2021 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)

Federated Few-Shot Learning with Adversarial Learning

Chenyou Fan[†] Jianwei Huang[‡,†]
{fanchenyou, jianweihuang}@cuhk.edu.cn
[†]Shenzhen Institute of Artificial Intelligence and Robotics for Society, China
[‡]School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China

***Abstract*—We are interested in developing a unified machine learning framework for effectively training machine learning models from many small data sources such as mobile devices. This is a commonly encountered situation in mobile computing scenarios, where data is scarce and distributed while the tasks are distinct. In this paper, we propose a federated few-shot learning (FedFSL) framework to learn a few-shot classification model that can classify unseen data classes with only a few labeled samples. With the federated learning strategy, FedFSL can utilize many data sources while keeping data privacy and communication efficiency. To tackle the issue of obtaining misaligned decision boundaries produced by client models, we propose to regularize local updates by minimizing the divergence of client models. We also formulate the training in an adversarial fashion and optimize the client models to produce a discriminative feature space that can better represent unseen data samples. We demonstrate the intuitions and conduct experiments to show our approaches outperform baselines by more than 10% in learning benchmark vision tasks and 5% in language tasks.**

***Index Terms*—federated learning, few-shot learning, adversarial optimization**

I. Introduction

Conventional distributed machine learning approaches [1], [2] require the data to be transferred from clients to a central server, which raises serious concerns of data privacy. A recent proposed approach to address this issue is federated learning (FedL) [3]–[5]. In the FedL paradigm, each participating client computes a local machine learning model with its own data, while a central server periodically coordinates client models by model aggregation without collecting the actual data.

However, existing FedL approaches assume each participating client has sufficient training data for the tasks of interest. For example, image classification, the common benchmark task of FedL studies [3]–[5], assumes the availability of thousands of labeled training samples for every class. In reality, each mobile user may own just one or a few samples of interested classes, and the user often does not have time or interest to label each of them. The huge gap between lab scenarios with abundant labeled data and real situations with scarce and mostly unlabeled data severely limits the practicality and scalability of FedL in real applications. It motivates us to consider the following question: **How to make FedL effective in data-scarce scenarios?**

A recently developed concept to address the issue of insufficient training data is few-shot learning (FSL) [6]–[8]. FSL aims to develop machine learning models to solve unseen tasks with very few examples, but often in the context of a single data source. In this paper, we propose a federated few-shot learning (FedFSL) framework for efficiently training models with scarce data sources in federation. The developed framework can tackle novel tasks that are never seen by any of the data sources. As shown in Fig. 1, the paradigm of FedFSL is to first perform local updates with few-shot tasks sampled from local data, then perform model aggregation and synchronization as in FedL which preserves the data privacy.

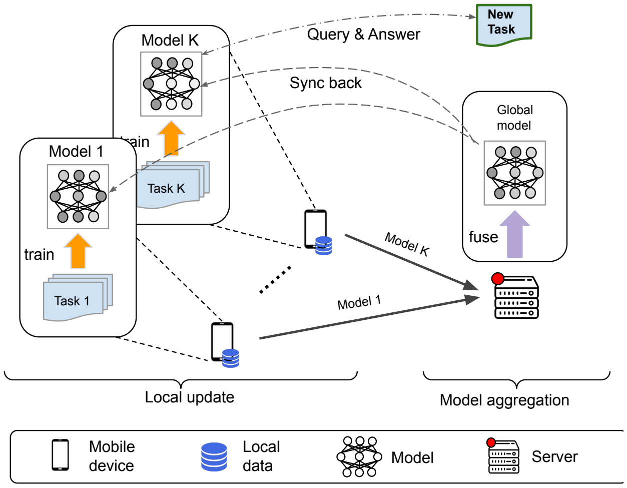

Fig. 1: Overview of a FedFSL system. Distributed client devices train models with sampled FSL tasks, send the models to the central server to aggregate for a central model, and synchronize back for the next round of local updates.

FedFSL has many potential applications for mobile computing. For example, a few-shot language model can be used to suggest words by learning from just a few typing records from each of many users; a few-shot face recognition model can identify users and their friends by learning from just few annotated photos by each of many mobile users.

There are two technical challenges to develop an effective FedFSL framework: 1) directly using the existing FedL approaches in the data-scarce scenarios may lead to misaligned decision boundaries produced by client models, and 2) constraining the decision boundaries to be similar over clients would develop a classifier overfit to training tasks but not transferable to unseen tasks. To address these issues, we first propose to regularize local updates by minimizing

ISBN 978-3-903176-37-9 © 2021 IFIP

the divergence between client models and the central model. Then we design an adversarial learning procedure to construct a discriminative feature space that can better generalize to unseen tasks. We demonstrate the intuitions and conduct experiments to show the effectiveness of our approaches.

Our contributions can be summarized as follows:

- We propose a novel learning framework that can perform effective federated learning on few-shot tasks. This represents the first step in addressing the scenarios where training data is scarce and testing tasks are distinct.
- We define a novel concept of mutual divergence of federated client models, which can be minimized to better coordinate the client training on scarce local data.
- We design a dedicated adversarial learning approach to construct a discriminative feature space, which better generalizes to unseen tasks compared with existing training procedures of FSL models.
- We evaluate our framework by modelling different types of structured data (such as images and sentences) and show its effectiveness and practical usability in various learning tasks in machine vision and NLP.
- Our approaches significantly outperform baselines that are either non-distributed or not aligning the feature space across the clients by more than 10% on vision tasks and 5% on language tasks.

II. Related work

We will briefly review recent related work in two categories: (i) studies either federated learning (FedL) (e.g., [3]–[5], [9]–[17]) or few-shot learning (FSL) (e.g., [6], [8], [18]–[29]), or both of them [30], [31] (ii) studies proposing similar ideas of minimizing model divergence to better learn individual models or an ensemble model.

To our best knowledge, training FSL models on distributed devices is still an under-explored open problem. The first work of this topic was from Chen *et al.* [30] who explored federated meta-learning by applying FedAvg on meta-learning approaches such as MAML [7] in a straightforward way. However, their goal is to improve supervised learning by better sharing models among federated clients, instead of learning few-shot tasks. They neither evaluated their models on FSL tasks, nor explicitly considered dealing with the underlying data heterogeneity in different devices (e.g., non-IID case) which can severely harm FSL. Our model captures the idea of federating transferable knowledge among distributed clients. We further explore the practical data-scarce scenarios and evaluate our models on challenging benchmark FSL datasets. In addition, we explicitly resolve the data heterogeneity issue by proposing a family of more effective meta-learning approaches designed for federated settings.

The other work that loosely connects FSL with FedL is Li *et al.* [32], which proposed a differentially private algorithm for securing parameter transfer across devices or learning stages. The authors considered FedL and FSL as two separated applications of their technique, and their goal is to secure data privacy during model sharing instead of performing FSL with federated devices.

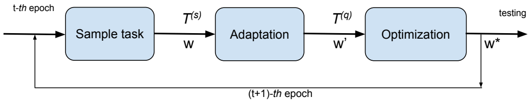

Fig. 2: Three-step meta-learning of FSL.

There are two recent topics relating to training coordinated client models with consistent feature space, though neither of them considered few-shot learning. Personalized FedL [33]–[35] aims to use FedL to learn different client models to better fit local data. Existing work either fails to consider few-shot scenarios [33], [34] or simply combines FSL and FedL in naive way without considering optimizing the feature space. The other related work of Zhang *et al.* [36] proposed to minimize the divergence of every client model pair to enhance ensemble learning. However, this imposes heavy computation and communication costs in distributed scenarios. On the contrary, we propose to approximate the pairwise client divergence by the divergence between the client model and the federated global model, which integrates into FedL seamlessly.

III. Federated Few-shot Learning

In this section, we will formulate a straightforward way of perform few-shot learning with federated learning setting, which we call FedFSL-naive. We will first review the general federated learning (FedL) and the general centralized few-shot learning (FSL) respectively, based on which we propose the FedFSL-naive formally.

A. General Federated Learning Objective

In a common federated learning (FedL) system of K clients, let n_k be number of data samples of client k, $n = \sum_k n_k$ be total samples across the devices, w be the learning model. The local objective for client k is the average loss over all data samples

$$\mathcal{L}_k(w) = \frac{1}{n_k} \sum_{i=1}^{n_k} f(x_i, y_i; w) \ , \tag{1}$$

in which f is a loss function that evaluates the prediction of model w on a data sample (x_i, y_i). The global target is a weighted average of local objectives

$$\min_w \mathcal{L}(w) = \sum_{k=1}^{K} p_k \mathcal{L}_k(w), \quad s.t. \ p_k = n_k/n \ . \tag{2}$$

Existing FedL approaches often assume that the clients always hold sufficient training data for a same task, i.e., the local objective $\mathcal{L}_k(w)$ can be well optimized with enough data samples. However, the realistic situation is each client may own a few labeled data samples for certain categories for training, and may encounter unlabeled data samples for testing with unseen true categories. This leads us to study the few-shot classification task which learns to classify on novel classes with few training samples in the following sections.

ISBN 978-3-903176-37-9 © 2021 IFIP

B. General Centralized Few-Shot Learning

Next, we briefly review centralized few-shot learning (FSL) procedures. FSL aims to learn a generic model which can adapt to unseen tasks with only a few labeled training samples. Following the convention, we define FSL task as an N-way P-shot Q-query learning task, i.e., training a model with P labeled data instances for each of N classes and then testing on Q unlabeled query instances for each class. P is typically very small such as 1 or 5 as "few-shot" implies.

Let us consider a toy example of classifying animal pictures with mobile devices. The user captures and labels several images for cat and dog, and wish to develop a machine learning model to classify rare classes such as tigers and wolves from just one captured image at the zoo. As the training instances for tiger and wolf are extremely scarce, supervised learning is infeasible.

We consider the recent state-of-the-art few-shot learning strategy called meta-learning [7], [8], which aims to learn transferable knowledge from few data samples and apply on unseen data. The training objective is to minimize the training loss over a batch of tasks $\mathcal{T} \in \mathcal{B}$ as follows

$$\min_{w} \mathcal{L}(w) = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{T} \in \mathcal{B}} \ell_{\mathcal{T}}(w), \tag{3}$$

in which ℓ is the task loss on training batch. This can be tackled with an iterative approach in which each iteration can be decomposed into three steps, as shown in Fig. 2.

- **Sampling step:** The first step is to sample a few-shot task $\mathcal{T}$, also called an *episode*, from base classes. For an N-way P-shot Q-query few-shot task, an episode consists of P data instances sampled from each of N distinct base classes as a support set $\mathcal{T}^{(s)}$, and Q data instances sampled from the same N classes as a query set $\mathcal{T}^{(q)}$.
- **Adaptation step:** The second step is to adapt the current model to the sampled task with gradient descents. This step uses the few labeled data in the support set $\mathcal{T}^{(s)}$ and performs one or several gradient steps towards optimizing the model weights to the sampled task such that

  $$w^{'} = w - \alpha \nabla_w f_{\mathcal{T}^{(s)}}(w), \tag{4}$$

  in which w' is the adapted model, α is the step size.
- **Optimization step:** The final step is to evaluate w' with more samples in the query set $\mathcal{T}^{(q)}$ with the empirical loss

  $$\ell_{\mathcal{T}}(w) = f_{\mathcal{T}^{(q)}}(w') = f_{\mathcal{T}^{(q)}}(w - \alpha \nabla f_{\mathcal{T}^{(s)}}(w)), \tag{5}$$

  During training, the query images $T^{(q)}$ are labeled to provide training signals to update the model w as follows

  $$w \leftarrow w - \beta \; \nabla_w \ell_{\mathcal{T}}(w), \tag{6}$$

  in which β is the learning rate. During inference, the labels of the query images are unknown, and we estimate them with the output of model w'.

The above procedures are summarized in Fig. 2.

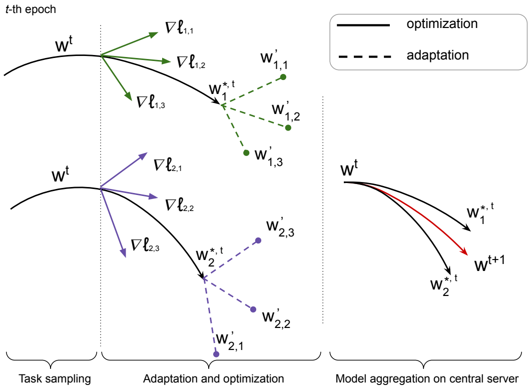

Fig. 3: Demo of a two-client case of FedFSL.

C. Federated Few-shot Learning (FedFSL)

As our purpose is to facilitate distributed devices to learn models for few-shot tasks, we need study how to design such a framework in which meta-learning procedures can be integrated in the federated learning. We propose Federated Few-shot Learning (FedFSL) in this section. The goal of FedFSL is to search for a *global* optimal model w^* learned on distributed data sources that can best perform few-shot tasks.

Suppose we have K participating clients and each of them can sample batches of few-shot tasks $\mathcal{T}_k \in \mathcal{B}_k$ from their local data sources as discussed in previous section. The local FSL objective of k-th client extends from (3) as follows

$$\mathcal{L}_k(w) = \frac{1}{|\mathcal{B}_k|} \sum_{\mathcal{T}_k \in \mathcal{B}_k} \ell_{\mathcal{T}_k}(w), \tag{7}$$

in which the subscript k of $\mathcal{T}_k$ emphasizes that it's sampled from local data source of client k. We search for a federated global model w^* which minimizes the weighted average of local FSL objectives as follows

$$w^* = \min_{w} \mathcal{L}(w) = \sum_{k=1}^{K} p_k \mathcal{L}_k(w) = \sum_{k=1}^{K} \frac{|\mathcal{B}_k|}{|\mathcal{B}|} \mathcal{L}_k(w) \tag{8}$$

Motivated by FedAvg [3], we propose a straightforward way of solving a surrogate objective of (8) to approximate the global solution, which we call **FedFSL-naive**. Similar approaches have also been mentioned in [30], [34]. As shown in Fig. 3, FedFSL-naive iteratively updates the central model w by (i) first optimizing each local objective of (7) in parallel, and (ii) aggregating local models to the central model, which update the global model and send it back to clients for the next round of optimization. Formally,

- At the t-th optimization round, each client k optimizes the following local objective

  $$w_k^{*,t} = \underset{w}{\text{argmin}} \; \mathcal{L}_k(w) = \underset{w}{\text{argmin}} \; \frac{1}{|\mathcal{B}_k|} \sum_{\mathcal{T}_k \in \mathcal{B}_k} \ell_{\mathcal{T}_k}(w) \;, \tag{9}$$

  in which the FSL loss $\ell(w)$ is given by (5). Fig. 3 shows a two-client example, in which each client updates on three sampled tasks with (9) and obtains local optimal models

ISBN 978-3-903176-37-9 © 2021 IFIP

$w_1^{*,t}$ and $w_2^{*,t}$. The clients then send these local parameters to the central server.
- Then the central server approximates the optimal global solution by averaging the client models such that

$$w^{t+1} = \sum_{k=1}^{C} \frac{|\mathcal{B}_k|}{|\mathcal{B}|} w_k^{*,t}, \tag{10}$$

and send to all clients for next round of optimization.

Steps (9)-(10) are repeated for multiple rounds until convergence. We show the convergence of FedFSL-naive as follows.

Proposition 1. *If loss function $f_{\mathcal{T}}(w)$ in (4) satisfies the strongly-convex conditions as in Corollary 1, Finn et al. [23]* [1] *, FedFSL-naive converges at a rate of $\mathcal{O}(\frac{1}{T})$ in which T is the total number of gradient updates during training.*

Proof. As $f_{\mathcal{T}}(w)$ is strongly-convex, Corollary 1, [23] implies that the local FSL objective $\mathcal{L}_k$ in (7) is also strongly-convex. FedFSL-naive can be taken as a FedAvg algorithm with a strongly-convex objective, thus Theorem 3, [37] implies that it converges at a rate of $\mathcal{O}(\frac{1}{T})$ in which T is total number of local gradient updates of all devices during training. □

Remark 1. *By (Theorem 4.5 [34]), non-convex models $f(w)$ satisfy bounded gradient $||\nabla f_i|| \leq B$, twice continuously differentiable, L-Lipschitz with ρ_i-Lipschitz Hessians of bounded variance, converge to some stationary points.*

IV. Improving FedFSL with better coordination

So far, we have provided a straightforward way of performing federated few-shot learning. However, as meta-learning depends on sampled episodes that contain only very few labeled data points, different data distributions over the clients as well as the high variance of the data may lead to quite distinct gradient descent directions, and thus the trained few-shot models could become quite distinct over the clients. This results in model divergence in aggregation. Similar observations were also found in FedL tasks with non-IID data [4], [9] but this problem could be amplified in the data-scarce scenarios we consider.

In Fig. 4(a), we illustrate a two-client case to show that the discrepancy between two client models makes them provide misaligned individual decision boundaries (left). The aggregated central model thus provides less optimal federated decision boundary (right) with lots of misclassified data samples.

In this section, we will discuss how to better coordinate client models with mutual information in IV-A, and we propose an adversarial learning procedures to further learn a discriminative feature space in IV-B.

A. FedFSL with Mutual Information (MI)

To better coordinate client models learned on distinct data sources, we propose the concept of *mutual information* (MI) as regularization term to measure and minimize the discrepancy of all the participating client models.

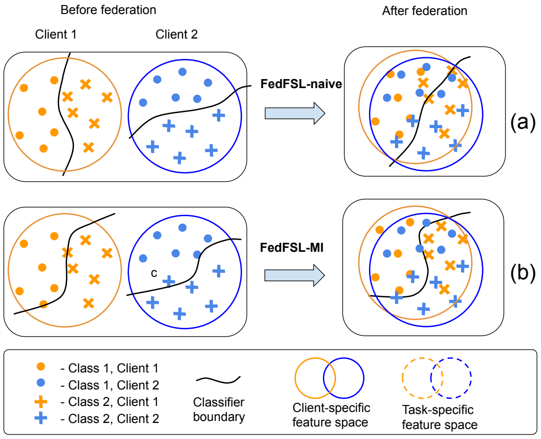

Fig. 4: Illustration of decision boundaries learned by (a) FedFSL-naive and (b) FedFSL-MI in two-client case.

Formally, we define the MI loss as the Kullback-Leibler (KL) divergence of probability outputs produced by the federated global model w^t and the client model w_k over sampled tasks at each round t such that

$$\begin{aligned} \mathcal{L}_k^{MI}(w^t, w_k) =& \frac{1}{|\mathcal{B}_k|} \sum_{\mathcal{T}_k} D_{KL}\left(p(w^t) \mid\mid p(w_k)\right) \\ =& \frac{1}{|\mathcal{B}_k|} \sum_{\mathcal{T}_k} \left(p(w^t) \cdot \log \frac{p(w^t)}{p(w_k)} \right), \end{aligned} \tag{11}$$

in which $p(\cdot)$ is the probability outputs of an FSL model. Given an N-way FSL task $\mathcal{T}_k$, $p(w)$ is the normalized N-way predictions over N classes that sums to one. We aim to minimize MI in order to reduce the discrepancy, and define the objective of Fed-MI as follows

$$\min_{w_k} \mathcal{L}^{Fed\text{-}MI}(w_k) = \mathcal{L}(w_k) + \gamma \mathcal{L}^{MI}(w^t, w_k) \ , \tag{12}$$

in which the weight $\gamma > 0$ can be searched by cross-validation.

To stabilize the stochastic optimization process, we clip the probability ratios between global and local predictions $\frac{p(w^t)}{p(w_k)}$ in (11) to be within interval $(1-\varepsilon, 1+\varepsilon)$, and $0 < \varepsilon < 1$. Similar practices are also used to stabilize generative adversarial learning [38] and reinforcement learning [39].

We call this new method **FedFSL-MI** (FedFSL with Mutual Information regularization), and illustrate the intuition in Figure 4(b). We encourage the decision boundaries to be consistent across the local clients so that the federated model can produce a better aligned decision boundary. We now prove the convergence of FedFSL-MI as follows.

Lemma 1. *If non-convex loss function $f_{\mathcal{T}}(w)$ satisfies conditions as in Theorem 4.5 [37]: $f(w)$ has bounded gradient $||\nabla f_i|| \leq B$, twice continuously differentiable and L-Lipschitz with ρ_i-Lipschitz Hessians of bounded variance, learning rate β and with term T defined in Theorem 4.5 [37], and we further assume the probability output function is also*

[1] f is *G-Lipschitz*, *β-smooth*, *ρ-Lipschitz* Hessians and *μ-strongly* convex

ISBN 978-3-903176-37-9 © 2021 IFIP

bounded $||\nabla p(w_k)|| \leq B_p$*, then the corresponding FedFSL-MI objective* (11) *satisfies the first-order stationary condition*

$$\begin{aligned}&\frac{1}{\tau K}\sum_{k=0}^{K-1}\sum_{t=0}^{\tau-1}E[||\nabla L(\bar{w}_{k+1,t})+\gamma L^{MI}(\bar{w}_{k+1,t})||^2]\\ \leq&\frac{4(L(w_0)-L^*)}{\beta\tau K}+O(1)(T+\gamma^2B_p^2(1+\varepsilon)^2)\ ,\end{aligned} \tag{13}$$

where $\bar{w}_{k+1,t}$ *is the average of each local update* $t \leq \tau$.

Proof. We consider the MI objective as follows

$$\begin{aligned}&E[||\nabla L(w_k)+\gamma L^{MI}(w^t,w_k)||^2]\\ \leq\ &2E[||\nabla L(w_k)||^2]+2\gamma^2E[||\nabla L^{MI}(w^t,w_k)||^2]\ .\end{aligned}$$

We can directly utilize Theorem 4.5 [37] to give an upper bound for the first term. Then the second term satisfies

$$\begin{aligned}&\gamma^2E[||\nabla L^{MI}(w^t,w_k)||^2]\\ =\ &\gamma^2E\left[\left\|\frac{p(w^t)}{p(w_k)}\nabla p(w_k)\right\|^2\right]\leq\gamma^2B_p^2(1+\varepsilon)^2\ ,\end{aligned}$$

since we clip the ratio $\frac{p(w^t)}{p(w)}$ to interval $(1-\varepsilon, 1+\varepsilon)$. We combine the upper bounds of first and second term, and we properly scale the learning rate, to complete the proof. □

Remark 2. *Based on Lemma 1, and Corollary 4.6 [37], and by letting* σ_G^2 *denote bounded stochastic gradient, we can properly set the batch size* D*, step size* α *of local updates, and communication rounds* τ *to find an* $O(\varepsilon+\frac{\alpha^2\sigma_G^2}{D})$*-first-order stationary point.*

B. Improving feature space with adversarial learning

One technical disadvantage of FedFSL-MI is that constraining the decision boundaries to be similar over clients would develop a complex classifier that overfits to training tasks, while makes the complex decision boundary not useful to unseen tasks. This also presents a key difference between FSL and conventional supervised learning.

1) Feature space: We aim to search for a representative feature space as transferable knowledge that can be used to learn unseen data samples. In an ideal feature space, samples of the same labels are close to each other, while samples of different labels are far away. For example, images of cats and tigers are close in feature space, while tigers and wolves could be far away due to their distinct visual features.

2) Learn a consistent feature space: We will show that such a feature space can be derived properly and efficiently in distributed scenarios without sharing local data. Motivated by recent progress in Generative Adversarial Networks [40], [41], we will decompose an FSL model as a feature generator and a classifier (i.e., discriminator) which can be optimized in an alternative and iterative fashion. This new adversarial learning approach is named as **FedFSL-MI-Adv** (FedFSL with Mutual Information regularization and Adversarial learning).

We first introduce the notations to facilitate discussion. Without loss of generality, a few-shot learning model can be represented as a feature generator Θ and a classifier θ. For a given data sample x, we denote its generated feature as $f_\Theta(x)$. The output logits of the classification model is derived by applying the classifier on the feature such that $f_\theta \circ f_\Theta(x)$. Thus the probabilistic distribution over N classes is denoted as $p(\Theta,\theta)=\sigma(f_\theta \circ f_\Theta(x))$ in which σ is the softmax function.

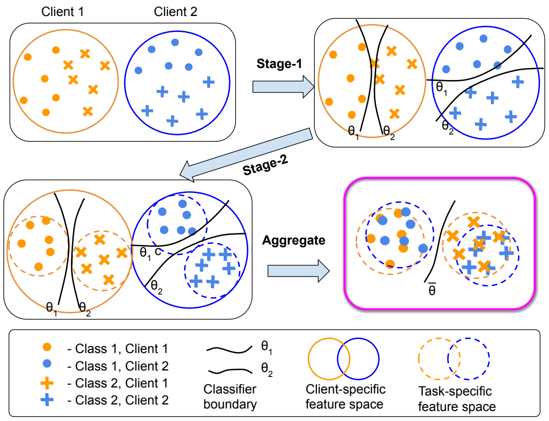

Fig. 5: An example of federated decision boundaries learned by FedFSL-MI-adv with two-stage adversarial learning on two clients of different data distributions.

3) Adversarial learning procedure: We design a two-stage adversarial learning procedure for the local update for explicitly learning a consistent feature space. The core idea is to alternatively train the classifier and the generator as two opponents: train the local classifiers to maximize the difference between their predictions and central model predictions, while train the client feature generators to minimize the difference.

In overall, the (t-1)-th communication round ends up by aggregating the client models to a central model in (10) and sending it back to clients as $w_k = [\Theta_k, \theta_k]$. In addition, we utilize a second classifier θ_k' for the learning process.

- **Training stage-1** is to train θ_k and θ_k' to produce *distinct* decision boundaries, in the motivation of detecting ambiguous data samples in current feature space. Ambiguous samples are lying near the decision boundaries which tend to be misclassified by two different classifiers, as shown in Fig. 5(stage-1). We define the adversarial loss to measure the difference of two classifiers θ_k and θ_k' by the KL divergence of their probabilistic outputs $p(\Theta,\theta)$ such that

$$\mathcal{L}_k^{adv}(\theta_k,\theta_k';\Theta_k)=\frac{1}{|\mathcal{B}_k|}\sum_{\mathcal{T}_k}D_{KL}\left(p(\Theta_k,\theta_k)\ ||\ p(\Theta_k,\theta_k')\right). \tag{14}$$

  We simultaneously minimize the FSL-MI local objective while *maximize* the adversarial loss to encourage the disagreement of the two classifiers. Formally, the learning objective of stage-1 is a combination of task objective $\mathcal{L}_k$ (12) and adversarial loss $\mathcal{L}_k^{adv}$ with weight $\eta > 0$ such that

$$\begin{aligned}&\min_{\theta_k,\theta_k'}\ \mathcal{L}_k^{st\text{-}1}(\theta_k,\theta_k';\Theta_k)\\ =\ &\mathcal{L}_k(\theta_k;\Theta_k)+\mathcal{L}_k(\theta_k';\Theta_k)-\eta\mathcal{L}_k^{adv}(\theta_k,\theta_k';\Theta_k).\end{aligned} \tag{15}$$

- **Training stage-2** is to optimize the generator Θ_k to minimize the discrepancy of the two classifiers θ_k and θ_k'. The

ISBN 978-3-903176-37-9 © 2021 IFIP

intuition is shown in Fig. 5(stage-2): by *minimizing* (14), the feature generator Θ is learning to push ambiguous data samples away from the decision boundaries, so that both classifiers could make the right predictions and their discrepancy gets reduced. As a result, the feature space (dashed circles) generated by Θ is trained to be discriminative which produces larger inter-class margins. Formally, we define the objective of stage-2 as a combination of local task objective $\mathcal{L}_k$ and adversarial loss $\mathcal{L}_k^{adv}$ with weight $\lambda > 0$ such that

$$\begin{aligned} &\min_{\Theta_k} \mathcal{L}_k^{st\text{-}2}(\Theta_k; \theta_k, \theta_k') \\ &= \mathcal{L}_k(\Theta_k; \theta_k) + \mathcal{L}_k(\Theta_k; \theta_k') + \lambda \mathcal{L}_k^{adv}(\Theta_k; \theta_k, \theta_k'). \end{aligned} \tag{16}$$

By training the classifiers and the feature generator in an adversarial manner, we iteratively optimize the model to learn a discriminative feature generator which helps boost few-shot learning on unseen tasks. We summarize in Algorithm 1.

Algorithm 1: FedFSL-MI-Adv algorithm.

```
Input: A set of $K$ federated clients. A local FSL objective
    $\mathcal{L}_k$ for each client $k$.
Output: A global model $w = [\Theta, \theta]$ optimized for FSL task.
Server executes:
    Initialize global model $w^0 = [\Theta^0, \theta^0]$
    $t \leftarrow 1$
    while $t \leq$ maximum rounds $T$ do
        for each client $k$ in parallel do
            $[\Theta_k^t, \theta_k^t] \leftarrow$ ClientUpdate($[\Theta^t, \theta^t]$)
        end
        Clients send models $[\Theta_{1...K}^t, \theta_{1...K}^t]$ to server
        $[\Theta^{t+1}, \theta^{t+1}] \leftarrow \sum_{k=1}^{K} \frac{|\mathcal{B}_k|}{|\mathcal{B}|}[\Theta_k^t, \theta_k^t]$
        The server sends $[\Theta^{t+1}, \theta^{t+1}]$ back to clients
        $t \leftarrow t + 1$
    end
    Return $[\Theta^t, \theta^t]$

ClientUpdate($[\Theta, \theta]$):
    Input: global model from previous round $[\Theta^t, \theta^t]$
    Output: updated local model $[\Theta_k^t, \theta_k^t]$
    Sample a batch of episodes $\mathcal{B}_k = \{\mathcal{T}_1, ..., \mathcal{T}_n\}$
    $[\Theta_k, \theta_k] \leftarrow [\Theta^t, \theta^t]$,  Initialize $\theta_k'$
    $\theta_k, \theta_k', \Theta_k \leftarrow$ Solve Eq.(15)-(16) alternatively
    Return $[\Theta_k, \theta_k]$
```

V. Experiments and Discussions

We first provide details of the model architecture, parameter settings, and datasets that we use in the experiments. Then we visualize the decision boundaries of our approaches with a toy example. We then demonstrate the performance of our proposed algorithms with two typical few-shot classification tasks – 5-way 1-shot and 5-way 5-shot – on three benchmark datasets which cover machine vision and NLP tasks. We will make in-depth discussions.

A. Model and datasets

We utilize a deep neural network (DNN) as our base model (adopted from ResNet-12 [42] which is commonly used for image classification tasks [20], [43], [44]). We set the adaptation step size $\alpha = 0.01$ in (4), the mutual information weight $\gamma = 0.2$, and the stage-1/-2 discrepancy loss weight $\eta = \lambda = 0.1$ in (15) and (16).

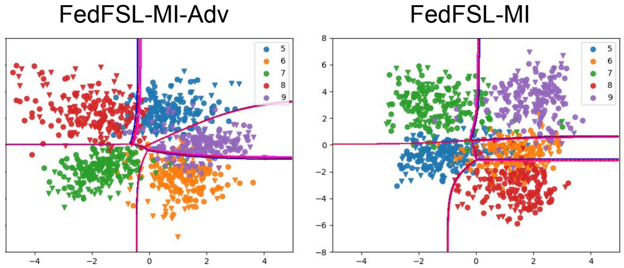

Fig. 6: Visualization of decision boundaries of FedFSL-KD and FedFSL-KD-Adv at different epochs.

We take three common benchmark datasets in evaluating both FSL and FedL in previous work [7], [8], [19], [30].

***mini*ImageNet** [6] is based on a small portion of the full ImageNet images [45]. It has 100 classes of images split to 64/16/20 as train/val/test sets. Each class has 600 images with a resolution of 84 × 84. **Sent140** [46] is a benchmark federated learning dataset for 2-way sentiment classification (positive and negative). We sampled from this dataset 10,000 annotated tweets provided by 310 twitter users and split them to train/val/test sets with provided tools. Each tweet has 1-20 English words. We tokenize the sentences and keep only common words which have GloVe representations [47].

B. MNIST Example

We provide a simple example on MNIST digit dataset in Fig. (6), to visualize and compare the decision boundaries of FedFSL-MI and FedFSL-MI-Adv. We consider the 5-way 1-shot FSL task here: train a digit classification model on data of digits 0-4, and test its few-shot classification capability on digits 5-9 by observing just one labeled sample per class. To better visualize the results, we manipulate the feature generator to produce a 2-dim feature for each input digit sample.

In Fig. 6, we plot the testing data samples from digit class 5-9 by projecting their features produced by the feature generators. We also depict the decision boundaries of the classifiers. Data samples of different classes are with different colors. We observe that FedFSL-MI-Adv (left) produces more distinguishable decision boundaries than FedFSL-MI (right) as expected. The two algorithms achieve a few-shot classification accuracy of 87.5% and 83.6%, respectively. The least accurate class recognized by FedFSL-MI-Adv is digit ’9’ (purple) of 73% correctness rate, with 18% misclassified as ’6’ (orange); while for FedFSL-MI is digit ’6’ (orange) of 64% correctness rate, with 23% misclassified as ’8’ (red). This indicates that our adversarial learning approach proposed in Section IV-B boosts the FedFSL task by constructing a more discriminative and transferable feature space for FSL.

C. Results on benchmark datasets

We experiment with our proposed three methods (FedFSL-naive, -MI, -MI-Adv) and two additional baselines (FSL-local and FedFSL-prox) for comparison.

- FSL-local is a non-distributed baseline of training an individual FSL model for each client on local data and averaging their results on the shared testing tasks.
- FedFSL-prox is a variant of FedFSL-naive by adding a weight regularization term as FedProx [9] in objective.

We partition the data samples in IID and non-IID ways. For IID partition, data samples of each class are uniformly distributed to each client. To perform non-IID partition, we follow [33], [48] by dividing data samples to all clients class-by-class with Dirichlet distribution of concentration parameter $\alpha = 1.0$. In Fig.7, we show an example of such a partition of 64 training classes of miniImageNet on a randomly chosen client, when total device number is 2 to 30.

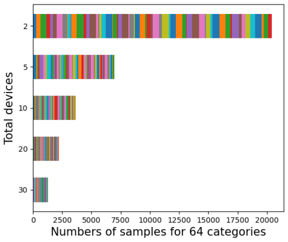

Fig. 7: Non-IID data.

Method	**Non-IID**	
	1-shot	5-shot
FedFSL-local	59.70%	66.68%
FedFSL-naive	68.85%	70.62%
FedFSL-prox	70.77%	72.25%
FedFSL-MI	70.37%	73.25%
FedFSL-MI-Adv	**71.35%**	**76.00%**

TABLE I: Sent140 results.

Method	**IID**		**Non-IID**	
	1-shot	5-shot	1-shot	5-shot
FSL-local	50.83%	67.47%	48.08%	63.25%
FedFSL-naive [30]	53.00%	67.63%	49.95%	66.11%
FedFSL-prox [9]	53.03%	69.05%	50.08%	68.53%
FedFSL-MI (ours)	54.98%	69.07%	51.07%	68.57%
FedFSL-MI-Adv (ours)	**56.42%**	**70.92%**	**53.69%**	**69.61%**

TABLE II: Results on benchmark datasets.

1) Results on Image Classification: In Table II, we compare our methods on miniImageNet with 1-/5-shot tasks learned by a federation of 10 clients. We observe

- *FedFSL-MI-Adv outperforms others.* For both IID and non-IID case, FedFSL-MI-Adv consistently outperforms others on both 1-shot and 5-shot tasks for both datasets.
- For 1-shot and 5-shot IID task on miniImageNet, FedFSL-MI-Adv achieves the best accuracy of 56.42% and 70.92%, which outperforms the second best FedFSL-MI by more than 2.6% and 2.7% respectively and relatively.
- For non-IID task, FedFSL-MI-Adv outperforms the second best FedFSL-MI by more than 5% in 1-shot case on miniImageNet, and outperforms FedFSL-naive by more than 7.2%. This indicates that our designed modules indeed help achieve a better federated model especially for non-IID case.
- Using FedProx [9] performs no better than our FedFSL-MI. This is because FedProx directly constrains client model weights to be closer to global model, while our FedFSL-MI softly optimizes the model outputs of them to be closer, which makes the training end-to-end and easier to optimize.

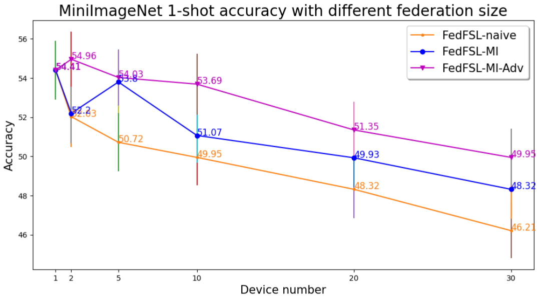

Fig. 8: FSL accuracy of 1-shot task on MiniImageNet w.r.t. number of devices in federation.

2) Results on Text Classification: In Table I, we compare our methods on Sent140 dataset with 1-shot / 5-shot tasks learned by a federation of 5 clients. Following the provided tool of partitioning the dataset, we distribute different users' data to each client without replacement. Since the data distributions vary for users, this sampling process provides non-IID data partitions. Our goal is to train an effective global sentiment classification model on one portion of users, which can be used to detect the sentiment on disjoint new users. This is particularly challenging because different users can use very distinct words and exclamations to express feelings.

In this task, our backbone model is a GRU (RNN) network with hidden size 128. We convert tweet sentences to sequences words as input to the model, and use a binary classifier to distinguish negative or positive sentiment. We examine the performance of baselines and our models and observe that

- FedFSL-MI-Adv outperforms the other approaches in this natural language understanding task, similar as in image classification task. It also shows that our FedFSL framework can be applicable to both CNN and RNN models.
- We found that the performance increases from 1-shot to 5-shot tasks are generally less than image classification tasks, e.g., less than 5% on Sent140 while more than 10% on miniImageNet. This is because the few-shot labelled sentences can only provide a few more words to help adapt to a user's emotion, while images can provide much richer details and patterns of a given object.

3) Different device number: We study the trend of accuracy of FedFSL with different number of participating devices $K = 2, 5, 10, 20, 30$, as well as $K = 1$ to simulate complete centralized training. We illustrate the results for non-IID 5-way 1-shot task with miniImageNet in Fig. 8 with detailed numbers. Note that the more participating devices, the fewer training samples each device holds. We observe that

- The overall trend is that more participating devices yielded decreased accuracy for all 3 approaches. The task becomes more difficult when K increases as the device coordination grows harder and the client model becomes less capable with less training data.
- *FedFSL-MI-Adv achieves the best results* on all cases, leading the second best FedFSL-MI by more 2-5% relatively.
- The performance of *FedFSL-MI-Adv decreases more slowly than other approaches* with the increase of K, which indicates the beneficial of learning a consistent feature space.
- *FedFSL-MI-Adv in 2-device federation works even better*

ISBN 978-3-903176-37-9 © 2021 IFIP

than 1-device centralized training, with accuracy 54.96% v.s. 54.41%. Note that the total training samples of each device get halved on each device when $K = 2$. The surprising result that distributed training outperforms centralized training can be explained that FSL is aiming to learn with very few training samples, instead of fitting a task with many samples as in supervised learning. Therefore, FSL is relative less sensitive to the number of examples in *base* classes on each client. Moreover, by utilizing our approach to align decision boundaries well, the two client models form an effective ensemble to enhance the overall performance, compared with a single-model case.

VI. Conclusion

In this paper, we proposed a framework that makes federated learning effective in data-scarce scenarios. We designed an adversarial learning strategy to construct a consistent feature space over the clients, to better learn from scarce data. Experimental results show that our adversarial learning based method outperforms baseline methods by 5%∼15% on benchmark datasets. Future work can investigate how to further extend FedFSL to regression and reinforcement learning scenarios.

Acknowledgements: This work is supported by the Shenzhen Institute of Artificial Intelligence and Robotics for Society, and the Presidential Fund from The Chinese University of Hong Kong, Shenzhen.

References

[1] M. Li, B.-Y. Su *et al.*, “Scaling distributed machine learning with the parameter server,” in *OSDI*, 2014.

[2] S. Zhang, A. E. Choromanska, and Y. LeCun, “Deep learning with elastic averaging sgd,” in *NeurIPS*, 2015.

[3] H. B. McMahan, S. Hampson *et al.*, “Communication-efficient learning of deep networks from decentralized data,” in *AISTATS*, 2017.

[4] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, “Federated learning with non-iid data,” *arXiv preprint arXiv:1806.00582*, 2018.

[5] T. Li and et al., “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Processing Magazine*, 2020.

[6] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, “Matching networks for one shot learning,” in *NIPS*, 2016.

[7] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” *ICML*, 2017.

[8] Z. Li, F. Zhou, F. Chen, and H. Li, “Meta-sgd: Learning to learn quickly for few-shot learning,” *arXiv preprint arXiv:1707.09835*, 2017.

[9] T. Li and et al., “Federated optimization in heterogeneous networks,” *arXiv preprint arXiv:1812.06127*, 2018.

[10] C. Dinh and et al., “Federated learning over wireless networks,” in *Proc. IEEE INFOCOM*, 2019.

[11] X. Zhang and et al., “Enabling execution assurance of federated learning at untrusted participants,” in *Proc. IEEE INFOCOM*, 2020.

[12] X. Zhang, M. Hong, S. V. Dhople, W. Yin, and Y. Liu, “Fedpd: A federated learning framework with optimal rates and adaptivity to non-iid data.” *arXiv preprint arXiv:2005.11418*, 2020.

[13] C. Fan and P. Liu, “Federated generative adversarial learning,” in *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, 2020.

[14] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, “Cost-effective federated learning design,” *IEEE InfoCom*, 2021.

[15] T. Wang, Y. Liu, X. Zheng, H.-N. Dai, W. Jia, and M. Xie, “Edge-based communication optimization for distributed federated learning,” *IEEE Transactions on Network Science and Engineering*, 2021.

[16] W. Sun, N. Xu, L. Wang, H. Zhang, and Y. Zhang, “Dynamic digital twin and federated learning with incentives for air-ground networks,” *IEEE Transactions on Network Science and Engineering*, 2020.

[17] Y. Gao, L. Liu, B. Hu, T. Lei, and H. Ma, “Federated region-learning for environment sensing in edge computing system,” *IEEE Transactions on Network Science and Engineering*, 2020.

[18] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in *NIPS*, 2017.

[19] T. Munkhdalai and H. Yu, “Meta networks,” in *ICML*, 2017.

[20] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, “Meta-transfer learning for few-shot learning,” *CVPR*, 2019.

[21] S. Gidaris and N. Komodakis, “Dynamic few-shot visual learning without forgetting,” in *CVPR*, 2018.

[22] F. Sung and et al., “Learning to compare: Relation network for few-shot learning,” *CVPR*, 2018.

[23] C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, “Online meta-learning,” *NIPS*, 2019.

[24] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, “Panet: Few-shot image semantic segmentation with prototype alignment,” in *ICCV*, 2019.

[25] H. Tang, Z. Li, Z. Peng, and J. Tang, “Blockmix: Meta regularization and self-calibrated inference for metric-based meta-learning,” in *ACM Multimedia*, 2020.

[26] J. He, R. Hong, X. Liu, M. Xu, Z.-J. Zha, and M. Wang, “Memory-augmented relation network for few-shot learning,” in *ACM Multimedia*, 2020.

[27] Z. Wang, Y. Zhao, J. Li, and Y. Tian, “Cooperative bi-path metric for few-shot learning,” in *ACM Multimedia*, 2020.

[28] A. Li, W. Huang, X. Lan, J. Feng, Z. Li, and L. Wang, “Boosting few-shot learning with adaptive margin loss,” in *CVPR*, 2020.

[29] C. Fan, J. Hu, and J. Huang, “Few-shot multi-agent perception,” *ACM Multimedia*, 2021.

[30] F. Chen and et al., “Federated meta-learning with fast convergence and efficient communication,” *arXiv preprint arXiv:1802.07876*, 2018.

[31] Z. Chen, P. Tian, W. Liao, and W. Yu, “Zero knowledge clustering based adversarial mitigation in heterogeneous federated learning,” *IEEE Transactions on Network Science and Engineering*, 2021.

[32] J. Li, M. Khodak, S. Caldas, and A. Talwalkar, “Differentially private meta-learning,” in *ICLR*, 2020.

[33] T. Yu, E. Bagdasaryan, and V. Shmatikov, “Salvaging federated learning by local adaptation,” *arXiv preprint arXiv:2002.04758*, 2020.

[34] Q. Wu, K. He, and X. Chen, “Personalized federated learning for intelligent iot applications: A cloud-edge based framework,” *IEEE Computer Graphics and Applications*, 2020.

[35] A. Fallah, A. Mokhtari, and A. E. Ozdaglar, “Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach,” in *NeurIPS*, 2020.

[36] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, “Deep mutual learning,” in *CVPR*, 2018.

[37] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, “On the convergence of fedavg on non-iid data,” in *ICLR*, 2019.

[38] Y. Wu and et al., “Improving gan training with probability ratio clipping and sample reweighting,” in *NeurIPS*, 2020.

[39] J. Schulman and et al., “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.

[40] I. Goodfellow and et al., “Generative adversarial nets,” in *NIPS*, 2014.

[41] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, “Maximum classifier discrepancy for unsupervised domain adaptation,” in *CVPR*, 2018.

[42] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.

[43] B. Oreshkin, P. R. López, and A. Lacoste, “Tadam: Task dependent adaptive metric for improved few-shot learning,” in *NIPS*, 2018.

[44] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Song, “Metagan: An adversarial approach to few-shot learning,” in *NIPS*, 2018.

[45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *CVPR*, 2009.

[46] S. Caldas and et al., “Leaf: A benchmark for federated settings,” *arXiv preprint arXiv:1812.01097*, 2018.

[47] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *EMNLP*, 2014.

[48] T.-M. H. Hsu, H. Qi, and M. Brown, “Measuring the effects of non-identical data distribution for federated visual classification,” *arXiv preprint arXiv:1909.06335*, 2019.