FLIX: A Simple and Communication-Efficient Alternative to Local Methods in Federated Learning

Elnur Gasanov
KAUST, Thuwal
Saudi Arabia

Ahmed Khaled
Cairo University
Egypt

Samuel Horváth
KAUST, Thuwal
Saudi Arabia

Peter Richtárik
KAUST, Thuwal
Saudi Arabia

Abstract

Federated Learning (FL) is an increasingly popular machine learning paradigm in which multiple nodes try to collaboratively learn under privacy, communication and multiple heterogeneity constraints. A persistent problem in federated learning is that it is not clear what the optimization objective should be: the standard average risk minimization of supervised learning is inadequate in handling several major constraints specific to federated learning, such as communication adaptivity and personalization control. We identify several key desiderata in frameworks for federated learning and introduce a new framework, FLIX, that takes into account the unique challenges brought by federated learning. FLIX has a standard finite-sum form, which enables practitioners to tap into the immense wealth of existing (potentially non-local) methods for distributed optimization. Through a smart initialization that does not require any communication, FLIX does not require the use of local steps but is still provably capable of performing dissimilarity regularization on par with local methods. We give several algorithms for solving the FLIX formulation efficiently under communication constraints. Finally, we corroborate our theoretical results with extensive experimentation.

1 Introduction

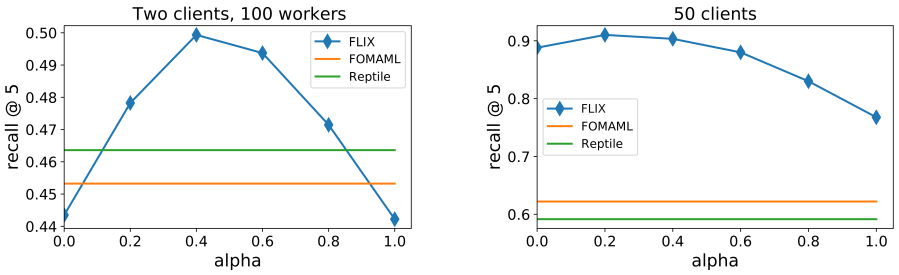
Federated Learning (FL) aims to enable machine learning in the decentralized setting while respecting data privacy. Application domains of federated learning include healthcare, learning language models for virtual keyboards, and speech recognition (Kairouz et al., 2019). The promise of federated learning is that by participating in a distributed training process, clients can learn better machine learning models than they can using only their own data. The main cost in using federated learning over local training lies in the network bandwidth used for the distributed training process. Hence, federated learning must be flexible enough to provide a benefit to users without a prohibitive communication cost. The standard formulation of FL is to cast it as an optimization problem of the form

$$\min_{x\in\mathbb{R}^d}\left[f(x)\stackrel{\text{def}}{=}\frac{1}{n}\sum_{i=1}^{n}f_i(x)\right], \tag{ERM}$$

where f_i is the loss function on client i. Thus, the goal of classical FL is for the n clients to collaboratively learn a single model, $x^*=\arg\min f$, to be deployed on all clients. Recent development shows that using a single model for all clients can be severely detrimental to individual performance on many clients (Yu et al., 2020), defeating the purpose of joining distributed training. Furthermore, (ERM) offers no clear tunable knobs that can accommodate constraints on the network bandwidth. *Can we find a formulation for federated learning that is flexible enough to accommodate the needs of federated learning, yet also solvable using standard methods?*

* The full paper is available at https://arxiv.org/pdf/2111.11556.pdf

1st NeurIPS Workshop on New Frontiers in Federated Learning (NFFL 2021), Virtual Meeting.



(a) 100 workers created out of two clients' data (b) 50 workers with distinct data distributions

Figure 1: Test accuracy of FLIX model for different personalization parameter values, FOMAML and Reptile. $\alpha_i = \alpha$ is set to the value indicated on horizontal axis. FOMAML and Reptile are independent from the personalization parameter α. Plots correspond to different data splittings.

1.1 Key properties of the FLIX framework

Our main contribution is *FLIX*, a novel and flexible formulation for federated learning: define $\alpha_i > 0$ to be the *personalization parameter for node* i, and let $x_i \overset{\text{def}}{=} \min_{x \in \mathbb{R}^d} f_i(x)$ be the local solution to the i-th objective– note that x_i can be found by solely running a local optimizer, and hence computing it requires no communication at all. The FLIX problem is

$$\min_{x \in \mathbb{R}^d} \tilde{f}(x) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(\alpha_i x + (1 - \alpha_i)\, x_i). \tag{FLIX}$$

Once we find a solution x_* of sufficient quality for (FLIX), we deploy $T_i(x_*) = \alpha_i x_* + (1 - \alpha_i)\, x_i$ on node i as its final personalized model. We now enumerate some of the key properties of (FLIX):

- **Efficiently solvable as a finite-sum problem.** (FLIX) preserves the standard finite-sum formulation of empirical risk minimization. Moreover, it preserves problem structure: we show (in Section 3) that when the f_i are smooth (and/or convex), $\tilde{f}$ is also smooth (resp. convex).
- **Adaptive to communication constraints.** Communication efficiency is an important concern in federated learning, as often bandwidth is valuable and limited (Kairouz et al., 2019; Konečný et al., 2016; Li et al., 2019). Computing x_i, a precondition to solving (FLIX), requires no communication at all, and can be done purely locally on node i. If $\alpha_i = 0$, then no communication at all is needed to compute the personalized model $T_i(x_*)$. By varying α_i between 0 and 1, we can control the amount of communication needed to compute $T_i(x_*)$. We show (in Section 3) that given a communication budget of R steps, we can find parameters α_i that allows us to solve (FLIX) in no more than R communication steps.
- **Adaptivity to personalization.** Our end-goal in federated learning is to generalize well *on each client*: this means that the solution deployed on node i should be tailored to its local data distribution, which may differ from the data distributions on other nodes. In FLIX, varying α_i enables us to amplify or reduce the effect of other objectives on the solution deployed on node i. In situations where the data on all of the nodes is sufficiently heterogeneous, we set α_i to be small and the effect of other data on node i will be neglibile. On the other hand, when the data on the different nodes is related we may set α_i to be closer to 1. We observe a benefit to varying α in this manner in practice: Figure 1 shows the effect of varying the α_i on real data (see Section 4 for the details and for other experiments).

FLIX fills a gap that is unsatisfied by existing methods. To the best of our knowledge, there is no other method for federated learning that is efficiently solvable via standard algorithms and also adaptive to communication and personalization constraints, and indeed both constraints are important in practice (Li et al., 2020). We believe the key properties we enumerate can also serve as natural desiderata in the development of new formulations and methods for federated learning.

1.2 Related work

Personalization has garnered significant recent interest in federated learning as personalized models often perform well in practice compared to non-personalized models (Jiang et al., 2019; Yu et al., 2020). FLIX is a *model mixture* method: the personalized solution is a mixture of a global model and a local model. In recent work, Deng et al. (2020) and Mansour et al. (2020) propose model mixture methods and prove their statistical benefits, while Zec et al. (2021) introduce a similar formulation based on the mixture of experts framework. Unfortunately, we show in the supplementary material that, from the perspective of optimization and without additional data, the formulations in all three works are trivially minimized at the local minimizers $x_1, \ldots, x_n$. An alternative to model mixing is mixing in function space, where we optimize a mixture of objectives rather than a model mixture. This mixture is often constructed to control model variance: examples of this approach can be found in (Dinh et al., 2021a;b; Hanzely and Richtárik, 2020; Huang et al., 2021). In FLIX, we take the model mixture approach as it allows us to use pretraining to better solve the problem while still regularizing model variance (see Section 2.1). A parallel line of work applies meta-learning methods like MAML to federated learning (Fallah et al., 2020a; Jiang et al., 2019): in Section 2.2 we motivate FLIX by taking MAML as our starting point. Chen et al. (2021) discuss the statistical limits of personalization and show that either solving empirical risk minimization or local training is optimal, depending on certain problem parameters; However, as of yet there is no single optimal adaptive algorithm (from the statistical perspective). There are several other techniques in federated learning that can be combined with our approach for better results, such as clustering (Sattler et al., 2020) or robust optimization (Reisizadeh et al., 2020).

2 The FLIX formulation

In this section we reintroduce and motivate the FLIX formulation in detail. We define the FLIX objective as

$$\tilde{f}(x; \alpha_1, \ldots, \alpha_n, x_1, \ldots, x_n) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(\alpha_i x + (1-\alpha_i) x_i), \tag{1}$$

where $\alpha_i \in (0,1)$ is the personalization coefficient for node i and x_i is the minimizer of f_i, for all $i = 1, 2, \ldots, n$. We will use $\tilde{f}(x)$ to refer to the objective in (1) when the α_i and x_i are clear from the context. The FLIX problem is then

$$\min_{x \in \mathbb{R}^d} \left[\tilde{f}(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(\alpha_i x + (1-\alpha_i) x_i) \right]. \tag{2}$$

Let $\alpha = [\alpha_1, \ldots, \alpha_n]$ be the vector of the personalization coefficients. If $x_* = x_*(\alpha)$ is a solution of (2), we call $T_i(x; \alpha_i, x_i) = \alpha_i x_* + (1-\alpha_i) x_i$ the *deployed solution on node* i. Like with $\tilde{f}$, we will refer to the deployed solution on node i as $T_i(x)$ when x_i and α_i are clear from the context.

2.1 Motivation 1: from Local GD to FLIX

The most popular algorithm for solving federated learning problems is the Federated Averaging algorithm (Kairouz et al., 2019), also known as Local (Stochastic) Gradient Descent (Local GD/SGD). Local GD alternates steps of local computation on each node with steps of communication and aggregation. More concretely, the Local GD update is:

$$x_{t+1}^i = \begin{cases} x_t^i - \gamma \nabla f_i(x_t^i) & \text{if } t \mod H \neq 0 \\ \frac{1}{n} \sum_{i=1}^n \left[x_t^i - \gamma \nabla f_i(x_t^i) \right] & \text{if } t \mod H = 0 \end{cases}, \tag{3}$$

where H is the number of local steps. Early papers on federated learning (such as e.g. (Konečný et al., 2016)) motivated local methods as communication-efficient ways of solving (ERM), but subsequent theoretical development reveals that local methods are, in fact, quite bad solvers for (ERM) whenever there is significant statistical heterogeneity among the clients (Woodworth et al., 2020). Moreover, Pathak and Wainwright (2020) show that for the linear least-squares problem, Local GD converges to a different point than the minimizer of (ERM). More generally, the fixed points of Algorithm (3) can be very different from the minimizer of (ERM) whenever $H > 1$ (Malinovskiy et al., 2020). Hanzely

and Richtárik (2020) show that a mild variant of Local GD can be interpreted as SGD applied on the nd-dimensional regularized objective f_λ defined by

$$f_\lambda(y_1, y_2, \ldots, y_n) \stackrel{\text{def}}{=} \left[\frac{1}{n} \sum_{i=1}^{n} f_i(y_i) + \frac{\lambda}{2n} \sum_{i=1}^{n} \|y_i - \bar{y}\|^2 \right], \tag{4}$$

where $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$ is the counterpart of x in (ERM), and where λ is a regularization parameter determined according to the number of local steps. Objective f_λ is the summation of two terms: the first asks that each node i finds a solution y_i that minimizes its local objective well, while the regularizer $\psi(y_1, \ldots, y_n) = \frac{1}{2n} \sum_{i=1}^{n} \|y_i - \bar{y}\|^2$ forces the solutions $y_1, y_2, \ldots, y_n$ to be close to their average $\bar{y}$. Hence, Local GD incentivizes finding *personalized* solutions $y_1, y_2, \ldots, y_n$ that have small population variance. Hanzely and Richtárik (2020) note that as the λ parameter varies between 0 and ∞, the solutions found by Local GD interpolate between the pure local optimal models (i.e. $x_i = \operatorname{argmin}_x f_i(x)$) and the solution of the global problem x_* (the minimizer of (ERM)). We observe that the solutions $y_1, \ldots, y_n$ found by Local GD are an *implicit mixture* of the local minimizers $x_1, \ldots, x_n$ and the global empirical risk minimizer x_*. Rather than seeking an implicit mixture of the local and global optimal models, we instead propose to find an *explicit* mixture of the local optimal models and a global model: given any global model x (not necessarily the empirical risk minimizer), we choose coefficients $\alpha_1, \alpha_2, \ldots, \alpha_n$ (all between 0 and 1) and then deploy on node i the mixture

$$T_i(x) = \alpha_i x + (1 - \alpha_i) x_i, \tag{5}$$

we may then choose x as *the best* such global model by explicitly solving the optimization problem $\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i\left(\alpha_i x + (1 - \alpha_i) x_i\right)$, and this is exactly the FLIX formulation. Observe that coefficients $\alpha_1, \alpha_2, \ldots, \alpha_n$ should regularize the population variance of the deployed solutions $T_1(x), T_2(x), \ldots, T_n(x)$, as in local methods. We show this rigorously for equal α_i in Proposition 3. Our development thus leads us to a natural framework that captures the strength of local methods while also satisfying the desiderata specified in Section 1.1.

2.2 Motivation 2: from model-agnostic meta-learning to FLIX

We now motivate FLIX differently by starting with *personalization via fine-tuning*. The ordinary formulation of the federated learning problem (ERM) asks for a single global model to be used on all clients. If the clients are sufficiently heterogeneous, a single model may perform badly on many of them (Jiang et al., 2019). Personalizing a global model to each of the users' custom data is often beneficial in practice; For example, Wang et al. (2019) study the benefits of personalizing language models for a virtual keyboard application used by tens of millions of users. They observe that a sizeable fraction of the users benefit from personalization. Personalization is often done in two steps:

Step I: initial model training. Find a "good" global model x_{global}.
Step II: fine-tuning. Personalize the global model x_{global} on each client to get the *personalized* local models x_i.

Methods that fit this framework are known as *finite-tuning approaches*: they include the model-agnostic meta-learning (MAML) family of methods (Finn et al., 2017). In addition to its practical popularity, recent theoretical investigations reveal that fine-tuning approaches, such as MAML, are also benefical from a statistical perspective (Chua et al., 2021; Fallah et al., 2021). In MAML, we find x_{global} by optimizing for the loss after a single step of gradient descent, i.e. the MAML objective is

$$\text{Find} \quad x_{\text{global}} \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(x - \gamma \nabla f_i(x)), \tag{6}$$

where γ is a given stepsize. Once x_{global} is found, we may then fine-tune it by running gradient descent for a number of steps on each node i locally using its own objective f_i (Finn et al., 2017). To gain further insight into what fine-tuning is doing, we now consider the case when each f_i is a quadratic function. Because this problem is amenable to analysis, several authors have used it to study the theoretical properties of MAML (Charles and Konečný, 2021; Collins et al., 2020; Gao and Sener, 2020), and we follow in their footsteps. Assume that each f_i is a quadratic function, suppose that we have some initial global model x^0, and we fine-tune it by running gradient descent for H steps on node i: the next proposition shows the final iterate is a matrix-weighted average of the initial solution and the optimal local solution:

Proposition 1. *Suppose that we run gradient descent for H steps on the quadratic objective $f_i = \frac{1}{2}x^T A_i x - b_i^T x + c$ starting from x^0 with stepsize $\gamma > 0$. Suppose that the stepsize satisfies $\gamma \leq \frac{1}{L_i}$, where $L_i = \lambda_{\max}(A_i)$. Then the final iterate x_i^H can be written as*

$$x_i^H = \left(I - J_i^H\right) x_i + J_i^H x^0,$$

where x_i minimizes f_i and $J_i \in \mathbb{R}^{d\times d}$ is a matrix with maximum eigenvalue smaller than 1, i.e. $\lambda_{\max}(J) < 1$.

The proof of Proposition 1 and all subsequent proofs are relegated to the supplementary. Plugging the result of Proposition 1 into Equation (6), observe that in MAML we find the initial model x^0 by solving the problem $\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n f_i((I - J_i)x_i + J_i x)$. Hence, MAML is optimizing *for a specific weighted average* of the initial model x^0 and the local solutions $x_1, x_2, \ldots, x_n$. We thus propose to dispense with the specific matrix J_i and instead optimize an average weighted with an arbitrary constant α_i: $\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n f_i(\alpha_i x + (1-\alpha_i)\, x_i)$, and this is exactly the FLIX formulation. Observe that by properly normalizing or whitening the data and tuning α_i we may accomplish a similar effect to multiplying by J_i^H for any H. This gives FLIX a new interpretation as an approximate generalized MAML, where we optimize the global model for performance after potentially many gradient descent steps rather than just a single step.

3 Theory and algorithms

In this section we aim to develop algorithms to solve (FLIX) in a communication-efficient manner. Before discussing concrete algorithms, we study a few algorithm-independent properties of (FLIX) that will come in handy for understanding the formulation and proving convergence bounds. The following proposition shows that the formulation preserves smoothness and convexity. This is in contrast, for example, to MAML, where the objective may be nonsmooth (Fallah et al., 2020b).

Proposition 2. *Suppose that each objective f_i is L_i-smooth. That is, for any $x, y \in \mathbb{R}^d$ we have $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i \|x - y\|$. Then the FLIX objective $\tilde{f}$ defined in (1) is L_α-smooth for $L_\alpha \overset{def}{=} \frac{1}{n}\sum_{i=1}^n \alpha_i^2 L_i$. If each f_i is convex, then $\tilde{f}$ is also convex. If each f_i is μ_i-strongly convex, then $\tilde{f}$ is μ_α strongly convex for $\mu_\alpha \overset{def}{=} \frac{1}{n}\sum_{i=1}^n \alpha_i^2 \mu_i$.*

Our next result offers some insight into the variance-regularizing effect of the α_i: in particular, when all the α_i are equal, increasing α in (FLIX) directly decreases the variance of the deployed local models from their mean. As discussed in Section 2.1, this is a key property of local descent methods that the FLIX formulation captures.

Proposition 3. *Suppose that $\alpha_1 = \alpha_2 = \ldots = \alpha_n = \beta$ in the FLIX formulation (FLIX). Let $T_1(x), T_2(x), \ldots, T_n(x)$ be the deployed models defined in (5). If $y_1, \ldots, y_n$ are vectors in $\mathbb{R}^d$ and $\bar{y}$ is their mean, we define $V(y_1, \ldots, y_n)$ as the population variance $V(y_1, \ldots, y_n) \overset{def}{=} \frac{1}{n}\sum_{i=1}^n \|y_i - \bar{y}\|^2$. Then, $V(T_1(x), T_2(x), \ldots, T_n(x)) = (1-\beta)^2 V(x_1, x_2, \ldots, x_n)$.*

One-shot learning is a learning paradigm where we may use only a single round of communication to solve the federated learning problem (Guha et al., 2019; Salehkaleybar et al., 2019). When the personalization parameters are small enough, we can provably solve the FLIX problem with a single round of communication by computing a certain weighted average of the local solutions $x_1, x_2, \ldots, x_n$.

Theorem 1. *Suppose that each objective f_i is L_i-smooth, let $\hat{L} \overset{def}{=} \frac{1}{n}\sum_{i=1}^n L_i$. Given the pure local models $x_1, x_2, \ldots, x_n$, define the weighted average $x^{\text{avg}} \overset{def}{=} \sum_{i=1}^n w_i x_i$, $w_i \overset{def}{=} \frac{\alpha_i^2 L_i}{n L_\alpha}$, $L_\alpha \overset{def}{=} \frac{1}{n}\sum_{i=1}^n \alpha_i^2 L_i$. We further define the constants $D \overset{def}{=} \max_{i,j=1,\ldots,n, i\neq j} \|x_i - x_j\|$, and, $V \overset{def}{=} \sum_{i=1}^n w_i \|x_i - x^{\text{avg}}\|^2$. Fix any $\epsilon > 0$. Assume that either $\max_{i=1,\ldots,n} \alpha_i \leq \sqrt{2\epsilon}/\sqrt{\hat{L}D}$, or $\alpha_i = \beta$ for all i and $\beta \leq \sqrt{2\epsilon}/\sqrt{\hat{L}D}$. Then x^{avg} is an ϵ-approximate minimizer of (FLIX).*

For α_i larger than this, we need more communication rounds. In the next subsection, we describe how distributed gradient descent can be used to solve the problem.

3.1 Distributed gradient descent

The simplest approach to solving (FLIX) is via distributed gradient descent (DGD): given the local models $x_1, x_2, \ldots, x_n$ (precomputed before starting the process) and an initial global model x^0, we run the update $x^{k+1} = x^k - \frac{\gamma}{n} \sum_{i=1}^{n} \alpha_i \nabla f_i \left(\alpha_i x^k + (1-\alpha_i)\, x_i\right)$. The next theorem shows that under smoothness and strong convexity, DGD converges linearly to the (FLIX) solution.

Theorem 2. *Suppose that each f_i in (FLIX) is L_i-smooth and μ_i-strongly convex. Define x^{avg}, L_α, $\hat{L}$, V and D as in Theorem 1. Suppose that we run DGD for K iterations starting from $x^0 = x^{\text{avg}}$. Then the following hold:*

i) If the α_i are allowed to be arbitrary, then for $\alpha_{\max} \overset{def}{=} \max_{i=1,\ldots,n} \alpha_i$ we have $\tilde{f}(x^k) - \min\limits_{x \in \mathbb{R}^d} \tilde{f}(x) \leq \left(1 - \frac{\mu_\alpha}{L_\alpha}\right)^K \frac{\alpha_{\max}^2 \hat{L} D}{2}$.

ii) Let $\hat{\mu} \overset{def}{=} \frac{1}{n} \sum_{i=1}^{n} \mu_i$. If $\alpha_i = \beta$ for all i, then

$$\tilde{f}(x^k) - \min_{x \in \mathbb{R}^d} \tilde{f}(x) \leq \left(1 - \frac{\hat{\mu}}{\hat{L}}\right)^K \frac{\beta^2 \hat{L} V}{2}. \tag{7}$$

There are four ways of making the right hand side in (7) (the communication complexity) small:

- **Communicate more.** Increase the number of communications K.
- **Homogeneous data**. The variance $V = \sum_{i=1}^{n} w_i \|x_i - x^{\text{avg}}\|^2$ can be seen as a measure of data heterogeneity. More homogeneous data means smaller V, which leads to better performance.
- **Train simpler models.** Focusing attention on models with smaller $\hat{L}$ (adjust model design), or larger $\hat{\mu}$ (e.g., add more regularization).
- **Put more weight on local models.** If we prefer local models to the global model, then α_i is small, and hence fewer communications are needed to achieve any given accuracy.

Armed with Theorem 2, we make good on our promise in Section 1.1 and show that FLIX can be solved using *any* communication budget. Looking at (7) we see that for any fixed $\epsilon > 0$ we have $\tilde{f}(x^k) - \min_{x \in \mathbb{R}^d} \tilde{f}(x) \leq \epsilon$ as long as $\beta \leq Aq^k$, where $A = \sqrt{2\epsilon/(\hat{L}V)}$ and $q = 1/\sqrt{1 - \frac{\hat{\mu}}{\hat{L}}}$.

Putting this together leads to the following observations:

- If $\beta = 0$, the problem can be solved with 0 communications (i.e. , each device i independently computes the pure local model x_i).
- If $0 < \beta \leq A$, the problem can be solved with 1 communication (i.e., compute x^{avg}). This follows from Theorem 1, and also from the more general result Theorem 2 by setting $K = 0$.
- If $A < \beta \leq Aq$, the problem can be solved with 2 communications (1 communication to compute $x^0 = x^{avg}$, followed by one iteration of distributed GD).
- If $Aq^{k-1} < \beta \leq Aq^k$, the problem can be solved with $k+1$ communications (1 communication to compute $x^0 = x^{avg}$, followed by K iterations of distributed gradient descent).
- If $\beta = 1$, we need 1 communication to compute $x^0 = x^{avg}$, followed by $k \geq \frac{\bar{L}}{\bar{\mu}} \log \frac{\bar{L}V}{2\varepsilon}$ iterations of distributed gradient descent. This is recovers the standard communication complexity of gradient descent needed to find the optimal solution of the average risk minimization problem (ERM).

In the supplementary, we develop other algorithms for solving (FLIX) such as distributed gradient descent with compression (Alistarh et al., 2017) and DIANA (Mishchenko et al., 2019). We note that because (FLIX) has a standard finite-sum form, many more algorithms can be used to solve it, e.g. accelerated minibatch SGD (Cotter et al., 2011) or SARAH (Nguyen et al., 2017).

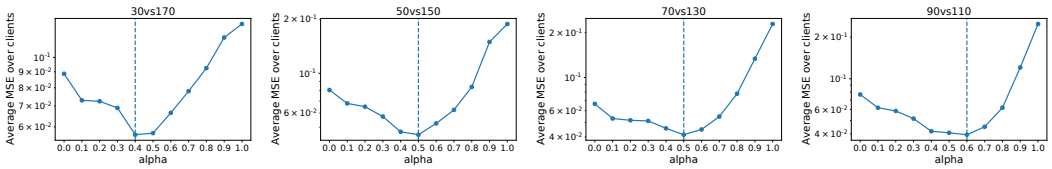


Figure 2: Average MSE vs. personalization parameter α.

4 Experiments

Generalization experiment 1: Fitting Sine Functions. Following Finn et al. (2017) and Zhou et al. (2019), we show the generalization advantages of FLIX on the following regression problem. We define i-th client's function $f_i(x) = a_i \sin(x + b_i)$, where amplitude a_i and phase b_i lie in the intervals $[0.1, 0.5]$ and $[0, 2\pi]$, respectively. For each client, we fix a_i and b_i and sample 50 points uniformly at random from the interval $[-5.0, 5.0]$. We measure regression fit in terms of mean squared error (MSE) loss. To train a model each client adopts a neural net with 2 hidden layers of size 40 with tanh activation. Further technical details are deferred to the appendix. For the experiment, we first sample 2 pairs $\{a_i, b_i\}$ and each of 200 clients is assigned one pair, we investigate different proportion– $(30, 170), (50, 150), (70, 130), (90, 110)$. We then train our FLIX formulation with $\alpha_i = \alpha = 0.1, 0.2, \ldots, 1$. For testing for each client generates a new dataset of size 2000. Figure 2 shows average MSE over clients against different values of α for different proportions. As this figure indicates, optimal α for which test average MSE is minimal can dramatically outperform the edge cases of either global model for all tasks or personalized model trained only on the local dataset.

Generalization experiment 2: Comparison to FOMAML and Reptile. Inspired by Reddi et al. (2021), we conduct a similar experiment to compare generalization capabilities, i.e., test accuracy, of FLIX and its two baselines FOMAML (Finn et al., 2017), and Reptile (Nichol et al., 2018). For the first experiment (see Figure 1a), we take 500 train data points of two clients (with client ids '00000267' and '00000459') from the Stack Overflow dataset (TensorFlow Developers, 2021) and divide them among 100 workers so that there are 50 workers with 10 train data points from the first client and another 50 with 10 data points from the second client. For the second experiment (see Figure 1b), a worker gets 90 train data points from a distinct client. For both experiments, each objective component f_i is a cross-entropy loss for multi-class logistic regression. Further technical details and the hyperparameters tuning for a fair comparison can be found in the supplementary. In the test phase, for each client, we used a hold-out testing dataset of size 300 (the same dataset has been used for workers related to the same client in the first experiment). It can be observed from Figure 1a, that for wide range of α_i, $\alpha_i \in \{0.2, 0.4, 0.6, 0.8\}$ FLIX exhibits a better generalization than its classical meta-learning competitors–FOMAML and Reptile, and it can lead to improvement of up to 11% in recall@5. Figure 1b shows that in the more real-world scenario FLIX outperforms FOMAML and Reptile while showing its best test accuracy in non-edge α.

References

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: communication-efficient SGD via gradient quantization and encoding. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1709–1720, 2017. (Cited on page 6)

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. (Cited on page 38)

Zachary Charles and Jakub Konečný. Convergence and accuracy trade-offs in federated learning and meta-learning. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2575–2583. PMLR, 13–15 Apr 2021. (Cited on page 4)

Shuxiao Chen, Qinqing Zheng, Qi Long, and Weijie J. Su. A theorem of the alternative for personalized federated learning. *arXiv preprint arXiv:2103.01901*, 2021. (Cited on page 3)

Kurtland Chua, Qi Lei, and Jason D. Lee. How fine-tuning allows for effective meta-learning. *arXiv preprint arXiv:2105.02221*, 2021. (Cited on page 4)

Liam Collins, Aryan Mokhtari, and Sanjay Shakkottai. Why does MAML outperform ERM? an optimization perspective. *arXiv preprint arXiv:2010.14672*, 2020. (Cited on page 4)

Andrew Cotter, Ohad Shamir, Nati Srebro, and Karthik Sridharan. Better mini-batch algorithms via accelerated gradient methods. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 1647–1655, 2011. (Cited on page 6)

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020. (Cited on pages 3 and 35)

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. (Cited on page 36)

Canh T. Dinh, Nguyen H. Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. *arXiv preprint arXiv:2006.08848*, 2021a. (Cited on page 3)

Canh T. Dinh, Tung T. Vu, Nguyen H. Tran, Minh N. Dao, and Hongyu Zhang. FedU: A unified framework for federated multi-task learning with laplacian regularization. *arXiv preprint arXiv:2102.07148*, 2021b. (Cited on page 3)

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020a. (Cited on page 3)

Alireza Fallah, Aryan Mokhtari, and Asuman E. Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 1082–1092. PMLR, 2020b. (Cited on page 5)

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Generalization of model-agnostic meta-learning algorithms: Recurring and unseen tasks. *arXiv preprint arXiv:2102.03832*, 2021. (Cited on page 4)

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 2017. (Cited on pages 4 and 7)

Katelyn Gao and Ozan Sener. Modeling and optimization trade-off in meta-learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. (Cited on page 4)

Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. A unified theory of sgd: Variance reduction, sampling, quantization and coordinate descent, 2019. (Cited on pages 20, 21, 22, and 25)

Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019. (Cited on page 5)

Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv:2002.05516*, 2020. (Cited on pages 3 and 4)

Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. *arXiv preprint arXiv:2007.03797*, 2021. (Cited on page 3)

Yihan Jiang, Jakub Konečný, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019. (Cited on pages 3 and 4)

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019. (Cited on pages 1, 2, and 3)

Ahmed Khaled and Peter Richtárik. Better theory for sgd in the nonconvex world, 2020. (Cited on pages 30 and 31)

Ahmed Khaled, Othmane Sebbouh, Nicolas Loizou, Robert M. Gower, and Peter Richtárik. Unified analysis of stochastic gradient methods for composite convex and smooth optimization, 2020. (Cited on pages 21, 27, and 30)

Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Private Multi-Party Machine Learning Workshop*, 2016. (Cited on pages 2 and 3)

Dingwei Li, Qinglong Chang, Lixue Pang, Yanfang Zhang, Xudong Sun, Jikun Ding, and Liang Zhang. More industry-friendly: Federated learning with high efficient design. *arXiv preprint arXiv:2012.08809*, 2020. (Cited on page 2)

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *arXiv preprint arXiv:1908.07873*, 2019. (Cited on page 2)

Zhize Li and Peter Richtárik. A unified analysis of stochastic gradient methods for nonconvex federated optimization, 2020. (Cited on page 32)

Grigory Malinovskiy, Dmitry Kovalev, Elnur Gasanov, Laurent Condat, and Peter Richtárik. From local SGD to local fixed-point methods for federated learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6692–6701. PMLR, 2020. (Cited on page 3)

Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020. (Cited on pages 3 and 35)

Konstantin Mishchenko, Eduard Gorbunov, Martin Takác, and Peter Richtárik. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019. (Cited on pages 6, 20, and 38)

Yurii Nesterov. *Lectures on Convex Optimization*. Springer International Publishing, 2018. doi: 10.1007/978-3-319-91578-4. (Cited on pages 13 and 19)

Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takác. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2613–2621. PMLR, 2017. (Cited on page 6)

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. (Cited on page 7)

Reese Pathak and Martin J. Wainwright. FedSplit: an algorithmic framework for fast federated optimization. *arXiv preprint arXiv:2005.05238*, 2020. (Cited on page 3)

Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. (Cited on pages 7 and 36)

Amirhossein Reisizadeh, Farzan Farnia, Ramtin Pedarsani, and Ali Jadbabaie. Robust federated learning: the case of affine distribution shifts. *arXiv preprint arXiv:2006.08907*, 2020. (Cited on page 3)

Saber Salehkaleybar, Arsalan Sharifnassab, and S. Jamaloddin Golestani. One-shot federated learning: Theoretical limits and algorithms to achieve them. *arXiv preprint arXiv:1905.04634*, 2019. (Cited on page 5)

Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–13, 2020. doi: 10.1109/TNNLS.2020.3015958. (Cited on page 3)

TensorFlow Developers. TensorFlow Federated Stack Overflow dataset, 2021. URL https://www.tensorflow.org/federated/api_docs/python/tff/simulation/datasets/stackoverflow/load_data. (Cited on page 7)

Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Françoise Beaufays, and Daniel Ramage. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019. (Cited on page 4)

Blake Woodworth, Kumar Kshitij Patel, and Nathan Srebro. Minibatch vs local SGD for heterogeneous distributed learning. *arXiv preprint arXiv:2006.04735*, 2020. (Cited on page 3)

Zheng Xu, Karan Singhal, Zachary Charles, Ziyu Liu, Advait Gadhikar, and Shanshan Wu. Federated optimization. https://github.com/google-research/federated/tree/master/optimization, 2021. (Cited on page 36)

Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020. (Cited on pages 1 and 3)

Edvin Listo Zec, Olof Mogren, John Martinsson, Leon René Sütfeld, and Daniel Gillblad. Specialized federated learning using a mixture of experts. *arXiv preprint arXiv:2010.02056*, 2021. (Cited on pages 3 and 35)

Pan Zhou, Xiaotong Yuan, Huan Xu, Shuicheng Yan, and Jiashi Feng. Efficient meta learning via minibatch proximal update. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 1534–1544. Curran Associates, Inc., 2019. (Cited on page 7)