, which can be parameterized by a vector $w_t^g \in \mathbb{R}^d$ and each client $i$ keeps a *local model*, parameterized by $w_{i,t} \in \mathbb{R}^{d_i}$. At each round $t$, client $i$ observes a loss function $\ell_{i,t} : \mathbb{R}^d \times \mathbb{R}^{d_i} \to \mathbb{R}$ and incurs the loss $\ell_{i,t}(w_t^g, w_{i,t})$. In the later sections, when we also consider communication delays, the client lacks access to the latest server model $w_t^g$ instead having only some delayed version from a previous round $w_{t-\tau_i}^g$, where $\tau_i$ is the communication latency between the server and client $i$, and similarly the server only observes delayed information about the client loss function.

As an example, the client $i$ might receive a feature vector $x_{i,t}$ at time $t$ and $\ell_{i,t}$ might be a prediction loss $\ell(y_{i,t}; f(x_{i,t}; w_t^g) + f(x_{i,t}; w_{i,t}))$, where $y_{i,t}$ is a label observed after making the prediction, $\ell$ is some loss function and $f$ is the functional form given the parameters. More generally, the local and global functions can even differ, such as by acting on different feature sets in case communicating high dimensional features or gradients to the server is prohibitive. We give more detailed examples instantiating these choices after we describe our solution concept next.

We use $\lambda_i \geq 0$ to denote a weight for client $i$. The goal is to have low regret against the optimal joint global and local models. The (average) regret is defined as

$$\text{Reg} = \sup_{u^g, u_i} \frac{1}{PT} \sum_{i=1}^{P} \lambda_i \sum_{t=1}^{T} \left( \ell_{i,t}(w_t^g, w_{i,t}) - \ell_{i,t}(u^g, u_i) \right). \tag{1}$$

It might appear that the model requires all clients to see the same number of examples as we draw a loss function $\ell_{i,t}$ for each client $i$ on every round $t$. We can easily circumvent this by setting the loss function to be identically 0 if no data was observed on some round. Formally, if $N_i$ non-zero samples are observed at the client $i$, then setting $\lambda_i = T/N_i$ turns the objective into a sum of the average losses incurred at each client. For simplicity, in the later text, we assume $\lambda_i = 1$. Below we give more concrete examples for our system.

**Example 1** (Linear regression)**.** In this case, we define $\ell_{i,t}(w^g, w_i) = \left(y_{i,t} - w^{g\top} x_{i,t}^g - w_i^\top x_{i,t}^l\right)^2$ for some sample $(x_{i,t}^g, x_{i,t}^l, y_{i,t}) \in \mathbb{R}^d \times \mathbb{R}^{d_i} \times \mathbb{R}$. Here, $y_{i,t} \in \mathbb{R}$ is the label; $x_{i,t}^g$ and $x_{i,t}^l$ are the features used by the global and local models respectively. Note that $x_{i,t}^g$ and $x_{i,t}^l$ can be identical, but we allow separate feature spaces for additional modeling flexibility.

We now consider a stylized specialization of the above example to motivate the need for local models when the data distribution is heterogenous across clients. While quite simple, this example is somewhat representative of making recommendations across polarized preferences.

**Example 2** (Need for local models)**.** In the setting of Example 1 assume further that there exist vectors $u^g, \{u_i\}_{i=1}^P$ such that $y_{i,t} = u^g \cdot x_{i,t} + u_i \cdot x_{i,t}$ for all $i = 1, \ldots, P$ and $t = 1, \ldots, T$ where local and global features are identical. Assume $P$ is an even number and there is a vector $v$ such that $u_i = v$ for $i \leq P/2$ and $u_i = -v$ for $i > P/2$. The distribution of the covariates $x_{i,t}$ is identical across rounds and clients. As $T$ becomes large, the optimal solution for our objective (1) coincides with the underlying parameters which generated the data. If we instead consider purely global training which would find $\min_w \sum_{i=1}^P \sum_{t=1}^T (y_{i,t} - w \cdot x_{i,t})^2$, then the solution of $w$ approaches $u^g$ as $T$ increases. However, when the model has converged, the clients still suffer a loss of $(v \cdot x_{i,t})^2$ for each sample. Thus, each client gets inaccurate predictions despite using a sufficiently expressive model.

[1] The server is not necessarily a single device though, and can be easily scaled in the cloud.

**Communication Delays** In this work, we account for the communication delay between the clients and the server. At each round, each client can upload data samples/ gradients to the server, and/or fetch global models to the client side. Following prior works [1, 7, 26], we assume that at time $t$, client $i$ is able to fetch an outdated global model that is constructed at time $t - \tau_i$, and samples or gradients sent at time $t$ by client $i$ are received by the server at time $t + \tau_i$. We focus on $\tau_i \equiv \tau$ for all $i$ for notational ease, though the general case can also be handled with our approach.[2]

**More notations and assumptions.** For a random vector $v$, we use $\mathbb{V}[v]$ to denote $\mathbb{E}[\|v - \mathbb{E}[v]\|^2] = \text{trace}\left(\text{Cov}[v]\right)$. Denote the gradient of the losses with respect to global parameters and local parameters by:

$$\nabla^g \ell_{i,t}(w^g, w_i) \triangleq \nabla_{w^g} \ell_{i,t}(w^g, w_i) \text{ and } \nabla^l \ell_{i,t}(w^g, w_i) \triangleq \nabla_{w_i} \ell_{i,t}(w^g, w_i).$$

$\nabla \ell_{i,t}(w^g, w_i)$ denotes $\nabla_{(w^g, w_i)} \ell_{i,t}(w^g, w_i)$. For the loss function, we make the following assumptions for any pair $(w^g, w_i)$ such that $\|w^g\|, \|w_i\| \leq D$:

- The value of the loss $\ell_{i,t}(w^g, w_i)$ lies in $[0, 1]$.
- The losses are convex and $\gamma$-smooth jointly in both parameters. $f$ is $\gamma$-smooth if for all $a, b$
$$f(a) - f(b) \leq \nabla f(b) \cdot (a - b) + \tfrac{\gamma}{2}\|a - b\|^2.$$
- The $\ell_2$-norm of the gradient of the loss $\|\nabla \ell_{i,t}(w^g, w_i)\|$ is upper bounded by $G$.[3]

We also assume that each client's data samples $(x^g_{i,t}, x^l_{i,t}, y_{i,t})$ are i.i.d. across time, but the distributions can differ across the different clients. We use $\Pi_D(v) \triangleq \text{argmin}_{u:\|u\| \leq D} \|u - v\|$ to denote that projection operator onto a ball of radius $D$.

# 3 Algorithms

We extend two common centralized statistical learning algorithms to the distributed setting. One is the empirical risk minimization (ERM) approach that is fully general in that it can be coupled with any centralized loss minimization scheme, while the second is a stochastic gradient descent (SGD) approach which is a computationally attractive incremental approach for large-scale settings. We introduce them in Sections 3.1 and 3.2 respectively.

When there is no communication delay, these algorithms are straightforward extensions of traditional statistical learning algorithms, though some care is needed in analysis due to model splitting. Making them further robust to communication delay is non-trivial both in terms of algorithm design and analysis. The way we treat delays is considered a primary innovation of this paper.

Note that in our setting, although the samples from a specific client are i.i.d. across time, the server and the client individually face non-i.i.d. loss functions, because the parameters of the other side are constantly changing. It has been shown in [26] that when an online learning algorithm with delayed feedback faces a non-i.i.d. loss sequence, the worst case regret lower bound would be $\Omega(\sqrt{\tau/T})$, where $\tau$ is the delay and $T$ is the horizon. On the other hand, as shown by [1, 7], when facing an i.i.d. loss sequence with smooth losses, the regret bound could become $O(\sqrt{1/T} + (\tau/T)^\alpha)$, where the effect of delay only appears in a lower order term ($\alpha > \frac{1}{2}$). Our goal is to achieve an $O(\sqrt{1/T} + (\tau/T)^\alpha)$ regret bound, even though the server and clients seemingly face non-i.i.d. losses.

## 3.1 Empirical risk minimization based approach

Empirical-risk minimization (ERM) is a simple and generic way of finding a good model given i.i.d. data samples. In the traditional centralized setting, the learner simply finds the model that minimizes the empirical loss on the previously observed data. We extend this algorithm to our setting as follows: in each round, client $i$ fetches the newest global model $w^g_{t-\tau}$, and then finds a local model $w_{i,t}$ which, together with $w^g_{t-\tau}$, jointly minimizes the empirical loss on all the previously observed data of client $i$ (Algorithm 1). On the server side, in each round, the server receives the newest loss functions $\ell_{i,t-\tau}(\cdot, \cdot)$ and local models $w_{i,t-\tau}$ from all clients, and then finds a global model $w^g_t$ that, together with all local models, jointly minimizes the total empirical loss across all the clients (Algorithm 2).

[2] We can extend the analysis to when the uplink and downlink latencies for the clients are different.

[3] Smoothness of $\ell$ gives that gradients exist almost everywhere so that we can avoid working with subgradients.

**Algorithm 1** Glocal.ERM.Client

**for** $t = 1, \ldots, T$ **do**

Fetch the global model $w^g_{t-\tau}$.

See loss function $\ell_{i,t}(\cdot,\cdot)$ and incur loss $\ell_{i,t}(w^g_{t-\tau}, w_{i,t})$.

Send $\ell_{i,t}$ and $w_{i,t}$ to the server.

Update the local model:

$$w_{i,t+1} = \operatorname*{argmin}_{w:\|w\|\le D} \sum_{s=1}^{t} \ell_{i,s}(w^g_{t-\tau}, w). \quad (2)$$

**Algorithm 2** Glocal.ERM.Server

**for** $t = 1, \ldots, T$ **do**

Receive $\ell_{i,t-\tau}$ and $w_{i,t-\tau}$ from all $i = 1, \ldots, P$.

Update

$$w^g_{t+1} = \operatorname*{argmin}_{w:\|w\|\le D} \sum_{i=1}^{P} \sum_{s=1}^{t-\tau} \ell_{i,s}(w, w_{i,t-\tau}). \quad (3)$$

Analyzing this algorithm is not as straightforward as the centralized ERM, because each client (server) is now facing a changing global (local) model, making the losses seen by the client (server) non-i.i.d. The algorithm is related to *alternating minimization*, whose offline convergence properties has been extensively studied in Beck [2]. Our analysis is inspired by Beck [2], but further complicated because we deal with the online setting and consider the presence of delay. The following theorem gives a regret bound for this algorithm.

**Theorem 1.** *Suppose the variance of the loss $\mathbb{V}[\ell_{i,t}(w^g, w_i)]$ is upper bounded by $\sigma^2$ for all $i, t$. Let $d_{total} := d + \sum_{i=1}^{P} d_i$. Then for any $w^g_*, w_{i,*}$, Glocal.ERM (Algorithm 1 and 2) guarantees*

$$\mathbb{E}\left[\frac{1}{PT}\sum_{i=1}^{P}\sum_{t=1}^{T} \ell_{i,t}\left(w^g_{t-\tau}, w_{i,t}\right) - \ell_{i,t}\Big(w^g_*, w_{i,*}\Big)\right] = \widetilde{\mathcal{O}}\left(\sqrt{\frac{d_{total}\,\sigma^2}{PT}} + poly(d, d_i, \gamma, D)\left(\frac{\tau}{T}\right)^{\frac{3}{4}}\right).$$

The exact lower-order term can be found in Appendix A. We can compare the dominant term in Theorem 1, with that of independent ERM at each client: $\mathcal{O}\left(\frac{1}{P}\sum_{i=1}^{P}\sqrt{(d+d_i)\sigma^2/T}\right) = \mathcal{O}\left(\sqrt{(d+d_\ell)\sigma^2/T}\right)$, if each client has the same number $d_\ell$ of features. One can see that the complexity of the global model is amortized among the clients in Theorem 1, unlike the above bound. Of course the server can learn both $w^g$ and $w_i$ in a fully centralized setting, but doing so severely increases the storage complexity of the server, as well as communication if $d_i$ is large relative to $d$.

One may wonder whether the following pair of update rules also work (*cf.* (2) and (3)):

$$w_{i,t+1} = \operatorname*{argmin}_{w:\|w\|\le D}\left\{\sum_{s=1}^{t}\ell_{i,s}(w^g_{s-\tau}, w)\right\} \quad \text{and} \quad w^g_{t+1} = \operatorname*{argmin}_{w:\|w\|\le D}\left\{\sum_{i=1}^{P}\sum_{s=1}^{t-\tau}\ell_{i,s}(w, w_{i,s})\right\}.$$

Clearly, this pair of update rules is more computationally efficient than (2) and (3) because the algorithm does not need to apply new model parameters to previous data samples. However, these updates converge very slowly even without delays ($\tau = 0$). See our discussion in Appendix C.

### 3.2 Sochatic gradient descent (SGD) based approach

A natural approach for Distributed Personalization is that upon receiving a new sample, the clients and the server perform individual SGD updates using the gradient with respect to local and global parameters, respectively. We begin with two natural update rules that implement this intuition, and highlight the hurdles in getting the desired regret bound that has $\tau$ in the lower order term. Then we describe our update rule which gets around these hurdles.

#### 3.2.1 Challenges with some baselines

Perhaps the most natural update rule for performing SGD on both client and server sides, in the presence of delays is the following:

$$w_{i,t+1} = w_{i,t} - \eta_i \nabla^l \ell_{i,t}(w^g_{t-\tau}, w_{i,t}), \quad \text{and} \quad w^g_{t+1} = w^g_t - \eta \textstyle\sum_{i=1}^{P} \nabla^g \ell_{i,t-\tau}(w^g_t, w_{i,t-\tau})$$

**Algorithm 3** Glocal.SGD.Client

**for** $t = 1, \ldots, T$ **do**

Fetch the global model $w^g_{t-\tau}$.

Observe a new loss function $\ell_{i,t}$ and incur a loss $\ell_{i,t}(w^g_{t-\tau}, w_{i,t})$.

Send the gradient $\nabla^g_{i,t} \triangleq \nabla^g \ell_{i,t}(w^g_{t-\tau}, w_{i,t})$ to the server.

Update local model:

$$w_{i,t+1} \leftarrow \Pi_D\Big\{w_{i,t} - \eta_i \nabla^l_{i,t-2\tau}\Big\}, \quad (4)$$

where $\nabla^l_{i,t} \triangleq \nabla^l \ell_{i,t}(w^g_{t-\tau}, w_{i,t})$.

**Algorithm 4** Glocal.SGD.Server

**for** $t = 1, \ldots, T$ **do**

Receive $\nabla^g_{i,t-\tau}$ from all $i = 1, \ldots, P$.

Update global model:

$$w^g_{t+1} \leftarrow \Pi_D\left\{w^g_t - \eta \sum_{i=1}^P \nabla^g_{i,t-\tau}\right\}. \quad (5)$$

where we recall that $\nabla^l$ and $\nabla^g$ are the gradients with respect to the local and global parameters, respectively. The problem of this update rule is that the updates of the clients and the server are *mis-aligned*. The prediction model pair is $(w^g_{t-\tau}, w_{i,t})$ on the client side, with the global model lagging the local model by $\tau$ rounds. However, the server performs SGD on the model pair $(w^g_t, w_{i,t-\tau})$, where the local model is behind the global model. This mismatch makes the global parameter update to a slightly incorrect direction, and leads to a regret bound in which $\tau$ appears in the dominant term.

A natural remedy to this mis-alignment is to instead perform the following updates:

$$w_{i,t+1} = w_{i,t} - \eta_i \nabla^l \ell_{i,t}(w^g_{t-\tau}, w_{i,t}), \quad \text{and} \quad w^g_{t+1} = w^g_t - \eta \textstyle\sum_{i=1}^P \nabla^g \ell_{i,t-\tau}(w^g_{t-2\tau}, w_{i,t-\tau}).$$

That is, the updates always utilize a gradient evaluated at a pair of models $(w^g_{t-\tau}, w_{i,t})$ for some client $i$ and time $t$. While this update rule has the right pairing of local and global models on both client and server, there is an asymmetry in the delays experienced by the two. For the clients, there is no delay in the local model used, but the server experiences a round-trip delay of $2\tau$ to maintain alignment with the most current client models it has access to. This asymmetry again results in a delay dependence on the dominant term in our regret analysis. We now present a different solution by creating a symmetric delayed setting on both client and server ends.

### 3.2.2 Our algorithm and results

To address these problems, we *align* the model updates as well as the delay structures on both client and server. That is, all gradients are taken on model pairs of the form $(w^g_{t-\tau}, w_{i,t})$ and the client also experiences a similar delay as the server. To achieve the latter, we let the client make *delayed updates* — the client performs a descent step using a gradient that is one round-trip delayed:

$$w_{i,t+1} = w_{i,t} - \eta_i \nabla^l \ell_{i,t-2\tau}(w^g_{t-3\tau}, w_{i,t-2\tau}), \text{ and } w^g_{t+1} = w^g_t - \eta \sum_{i=1}^P \nabla^g \ell_{i,t-\tau}(w^g_{t-2\tau}, w_{i,t-\tau}).$$

With this fix, we obtain the desired result — *the delay appears in a lower-order term of the regret*.

**Theorem 2.** *Suppose the variance of the gradient of the losses $\mathbb{V}[\nabla \ell_{i,t}(w^g, w_i)]$ is upper bounded by $\sigma^2$ for all $i, t$. Let $W^2 = \|w^g_*\|^2 + \sum_{i=1}^P \|w_{i,*}\|^2$. Then Glocal.SGD (Algorithms 3 and 4) guarantees*

$$\mathbb{E}\left[\frac{1}{PT}\sum_{i=1}^P\sum_{t=1}^T \ell_{i,t}\left(w^g_{t-\tau}, w_{i,t}\right) - \ell_{i,t}\Big(w^g_*, w_{i,*}\Big)\right] = \mathcal{O}\left(\sqrt{\frac{W^2\sigma^2}{PT}} + \textit{poly}\big(\gamma, D, G\big)\big(\frac{\tau}{T}\big)^{\frac{2}{3}}\right).$$

See the appendix for the precise lower order term. Similar to Theorem 1, this bound is an improvement over $\mathcal{O}\left(\frac{1}{P}\sum_{i=1}^P \sqrt{((\|w^g_*\|^2 + \|w_{i,*}\|^2))\sigma^2/T}\right) = \mathcal{O}\left(\sqrt{(\|w^g_*\|^2 + W^2_\ell)\sigma^2/T}\right)$, which is the achievable bound when all clients run independent SGD with a common local weight bound $W_\ell$ and compare their performance with the benchmark $(w^g_*, w_{i,*})$ with each client using the same $w^g_*$. The communication consists of gradients from client to server and global model in the other direction.[4]

[4] If gradients are high-dimensional, examples along with local predictions can be communicated instead.

**Effect of mini-batching.** The need for the client and server to communicate on each round might seem prohibitive. Taking a cue from prior works on distributed optimization, we can further reduce the amount of communication by computing mini-batch gradients at each client and communicating once per mini-batch. We provide the analysis for this version of our updates in Appendix D.

# 4 Experiments

We evaluate Glocal by testing it on several Distributed Personalization scenarios. The first set of scenarios creates multi-task learning problems from multiclass classification datasets, while the other two are previously used datasets in multitask and federated learning.

**Datasets.** The first setting adapts LIBSVM multiclass classification datasets to create related learning problems across clients. Specifically, we make each client solve a binary classification problem, in which all clients share the same class 0 while having possibly different class 1, receiving $M$ examples of each class. We describe details about the creation of the classification datasets in the appendix. The descriptions for the original datasets can be found in Chang and Lin [4]. We use $M = 10$ to mimic the case where each client only has a small amount of data.

The second dataset we use is Isolet, which is a standard multi-task learning dataset. It consists of acoustic features of 150 speakers uttering each of the 26 English letters twice. The speakers are grouped into 5 groups of 30 speakers, based on their pronunciations of the alphabet. The goal is to distinguish between the 26 letters. In our experiments, we assign one speaker's data to each client. More details of the Isolet dataset can be found in Fanty and Cole [10].

The third dataset we use is FEMNIST [3] (which is further derived from the EMNIST datasets [6]). This dataset consists of different persons' hand-written images of the 62 characters [0-9], [a-z], and [A-Z], each as a $28\times28$ pixels image, with the writers of each image recorded. In our experiment, we make each client correpond to a writer and restrict the data to digit classification (i.e., labels are [0-9]). The number of samples of each writer has mean 112.5 and standard deviation 17.8.

**Algorithms and implementation** We evaluate three algorithms under the SGD[5] framework:

1. **Independent**: Each client performs individual SGD on their own data.
2. **Central**: The server performs SGD over all data from all clients.
3. **Glocal**: Glocal.SGD: The SGD version of our approach (Algorithms 3 and 4).

The first two algorithms are baselines corresponding to fully local and central solutions respectively.

For binary classification , we use the linear regression implementation by Vowpal Wabbit (VW) [16] to regress on the labels 0 and 1. For multiclass classification (i.e., Isolet, FEMNIST), we use the one-versus-all reduction to binary problems. For Isolet and FEMNIST datasets, which originally have 617 and 784 features, we use PCA to reduce their dimension to 100.

**Comparison of methods without communication delays** For all the datasets, we partition each client's data into training, validation, and testing splits, and we randomly generate 5 different partitions, averaging the performance in the end. For each dataset, we train them with multiple passes of online learning on the data: 15 passes for LIBSVM, 2 for Isolet, and 6 for FEMNIST.[6] For each method, we perform a grid search on the learning rates within the range of $[0.002, 0.2]$ at a doubling increment, and report test error of the one with the best validation performance.

Table 1 compares the performance of Independent, Central, and Glocal in the case of zero delay. We see that Glocal outperforms the baselines when communication delay is not present with the relative ordering of Central and Independent approaches varying by the dataset.

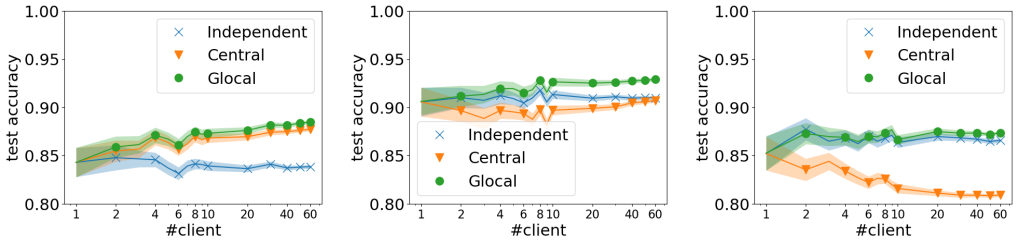
**The effect of the number of clients** We vary the number of clients (while maintaining the number of data samples seen by single client), and see how the number of clients affect the performance of each method. We see that in most of the cases, increasing the number of clients improves the performance of both Central and Glocal (as we show in Figure 1 (a) and (b) for covtype and mnist datasets respectively). This verifies the benefit of joint learning. However, there are also a few cases where increasing the number of clients does not improve the performance (as we show in Figure 1 (c) for the letter dataset). We believe that corresponds to the case where the tasks of the

[5] Our implementation uses the Adagrad [9] variant of SGD for improved performance across all algorithms.

[6] Isolet and FEMNIST have a lot more examples per client, hence the fewer epochs.

| | mnist | covtype | satimage | usps | sensorless | shuttle | pendigits | letter | isolet | femnist |
|---|---|---|---|---|---|---|---|---|---|---|
| Indep | 91.1 | 84.1 | 96.08 | 93.84 | 78.93 | 92.29 | 92.89 | 86.83 | 91.77 | 86.71 |
| Central | 90.06 | 87.42 | 94.96 | 92.88 | 79.51 | 95.55 | 88.03 | 80.88 | **94.44** | 87.68 |
| Glocal | **92.6** | **88.17** | **97.01** | **94.89** | **87.68** | **96.04** | **93.67** | **87.32** | 94.15 | **93.67** |

Table 1: Comparing the classification test accuracy between methods without communication delay. For the LIBSVM datasets (mnist - letter), the problem is binary classification, and #clients is 30. Isolet is a 26-class classification problem, with #clients being 5; FEMNIST is a 10-class classification problem, with #clients being 30.

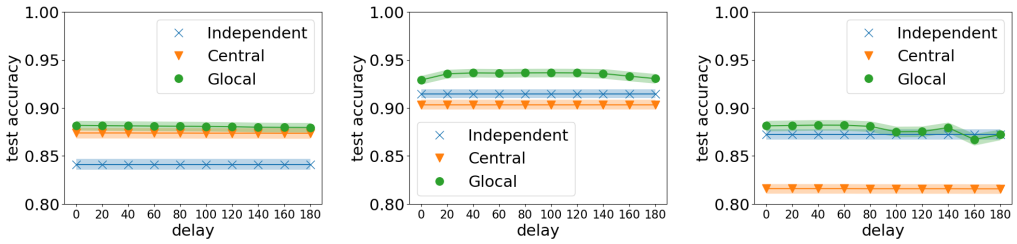



(a) covtype (b) mnist (c) letter

Figure 1: Test accuracy versus number of clients for LIBSVM datasets with no communication delay

clients are significantly different, so joint learning does not bring advantages, and might even hurt the performance of Central. However, Glocal still remains robust in such cases.

**The effect of the delay** Next, we inverstigate the effect of delay under a fixed number of clients. In




(a) covtype (b) mnist (c) letter

Figure 2: Test accuracy versus delays for LIBSVM datasets. The number of clients is fixed at 30.

the experiments in Figure 2, we run the training procedure for $T = 1200$ rounds, and see the effect of delay on the test performance of our method. While Theorem 2 suggests that there is an increasing penalty when delay increases, we observe that for a wide range of delays (in our case, the value of $\tau$ spans from 0 to $T/6$), the test performance is quite robust with only slight variation.

More plots for the remaining LIBSVM datasets under varying delays and clients, as well as the case when we use different features at client and server can be found in the appendix.

## 5 Discussion

In this paper, we present a novel approach for the setting of Distributed Personalization, which is increasingly relevant in many applications. Our algorithms show theoretical and empirical properties such as robustness to delay and strong performance across variation in datasets, which make them suited for real-world applications. In future work, it would be interesting to consider privacy challenges which are beyond the scope of this paper.

## Broader Impact

This paper provides new algorithms for a setting we call Distributed Personalization. The focus of the paper is theoretical and aims to develop general purpose algorithms useful in a variety of settings. Personalization of machine learning predictions has the potential for great benefits as well as can be harmful if not done right. Examples of benefits range from increased productivity through personalized interfaces and greater user satisfaction from personalized content to the potential for customized accessibility, such as speech-based inputs. On the other hand, the use of targeted advertising to cause societal harm and social polarization from recommendation systems are cautionary examples. Nevertheless, we believe that the type of on-device learning that our approach aims to enable is worth further development, particularly if it can be extended to handle privacy considerations.

## References


[1] Alekh Agarwal and John C Duchi. Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 873–881, 2011.

[2] Amir Beck. On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes. *SIAM Journal on Optimization*, 25(1):185–209, 2015.

[3] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[4] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[5] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. *CoRR*, abs/1802.07876, 2018. URL `http://arxiv.org/abs/1802.07876`.

[6] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[7] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(Jan):165–202, 2012.

[8] John C Duchi, Alekh Agarwal, and Martin J Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3):592–606, 2011.

[9] John C. Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011. URL `http://dl.acm.org/citation.cfm?id=2021068`.

[10] Mark Fanty and Ronald Cole. Spoken letter recognition. In *Advances in Neural Information Processing Systems*, pages 220–226, 1991.

[11] Lie He, An Bian, and Martin Jaggi. Cola: Decentralized linear learning. In *Advances in Neural Information Processing Systems*, pages 4536–4546, 2018.

[12] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

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

[14] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378*, 2019.

[15] Meghana Kshirsagar, Eunho Yang, and Aurélie C Lozano. Learning task clusters via sparsity grouped multitask learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 673–689. Springer, 2017.

[16] John Langford, Lihong Li, and Alex Strehl. Vowpal wabbit online learning project. *hunch.net*, 2007.

[17] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[18] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *arXiv preprint arXiv:1908.07873*, 2019.

[19] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[20] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *ICML*, 2019.

[21] Daniel Peterson, Pallika Kanani, and Virendra J Marathe. Private federated learning with domain adaptation. *International Workshop on Federated Learning for User Privacy and Data Confidentiality*, 2019.

[22] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[23] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. $D^2$: Decentralized training over decentralized data. *arXiv preprint arXiv:1803.07068*, 2018.

[24] Kilian Q. Weinberger, Anirban Dasgupta, John Langford, Alexander J. Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, pages 1113–1120, 2009.

[25] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

[26] Martin Zinkevich, John Langford, and Alex J Smola. Slow learners are fast. In *Advances in neural information processing systems*, pages 2331–2339, 2009.

# Appendix

We include the following items in the appendix:

A. The proof of Theorem 1 for the Glocal.ERM algorithm
B. The proof of Theorem 2 for the Glocal.SGD algorithm
C. Explaining the failure of the fictitious-play strategy described in Section 3.1 with simulation results (see the discussion below Theorem 1)
D. Showing how to use the technique of mini-batching to achieve communication efficiency and robustness to delay.
E. More experimental results that complement Section 4

## A Proofs for Theorem 1 (Glocal.ERM algorithm)

We define several notations to be used in the proofs.

**Definition 1.** *For any $w^g, w^l$,*

$$\begin{aligned} L_i(w^g, w^l) &\triangleq \mathbb{E}\left[\ell_{i,t}(w^g, w^l)\right] \\ (w^g_*, w_{1,*}, \ldots, w_{P,*}) &\triangleq \operatorname*{argmin}_{w^g, w_1, \ldots, w_P} \sum_{i=1}^{P} L_i(w^g, w_i) \\ \widehat{L}_{i,t}(w^g, w^l) &\triangleq \frac{1}{t-1}\sum_{s=1}^{t-1} \ell_{i,s}(w^g, w^l) \\ \Delta_{i,t}(w^g, w^l) &\triangleq \widehat{L}_{i,t}(w^g, w^l) - \widehat{L}_{i,t}(w^g_*, w_{i,*}). \end{aligned}$$

**Definition 2.** *Define*

$$\begin{aligned} \overline{\sigma} &= \sqrt{\frac{1}{P}\sum_{i=1}^{P} \sigma_i^2}, \\ \overline{d} &= \frac{1}{P}\left(d + \sum_{i=1}^{P} d_i\right), \end{aligned}$$

*where $\sigma_i$ is an upper bound for the variance of $\ell_{i,t}(w^g, w_i)$ for any $w^g, w_i$, and $d, d_1, \ldots, d_P$ are the dimensions of $w^g, w_1, \ldots, w_P$ respectively.*

First, we bound the difference between $\sum_{i=1}^{P} \widehat{L}_{i,t}$ and $\sum_{i=1}^{P} L_i$.

**Lemma 1.** *Suppose $D \leq T$. With probability $1 - \frac{1}{T}$, the following holds for all $t$ and all $(w^g, w_1, \ldots, w_P)$:*

$$\left|\sum_{i=1}^{P} \widehat{L}_{i,t}(w^g, w_i) - \sum_{i=1}^{P} L_i(w^g, w_i)\right| = \mathcal{O}\left(P \cdot \sqrt{\frac{\overline{\sigma}^2 \overline{d} \log T}{t}} + P^2 \cdot \frac{\overline{d}\log T}{t}\right).$$

*Proof.* We use Bernstein's inequality on the discretized space of $(w^g, w_1, \ldots, w_P)$. Recall that $(w^g, w_1, \ldots, w_P) \in \mathbb{R}^d \times \mathbb{R}^{d_1} \times \cdots \mathbb{R}^{d_P}$. We discretize each dimension into $T^2$ values, and so the total number of discretization points is $\left(T^2\right)^{d+\sum_{i=1}^{P} d_i}$. Suppose the nearest discretization point to $(w^g, w_1, \ldots, w_P)$ is $(\widehat{w}^g, \widehat{w}_1, \ldots, \widehat{w}_P)$. By Bernstein's inequality, with probability at least $1 - \frac{1}{T^2}$ the following holds for all discretization points:

$$\left|\sum_{i=1}^{P} \widehat{L}_{i,t}(\widehat{w}^g, \widehat{w}_i) - \sum_{i=1}^{P} L_i(\widehat{w}^g, \widehat{w}_i)\right|$$

$$
\begin{aligned}
&= \left| \sum_{i=1}^{P} \frac{1}{t-1} \sum_{s=1}^{t-1} \ell_{i,s}(\widehat{w}^g, \widehat{w}_i) - \sum_{i=1}^{P} L_i(\widehat{w}^g, \widehat{w}_i) \right| \\
&= \mathcal{O}\left( \sqrt{\frac{\left(\sum_{i=1}^{P} \sigma_i^2\right)\left(d + \sum_{i=1}^{P} d_i\right) \log T}{t-1}} + \frac{P\left(d + \sum_{i=1}^{P} d_i\right) \log T}{t-1} \right) \qquad (\ell_{i,t} \in [0,1]) \\
&= \mathcal{O}\left( P \cdot \sqrt{\frac{\overline{\sigma}^2 \overline{d} \log T}{t}} + P^2 \cdot \frac{\overline{d} \log T}{t} \right). \qquad (6)
\end{aligned}
$$

The first equality comes from the fact that all clients generate data independently, so the variance of $\sum_{i=1}^{P} \ell_{i,s}(\widehat{w}^g, \widehat{w}_i)$ is upper bounded by $\sum_{i=1}^{P} \sigma_i^2$. The $(d + \sum_{i=1}^{P} d_i) \log T$ factor comes from $\log\left((T^2)^{d+\sum_{i=1}^{P} d_i}\right)$. Since the distance between $(\widehat{w}^g, \widehat{w}_1, \ldots, \widehat{w}_P)$ and $(w^g, w_1, \ldots, w_P)$ is no more than $\frac{D}{T^2}$ in each dimension, the above implies that

$$
\left| \sum_{i=1}^{P} \widehat{L}_{i,t}(w^g, w_i) - \sum_{i=1}^{P} L_i(w^g, w_i) \right| = \mathcal{O}\left( P \cdot \sqrt{\frac{\overline{\sigma}^2 \overline{d} \log T}{t}} + P^2 \cdot \frac{\overline{d} \log T}{t} + \frac{PD\overline{d}}{T^2} \right)
$$

holds with probability $1 - \frac{1}{T^2}$ for all $w^g, w_i$. Using a union bound over $t$ finishes the proof. □

Next, we state a lemma that is useful for showing the convergence of alternating minimization, which is adapted from the analysis in [2].

**Lemma 2.** *Let $\ell(u, v)$ be a $\gamma$-smooth joint convex function of $u$ and $v$, and $\Omega_u$, $\Omega_v$ are convex feasible sets of $u$, $v$ respectively. Now fix $u = u_0$, and let $v_0 = \operatorname{argmin}_{v \in \Omega_v} \ell(u_0, v)$. Suppose $\sup_{u \in \Omega_u} \|u\| \le D$ and $\ell(u, v) \in [0, R]$ for any $u, v$. Then*

$$
\min_{u \in \Omega_u} \ell(u, v_0) \le \ell(u_0, v_0) - \frac{1}{18\gamma D^2 + 2R} \left[\ell(u_0, v_0) - \ell(u_*, v_*)\right]_+^2 .
$$

*for any $u_* \in \Omega_u, v_* \in \Omega_v$.*

*Proof.* Define

$$
u_1 = \operatorname*{argmin}_{u \in \Omega_u} \left\| u - u_0 + \frac{1}{\gamma} \nabla_u \ell(u_0, v_0) \right\|^2 .
$$

By the smoothness of $\ell$, we have

$$
\ell(u_1, v_0) \le \ell(u_0, v_0) + \nabla_u \ell(u_0, v_0)^\top (u_1 - u_0) + \frac{\gamma}{2} \|u_1 - u_0\|^2 . \qquad (7)
$$

By the optimality of $u_1$, we have

$$
\left( u_1 - u_0 + \frac{1}{\gamma} \nabla_u \ell(u_0, v_0) \right)^\top (u' - u_1) \ge 0 \qquad (8)
$$

for all $u' \in \Omega_u$.

Specially, by invoking (8) with $u' = u_0$, we can further upper bound the right-hand side of (7) by

$$
\ell(u_0, v_0) - \gamma \|u_1 - u_0\|^2 + \frac{\gamma}{2} \|u_1 - u_0\|^2 \le \ell(u_0, v_0) - \frac{\gamma}{2} \|u_1 - u_0\|^2 . \qquad (9)
$$

Below we further lower bound $\|u_1 - u_0\|^2$. Since $v_0$ is the minimizer of $\ell(u_0, \cdot)$ in $\Omega_v$, we have

$$
\nabla_v \ell(u_0, v_0)^\top (v' - v_0) \ge 0 \qquad (10)
$$

for all $v' \in \Omega_v$.

Define $(u_{\min}, v_{\min}) = \operatorname{argmin}_{u \in \Omega_u, v \in \Omega_v} \ell(u, v)$. With the above ingredients, we can bound

$$
\min_{u \in \Omega_u} \ell(u, v_0) - \ell(u_{\min}, v_{\min})
$$

$$
\begin{aligned}
&\le \ell(u_1, v_0) - \ell(u_{\min}, v_{\min}) \\
&\le \ell(u_0, v_0) - \ell(u_{\min}, v_{\min}) + \nabla_u \ell(u_0, v_0)^\top (u_1 - u_0) + \frac{\gamma}{2}\|u_1 - u_0\|^2. && \text{(by (7))} \\
&\le \nabla_u \ell(u_0, v_0)^\top (u_0 - u_{\min}) + \nabla_v \ell(u_0, v_0)^\top (v_0 - v_{\min}) + \nabla_u \ell(u_0, v_0)^\top (u_1 - u_0) + \frac{\gamma}{2}\|u_1 - u_0\|^2 \\
& && \text{(by the convexity of } \ell) \\
&\le \nabla_u \ell(u_0, v_0)^\top (u_1 - u_{\min}) + \frac{\gamma}{2}\|u_1 - u_0\|^2 && \text{(using (10) with } v' = v_{\min}) \\
&\le \gamma (u_1 - u_0)^\top (u_{\min} - u_1) + \frac{\gamma}{2}\|u_1 - u_0\|^2 && \text{(using (8) with } u' = u_{\min}) \\
&\le 2\gamma \|u_1 - u_0\| D + \gamma \|u_1 - u_0\| D \\
&\le 3\gamma \|u_1 - u_0\| D,
\end{aligned}
$$

which implies

$$
\|u_1 - u_0\|^2 \ge \frac{1}{9\gamma^2 D^2} \left[ \min_{u \in \Omega_u} \ell(u, v_0) - \ell(u_{\min}, v_{\min}) \right]^2 .
$$

Combine this with (7), (9), and using the fact $\min_{u \in \Omega_u} \ell(u, v_0) \le \ell(u_1, v_0)$, we get

$$
\begin{aligned}
\min_{u \in \Omega_u} \ell(u, v_0) &\le \ell(u_0, v_0) - \frac{1}{18\gamma D^2} \left[ \min_{u \in \Omega_u} \ell(u, v_0) - \ell(u_{\min}, v_{\min}) \right]^2 \\
&= \ell(u_0, v_0) - \frac{1}{18\gamma D^2} \left[ \ell(u_0, v_0) - \ell(u_{\min}, v_{\min}) \right]^2 \\
&\qquad + \frac{1}{18\gamma D^2} \left( \ell(u_0, v_0) - \min_{u \in \Omega_u} (u, v_0) \right) \left( \ell(u_0, v_0) + \min_{u \in \Omega_u} (u, v_0) - 2\ell(u_{\min}, v_{\min}) \right) \\
&\le \ell(u_0, v_0) - \frac{1}{18\gamma D^2} \left[ \ell(u_0, v_0) - \ell(u_{\min}, v_{\min}) \right]^2 + \frac{R}{9\gamma D^2} \left( \ell(u_0, v_0) - \min_{u \in \Omega_u} \ell(u, v_0) \right).
\end{aligned}
$$

Rearranging this gives

$$
\begin{aligned}
\min_{u \in \Omega_u} \ell(u, v_0) &\le \ell(u_0, v_0) - \frac{1}{18\gamma D^2 \left(1 + \frac{R}{9\gamma D^2}\right)} \left[ \ell(u_0, v_0) - \ell(u_{\min}, v_{\min}) \right]^2 \\
&= \ell(u_0, v_0) - \frac{1}{18\gamma D^2 + 2R} \left[ \ell(u_0, v_0) - \ell(u_{\min}, v_{\min}) \right]^2 \\
&\le \ell(u_0, v_0) - \frac{1}{18\gamma D^2 + 2R} \left[ \ell(u_0, v_0) - \ell(u_*, v_*) \right]_+^2 .
\end{aligned}
$$

□

**Lemma 3.** *Let $\delta_{t,t'}$ denote $\frac{1}{P}\sum_{i=1}^P \Delta_{i,t'}(w_t^g, w_{i,t'})$. Then Glocal.ERM (Algorithm 1 and 2) ensures that for any $t > 2\tau + 2$,*

$$
\delta_{t,t-\tau-1} \le \delta_{t-2\tau-2,t-\tau-1} - \frac{1}{18\gamma D^2 + 2} \left[ \delta_{t-2\tau-2,t-\tau-1} \right]_+^2 .
$$

*Proof.* By Algorithm 1, we have

$$
\begin{aligned}
w_t^g &= \operatorname*{argmin}_w \sum_{i=1}^P \widehat{L}_{i,t-\tau-1}(w, w_{i,t-\tau-1}) \\
w_{i,t-\tau-1} &= \operatorname*{argmin}_w \widehat{L}_{i,t-\tau-1}(w_{t-2\tau-2}^g, w).
\end{aligned}
$$

The second inequality implies that $(w_{1,t-\tau-1}, w_{2,t-\tau-1}, \ldots, w_{P,t-\tau-1})$ jointly minimizes $\sum_{i=1}^P \widehat{L}_{i,t-\tau-1}(w_{t-2\tau-2}^g, \cdot)$. Using Lemma 2 with $R = 1$, we get

$$
\frac{1}{P} \sum_{i=1}^P \widehat{L}_{i,t-\tau-1}(w_t^g, w_{i,t-\tau-1})
$$

$$\leq \frac{1}{P}\sum_{i=1}^{P}\widehat{L}_{i,t-\tau-1}(w^g_{t-2\tau-2}, w_{i,t-\tau-1}) - \frac{1}{18\gamma D^2+2}\left[\frac{1}{P}\sum_{i=1}^{P}\Delta_{i,t-\tau-1}(w^g_{t-2\tau-2}, w_{i,t-\tau-1})\right]_+^2.$$

Subtracting both sides with $\frac{1}{P}\sum_{i=1}^{P}\widehat{L}_{i,t-\tau-1}(w^g_*, w_{i,*})$ finishes the proof. □

**Lemma 4.** *Glocal.ERM (Algorithm 1 and 2) ensures that for any $t > 3\tau + 3$,*

$$\delta_{t-\tau-1,t} \leq \delta_{t-\tau-1,t-2\tau-2} + \mathcal{O}\left(\frac{\tau}{t}\sqrt{\frac{\overline{\sigma}^2\overline{d}}{t}} + P\cdot\frac{\tau\overline{d}}{t^2}\right)$$

*Proof.*

$$\begin{aligned}
&\sum_{i=1}^{P}\Delta_{i,t}(w^g_{t-\tau-1}, w_{i,t})\\
&= \sum_{i=1}^{P}\left(\widehat{L}_{i,t}(w^g_{t-\tau-1}, w_{i,t}) - \widehat{L}_{i,t}(w^g_*, w_{i,*})\right)\\
&\leq \sum_{i=1}^{P}\left(\widehat{L}_{i,t}(w^g_{t-\tau-1}, w_{i,t-2\tau-2}) - \widehat{L}_{i,t}(w^g_*, w_{i,*})\right)\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{(because } w_{i,t} \text{ is the minimizer of } \widehat{L}_{i,t}(w^g_{t-\tau-1},\cdot))\\
&= \sum_{i=1}^{P}\Delta_{t-2\tau-2}(w^g_{t-\tau-1}, w_{i,t-2\tau-2}) + \sum_{i=1}^{P}\left(\widehat{L}_{i,t}(w^g_{t-\tau-1}, w_{i,t-2\tau-2}) - \widehat{L}_{i,t-2\tau-2}(w^g_{t-\tau-1}, w_{i,t-2\tau-2})\right)\\
&\qquad + \sum_{i=1}^{P}\left(\widehat{L}_{i,t-2\tau-2}(w^g_*, w_{i,*}) - \widehat{L}_{i,t}(w^g_*, w_{i,*})\right). \qquad (11)
\end{aligned}$$

Now remains the bound the last two terms above. Note that they are of similar form. Then for any $t > 3\tau + 3$, any $(w^g, w_1, w_2, \ldots, w_P)$,

$$\begin{aligned}
&\sum_{i=1}^{P}\left(\widehat{L}_{i,t}(w^g, w_i) - \widehat{L}_{i,t-2\tau-2}(w^g, w_i)\right)\\
&= \sum_{i=1}^{P}\left(\frac{1}{t-1}\sum_{s=1}^{t-1}\ell_{i,s}(w^g, w_i) - \frac{1}{t-2\tau-3}\sum_{s=1}^{t-2\tau-3}\ell_{i,s}(w^g, w_i)\right)\\
&= \sum_{i=1}^{P}\frac{1}{t-1}\left(\sum_{s=1}^{t-1}\ell_{i,s}(w^g, w_i) - \frac{t-1}{t-2\tau-3}\sum_{s=1}^{t-2\tau-3}\ell_{i,s}(w^g, w_i)\right)\\
&= \sum_{i=1}^{P}\frac{1}{t-1}\left(\sum_{s=t-2\tau-2}^{t-1}\ell_{i,s}(w^g, w_i) - \frac{2\tau+2}{t-2\tau-3}\sum_{s=1}^{t-2\tau-3}\ell_{i,s}(w^g, w_i)\right)\\
&= \sum_{i=1}^{P}\frac{2\tau+2}{t-1}\left(\frac{1}{2\tau+2}\sum_{s=t-2\tau-2}^{t-1}\ell_{i,s}(w^g, w_i) - \widehat{L}_{i,t-2\tau-2}(w^g, w_i)\right) \qquad (12)
\end{aligned}$$

For the second term on the right-hand side of (11), we can now bound its expectation with the help of (12) and Lemma 1:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{P}\left(\widehat{L}_{i,t}(w^g_{t-\tau-1}, w_{i,t-2\tau-2}) - \widehat{L}_{i,t-2\tau-2}(w^g_{t-\tau-1}, w_{i,t-2\tau-2})\right)\right]\\
&= \mathbb{E}\left[\sum_{i=1}^{P}\frac{2\tau+2}{t-1}\left(\frac{1}{\tau}\sum_{s=t-2\tau-2}^{t-1}\ell_{i,s}(w^g_{t-\tau-1}, w_{i,t-2\tau-2}) - \widehat{L}_{i,t-2\tau-2}(w^g_{t-\tau-1}, w_{i,t-2\tau-2})\right)\right]
\end{aligned}$$

$$\leq \underbrace{\mathbb{E}\left[\sum_{i=1}^{P}\frac{2\tau+2}{t-1}\left(\frac{1}{2\tau+2}\sum_{s=t-2\tau-2}^{t-1}\ell_{i,s}(w^g_{t-\tau-1},w_{i,t-2\tau-2})-L_i(w^g_{t-\tau-1},w_{i,t-2\tau-2})\right)\right]}_{\textbf{term}_1}$$
$$+\mathcal{O}\left(P\cdot\sqrt{\frac{\overline{\sigma}^2\overline{d}\log T}{t-2\tau-2}}+P^2\cdot\frac{\overline{d}\log T}{t-2\tau-2}\right)\times\frac{2\tau+2}{t-1}$$

Notice that $w^g_{t-\tau}$ and $w_{i,t-2\tau}=w_{i,t-\tau}$ only depend on $\ell_{i,s}$ for $s<t-\tau$. Therefore, conditioned on $\{\ell_{i,s}\}_{s<t-\tau}$, the expectation of $\ell_{i,s}(w^g_{t-\tau},w_{i,t-2\tau})$ for $s\geq t-\tau$ is exactly $L_i(w^g_{t-\tau},w_{i,t-2\tau})$. Therefore, $\textbf{term}_1$ is zero. On the other hand, the expectation of the third term on the right-hand side of (11) is

$$\mathbb{E}\left[\sum_{i=1}^{P}\left(\widehat{L}_{i,t-2\tau}(w^g_*,w_{i,*})-\widehat{L}_{i,t}(w^g_*,w_{i,*})\right)\right]=0$$

because $\widehat{L}_{i,t}(w^g,w^l)$ is an unbiased estimator of $L_i(w^g,w^l)$ for fixed $(w^g,w^l)$. With all the above arguments, and using the fact that $t>3\tau+3$, we can bound the expectation of the last two summations in (11) by

$$\widetilde{\mathcal{O}}\left(P\cdot\sqrt{\frac{\overline{\sigma}^2\overline{d}}{t}}+P^2\cdot\frac{\overline{d}}{t}\right)\times\frac{\tau}{t},$$

which finishes the proof. □

We also need the following lemma to prove Theorem 1.

**Lemma 5.** *For any $(w^g,w_1,\ldots,w_P)$, with probability $1-\frac{1}{T^2}$,*

$$\left[\frac{1}{P}\sum_{i=1}^{P}\Delta_{i,t}(w^g,w_i)\right]_-\leq\widetilde{\mathcal{O}}\left(\sqrt{\frac{\overline{\sigma}^2\overline{d}}{t}}+P\cdot\frac{\overline{d}}{t}\right).$$

*Proof.* By the definition of $w^g_*$ and $w_{i,*}$, we have for all $w^g,w_i$,

$$\sum_{i=1}^{P}\left(L_i(w^g,w_i)-L_i(w^g_*,w_{i,*})\right)\geq 0.$$

Then by Lemma 1, we have with probability at least $1-\frac{1}{T}$,

$$\begin{aligned}
&\sum_{i=1}^{P}\Delta_{i,t}(w^g,w_i)\\
&=\sum_{i=1}^{P}\left(\widehat{L}_{i,t}(w^g,w_i)-\widehat{L}_{i,t}(w^g_*,w_{i,*})\right)\\
&\geq\sum_{i=1}^{P}\left(L_i(w^g,w_i)-L_i(w^g_*,w_{i,*})\right)-\widetilde{\mathcal{O}}\left(P\cdot\sqrt{\frac{\overline{\sigma}^2\overline{d}}{t-1}}+P^2\cdot\frac{\overline{d}}{t-1}\right)\\
&=-\widetilde{\mathcal{O}}\left(P\cdot\sqrt{\frac{\overline{\sigma}^2\overline{d}}{t-1}}+P^2\cdot\frac{\overline{d}}{t-1}\right).
\end{aligned}$$

□

Finally, we are now able to prove Theorem 1. We provide a complete statement of the theorem below.

**Theorem 1** *Suppose the variance of the loss $\mathbb{V}[\ell_{i,t}(w^g, w_i)]$ is upper bounded by $\sigma_i^2$, and suppose $\sigma_i^2 \le \sigma^2$ for all $i$. Then Glocal.ERM (Algorithm 1 and 2) guarantees*

$$\mathbb{E}\left[\frac{1}{PT}\sum_{i=1}^{P}\sum_{t=1}^{T}\left(\ell_{i,t}\left(w^g_{t-\tau-1}, w_{i,t}\right) - \ell_{i,t}\left(w^g_*, w_{i,*}\right)\right)\right] \tag{13}$$

$$= \widetilde{\mathcal{O}}\left(\sqrt{\frac{\left(d+\sum_{i=1}^{P} d_i\right)\sigma^2}{PT}} + \frac{(1+D^2\gamma)\tau^{\frac{3}{4}}}{T^{\frac{3}{4}}} + \frac{(1+D^2\gamma)\tau + \left(d+\sum_{i=1}^{P} d_i\right)}{T}\right) \tag{14}$$

*Proof of Theorem 1.* Let $C_0 = 18\gamma D^2 + 2$. Combining Lemma 3 and 4, we get that for $t > \max\{C_0\tau, 3\tau+3\}$

$$\begin{aligned}
&\mathbb{E}\left[\delta_{t-\tau-1,t}\right] \\
&\le \mathbb{E}\left[\delta_{t-\tau-1,t-2\tau-2}\right] + \frac{\tau}{t}\times\widetilde{\mathcal{O}}\left(\frac{\overline{\sigma}\sqrt{\overline{d}}}{\sqrt{t}} + \frac{P\overline{d}}{t}\right) && \text{(Lemma 4)} \\
&\le \mathbb{E}\left[\delta_{t-3\tau-3,t-2\tau-2}\right] - \frac{1}{C_0}\mathbb{E}\left[\left[\delta_{t-3\tau-3,t-2\tau-2}\right]_+^2\right] + \frac{\tau}{t}\times\widetilde{\mathcal{O}}\left(\frac{\overline{\sigma}\sqrt{\overline{d}}}{\sqrt{t}} + \frac{P\overline{d}}{t}\right) && \text{(Lemma 3)} \\
&= \mathbb{E}\left[\delta_{t-3\tau-3,t-2\tau-2}\right] - \frac{1}{C_0}\mathbb{E}\left[\delta^2_{t-3\tau-3,t-2\tau-2}\right] + \frac{\tau}{t}\times\widetilde{\mathcal{O}}\left(\frac{\overline{\sigma}\sqrt{\overline{d}}}{\sqrt{t}} + \frac{P\overline{d}}{t}\right) + \frac{1}{C_0}\mathbb{E}\left[\left[\delta_{t-3\tau-3,t-2\tau-2}\right]_-^2\right] \\
&\le \mathbb{E}\left[\delta_{t-3\tau-3,t-2\tau-2}\right] - \frac{1}{C_0}\mathbb{E}\left[\delta^2_{t-3\tau-3,t-2\tau-2}\right] + \frac{\tau}{t}\times\widetilde{\mathcal{O}}\left(\frac{\overline{\sigma}\sqrt{\overline{d}}}{\sqrt{t}} + \frac{P\overline{d}}{t}\right) + \frac{1}{C_0}\times\widetilde{\mathcal{O}}\left(\frac{\overline{\sigma}^2\overline{d}}{t} + \frac{P^2\overline{d}^2}{t^2}\right) \\
& && \text{(Lemma 5)}
\end{aligned}$$

Now we focus on $t$'s that can be represented as $t = (2n+1)(\tau+1)$ with integer $n$. Define $B_n = \delta_{2n(\tau+1),(2n+1)(\tau+1)}$. Then the above implies

$$B_n \le B_{n-1} - \frac{1}{C_0}B_{n-1}^2 + \widetilde{\mathcal{O}}\left(\frac{\overline{\sigma}^2\overline{d}}{(n-1)C_0\tau} + \frac{\overline{\sigma}\sqrt{\overline{d}}}{(n-1)^{\frac{3}{2}}\sqrt{\tau}} + \frac{P\overline{d}}{(n-1)^2\tau} + \frac{P^2\overline{d}^2}{(n-1)^2C_0\tau^2}\right).$$

Define $C_1 = \frac{1}{C_0}, C_2 = \frac{\overline{\sigma}^2\overline{d}}{C_0\tau}, C_3 = \frac{\overline{\sigma}\sqrt{\overline{d}}}{\sqrt{\tau}}, C_4 = \frac{P\overline{d}}{\tau} + \frac{P^2\overline{d}^2}{C_0\tau^2}$. Then the above can be written as

$$B_n \le B_{n-1} - C_1B_{n-1}^2 + \widetilde{\mathcal{O}}\left(\frac{C_2}{n-1} + \frac{C_3}{(n-1)^{\frac{3}{2}}} + \frac{C_4}{(n-1)^2}\right).$$

Then using the Lemma 6 below, we have

$$\begin{aligned}
B_n &\le \widetilde{\mathcal{O}}\left(\frac{\overline{\sigma}\sqrt{\overline{d}}}{\sqrt{n\tau}}\right) + \widetilde{\mathcal{O}}\left(\frac{C_0}{n^{\frac{3}{4}}} + \frac{\sqrt{C_0\overline{\sigma}}\cdot\overline{d}^{\frac{1}{4}}}{n^{\frac{3}{4}}\tau^{\frac{1}{4}}}\right) + \widetilde{\mathcal{O}}\left(\frac{C_0}{n} + \frac{\sqrt{C_0P\overline{d}}}{n\sqrt{\tau}} + \frac{P\overline{d}}{n\tau}\right) \\
&= \widetilde{\mathcal{O}}\left(\frac{\overline{\sigma}\sqrt{\overline{d}}}{\sqrt{n\tau}}\right) + \widetilde{\mathcal{O}}\left(\frac{C_0}{n^{\frac{3}{4}}}\right) + \widetilde{\mathcal{O}}\left(\frac{C_0}{n} + \frac{P\overline{d}}{n\tau}\right) \\
&\qquad \text{(simplify the bound using } \tfrac{\overline{\sigma}\sqrt{\overline{d}}}{\sqrt{n\tau}} + \tfrac{C_0}{n^{\frac{3}{4}}} \ge 2\cdot\tfrac{\sqrt{C_0\overline{\sigma}}\cdot\overline{d}^{\frac{1}{4}}}{n^{\frac{3}{4}}\tau^{\frac{1}{4}}} \text{ and } \tfrac{C_0}{n} + \tfrac{P\overline{d}}{n\tau} \ge 2\cdot\tfrac{\sqrt{C_0P\overline{d}}}{n\sqrt{\tau}}\text{)}
\end{aligned}$$

Replacing $(2n+1)(\tau+1)$ back to $t$, we get

$$\mathbb{E}\left[\frac{1}{P}\sum_{i=1}^{P}\Delta_{i,t}(w^g_{t-\tau-1}, w_{i,t})\right] = \widetilde{\mathcal{O}}\left(\frac{\overline{\sigma}\sqrt{\overline{d}}}{\sqrt{t}}\right) + \widetilde{\mathcal{O}}\left(\frac{C_0\tau^{\frac{3}{4}}}{t^{\frac{3}{4}}}\right) + \widetilde{\mathcal{O}}\left(\frac{C_0\tau + P\overline{d}}{t}\right). \tag{15}$$

For $t = 2n(\tau+1), \ldots, 2n(\tau+1)+2n-1$, we can use the same approach to prove it. Thus, (15) actually holds for all $t > \max\{C_0\tau, 3\tau+3\}$. Finally, by Lemma 1, we have

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{P}\sum_{i=1}^{P}\left(\ell_{i,t}(w^g_{t-\tau-1}, w_{i,t}) - \ell_{i,t}(w^g_*, w_{i,*})\right)\right] \\
&= \mathbb{E}\left[\frac{1}{P}\sum_{i=1}^{P}\left(L_i(w^g_{t-\tau-1}, w_{i,t}) - L_i(w^g_*, w_{i,*})\right)\right] \\
&\le \mathbb{E}\left[\frac{1}{P}\sum_{i=1}^{P}\left(\widehat{L}_{i,t}(w^g_{t-\tau-1}, w_{i,t}) - \widehat{L}_{i,t}(w^g_*, w_{i,*})\right)\right] + \widetilde{\mathcal{O}}\left(\frac{\overline{\sigma}\sqrt{\overline{d}}}{t} + \frac{P\overline{d}}{t}\right) \\
&= \mathbb{E}\left[\frac{1}{P}\sum_{i=1}^{P}\Delta_{i,t}(w^g_{t-\tau-1}, w_{i,t})\right] + \widetilde{\mathcal{O}}\left(\frac{\overline{\sigma}\sqrt{\overline{d}}}{t} + \frac{P\overline{d}}{t}\right).
\end{aligned}$$

Combining this with (15), and summing over $t > \max\{C_0\tau, 3\tau+3\}$ finish the proof.

□

**Lemma 6.** *Suppose $B_n \le B_{n-1} - C_1 B_{n-1}^2 + \frac{C_2}{n-1} + \frac{C_3}{(n-1)^{\frac{3}{2}}} + \frac{C_4}{(n-1)^2}$ holds for all $n > n_0 \ge 1$ with $C_1, C_2, C_3, C_4 > 0$, and $B_{n_0} \le R$. Then for all $n \ge n_0$,*

$$B_n \le \frac{D_1}{\sqrt{n}} + \frac{D_2}{n^{\frac{3}{4}}} + \frac{D_3}{n}. \tag{16}$$

*where $D_1 = \sqrt{\frac{2C_2}{C_1}}$, $D_2 = \frac{1+\sqrt{1+2C_1(D_1+C_3)}}{C_1}$, $D_3 = \frac{1+\sqrt{1+4C_1C_4}}{C_1} + n_0 R$.*

*Proof.* We use induction. When $n = n_0$, $B_{n_0} \le R \le \frac{D_3}{n_0}$ by our assumption. Suppose (16) holds for $n-1$, then

$$B_n \le \frac{D_1}{\sqrt{n-1}} + \frac{D_2}{(n-1)^{\frac{3}{4}}} + \frac{D_3}{n-1} - C_1\left(\frac{D_1^2}{n-1} + \frac{D_2^2}{(n-1)^{\frac{3}{2}}} + \frac{D_3^2}{(n-1)^2}\right) + \frac{C_2}{n-1} + \frac{C_3}{(n-1)^{\frac{3}{2}}} + \frac{C_4}{(n-1)^2} \tag{17}$$

where we use that for $a, b, c > 0$, $(a+b+c)^2 \ge a^2 + b^2 + c^2$. Now we prove that the right-hand side of (17) is upper bounded by $\frac{D_1}{\sqrt{n}} + \frac{D_2}{n^{\frac{3}{4}}} + \frac{D_3}{n}$. This is equivalent to

$$\begin{aligned}
&D_1\left(\frac{1}{\sqrt{n-1}} - \frac{1}{\sqrt{n}}\right) + D_2\left(\frac{1}{(n-1)^{\frac{3}{4}}} - \frac{1}{n^{\frac{3}{4}}}\right) + D_3\left(\frac{1}{n-1} - \frac{1}{n}\right) + \frac{C_2}{n-1} + \frac{C_3}{(n-1)^{\frac{3}{2}}} + \frac{C_4}{(n-1)^2} \\
&\le C_1\left(\frac{D_1^2}{n} + \frac{D_2^2}{n^{\frac{3}{2}}} + \frac{D_3^2}{n^2}\right).
\end{aligned} \tag{18}$$

Using the inequality $\frac{1}{(n-1)^k} - \frac{1}{n^k} \le \frac{k}{n(n-1)^k}$ for $0 \le k \le n$, we can bound left-hand side of (18) by

$$\frac{D_1}{n\sqrt{n-1}} + \frac{D_2}{n(n-1)^{\frac{3}{4}}} + \frac{D_3}{n(n-1)} + \frac{C_2}{n-1} + \frac{C_3}{(n-1)^{\frac{3}{2}}} + \frac{C_4}{(n-1)^2} \le \frac{2C_2}{n} + \frac{2(D_1+D_2+C_3)}{n^{\frac{3}{2}}} + \frac{2D_3+4C_4}{n^2}.$$

Therefore, we only need to prove

$$2C_2 \le C_1 D_1^2, \qquad 2(D_1+D_2+C_3) \le C_1 D_2^2, \qquad 2D_3 + 4C_4 \le C_1 D_3^2.$$

They are indeed satisfied by our choice of $D_1, D_2, D_3$. □

# B Proofs for Theorem 2 (Glocal.SGD algorithm)

The complete statement of Theorem 2 is as follows. Note that as stated in Theorem 2, the $\sigma_i$ is defined slightly different from that in Definition 2. Also, note that our Glocal.SGD can deal with more general cases than Glocal.ERM in the sense that the delays $\alpha_i, \beta_i$ can be different for different clients.

**Theorem 2** *Suppose the variance of the gradient of the losses of client $i$, $\mathbb{V}[\nabla \ell_{i,t}(w^g, w_i)]$, is upper bounded by $\sigma_i^2$, and suppose $\sigma_i^2 \leq \sigma^2$. Then Glocal.SGD (Algorithm 3 and 4) guarantees that*

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{PT}\sum_{i=1}^{P}\sum_{t=1}^{T}\ell_{i,t}\left(w_{t-\tau}^g, w_{i,t}\right) - \ell_{i,t}\Big(w_*^g, w_{i,*}\Big)\right] \\
&= \frac{1}{PT} \times \mathcal{O}\left(\frac{\|w_*^g\|^2}{\eta} + \sum_{i=1}^{P}\frac{\|w_{i,*}\|^2}{\eta_i} + \eta T\sum_{i=1}^{P}\sigma_i^2 + T\sum_{i=1}^{P}\eta_i\sigma_i^2 + \gamma\eta^2P^3G^2T\tau^2 + \gamma G^2T\sum_{i=1}^{P}\eta_i^2\tau^2 + PDG\tau\right).
\end{aligned} \tag{19}$$

*Picking*

$$\eta = \eta_i = \min\left\{\sqrt{\frac{\|w_*^g\|^2 + \sum_{i=1}^{P}\|w_{i,*}\|^2}{TP\sigma^2}}, \sqrt[3]{\frac{\|w_*^g\|^2 + \sum_{i=1}^{P}\|w_{i,*}\|^2}{\gamma P^3G^2\tau^2T}}\right\},$$

*the above regret can be further upper bounded by*

$$\mathcal{O}\left(\sqrt{\frac{\left(\|w_*^g\|^2 + \sum_{i=1}^{P}\|w_{i,*}\|^2\right)\sigma^2}{PT}} + \frac{\left(\gamma D^4G^2\tau^2\right)^{\frac{1}{3}}}{T^{\frac{2}{3}}} + \frac{DG\tau}{T}\right). \tag{20}$$

*Proof of Theorem 2.* The objective is

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{P}\left(\ell_{i,t}(w_{t-\tau}^g, w_{i,t}) - \ell_{i,t}(w_*^g, w_{i,*})\right)\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{P}\left(L_i(w_{t-\tau}^g, w_{i,t}) - L_i(w_*^g, w_{i,*})\right)\right] \\
&\leq \mathbb{E}\Bigg[\underbrace{\sum_{t=1}^{T}\sum_{i=1}^{P}(w_{t-\tau}^g - w_*^g)\cdot\nabla^g L_i(w_{t-\tau-1}^g, w_{i,t-1})}_{\text{see Lemma 7}}\Bigg] + \mathbb{E}\Bigg[\underbrace{\sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,*})\cdot\nabla^\ell L_i(w_{i,t-\tau-1}^g, w_{i,t-1})}_{\text{see Lemma 8}}\Bigg] \\
&\quad + \frac{\gamma}{2}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{P}\|w_{t-\tau}^g - w_{t-\tau-1}^g\|^2\right] + \frac{\gamma}{2}\left[\sum_{t=1}^{T}\sum_{i=1}^{P}\|w_{i,t} - w_{i,t-1}\|^2\right].
\end{aligned}$$

(by Lemma 11)

By our update rules (4), (5), the third and the fourth terms above can be upper bounded by $\mathcal{O}(\gamma TP\sup_t\|w_t^g - w_{t-1}^g\|^2) = \mathcal{O}(\gamma TP(\eta PG)^2)$ and $\mathcal{O}(\gamma T\sum_{i=1}^{P}\eta_i^2G^2)$ respectively. Combining them with the following Lemma 7, 8, and 9, we can bound the last expression by

$$\mathcal{O}\left(\frac{\|w_*^g\|^2}{\eta} + \sum_{i=1}^{P}\frac{\|w_{i,*}\|^2}{\eta_i} + \eta T\sum_{i=1}^{P}\sigma_i^2 + T\sum_{i=1}^{P}\eta_i\sigma_i^2 + \gamma\eta^2P^3G^2T\tau^2 + \gamma G^2T\sum_{i=1}^{P}\eta_i^2\tau^2 + PDG\tau\right).$$

□

The following two lemmas deal with two unprocessed terms in the proof of Theorem 2.

**Lemma 7.**

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{P}(w_{t-\tau}^g - w_*^g)\cdot\nabla^g L_i(w_{t-\tau-1}^g, w_{i,t-1})\right] \\
&\leq \frac{\|w_*^g\|^2}{2\eta} + \eta T\sum_{i=1}^{P}\sigma_i^2 + \mathbb{E}\left[\sum_{i=1}^{P}\sum_{t=1}^{T}(w_{t-\tau}^g - w_{t+\tau-1}^g)\cdot\nabla^g L_i(w_{t-\tau-1}^g, w_{i,t-1})\right] + \mathcal{O}\left(PDG\tau\right).
\end{aligned}$$

*Proof.*

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{i=1}^{P}(w_{t-\tau}^g - w_*^g)\cdot\nabla^g L_i(w_{t-\tau-1}^g, w_{i,t-1}) \\
&= \sum_{t=1}^{T}(w_t^g - w_*^g)\cdot\underbrace{\sum_{i=1}^{P}\nabla^g L_i(w_{t-1}^g, w_{i,t+\tau-1})}_{a_t} + \mathcal{O}\left(PDG\tau\right) \\
&= \sum_{t=1}^{T}(w_t^g - w_*^g)\cdot\underbrace{\left(\sum_{i=1}^{P}\nabla_{i,t-\tau}^g\right)}_{b_t} + \sum_{t=1}^{T}(w_t^g - w_*^g)\cdot(a_t - b_t) + \mathcal{O}\left(PDG\tau\right) \\
&\le \sum_{t=1}^{T}\frac{\|w_*^g - w_{t-1}^g\|^2 - \|w_*^g - w_t^g\|^2 - \|w_{t-1}^g - w_t^g\|^2}{2\eta} + \underbrace{\sum_{t=1}^{T}(w_t^g - w_*^g)\cdot(a_t - b_t)}_{\textbf{term}_1} + \mathcal{O}\left(PDG\tau\right).
\end{aligned} \tag{21}$$

Note that $b_t$ is the gradient that is used to update the global model from $w_{t-1}^g$ to $w_t^g$ (Eq.(5)). Therefore using Lemma 10 we have the last equality.

We continue to bound $\textbf{term}_1$. We use $c_t$ to denote the expectation of $b_t$ conditioned on all examples that reach the server before time $t$. That is,

$$\begin{aligned}
c_t &= \mathbb{E}\Big[b_t \;\Big|\; \ell_{i,s} : s < t-\tau\Big] \\
&= \mathbb{E}\left[\sum_{i=1}^{P}\Delta_{i,t-\tau}^g \;\Big|\; \ell_{i,s} : s < t-\tau\right] \\
&= \sum_{i=1}^{P}\mathbb{E}\left[\nabla^g \ell_{i,t-\tau}(w_{t-2\tau}^g, w_{i,t-\tau}) \;\Big|\; \ell_{i,s} : s < t-\tau\right] \\
&= \sum_{i=1}^{P}\nabla^g L_i(w_{t-2\tau}^g, w_{i,t-\tau}).
\end{aligned} \tag{22}$$

The last equality comes from the fact that $w_{t-2\tau}^g$ and $w_{i,t-\tau}$ only depend on $\ell_{i,s}$ with $s < t-\tau$ (see update rules (4), (5)). Then we can decompose $\textbf{term}_1$ as follows:

$$\begin{aligned}
\textbf{term}_1 &= \sum_{t=1}^{T}(w_t^g - w_*^g)\cdot(a_t - b_t) \\
&= \sum_{t=1}^{T}(w_t^g - w_*^g)\cdot(a_t - c_t) + \sum_{t=1}^{T}(w_{t-1}^g - w_*^g)\cdot(c_t - b_t) + \sum_{t=1}^{T}(w_t^g - w_{t-1}^g)\cdot(c_t - b_t).
\end{aligned} \tag{23}$$

Since $w_{t-1}^g$ only depends on $\{\ell_{i,s} : s < t-\tau\}$ (by Algorithm 4), the conditional expectation of the second term in (23) is

$$\mathbb{E}\left[\sum_{t=1}^{T}(w_{t-1}^g - w_*^g)\cdot(c_t - b_t) \;\Big|\; \ell_{i,s} : s < t-\tau\right] = \sum_{t=1}^{T}(w_{t-1}^g - w_*^g)\cdot\mathbb{E}\Big[c_t - b_t \;\Big|\; \ell_{i,s} : s < t-\tau\Big] = 0 \tag{24}$$

by Eq.(22). The third term in (23) can be bounded as

$$\sum_{t=1}^{T}(w_t^g - w_{t-1}^g)\cdot(c_t - b_t) \le \sum_{t=1}^{T}\frac{\|w_t^g - w_{t-1}^g\|^2}{4\eta} + \eta\sum_{t=1}^{T}\|b_t - c_t\|^2. \tag{25}$$

Observe that $\mathbb{E}\left[\|b_t - c_t\|^2 \mid \ell_{i,s} : s < t-\tau\right] = \mathbb{V}\left[b_t \mid \ell_{i,s} : s < t-\tau\right]$. By the independence among the examples from different clients, we can bound

$$\mathbb{E}\left[\sum_{t=1}^{T}\|b_t - c_t\|^2\right] = \sum_{t=1}^{T}\sum_{i=1}^{P}\mathbb{V}[\nabla^g_{i,t-\tau}] \le T\sum_{i=1}^{P}\sigma_i^2. \tag{26}$$

Now we deal with the first term in (23):

$$\begin{aligned}
&\sum_{t=1}^{T}(w_t^g - w_*^g)\left(\sum_{i=1}^{P}\nabla^g L_i(w_{t-1}^g, w_{i,t+\tau-1}) - \sum_{i=1}^{P}\nabla^g L_i(w_{t-2\tau}^g, w_{i,t-\tau})\right) \\
&= \sum_{i=1}^{P}\sum_{t=1}^{T}(w_{t-\tau}^g - w_{t+\tau-1}^g)\cdot\nabla^g L_i(w_{t-\tau-1}^g, w_{i,t-1}) + \mathcal{O}\left(PDG\tau\right). \qquad \text{(re-indexing)}
\end{aligned} \tag{27}$$

Combining Eq.(21)-(27), we see that the right-hand side of (21), after taking expectation, is upper bounded by

$$\begin{aligned}
&\frac{\|w_0^g - w_*^g\|^2}{2\eta} + \sum_{t=1}^{T}\left(\frac{-1}{2\eta} + \frac{1}{4\eta}\right)\mathbb{E}\left[\|w_t^g - w_{t-1}^g\|^2\right] + \eta T\sum_{i=1}^{P}\sigma_i^2 \\
&\quad + \mathbb{E}\left[\sum_{i=1}^{P}\sum_{t=1}^{T}(w_{t-\tau}^g - w_{t+\tau-1}^g)\cdot\nabla^g L_i(w_{t-\tau-1}^g, w_{i,t-1})\right] + \mathcal{O}\left(PDG\tau\right) \\
&\le \frac{\|w_*^g\|^2}{2\eta} + \eta T\sum_{i=1}^{P}\sigma_i^2 + \mathbb{E}\left[\sum_{i=1}^{P}\sum_{t=1}^{T}(w_{t-\tau}^g - w_{t+\tau-1}^g)\cdot\nabla^g L_i(w_{t-\tau-1}^g, w_{i,t-1})\right] + \mathcal{O}\left(PDG\tau\right).
\end{aligned}$$

□

**Lemma 8.**

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,*})\cdot\nabla^l L_i(w_{i,t-\tau-1}^g, w_{i,t-1})\right] \\
&\le \sum_{i=1}^{P}\frac{\|w_{i,*}\|^2}{2\eta_i} + T\sum_{i=1}^{P}\eta_i\sigma_i^2 + \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,t+2\tau-1})\cdot\nabla^\ell L_i(w_{i,t-\tau-1}^g, w_{i,t-1})\right] + \mathcal{O}\left(PDG\tau\right).
\end{aligned}$$

*Proof.* This proof goes through almost the same procedure as in Lemma 7's proof.

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,*})\cdot\underbrace{\nabla^l L_i(w_{i,t-\tau-1}^g, w_{i,t-1})}_{d_{i,t}} \\
&= \sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,*})\cdot\underbrace{\nabla^l_{i,t-2\tau}}_{e_{i,t}} + \sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,*})\cdot(d_{i,t} - e_{i,t}) \\
&\le \sum_{t=1}^{T}\sum_{i=1}^{P}\frac{\|w_{i,t-1} - w_{i,*}\|^2 - \|w_{i,t} - w_{i,*}\|^2 - \|w_{i,t-1} - w_{i,t}\|^2}{2\eta_i} + \underbrace{\sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,*})\cdot(d_{i,t} - e_{i,t})}_{\textbf{term}_2}.
\end{aligned} \tag{28}$$

The last inequality is by Lemma 10 and the fact that $e_{i,t}$ is the gradient that is used to update the local model from $w_{i,t-1}$ to $w_{i,t}$. To bound $\textbf{term}_2$, we define

$$\begin{aligned}
f_{i,t} &= \mathbb{E}[e_{i,t} \mid \ell_{i,s} : s < t-2\tau] \\
&= \mathbb{E}\left[\nabla^l \ell_{i,t-2\tau}(w_{t-3\tau}^g, w_{i,t-2\tau}) \,\middle|\, \ell_{i,s} : s < t-2\tau\right]
\end{aligned}$$

$$= \nabla^l L_i(w^g_{t-3\tau}, w_{i,t-2\tau})$$

because $w^g_{t-3\tau}$ and $w_{i,t-2\tau}$ only depend on $\ell_{i,s}$ with $s < t - 2\tau$. Then we make the following decomposition:

$$\textbf{term}_2 = \sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,*}) \cdot (d_{i,t} - f_{i,t}) + \sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t-1} - w_{i,*}) \cdot (f_{i,t} - e_{i,t}) + \sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,t-1}) \cdot (f_{i,t} - e_{i,t}). \tag{29}$$

The second term in (29) has zero expectation because

$$\begin{aligned}&\mathbb{E}[(w_{i,t-1} - w_{i,*}) \cdot (f_{i,t} - e_{i,t}) \mid \ell_{i,s} : s < t - 2\tau] \\ &= (w_{i,t-1} - w_{i,*}) \cdot \mathbb{E}[(f_{i,t} - e_{i,t}) \mid \ell_{i,s} : s < t - 2\tau] = 0.\end{aligned} \tag{30}$$

The third term in (29) can be upper bounded as

$$\sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,t-1}) \cdot (f_{i,t} - e_{i,t}) \le \sum_{t=1}^{T}\sum_{i=1}^{P}\left(\frac{\|w_{i,t} - w_{i,t-1}\|^2}{4\eta_i} + \eta_i\|f_{i,t} - e_{i,t}\|^2\right), \tag{31}$$

and we note that $\mathbb{E}[\|f_{i,t} - e_{i,t}\|^2 \mid \ell_{i,s} : s < t - 2\tau] = \mathbb{V}[e_{i,t} \mid \ell_{i,s} : s < t - 2\tau]$ is the conditional variance of $e_{i,t}$. Since all samples are independent, we can bound

$$\mathbb{E}\left[\sum_{t=1}^{T}\|f_{i,t} - e_{i,t}\|^2 \,\middle|\, \ell_{i,s} : s < t - 2\tau\right] \le \sum_{t=1}^{T}\mathbb{V}\left[\nabla_{i,t-2\tau} \,\middle|\, \ell_{i,s} : s < t - 2\tau\right] \le T\sigma_i^2.$$

The first term in (29) is

$$\begin{aligned}&\sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,*}) \cdot (d_{i,t} - f_{i,t}) \\ &= \sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,*}) \cdot \left(\nabla^l L_i(w^g_{i,t-\tau-1}, w_{i,t-1}) - \nabla^l L_i(w^g_{t-3\tau}, w_{i,t-2\tau})\right) \\ &= \sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,t+2\tau-1}) \cdot \nabla^\ell L_i(w^g_{i,t-\tau-1}, w_{i,t-1}) + \mathcal{O}\left(PDG\tau\right).\end{aligned}$$

(telescoping and reindexing) (32)

Combining (28)-(32), we get that the left-hand side of (28), after taking expectation, is upper bounded by

$$\begin{aligned}&\sum_{i=1}^{P}\frac{\|w_{i,0} - w_{i,*}\|^2}{2\eta_i} + \sum_{t=1}^{T}\sum_{i=1}^{P}\left(-\frac{1}{2\eta_i} + \frac{1}{4\eta_i}\right)\mathbb{E}[\|w_{i,t} - w_{i,t-1}\|^2] + T\sum_{i=1}^{P}\eta_i\sigma_i^2 \\ &\quad + \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,t+2\tau-1}) \cdot \nabla^\ell L_i(w^g_{i,t-\tau-1}, w_{i,t-1})\right] + \mathcal{O}\left(PDG\tau\right) \\ &\le \sum_{i=1}^{P}\frac{\|w_{i,*}\|^2}{2\eta_i} + T\sum_{i=1}^{P}\eta_i\sigma_i^2 + \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,t+\beta_i+\alpha_i-1}) \cdot \nabla^\ell L_i(w^g_{i,t-\tau-1}, w_{i,t-1})\right] + \mathcal{O}\left(PDG\tau\right).\end{aligned}$$

□

The following lemma further deals with the unprocessed terms in Lemma 7 and Lemma 8.

**Lemma 9.**

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{P}(w^g_{t-\tau} - w^g_{t+\tau-1}) \cdot \nabla^g L_i(w^g_{t-\tau-1}, w_{i,t-1}) + \sum_{t=1}^{T}\sum_{i=1}^{P}(w_{i,t} - w_{i,t+2\tau-1}) \cdot \nabla^l L_i(w^g_{i,t-\tau-1}, w_{i,t-1})\right] \\ &= \mathcal{O}\left(\gamma\eta^2P^3G^2T\tau^2 + \gamma G^2T\sum_{i=1}^{P}\eta_i^2\tau^2 + PDG\tau\right).\end{aligned}$$

*Proof.* Define the joint parameter $u_{i,t} = (w^g_{t-\tau}, w_{i,t})$. Then the left-hand side can be written as

$$\sum_{t=1}^{T}\sum_{i=1}^{P}(u_{i,t} - u_{i,t+2\tau-1}) \cdot \nabla L_i(u_{i,t-1}) = \sum_{t=1}^{T}\sum_{i=1}^{P}(u_{i,t} - u_{i,t+\tau_i-1}) \cdot \nabla L_i(u_{i,t-1}).$$

By Lemma 11, we can bound it by

$$\begin{aligned}\sum_{t=1}^{T}\sum_{i=1}^{P}(u_{i,t} - u_{i,t+\tau-1}) \cdot \nabla L_i(i, u_{t-1}) &\le \sum_{t=1}^{T}\sum_{i=1}^{P}\left(L_i(u_{i,t}) - L_i(u_{i,t+\tau-1}) + \frac{\gamma}{2}\|u_{i,t+\tau_i-1} - u_{i,t}\|^2\right) \\ &= \frac{\gamma}{2}\sum_{t=1}^{T}\sum_{i=1}^{P}\|u_{i,t+\tau-1} - u_{i,t}\|^2 + \mathcal{O}\left(PDG\tau\right).\end{aligned}$$

By our update rule, we have $\|w^g_{t+\tau-1} - w^g_t\|^2 \le (\eta\tau PG)^2$ (from $t$ to $t+\tau-1$, there are $\tau P$ gradient updates for $w^g$) and $\|w_{i,t+\tau-1} - w_{i,t}\|^2 \le (\eta_i\tau G)^2$. Combining them finishes the proof. □

**Lemma 10.** *Let $w' = \Pi_\Omega(w - \eta g)$, where $\Pi_\Omega : \mathbb{R}^d \to \mathbb{R}^d$ is the projection operator that projects the input vector to the convex set $\Omega \subset \mathbb{R}^d$, and $w \in \Omega$, $g \in \mathbb{R}^d$, $\eta > 0$. Then we have for any $w_* \in \Omega$,*

$$(w' - w_*) \cdot g \le \frac{\|w - w_*\|^2 - \|w' - w_*\|^2 - \|w' - w\|^2}{2\eta}.$$

*Proof.* By the definition of $w'$, it is the minimizer of $\|w' - w + \eta g\|^2$ over $\Omega$. Therefore, by the first-order optimality condition, we have for any $w_* \in \Omega$,

$$(w' - w + \eta g) \cdot (w' - w_*) \le 0.$$

Rearranging it we get

$$(w' - w_*) \cdot g \le \frac{(w - w') \cdot (w' - w_*)}{\eta} = \frac{\|w - w_*\|^2 - \|w' - w_*\|^2 - \|w' - w\|^2}{2\eta},$$

where the last equality can be obtained by direct expansion. □

**Lemma 11.** *For any $\gamma$-smooth convex function $f$, and any $a, b, c$,*

$$f(a) - f(b) \le (a - b) \cdot \nabla f(c) + \frac{\gamma}{2}\|a - c\|^2.$$

*Proof.* By the convexity and the $\gamma$-smoothness of $f$, we have

$$\begin{aligned}f(c) - f(b) &\le (c - b) \cdot \nabla f(c), \\ f(a) - f(c) &\le (a - c) \cdot \nabla f(c) + \frac{\gamma}{2}\|a - c\|^2.\end{aligned}$$

Adding up two inequalities we get the desired inequality. □

# C The Failure of the Fictitious-Play Variant of the ERM Algorithm

In this section, we experimentally compare Glocal.SGD (Algorithm 3, 4), Glocal.ERM (Algorithm 1, 2), and the *fictitious play* variant of the ERM algorithm that we describe at the end of Section 3.1. The goal is to show that the last one may take significantly more rounds to converge.

## C.1 Data Generation

Suppose there is only one client. The feature dimensions are 2 for both global and local features. The feature vectors $(x^g_t, x^l_t)$ and the label $y_t$ are generated i.i.d. according to

$$\begin{aligned}a_t &\sim \mathcal{N}(0, 1) \\ b_t &\sim \mathcal{N}(0, 1) \\ x^g_t &= \begin{bmatrix} a_t + \epsilon_t \\ b_t \end{bmatrix} \qquad \text{where } \epsilon_t \sim \mathcal{N}(0, 0.25)\end{aligned}$$

$$x_t^l = \begin{bmatrix} 1 - a_t \\ 1 - b_t \end{bmatrix}$$
$$y_t = 1$$

The loss is defined as $\ell_t(w^g, w^l) = (y_t - w^g \cdot x_t^g - w^l \cdot x_t^\ell)^2$. Clearly, the best pair of regressors is $w_*^g = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $w_*^l = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, and this pair gives zero average loss. We run three algorithms for $T = 20000$ steps.

### C.2 Algorithms

We let the parameters be initialized as $w_1^g = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, w_1^l = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. Then the goal of the algorithms is to adjust both $w_t^g$ and $w_t^l$ from $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ to $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ since the latter is the optimal solution.

Assume no delays. Then the three algorithms we compare can be simplified as in Algorithm 5, 6, 7. The main difference between Algorithm 6 and 7 is that in the former, the server (client) re-applies the new parameters from the client (server) to the old samples, but the latter does not. As we mentioned in Section 3.1, in terms of computational and communication efficiency, Algorithm 7 is actually preferred over Algorithm 6.

**Algorithm 5** Glocal.SGD

Let $\eta = 1.0$ (an arbitrary choice).
**for** $t = 1, \ldots, T$ **do**
  Suffer loss $\ell_t(w_t^g, w_t^l)$ and make updates:

$$w_{t+1}^g = w_t^g - \eta \nabla^g \ell_t(w_t^g, w_t^l)$$
$$w_{t+1}^l = w_t^l - \eta \nabla^l \ell_t(w_t^g, w_t^l)$$

**Algorithm 6** Glocal.ERM

**for** $t = 1, \ldots, T$ **do**
  Suffer loss $\ell_t(w_t^g, w_t^l)$ and make updates:

$$w_{t+1}^g = \operatorname*{argmin}_{w^g} \sum_{s=1}^{t} \ell_s(w^g, w_t^l)$$
$$w_{t+1}^l = \operatorname*{argmin}_{w^l} \sum_{s=1}^{t} \ell_s(w_t^g, w^l)$$

**Algorithm 7** Fictitious Play

**for** $t = 1, \ldots, T$ **do**
  Suffer loss $\ell_t(w_t^g, w_t^l)$ and make updates:

$$w_{t+1}^g = \operatorname*{argmin}_{w^g} \sum_{s=1}^{t} \ell_s(w^g, w_s^l)$$
$$w_{t+1}^l = \operatorname*{argmin}_{w^l} \sum_{s=1}^{t} \ell_s(w_s^g, w^l)$$

### C.3 Comparing the performance

We compare the average loss performances of the three algorithms over time, and observe that the Fictitious-play strategy is highly sub-optimal (Figure 3). All plots in this section are an average over 50 random rollouts.

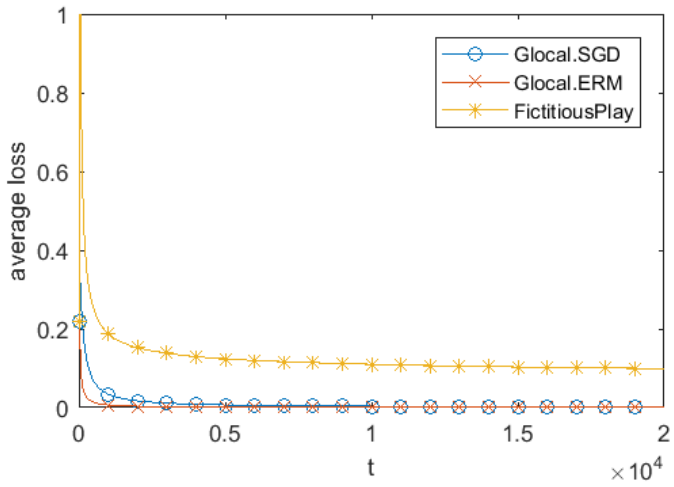



Figure 3: Comparing the average loss performance among Glocal.SGD (Algorithm 5), Glocal.ERM (Algorithm 6) and the fictitious-play strategy (Algorithm 7)

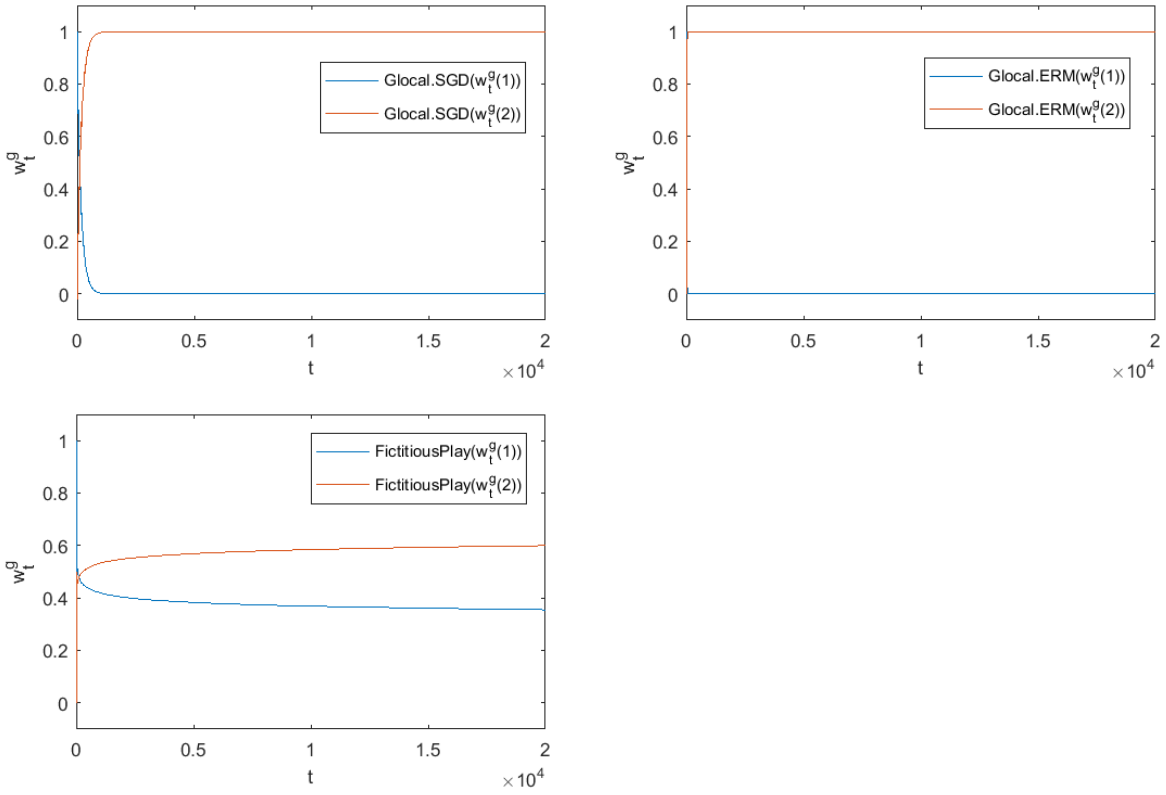



Figure 4: The change of $w_t^g$ over time (better viewed with color). Each sub-figure is for one algorithm. The blue lines plot the first component of $w_t^g$, which is initialized as 1 and the learner should adjust it to 0; the red lines plot the second component of $w_t^g$, which is initialized as 0 and should be adjusted to 1.

Recall that the goal of the algorithms is to change both $w_t^g$ and $w_t^l$ from $\begin{bmatrix}1\\0\end{bmatrix}$ to $\begin{bmatrix}0\\1\end{bmatrix}$. We plot the changes of the components of $w_t^g$ over time for three algorithms in Figure 4. From Figure 4 we see that while Glocal.SGD and Glocal.ERM can quickly find the optimal solutions, the fictitious-play strategy gets stuck before reaching the optimum. Our explanation for this phenomenon is below. Observe that by our construction of $(x_t^g, x_t^l, y_t)$, if $w_t^g$ is of the form $\begin{bmatrix}z\\1-z\end{bmatrix}$ (e.g., in the beginning, $z$ is 1), then it creates a loss for $w_t^l$ as

$$\left(1 - \begin{bmatrix}z\\1-z\end{bmatrix} \cdot \begin{bmatrix}a_t\\b_t\end{bmatrix} - \begin{bmatrix}w_t^l(1)\\w_t^l(2)\end{bmatrix} \cdot \begin{bmatrix}1-a_t\\1-b_t\end{bmatrix}\right)^2 = \Big((1 - w_t^l(1) - w_t^l(2)) + a_t(w_t^l(1) - z) + b_t(w_t^l(2) - 1 + z)\Big)^2,$$

whose expectation is minimized when $w_t^l(1) = z$ and $w_t^l(2) = 1 - z$; that is, when $w_t^l = w_t^g$, the expected loss is minimized. Similarly, when $w_t^l$ is fixed, the expected loss minimizer for $w_t^g$ is $w_t^l$. Since the fictitious-play strategy memorizes all previous losses under the outdated parameters, $w_t^g$ tends to be close to the average of $w_s^l$'s with $s < t$; similarly, $w_t^l$ tends to be close to the average of previous $w_s^g$'s. Therefore, the server and the client tend to lock each other, and this makes their updates very slow, which results in the learning curve of the fictitious-play strategy that we observe in Figure 4.

## D Reducing the communication through mini-batches

Our algorithms have heavy communication since the clients fetch a new global model each round. This communication cost can be reduced by using mini-batches where both the clients and the server

update their models once per batch. This can thus largely reduce the downlink communication because the client only needs to fetch the global model once per batch. The analysis in this section is inspired by the work of Dekel et al. [7].

To analyze the algorithm with mini-batches, we can reuse our theorems developed in the previous sections. For example, in the Glocal.SGD algorithm, if we use mini-batches of size $b$, we can define the aggregated loss

$$\widehat{\ell}_{i,n}(w^g, w_i) = \frac{1}{b} \sum_{t=(n-1)b+1}^{nb} \ell_{i,t}(w^g, w_i), \tag{33}$$

and run Glocal.SGD for rounds $n = 1, \ldots, \frac{T}{b}$. In the original algorithm, the clients accesses the global model $T$ times, but in the mini-batched algorithm, the clients only accesses $\frac{T}{b}$ times. We can also reuse Theorem 2 to analyze the regret of the batched algorithm. Applying Theorem 2 to the aggregated loss sequence defined in (33), we get

$$\begin{aligned}
&\mathbb{E}\left[\frac{b}{PT}\sum_{i=1}^{P}\sum_{n=1}^{T/b}\widehat{\ell}_{i,n}(w^g_{n-\tau'}, w_{i,n}) - \widehat{\ell}_{i,n}(w^*_g, w_{i,*})\right] \\
&= \mathcal{O}\left(\sqrt{\frac{\left(\|w^g_*\|^2 + \sum_{i=1}^{P}\|w_{i,*}\|^2\right)\sigma'^2 b}{PT}} + \frac{(\gamma D^4 G^2 \tau'^2)^{\frac{1}{3}} b^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{DG\tau' b}{T}\right)
\end{aligned}$$

where $\tau' = \frac{\tau}{b} + 1$ is the delay counted in batches and $\sigma'^2 = \frac{\sigma^2}{b}$ is the variance of the $\widehat{\ell}_{i,t}(w^g, w_i)$. The left-hand side turns out to be the true average loss of the learner, and the right-hand side is

$$\mathcal{O}\left(\sqrt{\frac{\left(\|w^g_*\|^2 + \sum_{i=1}^{P}\|w_{i,*}\|^2\right)\sigma^2}{PT}} + \frac{\left(\gamma D^4 G^2 (b+\tau)^2\right)^{\frac{1}{3}}}{T^{\frac{2}{3}}} + \frac{DG(b+\tau)}{T}\right).$$

As one can see, the dominant term remains the same order, and the lower-order term is unaffected if $b < \tau$.

# E More Experimental Results

## E.1 Details of Generating LIBSVM datasets

We create binary classification from real multiclass classification datasets provided in LIBSVM Dataset [4] as follows:

- For a multiclass classification dataset with the set of classes being $[K] = \{1, 2, \ldots, K\}$, we randomly pick a subset $\mathcal{A}$ of it. All data samples from $\mathcal{A}$ are merged as a new class $C_0$.
- For each client, its assigned task is a binary classification problem between class $C_0$ and a random class from $[K]\backslash\mathcal{A}$.

As can be seen, different clients face different classification problems which might be related: Suppose Client 1's task is to distinguish $C_0$ from class $A$; Client 2's task is to distinguish $C_0$ from class $B$. When there exists a single hyperplane that saperates $C_0$ from $A$ and $B$ well, then the two clients' task are closely related, although this is not guaranteed in the datasets we generate.

We then assign data to workers so that the following two properties are satisfied:

1. Different clients may work on the same task (i.e., the same random class from $[K]\backslash\mathcal{A}$), but the examples they are assigned to are guaranteed to be disjoint.
2. The positive and negative examples assigned to each client are equal.

In order to let the property 1 above hold, each client is assigned at most $\frac{\text{\# examples belonging to } \mathcal{A}}{\text{\# clients}}$ data samples. In order to make this large enough for experimental purpose, $|\mathcal{A}|$ should not be too small;

on the other hand, in order to keep the task diversity of the clients, $K - |\mathcal{A}|$ should also not be too small. We simply make a balanced choice of $|\mathcal{A}| = \lfloor 0.3K \rfloor$.

In order to satisfy the two properties, we distribute the data to clients following the procedures below:

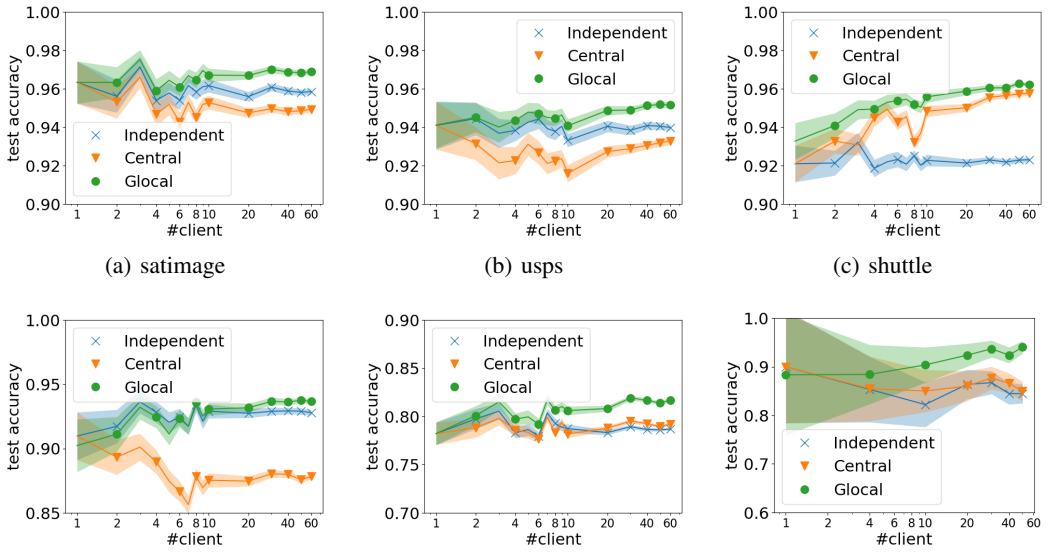
1. Uniformly randomly distribute the samples of $\mathcal{A}$ to all clients, making each client receive $M$ samples.
2. Create *buckets* of data samples from $[K] \backslash \mathcal{A}$. Each bucket contains $M$ single-class samples.
3. Each client is randomly assigned a bucket.

At the end, each client has $2M$ samples with balanced classes.

In order to maintain the diversity of tasks, we pick from LIBSVM multiclass classification datasets that have no less than 6 classes, and have enough samples that allow us to perform the above procedure.

### E.2 The effect of the number of clients

In Figure 1, we have shown the effect of the number of workers for three of the datasets we test on. In the Figure 5, we provide the plots for other datasets that we use. We do not include the result for Isolet because the original Isolet dataset is naturally partitioned into 5 subsets (corresponding to 5 clients in our experiments), and it is not clear how to further divide each client's data into multiple parts in a reasonable way.



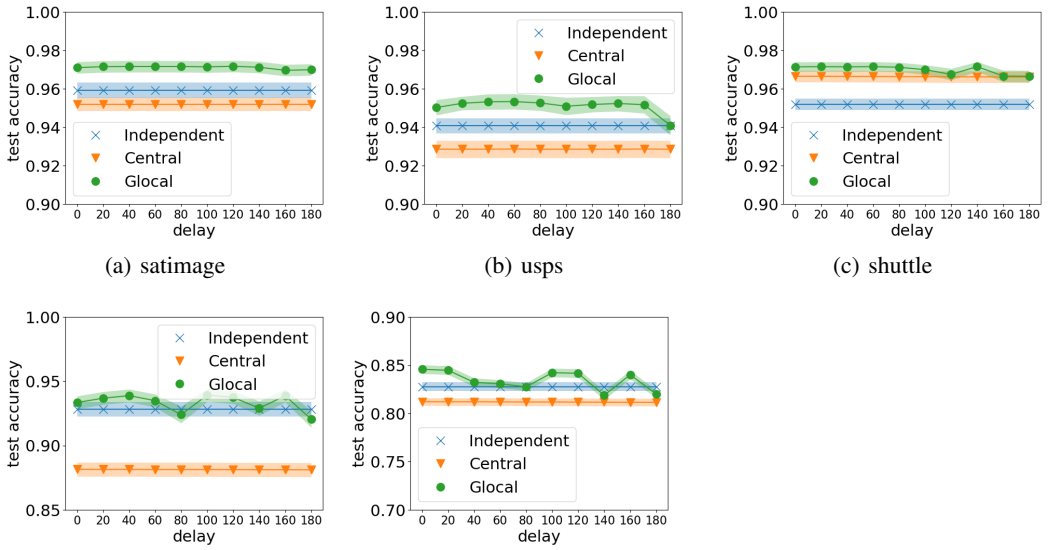

(a) satimage (b) usps (c) shuttle

(d) pendigits (e) sensorless (f) femnist

Figure 5: Test accuracy versus the numbers of clients. In these experiments, we let the delay be zero. Each data point is an average over 10 random trials.

### E.3 The effect of delay

We complement the results in Figure 2 with other datasets that we use. Figure 6 shows the relation between test accuracy and communication delay, verifying the robustness of Glocal under a wide range of delays. In the experiments, we fix the number of clients as 20 (as seen in Figure 5, the performance is relatively stable when #client $\geq 20$).



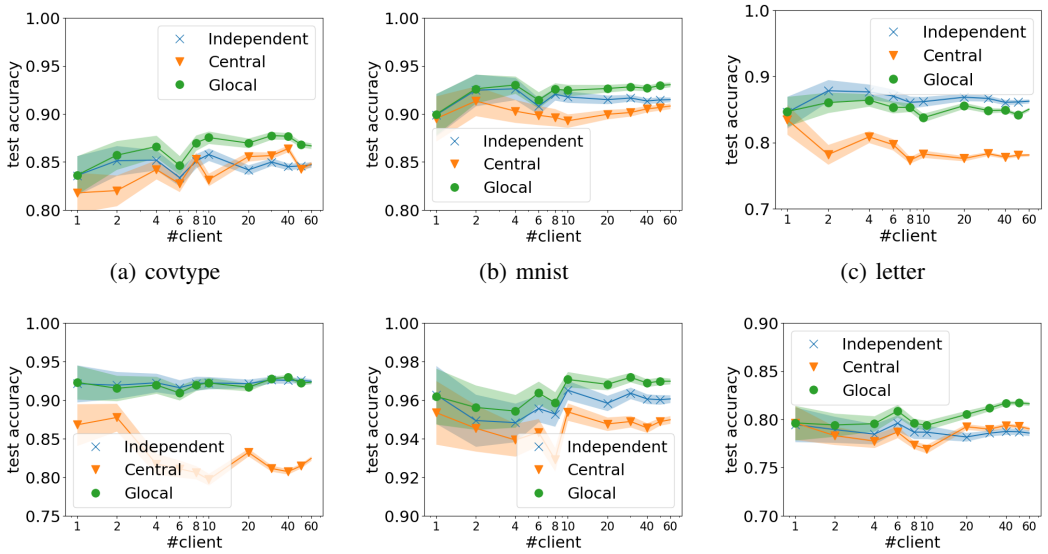

(a) satimage (b) usps (c) shuttle

(d) pendigits (e) sensorless

Figure 6: Test accuracy versus delay for satimage, usps, shuttle, pendigits, sensorless. We let the number of clients be 20. Each data point is an average over 10 random trials.

### E.4 The effect of feature splitting

In this subsection, we demonstrate that Glocal also outperforms baselines under *feature splitting*. As explained in Example 1, Glocal applies to both the case where the global model and local model share same features, and the case where they use separate features. *Feature splitting* refers to the latter. Feature splitting can be applied to the scenario when a set of features is only relevant to specific clients, or when the clients do not want to share some specific features due to privacy consideration. In the experiments in this subsection, we test the performance of feature splitting on LIBSVM datasets. We randomly divide the features into *global features* and *local features*. The three algorithms that we implement are modified as follows: the **Independent** algorithm learns both on global and local features; the **Central** algorithm only uses global features (because local features are not shared with the server); for the **Glocal** algorithm, the global model uses global features and the local model uses the local features. The results under different number of clients are shown in Figure 7.




(a) covtype (b) mnist (c) letter

(d) pendigits (e) satimage (f) sensorless

Figure 7: Test accuracy with feature splitting. In these experiments, we let the delay be zero. Each data point is an average over 10 random trials.

Different from the setting where the whole set of features is always use, in the feature splitting case, the Central algorithm has an intrinsic disadvantage because of the lack of local features. In the experiments in Figure 7, we show an phenomenon that when the global features are relatively weak (e.g. in letter, pendigits), Glocal still performs close to the Independent algorithm; on the other hand, when global features are relatively strong, Glocal get the advantage of joint training and outperforms both.