FedFM: Anchor-Based Feature Matching for Data Heterogeneity in Federated Learning

Rui Ye, Zhenyang Ni, Chenxin Xu, Jianyu Wang, Siheng Chen, *Senior Member, IEEE*, and Yonina C. Eldar, *Fellow, IEEE*

***Abstract*—One of the key challenges in federated learning (FL) is local data distribution heterogeneity across clients, which may cause inconsistent feature spaces across clients. To address this issue, we propose Federated Feature Matching (FedFM), which guides each client's features to match shared category-wise anchors (landmarks in feature space). This method attempts to mitigate the negative effects of data heterogeneity in FL by aligning each client's feature space. We tackle the challenge of varying objective functions in theoretical analysis and provide convergence guarantee for FedFM. In FedFM, to mitigate the phenomenon of overlapping feature spaces across categories and enhance the effectiveness of feature matching, we propose a feature matching loss called contrastive-guiding (CG), which guides each local feature to match with the corresponding anchor while keeping away from non-corresponding anchors. Additionally, to achieve higher efficiency and flexibility, we propose a FedFM variant, called FedFM-Lite, which enables flexible trade-off between algorithm utility and communication bandwidth cost. Through extensive experiments, we demonstrate that FedFM with CG outperforms seven classical and representative works by quantitative and qualitative comparisons. FedFM-Lite can achieve better performance than state-of-the-art methods with five to ten times less communication costs.**

***Index Terms*—Federated learning, data heterogeneity, feature matching, contrastive-guiding.**

I. Introduction

Most existing deep learning models are trained in a centralized manner. However, in practice, data may be distributed on several parties and may not be collected due to increasing privacy concerns. Federated learning (FL) [1] is proposed to address this issue and has become an emerging research topic [1], [2], [3], [4], [5], [6], [7]. In standard FL [1], each client first downloads the same global model from the server and conducts local model training on its private dataset. Then, clients upload their trained local models to the server, where a global model is updated via aggregating local models. This process is conducted iteratively to obtain a final global model. This privacy-preserving method has been widely explored applied to many tasks, such as image classification [8], [9], language modeling [10], speech recognition [11].

One of the key challenges that hinders FL from performing as well as centralized learning is data distribution heterogeneity across clients [3], [12], [13]. Due to diverse conditions of devices and application scenarios, data might not be independent and identical distributed (IID) across local clients. This may result in large variations in the locally trained models on clients and slow down convergence of the global model [14], [15]. This phenomenon is also referred to as *client drift* [7], [13].

To tackle the above mentioned data heterogeneity issue, most previous works [6], [7], [16] focus on model-level corrections, which intend to reduce the variations in locally trained models. However, these algorithms fail to ensure the consistency of multiple local models' feature spaces. It is possible that different local models have drastically misaligned feature spaces. This could lead to unclear decision boundaries and cause misclassification, which significantly differs from centralized learning. Fig. 1(a) empirically shows the T-SNE [17] of two local clients' features in FedAvg [1], where the color indicates categories and the shape indicates clients. We see that data samples with the same color, yet different shapes do not overlap, reflecting the two local models fail to share a consistent feature space. In addition, data samples with the same shape, yet different colors overlap with each other. This can be detrimental to classification tasks. Motivated by this, our work focuses on mitigating data heterogeneity in federated classification tasks through aligning the feature spaces across multiple local models.

In this article, we propose an anchor-based **Fed**erated **F**eature **M**atching (FedFM) method, the key idea of which is to leverage landmarks shared by all clients to provide global positioning, promoting a more consistent feature space. As a core concept of FedFM, we define landmarks as the average of features for the same class/category and name them as *anchors*. In each round

Manuscript received 12 October 2022; revised 17 April 2023 and 16 June 2023; accepted 12 August 2023. Date of publication 16 October 2023; date of current version 20 November 2023. This work was supported in part by the National Key R&D Program of China under Grant 2021ZD0112801, in part by the NSFC under Grant 62171276, and the Science and Technology Commission of Shanghai Municipal under Grants 21511100900 and 22DZ2229005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Meisam Razaviyayn. *(Corresponding author: Siheng Chen.)*

Rui Ye, Zhenyang Ni, and Chenxin Xu are with the Cooperative Medianet Innovation Center (CMIC), Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yr991129@sjtu.edu.cn; 0107nzy@sjtu.edu.cn; xcxwakaka@sjtu.edu.cn).

Jianyu Wang is with Meta Platforms, Menlo Park, CA 94025 USA (e-mail: jianyuwang@meta.com).

Siheng Chen is with Shanghai Jiao Tong University, Shanghai 200240, China and also with Shanghai AI Laboratory, Shanghai 200232, China (e-mail: sihengc@sjtu.edu.cn).

Yonina C. Eldar is with the Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 7610001, Israel (e-mail: yonina.eldar@weizmann.ac.il).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2023.3314277, provided by the authors.

Digital Object Identifier 10.1109/TSP.2023.3314277

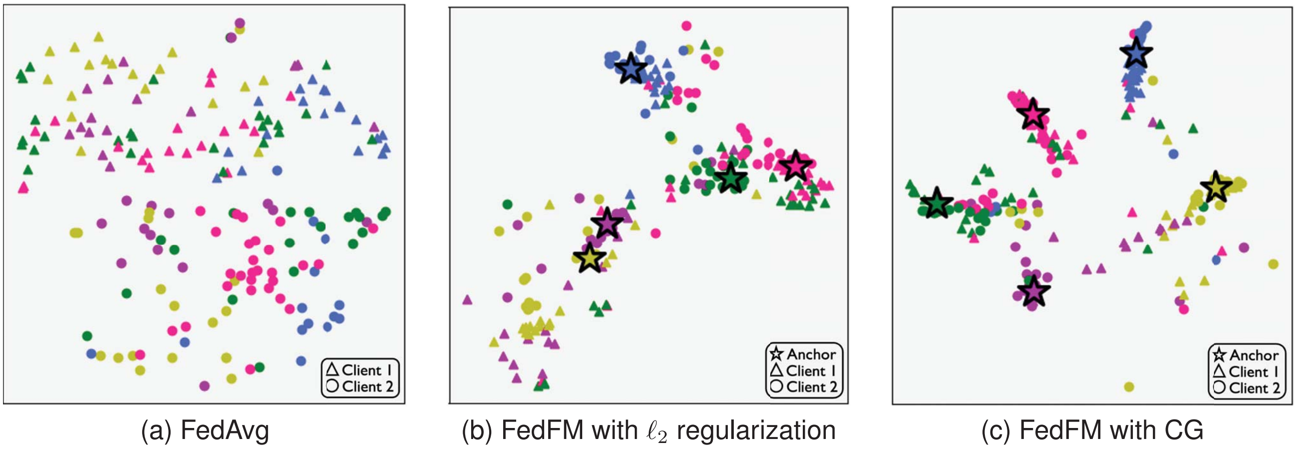


Fig. 1. FedFM alleviates the inconsistency in feature space by anchor-based feature matching. Here, we plot features of five categories for simplicity and each color denotes one category. (a) In existing methods, there is a large gap between samples of two clients (triangle and circle) in the feature space. (b) With simple ℓ_2 regularization, our FedFM leverages anchors (stars) to align the feature spaces of two clients. (c) With the proposed contrastive-guiding (CG) method, FedFM achieves more precise and compact matching. Quantitatively, the normalized mutual information (NMI) results are $0.30, 0.37, 0.40$, respectively; while the silhouette scores (SS) [18] for the three methods are $-0.07, -0.03, 0.05$, respectively.

of FedFM, there are two key steps: (1) anchor updating; and (2) anchor-based model updating. In the anchor updating step, first, each client calculates the local anchors; second, by interacting with the server, global anchors are updated by aggregating local anchors and sent back to each client. In the anchor-based model updating step, each client's feature is pushed to match with the global anchor of the corresponding category during local model training; see the significant improvement in Fig. 1(b), where we regularize the ℓ_2 distance between a feature and its corresponding global anchor to enhance the consistency of feature spaces across clients. Global anchors are denoted by star shape.

We then provide a convergence guarantee for our proposed FedFM algorithm. Unlike most existing literature that analyzes fixed objective functions, the analysis of FedFM faces a distinctive challenge of time-varying objective functions over rounds. This is because of the varying global anchors, which are updated at each round. We overcome this challenge by proving a key lemma, which suggests that updating of global anchors also contributes to optimizing the global objective. Based on standard assumptions and this key lemma, we proceed the convergence analysis of FedFM. The theoretical results show that the proposed FedFM converges at a rate that accords with the convergence results in many existing FL literature [14], [19].

To promote more precise and effective feature matching and push the feature spaces of different categories to be far away from each other, we further propose contrastive-guiding (CG) for feature matching in FedFM. The proposed CG guides each client's local feature to match with the corresponding global anchor while keeping away from non-corresponding global anchors. Comparing with the standard ℓ_2 regularization, CG contributes to more precise feature matching, which results in more distant and compact category-wise feature space; see its improvement over ℓ_2 regularization in Fig. 1(c).

To achieve higher efficiency and flexibility, we empirically propose a variant of FedFM, called FedFM-Lite. Compared with FedFM, FedFM-Lite communicates one time within one FL round and thus requires less synchronization times (handshakes) among clients and server. FedFM-Lite is also more flexible to communicate anchors and models at different frequency. Since models have significantly more communication bandwidth cost, we propose to communicate models at a relatively lower frequency, which is capable to accord with various real-world communication budgets. Generally, FedFM-Lite is a more flexible in practice though its convergence can not be guaranteed in the current theoretical framework.

Finally, through extensive experiments, we verify that FedFM with CG outperforms state-of-the-art FL methods, including FedAvg [1], FedAvgM [9], FedProx [6], SCAFFOLD [7], FedDyn [16], FedNova [19] and MOON [20], on three data heterogeneity types and three datasets, including CIFAR-10 [21], CINIC-10 [22] and CIFAR-100. We further visualize the feature space constructed by the proposed FedFM with CG, which qualitatively demonstrates its effectiveness. We also see that FedFM-Lite can achieve better performance than existing methods with five to ten times less communication costs and comparable performance compared with FedFM with half of the synchronization times.

Our main contributions are as follows:

1) We propose an anchor-based federated feature matching (FedFM) method and a contrastive-guiding (CG) technique in FedFM, which pushes each client's local feature to match with corresponding shared global anchor while keeping away from non-corresponding anchors, promoting a consistent feature space across clients and mitigating the notorious data heterogeneity issue;
2) We tackle the distinctive challenge of varying objective function in the theoretical analysis of FedFM and provide a convergence guarantee;
3) We propose an efficient and flexible variant, FedFM-Lite, which can be easily adjusted to accord with various real-world communication budgets;
4) We conduct extensive experiments and show that FedFM with CG (and FedFM-Lite) can significantly outperform state-of-the-art methods.

This article is organized as the following. Section II reviews related works. Section III presents several preliminaries including notations and motivations. Section IV describes and discusses our proposed FedFM method and the CG technique. Section V provides convergence analysis for FedFM. Section VI proposes a variant of FedFM, FedFM-Lite. Section VII shows the experimental results.

II. Related Work

A. Federated Learning

Federated learning (FL) is proposed in [1] and has been widely applied to many fields. In image processing, it is widely adopted since it takes advantage of the computational ability and locally-stored data of edge devices [8], [9]. It also attracts much attention in healthcare due to its privacy-preserving property [23], [24], [25]. But when the standard FL method FedAvg [1] meets the situation of data distribution heterogeneity across clients, the global model could move far away from the true global optimum due to the large variations in each local optima, which is referred to as client drift [7]. There have been numerous works trying to tackle this issue and two key approaches are local correction and global adjustment.

Local Correction. One main approach is conducting correction during the process of local model training, which aims to reduce the difference among trained local models. Most previous works conduct correction on the *model-level*. FedProx [6] applies a l_2-norm distance regularization between the current local model and the previous global model. FedDyn [16] proposes a dynamic regularizer to align local and global solutions. Variance reduction methods, such as SCAFFOLD [7] and VRLSGD [26], utilize the previous difference between local and global gradient to debias the gradient at each local training step. MOON [20] maximizes the similarity between the feature (intermediate layer output) of current local model and that of the previous global model, which requires twice more computation cost than FedAvg and has no convergence guarantee. As a concurrent work, FedFA [27] applies ℓ_2 regularization on feature space but has no theoretical convergence analysis. Our proposed FedFM conducts local correction at the level of *feature*, which employs shared category-wise global anchors to guide local feature learning. That is, MOON [20] aligns features of two models that belong to the same sample and FedFM aligns features of all samples that belong to the same category. We also provide a convergence guarantee of FedFM, which is the first in feature-based generalized FL methods.

Global Adjustment. Another key direction is global adjustment during the process of model interaction, which aims to obtain a better global model utilizing the uploaded local models [28], [29]. FedAvgM [9] and FedOPT [30] introduce momentum to global model updating, which stabilizes the global model optimization. FedNova [19] normalizes local updates according to the number of SGD steps, which eliminates objective inconsistency and achieves fast convergence. Using the Knowledge-Distillation technique, FedGen [31] learns a generator to assist local model training. FedDF [32] and FedFTG [33] refine the global model by learning from the uploaded local models. Our proposed FedFM is orthogonal to these global adjustment methods and can be easily incorporated with these techniques.

TABLE I
Related Work Comparisons. Conv. Denotes Convergence Guarantee. Mem. and Band. Denote Memory Cost and Bandwidth Cost Roughly Compared With FedAvg

Method	Local Correction	Conv.	Mem.	Band.
FedAvg [1]	-	✓	× 1	× 1
FedAvgM [9]	-	✓	× 1	× 1
FedProx [6]	Model	✓	× 2	× 1
SCAFFOLD [7]	Model	✓	× 2	× 2
FedDyn [16]	Model	✓	× 2	× 1
FedNova [19]	-	✓	× 1	× 1
MOON [20]	Feature	-	× 3	× 1
FedFM (Ours)	Feature	✓	× 1	× 1

As the above local correction and global adjustment methods, the focus of this article is generalized FL, which aims at collaboratively training one global model. Personalized FL aims at collaboratively training multiple personalized local models, including FedRep [34], FedAMP [35], pFedMe [36] and Personalized FedAvg [37]. Targeting personalized FL, FedProto [38] and FedPAC [39] utilize prototype to provide extra feature information from other clients and FLT [40] utilizes prototypes to calculate relatedness among clients to enhance personalization of each client. In comparison, our FedFM targets generalized FL, which uses anchors as landmarks to align clients' category-wise feature spaces to enhance generalization of all clients. We also propose a new contrastive-guiding (CG) technique. CG pushes local features close to the corresponding anchor and keeps them far away from non-corresponding anchors. Some works such as k-FED [41] focus on unsupervised tasks by using anchors as measurements for clustering. While we focus on supervised task by using anchors for aligning feature spaces of clients.

We compare FedFM with several representative methods in generalized FL in Table I.

B. Contrastive Learning

Contrastive loss [42] is proposed for deep metric learning, which aims to minimize the embedding distance between two inputs from the same category but maximizes the distance otherwise. Following this, many variants of contrastive loss are proposed [43], [44], [45], [46], [47], [48]. This technique is then applied to both self-supervised learning [49], [50], [51] and supervised learning [52], [53], [54] to enhance feature learning. These works all focus on centralized learning without considering privacy while our works apply contrastive feature learning for supervised FL in a privacy-preserving way. Recently, two concurrent works consider contrastive learning in FL. FedProc [55] empirically proposes prototypes-based contrastive learning while we propose an optimization problem and derive FedFM from it, where the contrastive-guiding loss is one variant of our framework. FedPCL [56] focuses on personalized FL and relies on pre-trained models, while ours focuses on generalized FL and is applicable whether pre-trained models are available. More importantly, both FedProc and FedPCL do not provide any convergence analysis. To the best of our knowledge, among those works that conduct feature-level alignment in generalized

FL [20], [27], we are the first one to provide a theoretical convergence guarantee, which could potentially motivate more theoretical and empirical future works.

III. Preliminaries

In this section, we present several key notations and the general process of FL. Then, we demonstrate two key empirical observations through preliminary experiments, including inconsistent feature spaces across clients and overlapping feature spaces across categories, which motivate the proposal of our FedFM method and CG loss.

A. Background of FL

Suppose there are K clients, where the kth client holds a local dataset $\mathcal{B}_k = \{(\mathbf{x}_i, c_i) | i = 1, 2, ..., |\mathcal{B}_k|\}$. Here, $\mathbf{x}_i$ and c_i are the data and the label of the ith sample, respectively. FL aims to leverage the local datasets at multiple clients to collectively train a global model $\mathbf{w}$ in the server without sharing raw data [1]. We focus on a C-classification task and each local dataset $\mathcal{B}_k$ can be further split to C category-wise sub-datasets, each of which is $\mathcal{B}_{k,c} = \{(\mathbf{x}_i, c_i) \in \mathcal{B}_k | c_i = c\}$. Let $f_{\text{full}}(\cdot, \cdot)$ be an end-to-end classification model with $f_{\text{full}}(\mathbf{w}, \mathbf{x}) \in \mathbb{R}^C$ the final classification output given the input data sample $\mathbf{x}$ and model parameters $\mathbf{w}$. We also consider the intermediate features as $f_{\text{mid}}(\mathbf{w}, \mathbf{x}) \in \mathbb{R}^d$, where $f_{\text{mid}}(\cdot, \cdot)$ denotes the feature-extract module in the full classification model $f_{\text{full}}(\cdot, \cdot)$. A standard global objective of FL is

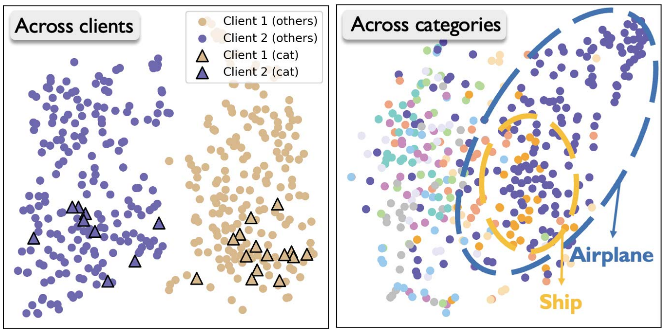
$$F(\mathbf{w}) = \sum_{k=1}^{K} p_k F_k(\mathbf{w}) = \sum_{k=1}^{K} \frac{p_k}{|\mathcal{B}_k|} \sum_{(\mathbf{x},c)\in\mathcal{B}_k} \ell(f_{\text{full}}(\mathbf{w}, \mathbf{x}), c), \tag{1}$$

where $p_k = |\mathcal{B}_k| / \sum_{k=1}^{K} |\mathcal{B}_k|$ is the aggregation weight of the kth client and $\ell(\cdot)$ is the task-specific loss function. To optimize the objective in a federated setting, at every communication round t, each client k downloads the same global model $\mathbf{w}^{(t)}$ and conducts τ iterations of SGD. Normally, the supervision is applied on final output $f_{\text{full}}(\mathbf{w}, \mathbf{x})$ while we could also consider supervision on the feature $f_{\text{mid}}(\mathbf{w}, \mathbf{x})$. Then, each client k uploads the updated model $\mathbf{w}_k^{(t,\tau)}$ to the server, which is aggregated to update the global model $\mathbf{w}^{(t+1)}$ for the next round.

Since each local dataset $\mathcal{B}_k$ could have different data distributions, the standard method (1) could result in a divergent global model and degraded performance. In this work, our goal is to introduce a regularization term to the global objective, mitigating the effects of data heterogeneity.

B. Motivation

Most FL methods do not explore clients' behavior in feature space. We conduct the following FL (FedAvg [1]) experiment on CIFAR-10 with two clients. Each holds an imbalanced dataset of size 5000, where Client 1 (2) has 50% of data in airplane (car) category while the rest is equally distributed to 9 other categories. Each client runs 10 epochs of local model training based on the same initial model. We use a ResNet18 [57] model to implement $f_{\text{full}}(\cdot, \cdot)$, where the feature-extraction module $f_{\text{mid}}(\cdot, \cdot)$ is ResNet18 without the last fully-connected layer. We then visualize intermediate features of several validation sample by T-SNE [17]. From the experimental results, we notice two unsatisfying phenomena in feature space that might cause bad performance in FL.



(a) Inconsistent features (b) Overlapping features

Fig. 2. Motivating examples. (a) Visualizes the scatter plot of intermediate features of all the samples across two clients, where each color denotes one client. The triangles are the features of cat category. This shows that although each client has learned well-clustered features, the features are quite inconsistent across clients. (b) Shows the scatter plot of intermediate features of all samples in Client 1, while each color denotes one category. This shows that airplane category occupies larger feature space and overlap with that of ship category.

Inconsistent feature spaces across clients. Fig. 2(a) illustrates the scatter plot of several samples' features from two clients, where two colors indicate two different clients and cat category is highlighted in the triangle shape. We see that the samples from cat category across two local clients have a huge gap, reflecting that the feature spaces of two local clients are seriously inconsistent. This distinctly differs from centralized training, where samples of the same category are gathered without an obvious gap. The intuition behind this phenomenon might be that two clients have two significantly different data distributions, resulting in distinct local models and therefore inconsistent behaviors in feature space. Motivated by this, we propose FedFM (anchor-based federated feature matching) to align the category-wise feature spaces across clients. The core idea is to establish shared global anchors as the landmarks in the feature space to guide feature learning across multiple clients; see details in Section IV-B.

Overlapping feature spaces across categories. Fig. 2(b) illustrates the scatter plot of several samples' features from various categories within Client 1, where various colors indicate different categories. We see that the samples from two categories (airplane and ship) greatly overlap, causing misclassification. This phenomenon often happens when the sample sizes across multiple categories are highly imbalanced. Motivated by this, we further propose a contrastive-guiding method in FedFM, which pushes each feature close to its corresponding anchor while keeping far away from non-corresponding anchors to avoid overlapping. This also ends up enlarging the distance between two distinct categories in the feature space and further mitigates overlapping; see details in Section IV-C.

IV. Methodology

This section introduces the proposed federated learning with anchor-based feature matching (FedFM) from both aspects of mathematical optimization and federated implementation. Based on the proposed framework, we further propose a contrastive-guiding (CG) loss to mitigate overlapping feature spaces across categories. Finally, we discuss the communication cost and privacy concerns.

A. Optimization Problem

To address the issue of inconsistent feature spaces across clients, we propose anchor-based feature matching, which introduces anchors to serve as the shared landmarks for aligning all the clients' feature spaces. Mathematically, let $\mathcal{A} = \{\mathbf{a}_c\}_{c=1}^{C}$ be a global anchor set, where $\mathbf{a}_c$ is the anchor of the cth category. The overall optimization problem with respect to the model parameter $\mathbf{w}$ and the anchor set $\mathcal{A}$ is

$$\begin{aligned}\min_{\mathbf{w},\mathcal{A}} \Phi(\mathbf{w};\mathcal{A}) &= \min_{\mathbf{w},\mathcal{A}} \sum_{k=1}^{K} p_k \Phi_k(\mathbf{w};\mathcal{A}) \\ &= \min_{\mathbf{w},\mathcal{A}} \sum_{k=1}^{K} p_k \Big(F_k(\mathbf{w}) + \lambda Q_k(\mathbf{w};\mathcal{A}) \Big), \end{aligned} \tag{2}$$

where p_k is the predefined aggregation weight of the kth client, with relative dataset size a standard choice $|\mathcal{B}_k|/\sum_{k=1}^{K}|\mathcal{B}_k|$, $\Phi_k(\cdot)$ is the kth client's objective, λ is a hyperparameter to balance the task-specific loss and the regularization term, $F_k(\mathbf{w}) = \sum_{(\mathbf{x},c)\in\mathcal{B}_k} \ell\left(f_{\text{full}}(\mathbf{w},\mathbf{x}), c\right)/|\mathcal{B}_k|$ is the task-specific loss at the kth client with $\mathcal{B}_k$ the kth client's local dataset and

$$\begin{aligned} Q_k(\mathbf{w};\mathcal{A}) &= \sum_{(\mathbf{x},c)\in\mathcal{B}_k} \frac{1}{|\mathcal{B}_k|} q(f_{\text{mid}}(\mathbf{w},\mathbf{x}), \mathcal{A}|c) \\ &= \sum_{(\mathbf{x},c)\in\mathcal{B}_k} \frac{1}{|\mathcal{B}_k|} \|f_{\text{mid}}(\mathbf{w},\mathbf{x}) - \mathbf{a}_c\|_2^2 \end{aligned} \tag{3}$$

is the kth client's anchor-based feature matching term, which forces each sample to match with the corresponding category-wise global anchor at each client. Since each global anchor is the proxy of each category and is shared across all the clients, with the anchor-based matching, the feature space at each client is evolving towards the same formation, enhancing the feature consistency across clients. Without anchor-based regularization term $Q_k(\mathbf{w};\mathcal{A})$, the overall objective degenerates to the standard federated learning objective.

To solve the optimization (2), we sequentially optimize the anchor set $\mathcal{A}$ and the model parameter $\mathbf{w}$ at each round t.

a) Optimizing global anchors $\mathcal{A}$: Fixing the model parameter at the previous round, $\mathbf{w}^{(t)}$, we optimize over the anchor set $\mathcal{A}$ by solving

$$\mathcal{A}^{(t)} = \arg\min_{\mathcal{A}} \Phi(\mathbf{w}^{(t)};\mathcal{A}) = \sum_{k=1}^{K} p_k \Phi_k(\mathbf{w}^{(t)};\mathcal{A}). \tag{4}$$

Since the task-specific loss has nothing to do with the anchors, the optimal anchor only relates to the anchor-based regularization term. Furthermore, the global anchor of the cth category only depends on the data sample belonging to the cth category. Mathematically, the global anchor of the cth category has a straightforward closed-form solution as

$$\begin{aligned} \mathbf{a}_c^{(t)} &= \arg\min_{\mathbf{a}} \sum_{k=1}^{K} \sum_{(\mathbf{x},c)\in\mathcal{B}_k} \left\| f_{\text{mid}}(\mathbf{w}^{(t)},\mathbf{x}) - \mathbf{a}_c \right\|_2^2 \\ &= \frac{1}{\sum_{k=1}^{K}|\mathcal{B}_{k,c}|} \sum_{k=1}^{K} \sum_{(\mathbf{x},c)\in\mathcal{B}_{k,c}} f_{\text{mid}}(\mathbf{w}^{(t)},\mathbf{x}), \end{aligned} \tag{5}$$

where $\mathcal{B}_{k,c}$ is the kth client's local dataset that belongs to the cth category. However, in the federated learning setting, since the global server cannot directly access the local data, each global anchor cannot be directly computed in the global server as in (5). In Section IV-B, we will consider a federated implementation.

b) Optimizing global model $\mathbf{w}$: Fixing the global anchors $\mathcal{A}^{(t)}$, we optimize over the model parameter $\mathbf{w}$ by solving

$$\mathbf{w}^{(t+1)} = \arg\min_{\mathbf{w}} \Phi(\mathbf{w};\mathcal{A}^{(t)}) = \sum_{k=1}^{K} p_k \Phi_k(\mathbf{w};\mathcal{A}^{(t)}). \tag{6}$$

We consider an iterative solver based on the standard gradient descent. Mathematically, the global model parameters are updated as

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \sum_{k=1}^{K} p_k \left(\frac{\partial F_k(\mathbf{w})}{\partial \mathbf{w}} + \lambda \frac{\partial Q_k(\mathbf{w};\mathcal{A}^{(t)})}{\partial \mathbf{w}} \right), \tag{7}$$

where η is the step size. Similarly to the optimization of global anchors, the model parameters cannot be directly updated in the global server as in (7). A federated implementation is introduced next.

B. Federated Implementation

Previously, we propose the mathematical optimization of federated learning with anchor-based feature matching. However, the solver is impractical due to the federated setting. Here we introduce a federated implementation, called FedFM.

1) Overview: Fig. 3 overviews the proposed FedFM. In each communication round, it consists of two main steps: anchor updating and model updating, where anchor updating solves the subproblem (4) and model updating solves the subproblem (6).

In the step of **anchor updating**, after downloading the same global model, each client calculates local anchors and uploads them to the server, where global anchors are updated by aggregating local anchors. These global anchors are then broadcast to be shared by all clients. In the step of **model updating**, each client conducts several iterations of local model training supervised by task-driven loss as well as an anchor-based feature matching loss, after which the updated local model is uploaded to server. Then, the server updates the global model by aggregating local models.

We now illustrate these two steps in detail.

a) Step 1: Anchor Updating: Here we aim to implement (5) in a federated fashion, which requires the coordination of both the clients and the server. This involves two substeps: local anchors calculation for integrating features of the same

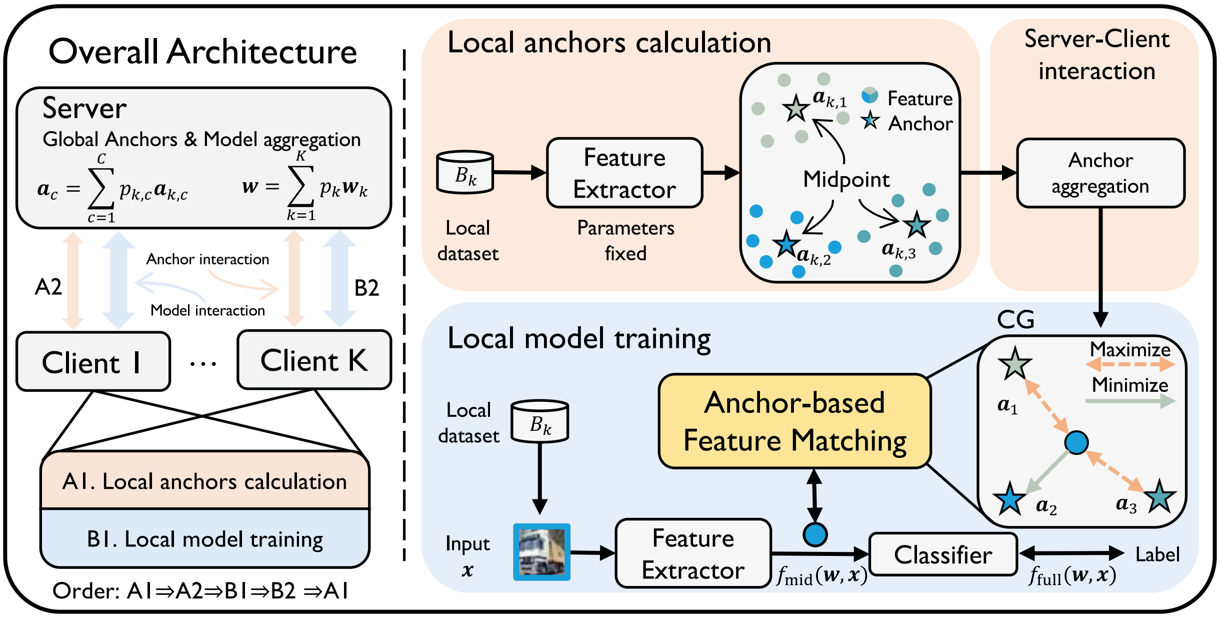


Fig. 3. Overview of FedFM for a 3-classification task. The left shows the overall architecture. The right shows two key steps in detail, where local anchors calculation generates feature anchors for each category and local model training utilizes these anchors to conduct anchor-based feature matching (e.g. contrastive guiding in the figure, which is described in Section IV-C). Here, a feature (in circle shape) is the intermediate layer output of a model and an anchor (in star shape) is an integration of features that belong to the same category.

Algorithm 1 FedFM

```
Initialization: Global model $\mathbf{w}^{(0)}$.
for $t = 0, 1, ..., T-1$ do
    Sends global model $\mathbf{w}^{(t)}$ to initialize each client $\mathbf{w}_k^{(t,0)}$
    $\mathcal{A}_k^{(t)} \leftarrow$ Local Anchors Calculation ($\mathbf{w}_k^{(t,0)}$) using (8)
    $\mathcal{A}^{(t)} \leftarrow$ Global Anchors Aggregation ($\{\mathcal{A}_k^{(t)}\}_{k=1}^K$) using (9)
    $\mathbf{w}_k^{(t,\tau)} \leftarrow$ Local Model Training ($\mathbf{w}_k^{(t,0)}$,$\mathcal{A}^{(t)}$) for $\tau$ iterations using (10)
    $\mathbf{w}^{(t+1)} \leftarrow$ Global Model Aggregation ($\{\mathbf{w}_k^{(t,\tau)}\}_{k=1}^K$) using (11)
end for
return final global model $\mathbf{w}^{(T)}$
```

category, working on the client side, and global anchors aggregation for aggregating anchors from all clients, working on the server side.

Local anchors calculation. After downloading the global model, at the start of each new round, each client conducts local anchors calculation by computing the category-wise midpoints of features; that is, local anchors are integration of features from the same category in each client. Mathematically, let $\mathbf{w}_k^{(t,r)}$ be the kth client's model parameter at the tth communication round with iteration r and $\mathbf{w}_k^{(t,0)} := \mathbf{w}^{(t)}$, which means that the local client's model parameter at iteration 0 is initialized by the global model in the previous communication round. Then, the local anchor of category c in client k at round t is calculated as

$$\mathbf{a}_{k,c}^{(t)} = \frac{1}{|\mathcal{B}_{k,c}|}\sum_{(\mathbf{x},c)\in\mathcal{B}_{k,c}} f_{\text{mid}}(\mathbf{w}^{(t)}, \mathbf{x}) \in \mathbb{R}^d, \tag{8}$$

where $f_{\text{mid}}(\mathbf{w}, \mathbf{x})$ is the intermediate layer output (feature) of a model $\mathbf{w}$ given a sample $\mathbf{x}$. Client k performs the calculation for each category c and sends all C local anchors to the server.

Global anchors aggregation. The server conducts global anchors aggregation by aggregating local anchors from clients so that global anchors are integration of features from the same category across all clients. Mathematically, receiving the local anchors from all the K clients, the server conducts the following dataset size weighted aggregation for each category c to obtain global anchor; that is,

$$\begin{aligned}\mathbf{a}_c^{(t)} &:= \frac{1}{\sum_{k=1}^K |\mathcal{B}_{k,c}|}\sum_{k=1}^K |\mathcal{B}_{k,c}|\mathbf{a}_{k,c}^{(t)} \\ &= \frac{1}{\sum_{k=1}^K |\mathcal{B}_{k,c}|}\sum_{k=1}^K \sum_{(\mathbf{x},c)\in\mathcal{B}_{k,c}} f_{\text{mid}}(\mathbf{w}^{(t)}, \mathbf{x}) \in \mathbb{R}^d. \end{aligned} \tag{9}$$

We see that each global anchor $\mathbf{a}_c^{(t)} \in \mathbb{R}^d$ still matches with the optimized results in (5) and is the midpoint of features of all data that belongs to $\mathcal{B}_{k,c}$, $k \in \{1, 2, ..., K\}$. All these global anchors $\mathcal{A}^{(t)} = \{\mathbf{a}_c^{(t)}\}_{c=1}^C$ are then broadcast to be shared by all clients. At this point, the acquired shared anchors are representative of the whole dataset. These are then used to guide the feature learning at each client's local model training.

b) Step 2: Model Updating: Here we implement the part of the update in (7) in a federated fashion, which involves two substeps: local model training for updating local models, working on the client side, and global model aggregation for aggregating local models, working on the server side.

Local model training. Each client conducts local model training with more than one SGD step on its private dataset with the task supervision and anchor-based feature matching loss.

The kth local client's model parameter at communication round t with iteration r is updated as

$$\mathbf{w}_k^{(t,r+1)} = \mathbf{w}_k^{(t,r)} - \eta\Big(\frac{\partial F_k(\mathbf{w})}{\partial \mathbf{w}} + \lambda \frac{\partial Q_k(\mathbf{w};\mathcal{A}^t)}{\partial \mathbf{w}}\Big), \tag{10}$$

where $F_k(\cdot)$ and $Q_k(\cdot)$ are the kth client's task-specific loss and anchor-based feature matching loss (2), respectively. After τ iterations of local training, each client k obtains a local model with parameters $\mathbf{w}_k^{(t,\tau)}$ and uploads it to the server.

Global model aggregation. The server receives and aggregates the local models from all the clients and obtains an updated global model for the next round:

$$\mathbf{w}^{(t+1)} \leftarrow \sum_{k=1}^{K} p_k \mathbf{w}_k^{(t,\tau)}, \tag{11}$$

where p_k is the predefined aggregation weight.

2) Strengths of Feature Matching: The proposed anchor-based feature matching can significantly relieve the inconsistency phenomenon in Section III-B. Since each client's features are trained to match the shared global anchors, the discrepancy of learned feature spaces across clients is reduced. Therefore, clients' models establish a more consistent feature space of every category. Furthermore, global anchors contain overall information since they are obtained by aggregating all local anchors. With global anchors, information of other clients is infused into each client, which provides additional guidance on local feature learning especially those categories with relatively few data samples.

C. Contrastive-Guiding Loss

1) Method: To address the problem of overlapping feature space across categories in Section III-B, we further propose contrastive-guiding (CG) loss to replace the ℓ_2-based loss in the feature matching term (3). The idea is to force each feature to be close to the corresponding anchor while keeping far away from non-corresponding anchors. Let $\mathcal{A}^{(t)} = \{\mathbf{a}_n^{(t)}\}_{n=1}^{C}$ be the global anchors and $\mathbf{w}_k^{(t,r)}$ be the local model for client k at training round t and iteration r. For data sample $\mathbf{x}$, the feature matching loss is

$$q\left(f_{\text{mid}}(\mathbf{w}_k^{(t,r)}, \mathbf{x}), \mathcal{A}^{(t)}|c\right) \; = \; \mathcal{L}_{CE}(\mathbf{s}, c),$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss function and $\mathbf{s} = [s_1, s_2, \ldots, s_C] \in \mathcal{A}^C$ is a similarity vector, whose nth element measures the distance with the nth anchor:

$$s_n = \frac{\exp(\langle \mathbf{a}_n^{(t)}, f_{\text{mid}}(\mathbf{w}_k^{(t,r)}, \mathbf{x})/\alpha\rangle)}{\sum_{i=1}^{C}\exp(\langle \mathbf{a}_i^{(t)}, f_{\text{mid}}(\mathbf{w}_k^{(t,r)}, \mathbf{x})/\alpha\rangle)},$$

with a temperature value α determining the level of concentration and $\langle\cdot,\cdot\rangle$ is inner product. By minimizing the cross entropy, we both maximize the similarity between the feature and its corresponding anchor $\mathbf{a}_c^{(t)}$ and minimize the similarity between this feature and each non-corresponding anchor in $\{\mathbf{a}_n^{(t)}\}_{n \neq c}$.

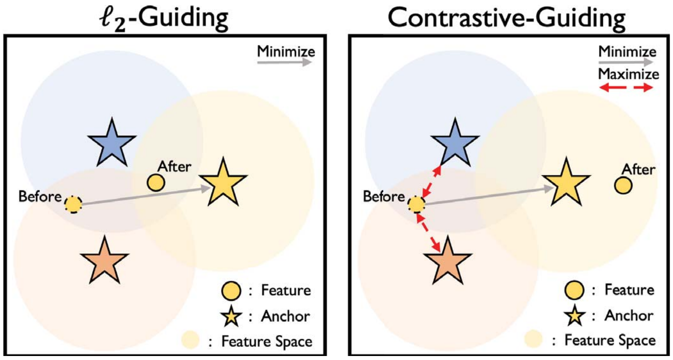


Fig. 4. Illustration of ℓ_2-Guiding and contrastive-guiding. Each color denotes one category. ℓ_2-Guiding only minimizes the distance between the feature and corresponding anchor, which ends up locating the feature at the feature space overlap of categories. However, contrastive-guiding also maximizes the distance between the feature and non-corresponding anchors, which feasibly locates the feature at space that merely belongs to the corresponding category.

2) Strengths of CG: Fig. 4 compares the aforementioned ℓ_2-Guiding and CG. We see that the CG loss achieves more effective feature matching and therefore better targets the overlapping phenomenon from the following two perspectives.

i) CG can provide a more precise target. As shown in the figure, there could be overlap among feature spaces of different categories. For ℓ_2 loss, only minimizing the distance between the feature and its corresponding anchor could end up locating the feature at that feature space overlap of several categories. However, CG provides a more precise target by simultaneously minimizing the distance between the feature and corresponding anchor and maximizing the distance between the feature and non-corresponding anchors, which feasibly locates the feature at space that merely belongs to the corresponding category.

ii) CG can enlarge the gap across categories. In each round, each feature is further pushed away from non-corresponding anchors, that is, more features are pushed to the non-overlap area as shown in Fig. 4. After this, each anchor (category-wise feature midpoint) is recalculated. Since more features are pushed to the non-overlap area, each recalculated anchor also moves towards the non-overlap area, which ends up enlarging the distance between anchors. This process repeats and eventually the feature space of different categories would be enlarged distinctly, which alleviates the overlap phenomenon.

D. Further Discussions

1) Communication Cost: The FedFM method involves two streams of communication, model parameters communication, which is required for most FL methods, and anchors communication, which is relatively negligible. Here, we take a normal setting as an example, where the model is ResNet18 [57], category number $C = 10$, feature dimension $d = 512$. In this case, the anchors cost of each client is $C \times d = 5.12 \times 10^3$ units while the model cost of each client is 1.17×10^7 units. As a result, communicating anchors only requires approximate 0.04% more bandwidth cost.

2) Privacy Analysis: While some feature inversion methods [58] attempt to reconstruct an image from a single feature, the

TABLE II
COMPLEXITY COMPARISON. N DENOTES THE NUMBER OF MODEL PARAMETERS AND D DENOTES THE FLOATING NUMBERS REQUIRED FOR ANCHORS, WHERE $N \gg D$

Method	Memory	Communication	Inference
FedAvg	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$
FedProx	$\mathcal{O}(2 \cdot N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$
SCAFFOLD	$\mathcal{O}(2 \cdot N)$	$\mathcal{O}(2 \cdot N)$	$\mathcal{O}(N)$
FedFM	$\mathcal{O}(N + D)$	$\mathcal{O}(N + D)$	$\mathcal{O}(N)$

communicated anchors are the average of a number of features, which makes the reconstruction difficult. This is also verified by [59], making FedFM a privacy-preserving method. Meanwhile, Secure Aggregation [60] is often used in practice to ensure the safety of model parameters, which can also be adopted to further secure the anchors communication. As CCVR [59] and FedFTG [33], the previously illustrated process requires uploading clients' category distributions for weighted aggregating local anchors. For cases when this is prohibited, we can directly compute the simple arithmetic mean, where clients are not asked to upload category distributions. We empirically verify that simple arithmetic mean of anchors achieves comparable performance in Section VII.D.1.

3) Complexity: The complexity of FedFM regarding to memory, communication and inference is $\mathcal{O}(N + D)$, $\mathcal{O}(N + D)$ and $\mathcal{O}(N)$, where N is the number of model parameters and D is the number of floating numbers required for anchors ($N \gg D$). Table II further compares it with some classical FL works, including FedAvg [1], FedProx [6] and SCAFFOLD [7]. From the table, we see that compared to other methods that focus on data heterogeneity, our method introduces acceptable memory and communication complexity, and achieves the same inference complexity as FedAvg. Specifically, compared to SCAFFOLD that performs decently, the memory and communication complexity of FedFM is significantly smaller since $N \gg D$. Similar to many classical FL works, such as FedProx [6] and SCAFFOLD [7], our method aims to mitigate the negative effects of data heterogeneity by inevitably introducing some additional memory or communication cost during federated training time, while preserving the complexity during inference time.

V. CONVERGENCE ANALYSIS

This section provides theoretical convergence analysis of FedFM, including the required assumptions, lemmas and the derived theorem and corollary.

We provide convergence analysis of the global objective function $\Phi(\mathbf{w}; \mathcal{A})$ in (2), which relies on the following 4 assumptions. Assumption 1 is based on smoothness of $F_k(\mathbf{w})$ as used in standard analysis of SGD and $f_{\text{mid}}(\mathbf{w}_k, \mathbf{x})$ as we need to additionally analyze $\Phi_k(\mathbf{w}; \mathcal{A})$. Assumptions 2, 3 and 4 are commonly used in the FL literature [6], [7], [14], [19], [30]. Here, ℓ_2 loss is applied for simplicity. A comprehensive proof can be found in Appendix, Section C (see supplementary material).

Assumption 1: (Smoothness). Each loss function $F_k(\mathbf{w})$ is Lipschitz-smooth. Feature function $f_{\text{mid}}(\mathbf{w}_k, \mathbf{x})$ is Lipschitz-continuous and Lipschitz-smooth.

Assumption 2: (Bounded Scalar). $\Phi_k(\mathbf{w}; \mathcal{A})$ is bounded below by Φ_{inf}.

Assumption 3: (Unbiased Gradient and Bounded Variance). For each client, the stochastic gradient is unbiased: $\mathbb{E}_{\xi}[g_k(\mathbf{w}|\xi)] = \nabla\Phi_k(\mathbf{w}; \mathcal{A})$, and has bounded variance: $\mathbb{E}_{\xi}[||g_k(\mathbf{w}|\xi) - \nabla\Phi_k(\mathbf{w}; \mathcal{A})||^2] \leq \sigma^2$.

Assumption 4: (Bounded Dissimilarity). For any set of weights $\{p_k \geq 0\}_{k=1}^K$ subject to $\sum_{k=1}^K p_k = 1$, there exists constants $\beta^2 \geq 1$ and $\kappa^2 \geq 0$ such that $\sum_{k=1}^K p_k ||\nabla\Phi_k(\mathbf{w}; \mathcal{A})||^2 \leq \beta^2 ||\nabla\Phi(\mathbf{w}; \mathcal{A})||^2 + \kappa^2$.

The smoothness property of $\Phi_k(\mathbf{w}; \mathcal{A})$ is necessary for convergence analysis. Since $\mathcal{A}$ changes over communication round t, assuming smoothness on $Q_k(\mathbf{w}; \mathcal{A})$ would be too strong as it requires T assumptions. Thus, we only make one minor assumption on the feature function $f_{\text{mid}}(\mathbf{w}_k, \mathbf{x})$ in Assumption 1 and prove the smoothness of $\Phi_k(\mathbf{w}; \mathcal{A})$ as stated in Lemma 1.

Lemma 1: The local objective function $\Phi_k(\mathbf{w}; \mathcal{A})$ is Lipschitz-smooth: $||\nabla\Phi_k(\mathbf{x}; \mathcal{A}) - \nabla\Phi_k(\mathbf{y}; \mathcal{A})|| \leq L||\mathbf{x} - \mathbf{y}||$ for some L.

The fact that $\mathcal{A}$ changes over round t makes it challenging to prove convergence as it changes the global loss function at each round t. In Lemma 2, we show that at each communication point, the aggregation and updating of anchors reduces (or keeps) the global loss value.

Lemma 2: The global loss function is non-increasing when updating global anchors. That is: $\Phi(\mathbf{w}^{(t+1)}; \mathcal{A}^{(t+1)}) \leq \Phi(\mathbf{w}^{(t+1)}; \mathcal{A}^{(t)})$.

Based on this key lemma, we derive our main Theorem, which is stated as follows:

Theorem 1: (Optimization bound of the global objective function). Under Assumptions 1 to 4, if we set $\eta L \leq min\{\frac{1}{2\tau}, \frac{1}{\sqrt{2\tau(\tau-1)(2\beta^2+1)}}\}$, the optimization error will be bounded as follows:

$$\begin{aligned}\min_t \mathbb{E}||\nabla\Phi(\mathbf{w}^{(t)}; \mathcal{A}^{(t)})||^2 &\leq \frac{4[\Phi(\mathbf{w}^{(0)}; \mathcal{A}^{(0)}) - \Phi_{inf}]}{\tau\eta T} + 4\eta L\sigma^2 \sum_{k=1}^K p_k^2 \\ &\quad + 3(\tau-1)\eta^2\sigma^2 L^2 + 6\tau(\tau-1)\eta^2 L^2\kappa^2,\end{aligned}$$

where η is the client learning rate and τ is the number of local iterations.

Theorem 1 indicates that as $T \to \infty$, the expectation of the optimization error will be bounded by a constant number for fixed η. Detailed proof is included in Appendix, Section C-3 (available online).

When setting a proper learning rate η, we have the following corollary:

Corollary 1: **(Convergence of the global objective function)** By setting $\eta = \frac{1}{\sqrt{\tau T}}$, FedFM can converge to a stationary point given an infinite number of communication rounds T. Specifically, the bound could be rewritten as:

$$\min_t \mathbb{E}||\nabla\Phi(\mathbf{w}^{(t)}; \mathcal{A}^{(t)})||^2 \leq \frac{4[\Phi(\mathbf{w}^{(0)}; \mathcal{A}^{(0)}) - \Phi_{inf}] + 4L\sigma^2 \sum_{k=1}^K p_k^2}{\sqrt{\tau T}}$$

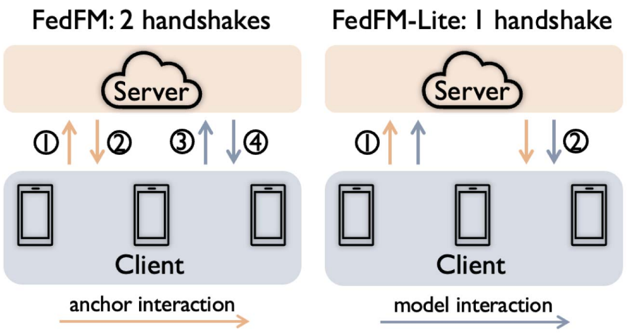


Fig. 5. Comparison between FedFM and FedFM-Lite. FedFM introduces minor bandwidth cost but requires 2 handshakes each round. FedFM-Lite further eliminates this issue, which introduces minor bandwidth cost and requires only 1 handshake.

$$+\frac{3(\tau-1)\sigma^2L^2}{\tau T}+\frac{6(\tau-1)L^2\kappa^2}{T}$$
$$=\mathcal{O}(\frac{1}{\sqrt{\tau T}})+\mathcal{O}(\frac{1}{T})+\mathcal{O}(\frac{\tau}{T}).$$

This corollary indicates that as $T\rightarrow\infty$, the error's upper bound approaches 0. Also, given a finite T, there exists a best τ that minimizes the error's upper bound. These analyses show that FedFM can achieve the same convergence rate as most methods, such as FedAvg [14], [19]. Therefore, FedFM achieves feature matching across clients without affecting its convergence. The detailed proof is included in the Appendix, Section C-4 (available online).

VI. An Efficient Variant: FedFM-Lite

We next propose an efficient and flexible variant of FedFM, called FedFM-Lite. The previously proposed FedFM involves two separate communication flows, anchors communication before local model training and models communication after local model training. Though we have discussed that the anchors communication introduces minor bandwidth cost, FedFM requires twice handshakes between client and server within a federated round, which could be a drawback in real-world application since that each handshake requires some synchronization time. Thus, to mitigate this issue, we propose a more efficient variant, FedFM-Lite, which requires one handshake between client and server. We compare FedFM and FedFM-Lite in Fig. 5.

In FedFM-Lite, each client computes local anchors after local model training and sends the local anchors together with model parameters. In this way, models and anchors are communicated within one handshake, which saves synchronization time and makes it more efficient. Beside higher efficiency, FedFM-Lite is also more flexible for real-world implementation since that models and anchors can be communicated at different frequencies. As discussed before, communicating anchors requires much less costs compared with model parameters. This motivates us to consider reducing the frequency of models communication and utilize anchors communication as compensation. Specifically, in a T rounds FL process, we can communicate anchors for each round while communicate models for every a rounds. This results in roughly a times less communication costs compared with most existing methods, such as FedAvg [1], FedProx [6], FedDyn [16] and $2\times a$ less than SCAFFOLD [7].

VII. Experiments

This section presents experimental details and results. We compare our proposed FedFM with state-of-the-art methods on various heterogeneous settings and datasets, which are evaluated by accuracy, feature space quality, memory and communication bandwidth cost.

A. Experimental Setup

a) Federated Setting: We set the number of clients $K=10$ and conduct experiments on datasets including CIFAR-10 [21], CINIC-10 [22] and CIFAR-100. Here we consider three data heterogeneity (non-IID) settings. 1) NIID-1: the category distributions of clients follow a Dirichlet distribution $Dir_{10}(\beta)$, where β (default 0.5) correlates to the heterogeneity level, which is a widely considered setting [61], [62]; 2) NIID-2: each client has several dominant categories (with much more data samples) while we keep the dataset size of each client the same. We consider this setting to focus on the distribution heterogeneity but not quantity imbalance; 3) NIID-3: each client has no data sample from several categories, which is also considered in [1], [6]. Fig. 9 in Appendix (available online) shows the data distribution of these three Non-IID settings on CIFAR-10.

b) Implementation: All the experiments are conducted using PyTorch API [63] on one Nvidia GeForce RTX 3090, and the required memory is around $2000-6000$ MB. We run $T=100$ communication rounds for all experiments. In each round, every client runs for 10 local epochs with a batch size of 64. We apply ResNet18 [57] for CIFAR-10 [21] and CINIC-10 [22], ResNet50 for CIFAR-100 [21]. We use SGD optimizer with learning rate 0.01, weight decay rate $1e^{-5}$ and SGD momentum 0.9. These are commonly used experimental settings [20], [59]. For evaluation, we hold out a testing dataset at the server side and conduct the above non-IID partitions on the training set. For each client, 20% of the training set is held out for validation. We average the results on each local validation set and save the best model. Finally, we report the testing accuracy of the best model on the testing dataset.

We consider $f_{\text{full}}(\cdot,\cdot)$ as a standard ResNet and the feature extractor $f_{\text{mid}}(\cdot,\cdot)$ as $f_{\text{full}}(\cdot,\cdot)$ without the last fully-connected layer. For feature matching, the feature is normalized before applying the feature matching loss term. FedFM denotes FedFM with CG unless explicitly specified. We run FedAvg for the first T_s rounds and then launch our proposed FedFM. For all methods, we tune the hyper-parameters in a reasonable range and report the best results. Generally, for FedFM, $\lambda=50.0$ and $T_s=20$ is a relatively better choice.

B. Main Results

We compare FedFM with seven existing classical methods, including FedAvg [1], FedAvgM [9], FedProx [6], SCAFFOLD [7], FedDyn [16], FedNova [19] and MOON [20] on various

TABLE III

Classification Accuracy (%) Under NIID-1 and NIID-2 Settings on CIFAR-10 [21], CINIC-10 [22] and CIFAR-100. NIID-1 is Under Dirichlet Distribution $Dir_{10}(0.5)$ and NIID-2 is the Distribution Where Each Client Has One Dominant Category. Memory Shows the Required Number of Floating Numbers of Each Client in Each Round ($\times 10^3$). FedFM Consistently Outperforms Other State-of-the-Art Methods With Relatively Less Memory Cost Across Various Settings

Method	CIFAR-10			CINIC-10			CIFAR-100		
	NIID-1	NIID-2	Memory	NIID-1	NIID-2	Memory	NIID-1	NIID-2	Memory
FedAvg [1]	66.69 ±0.69	69.47 ±0.48	11,182	55.96 ±0.16	58.56 ±0.22	11,182	62.16 ±0.04	62.33 ±0.27	23,705
FedAvgM [9]	66.85 ±0.42	67.87 ±0.17	11,182	56.15 ±0.45	58.79 ±0.30	11,182	61.23 ±0.12	61.30 ±0.27	23,705
FedProx [6]	66.99 ±0.26	69.42 ±0.38	22,364	55.58 ±0.13	58.32 ±0.11	22,364	61.96 ±0.05	62.20 ±0.28	47,410
SCAFFOLD [7]	69.91 ±0.54	71.48 ±0.23	22,364	58.60 ±0.27	60.78 ±0.32	22,364	67.32 ±0.29	67.24 ±0.03	47,410
FedDyn [16]	68.32 ±0.34	67.63 ±0.16	22,364	56.71 ±0.50	59.92 ±0.15	22,364	43.41 ±0.54	46.44 ±0.87	47,410
FedNova [19]	66.80 ±0.81	69.45 ±0.49	11,182	55.67 ±0.24	58.63 ±0.22	11,182	62.35 ±0.20	62.31 ±0.26	23,705
MOON [20]	67.74 ±0.30	71.09 ±0.22	33,546	57.25 ±0.07	59.28 ±0.03	33,546	62.56 ±0.22	62.99 ±0.13	71,115
FedFM (Ours)	**72.89**±0.22	**74.52**±0.21	11,187	**62.56**±0.40	**65.75**±0.46	11,187	**71.48**±0.25	**72.13**±0.45	23,909

TABLE IV

Classification Accuracy (%) Under NIID-3 Setting on CIFAR-10. Missing x Setting Represents That Each Client Has No Data Sample of x Categories. Memory and Bandwidth Show the Required Number of Floating Numbers of Each Client in Each Round ($\times 10^3$). Our Proposed FedFM Consistently Outperforms Other State-of-the-Art Methods With Minor Additional Resource Cost

Method	Missing 1	Missing 2	Missing 3	Missing 5	Missing 7	Memory	Bandwidth
FedAvg [1]	70.54 ±0.22	70.50 ±0.24	69.87 ±0.30	67.25 ±0.54	59.52 ±0.59	11,182	11,182
FedAvgM [9]	70.02 ±0.40	69.93 ±0.57	69.34 ±0.37	67.04 ±0.47	57.08 ±0.66	11,182	11,182
FedProx [6]	71.16 ±0.42	70.72 ±0.35	69.82 ±0.23	67.25 ±0.54	58.58 ±0.23	22,364	11,182
SCAFFOLD [7]	72.67 ±0.39	72.94 ±0.30	72.60 ±0.22	71.43 ±0.05	64.28 ±0.60	22,364	22,364
FedDyn [16]	67.43 ±0.51	67.76 ±0.64	67.78 ±0.28	69.53 ±0.59	64.75 ±0.30	22,364	11,182
FedNova [19]	70.56 ±0.25	70.48 ±0.23	70.05 ±0.15	67.56 ±0.52	59.66 ±0.42	11,182	11,182
MOON [20]	72.64 ±0.25	72.21 ±0.22	71.57 ±0.23	68.86 ±0.27	57.80 ±1.02	33,546	11,182
FedFM (Ours)	**75.97**±0.44	**75.84**±0.23	**75.04**±0.29	**73.23**±0.35	**65.24**±0.52	11,187	11,187

non-IID settings and datasets. We first show accuracy comparisons quantitatively and then demonstrate qualitative comparisons in feature space by T-SNE [17] visualization.

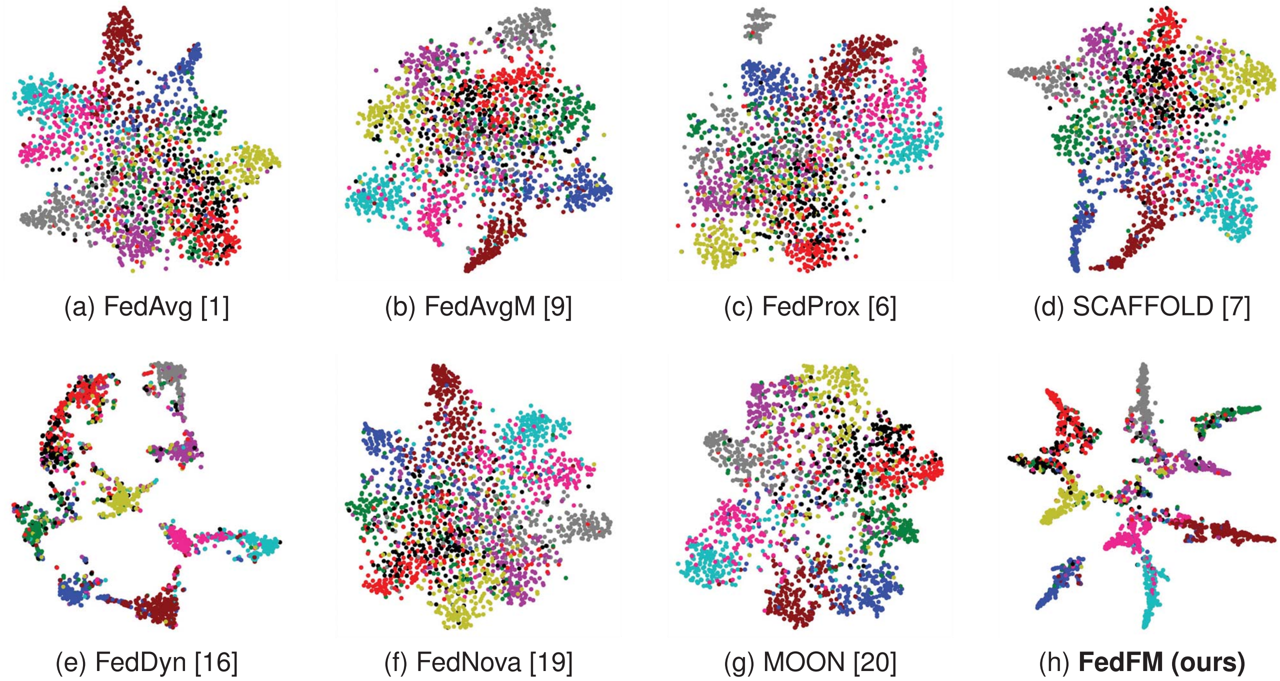
1) Quantitative Analysis: Table III presents accuracy comparisons on three datasets under both NIID-1 and NIID-2 settings. For each entry in the table, we run three independent trials and report the mean and standard deviation results. We see that i) FedFM consistently outperforms other state-of-the-art methods on all tasks. ii) On the relatively more complicated task, NIID-1 setting on CIFAR100, the proposed FedFM significantly outperforms other methods. Specifically, compared with standard FL, FedAvg [1], FedFM achieves 9.40% higher accuracy. iii) On the six different tasks, FedFM outperforms the second-best method (SCAFFOLD [7]) $2.98\%, 3.04\%, 3.96\%, 4.97\%, 4.16\%, 4.89\%$, respectively. It is also worth mentioning that SCAFFOLD [7] requires roughly twice the memory and communication bandwidth costs.

Table IV shows the accuracy, memory, bandwidth comparisons under NIID-3 setting on CIFAR-10. In NIID-3, each client has no data sample from several (x) categories, which is denoted as Missing x setting. We conduct experiments on different $x \in \{1, 2, 3, 5, 7\}$. We see that i) the performances of all methods degrade as x increases since larger x corresponds to a more heterogeneous setting. This verifies that data heterogeneity significantly affects the performance of FL. ii) FedFM consistently outperforms other baselines. Specifically, it outperforms FedAvg [1] by 5.53% and SCAFFOLD [7] by 2.28% on average. iii) Compared with FedAvg [1], FedFM achieves significantly better performance while introducing minor memory and bandwidth costs. Compared with SCAFFOLD [7], FedFM achieves better performance with nearly half of the memory and bandwidth costs.

Table XII in Appendix (available online) shows our proposed FedFM maintains pleasant performance under homogeneous data, indicating its broad applicability.

2) Qualitative Analysis: Fig. 6 presents T-SNE [17] visualization results in feature space of each method, where different color denote different category. All the sampled data is fed into the final global model of each method to obtain the corresponding features, which are then plot using T-SNE [17]. We see that i) most previous methods suffer from slack category-wise feature space while our FedFM establishes significantly more compact category-wise feature space, which reflects the effectiveness of utilizing anchors to conduct feature matching. ii) Most previous methods suffer from ambiguous boundaries. However, our FedFM establishes clusters with clear boundaries and large gap which are contributed by using anchors to attract features and the contrastive-guiding loss. These two phenomena indicate that our proposed FedFM indeed benefits the establishment of feature space and gives the evidence of significantly improved performance. Note that though FedDyn [16] seems to establish a good feature space, it only achieves a 68.32% accuracy, which is significantly lower than that of our proposed FedFM (72.89%).

For more comprehensive comparisons, we also evaluate the quality of feature space in Fig. 6 using normalized mutual information (NMI) and silhouette score (SS) [18]. NMI is capable of measuring the quality of clustering. SS is a measure of how similar an object is to its own cluster compared to other clusters. Note that for NMI, we first apply the K-Means [64]



(a) FedAvg [1]

(b) FedAvgM [9]

(c) FedProx [6]

(d) SCAFFOLD [7]

(e) FedDyn [16]

(f) FedNova [19]

(g) MOON [20]

(h) **FedFM (ours)**

Fig. 6. Qualitative comparisons among methods through T-SNE [17] visualization. Each dot represents the feature of one data sample, whose color denotes its category. FedFM establishes the most compact and distinct clusters in feature space. FedDyn [16] has moderately good visualization results but only achieves 68.32% accuracy while FedFM achieves 72.89%.

TABLE V

NUMERICAL QUALITY EVALUATION OF FEATURE SPACES. HIGHER NMI AND SS CORRESPOND TO HIGHER QUALITY OF FEATURE SPACES. OUR PROPOSED FEDFM ACHIEVES SIGNIFICANTLY HIGHEST NMI AND SS

Metric	FedAvg [1]	FedAvgM [9]	FedProx [6]	SCAFFOLD [7]	FedDyn [16]	FedNova [19]	MOON [20]	**FedFM (ours)**
NMI	0.413	0.411	0.397	0.432	0.485	0.416	0.481	**0.557**
SS	0.036	0.038	0.006	0.056	0.136	0.049	0.068	**0.173**

to perform clustering on sampled features of all methods and then use NMI to measure the clustering quality. Both NMI and SS are measured using Scikit-learn [65]. Higher NMI and SS correspond to higher quality of feature spaces. We present the evaluation results in Table V. The table shows that our proposed FedFM achieves significantly higher NMI and SS. Specifically, compared with FedDyn [16], FedFM achieves 14.8% higher NMI and 27.2% higher SS. This gives evidence for the seemly great feature space but ordinary accuracy performance of FedDyn [16] in a way.

C. Further Comparisons

1) Comparisons With FedProto: Targeting a different task, personalization in FL, FedProto [38] uses prototype to provide feature information from others to enhance personalization while FedFM focuses on generalization in FL. To further verify their difference, we implement a generalized version of FedProto [38] and compare it with FedFM under NIID-1 setting on CIFAR-10 [21]. Experiments show that generalized FedProto achieves $67.33 \pm 0.49\%$ accuracy, which is outperformed by SCAFFOLD [7], FedDyn [16] and MOON [20]. Our proposed FedFM achieves $72.89 \pm 0.22\%$ accuracy, which outperforms generalized FedProto [38] by 5.56%.

2) Performance and Resource Costs: FedFM introduces minor resource costs while bringing significantly better performance. To verify this point, we conduct experiments on CIFAR-100 [21] under various client numbers ($K \in \{20, 30, 50, 100\}$). Beside accuracy comparisons, we also compare the memory and bandwidth cost of these methods, which are evaluated by the number of required floating numbers ($\times 10^8$) for the overall FL process. We present the results in Table VI. Experiments show that i) our proposed FedFM consistently outperforms state-of-the-art methods for various client numbers, indicating its applicability to scenario with large client number. ii) FedFM achieves significantly better performance with minor additional memory and bandwidth overhead. Specifically, FedFM takes only 0.86% more communication overhead to achieve 13.97% better classification performance than FedAvg [1] when $K = 100$. Compared with the second-best method (SCAFFOLD [7]), FedFM achieves 7.68% higher accuracy when $K = 100$ with only half the memory and bandwidth costs.

3) FL on Large-Scale FL Dataset: Here we conduct experiments on the frequently used [6] dataset FEMNIST in Leaf [66]. FEMNIST is an FL dataset, which has 62 different classes (10 digits, 26 lowercase, 26 uppercase) and consists of 3500 clients. We hold out 10% of the clients for testing. For each round of FL, we sample 0.2% clients. We compare our proposed FedFM with FedAvg and SCAFFOLD, where FedAvg is the standard FL method and SCAFFOLD is shown to perform the

TABLE VI

COMPARISONS OF ACCURACY, MEMORY AND BANDWIDTH COSTS ON CIFAR-100. EACH ENTRY SHOWS CLASSIFICATION ACCURACY (%). WITHIN PARENTHESES, IT SHOWS THE REQUIRED MEMORY / BANDWIDTH COST, EVALUATED BY FLOATING NUMBERS ($\times 10^8$). WHEN $K = 100$, I) FEDFM TAKES ONLY 0.86% MORE RESOURCE OVERHEAD TO ACHIEVE 13.97% HIGHER ACCURACY THAN FEDAVG [1], II) FEDFM ACHIEVES 7.68% HIGHER ACCURACY WITH ONLY HALF THE MEMORY AND BANDWIDTH COSTS COMPARED WITH SCAFFOLD [7]

K	20	30	50	100
FedAvg [1]	58.48 (474 / 474)	54.46 (711 / 711)	50.20 (1,185 / 1,185)	41.41 (2,370 / 2,370)
FedAvgM [9]	58.36 (474 / 474)	54.48 (711 / 711)	52.86 (1,185 / 1,185)	46.72 (2,370 / 2,370)
FedProx [6]	58.27 (948 / 474)	54.50 (1,422 / 711)	50.55 (2,370 / 1,185)	40.62 (4,740 / 2,370)
SCAFFOLD [7]	64.65 (948 / 948)	61.82 (1,422 / 1,422)	56.71 (2,370 / 2,370)	47.70 (4,740 / 4,740)
FedDyn [16]	41.90 (948 / 474)	41.13 (1,422 / 711)	39.30 (2,370 / 1,185)	31.21 (4,740 / 2,370)
FedNova [19]	58.01 (474 / 474)	53.83 (711 / 711)	50.34 (1,185 / 1,185)	42.61 (2,370 / 2,370)
MOON [20]	57.63 (1,422 / 474)	52.71 (2,133 / 711)	47.84 (3,555 / 1,185)	38.45 (7,110 / 2,370)
FedFM (ours)	**69.49** (478 / 478)	**67.70** (717 / 717)	**64.22** (1,195 / 1,195)	**55.38** (2,390 / 2,390)

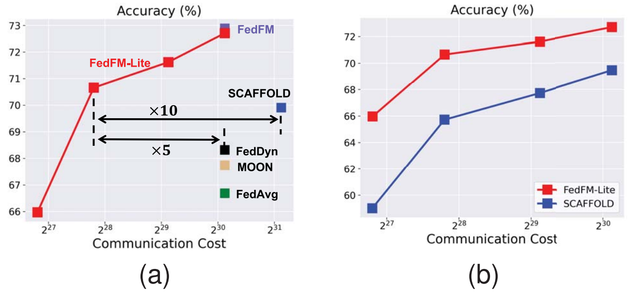


Fig. 7. Comparison of performances and communication costs among FedFM-Lite and several classical methods. (a) Our proposed FedFM-Lite can achieve significantly better performance while saving 5 or even 10 times communication costs compared with existing techniques. (b) FedFM-Lite consistently outperforms SCAFFOLD at the same communication cost.

TABLE VII

ACCURACY COMPARISON ON LARGE-SCALE FEMNIST DATASET IN LEAF. FEDFM CONSISTENTLY PERFORMS THE BEST

Model	FedAvg	SCAFFOLD	**FedFM (ours)**
ResNet18-v1	79.93	67.61	**81.62**
ResNet18-v2	81.85	74.63	**83.05**

second-best previously. We consider two versions of ResNet18, where ResNet18-v1 is the vanilla backbone in PyTorch [63] and ResNet18-v2 replaces the first convolution layer with smaller kernel size (from 7 to 3).

Table VII shows the accuracy comparison, we see that 1) FedFM consistently performs the best across different models, while SCAFFOLD even performs worse than FedAvg in these cases, indicating the effectiveness and stability of FedFM. 2) We notice that the model parameters of SCAFFOLD become ’NaN’ at the later period of training, which contributes to the low accuracy. In contrast, FedFM is much more stable throughout the entire training process.

4) Performance of an Efficient Variant: FedFM-Lite: In Section VI, we propose an efficient variant, called FedFM-Lite, which is more efficient and flexible in practice. Since anchors require less communication bandwidth costs than model communication, we propose to reduce the communication frequency of model communication while keeping the communication frequency of anchor communication, which can be easily achieved in FedFM-Lite. This modification saves communication bandwidth costs to a large extent.

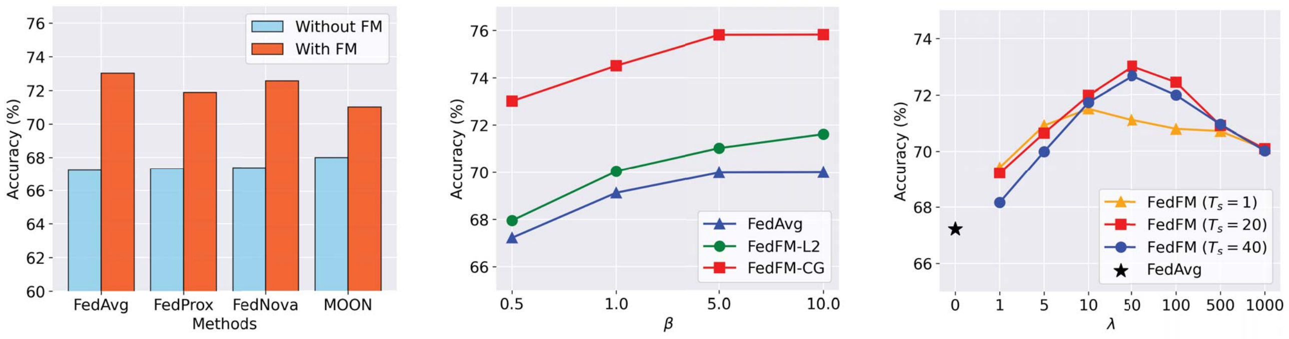
To empirically verify this efficiency and flexibility, we conduct the following experiments. We communicate anchors for each round while we communicate models every $a \in \{1, 2, 5, 10\}$ round(s), that is, larger a corresponds to less communication cost. We show the communication cost and final performance of each trial in Fig. 7. We also present several representative methods for comparison.

Fig. 7(a) shows that when we communicate models every $a \in \{1, 2, 5\}$ communication round(s), FedFM-Lite can significantly outperform compared methods. Specifically, when $a = 5$, FedFM-Lite outperforms FedDyn [16] by 2.34% with 5 times less communication costs and SCAFFOLD [7] by 0.75% with 10 times less communication costs. These experiments show that for bandwidth limited scenarios, FedFM-Lite can be an efficient candidate algorithm. From Fig. 7(b), we see that FedFM-Lite consistently outperforms SCAFFOLD at the same communication cost.

D. Ablation Study

1) Effects of Global Anchors Aggregation Manner: Here, we show that FedFM can still achieve great performance without uploading clients’ category distributions as discussed in Section IV.D.2. We compare two manners of global anchors aggregation, sample-number-based weighted aggregation and uniform aggregation. For the weighted aggregation, each category-wise global anchor is updated by weighted aggregating local anchors according to each client’s number of data samples of the corresponding category. This aggregation manner might not be allowed for its requirement for uploading clients’ category distributions. For the uniform aggregation, each category-wise global anchor is updated by uniform aggregating local anchors of the corresponding category, which relieves the above issue. Here, for those categories where a client has no data sample, we adopt the corresponding global anchors as the local anchors for aggregation.

We conduct experiments under NIID-1 on CIFAR-10 [21] and present the results in Table VIII. Experiments show that FedFM with uniform aggregation performs comparably to FedFM with weighted aggregation (only 0.02% performance drop). The reason behind this could be that for each category, all clients’ features are pushed to the same shared global anchor. As a result, all clients’ local anchors of the same category are close to each other, making it similar between applying weighted aggregation and uniform aggregation.



(a) Modularity of FM (b) ℓ_2-Guiding and CG (c) T_s and λ

Fig. 8. Ablation study. (a) Shows that FM consistently brings performance gain to four methods. (b) Shows that CG significantly improve the effectiveness of FM. (c) Shows that $T_s = 20$ and $\lambda = 10 \sim 100$ is roughly an optimal solution.

TABLE VIII
EFFECTS OF GLOBAL ANCHORS AGGREGATION MANNER. (WEIGHTED) DENOTES AGGREGATING CATEGORY-WISE ANCHORS ACCORDING TO EACH CLIENT'S NUMBER OF CORRESPONDING SAMPLES. (UNIFORM) DENOTES COMPUTING SIMPLE ARITHMETIC MEAN OF ANCHORS. EXPERIMENTS SHOW THAT FEDFM (UNIFORM) PERFORMS COMPARABLY WITH FEDFM (WEIGHTED) WHILE FEDFM (UNIFORM) DOES NOT NEED UPLOADING CLIENT'S CATEGORY DISTRIBUTION

Method	FedAvg [1]	FedFM (Weighted)	FedFM (Uniform)
Accuracy	66.69 ±0.69	72.89 ±0.22	72.87 ±0.26

2) Modularity of Feature Matching: One advantage of our anchor-based feature matching (FM) method is its modularity, that is, it can be combined with most existing methods. Fig. 8(a) shows the performances before and after feature matching (FM) with contrastive-guiding combined with several existing methods. Here, we take FedAvg [1], FedProx [6], FedNova [19] and MOON [20] as example and conduct experiments under NIID-1 setting on CIFAR10 [21]. Note that the previous explored FedFM corresponds to FedAvg [1] incorporated with FM.

From the figure, we see that applying our FM consistently brings performance gain to these four methods, achieving 4.63% higher accuracy than corresponding baselines on average. Note that all these methods with FM outperforms the state-of-the-art performance 69.91% (SCAFFOLD [7]).

3) Effects of Contrastive-Guiding Loss: Fig. 8(b) presents the performance of FedAvg [1], FedFM with ℓ_2 loss and FedFM with contrastive-guiding (CG) loss under different heterogeneous levels. Note that smaller β corresponds to more severe heterogeneous level. We see that i) the performance of all methods degrades as the heterogeneous level increases (β decreases), which verifies that data heterogeneity affects the performance of FL. ii) Both FedFM ℓ_2 and CG loss outperform baseline FedAvg [1], indicating that anchor-based feature matching brings performance improvement to standard FL. iii) FedFM with CG significantly enhances the performance compared with FedFM with ℓ_2, which indicates the effectiveness of our proposed CG loss.

4) Effects of the Weight of Feature Matching Loss λ: Fig. 8(c) presents the relationship between the final results and weight of feature matching loss λ. Specifically, for each curve in the figure, the launching round T_s of FedFM is fixed while the λ is tuned in $\{1, 5, 10, 50, 100, 500, 1000\}$. Note that when $\lambda = 0$, FedFM reduces to FedAvg [1], which is denoted by a star. We see that i) applying feature matching brings performance gain over FedAvg [1] for a wide range of λ, indicating the effectiveness of feature matching; ii) a moderate λ ranging from $10 \sim 100$ tends to perform better.

TABLE IX
ABLATION STUDY ON PARAMETER α IN PRACTICE. AS A REFERENCE, ON CIFAR-10, FEDAVG: 66.69, SCAFFOLD 69.91; ON CIFAR-100, FEDAVG: 62.16, SCAFFOLD: 67.32

Dataset	CIFAR-10				CIFAR-100			
α	0.01	0.1	1.0	10.0	0.01	0.1	1.0	10.0
Acc	70.44	72.41	71.46	69.17	68.10	71.41	68.24	65.05

5) Effects of the Round T_s to Launch FedFM: Fig. 8(c) presents the relationship between the final results and launching round T_s of FedFM. Specifically, for each fixed λ, we compare the performance of three different $T_s \in \{1, 20, 40\}$. We see that a moderate $T_s = 20$ tends to perform better. This is reasonable since at the initial rounds of FL, the established anchors are less representative and still in drastic change, which makes such feature matching less effective.

6) Effects of Temperature α in CG-Loss: During the experiments, we do not carefully tune this parameter and instead use a default setting $\alpha = 0.1$, which is commonly used in literature [49]. Here, we conduct experiments under NIID-1 setting of CIFAR-10 and CIFAR-100. As a reference, on CIFAR-10, the performances of FedAvg and SCAFFOLD are 66.69 and 69.91. On CIFAR-100, the performances of FedAvg and SCAFFOLD are 62.16 and 67.32. From Table IX, we see that 1) $\alpha = 0.1 \sim 1.0$ tend to perform better generally. 2) For a wide range, $\alpha = 0.01 \sim 10.0$, FedFM tend to consistently outperforms FedAvg, indicating the effectiveness of FedFM. On the other hand, FedFM outperforms SCAFFOLD for a wide range $\alpha = 0.01 \sim 1.0$.

We introduce this parameter to increase the flexibility of our method such that the performance is potential to be further improved by tuning. However, for anyone who is concerned about this hyper-parameter, we can just eliminate this α, which is equivalent to $\alpha = 1.0$ and preserves the high performance of FedFM. We can see from the table that $\alpha = 1.0$ is already capable of outperforming the second-best baseline, SCAFFOLD. Besides, it also outperforms FedFM with squared penalty (67.98% on CIFAR-10).

We also explore the effects of the number of epochs of local model training and the performance under partial client participation scenario in Tables X and XI in Appendix (available online).

VIII. Conclusion

Facing statistical data heterogeneity, there are two unsatisfying phenomena in feature space (inconsistent features across clients and overlapping features across categories) for existing federated learning methods. Motivated by this, we propose an anchor-based federated feature matching (FedFM) method, which utilizes shared anchors to guide feature learning at multiple local models, promoting a consistent feature space. Tackling the theoretical challenge of varying objective function, we prove the convergence of FedFM. For more precise guiding, we further propose a contrastive-guiding (CG) loss, which guides the feature of each sample to match with the corresponding anchor while keeping far away from non-corresponding anchors. We propose a more efficient and flexible variant of FedFM, FedFM-Lite, which is capable of communicating anchors and models at different frequency. Experiments show that FedFM with CG (and FedFM-Lite) consistently outperform state-of-the-art methods.

The current work focuses on supervised uni-modality FL. These ideas might be extended to self-supervised or semi-supervised learning, where the feature inconsistency could also exist in FL and feature matching should benefit representation learning. Feature matching can also be applied to multi-modality (e.g., image and text) tasks to align both features of image and text.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist*. PMLR, 2017, pp. 1273–1282.

[2] T. Gafni, N. Shlezinger, K. Cohen, Y. C. Eldar, and H. V. Poor, "Federated learning: A signal processing perspective," *IEEE Signal Process. Mag.*, vol. 39, no. 3, pp. 14–41, May 2022.

[3] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends® Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, 2021.

[4] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[5] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30.

[6] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proc. Mach. Learn. Syst.*, vol. 2, pp. 429–450, 2020.

[7] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn*. PMLR, 2020, pp. 5132–5143.

[8] T.-M. H. Hsu, H. Qi, and M. Brown, "Federated visual classification with real-world data distribution," in *Proc. Eur. Conf. Comput. Vision*. Springer, 2020, pp. 76–92.

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

[10] A. Hard et al., "Federated learning for mobile keyboard prediction," 2018, *arXiv:1811.03604*.

[11] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2019, pp. 6341–6345.

[12] S. AbdulRahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5476–5497, Apr. 2021.

[13] J. Wang et al., "A field guide to federated optimization," 2021, *arXiv:2107.06917*.

[14] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *Proc. Int. Conf. Learn. Representations*, 2019.

[15] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, *arXiv:1806.00582*.

[16] D. A. E. Acar, Y. Zhao, R. Matas, M. Mattina, P. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," in *Proc. Int. Conf. Learn. Representations*, 2020.

[17] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[18] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987.

[19] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "A novel framework for the analysis and design of heterogeneous federated learning," *IEEE Trans. Signal Process.*, vol. 69, pp. 5234–5249, 2021.

[20] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2021, pp. 10713–10722.

[21] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," 2009. [Online]. Available: http://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf

[22] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey, "CINIC-10 is not imageNet or CIFAR-10," 2018, *arXiv:1810.03505*.

[23] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Mach. Intell.*, vol. 2, no. 6, pp. 305–311, 2020.

[24] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "FedHealth: A federated transfer learning framework for wearable healthcare," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 83–93, Jul./Aug. 2020.

[25] Z. Chen, C. Yang, M. Zhu, Z. Peng, and Y. Yuan, "Personalized retrogress-resilient federated learning toward imbalanced medical data," *IEEE Trans. Med. Imag.*, vol. 41, no. 12, pp. 3663–3674, Dec. 2022.

[26] X. Liang, S. Shen, J. Liu, Z. Pan, E. Chen, and Y. Cheng, "Variance reduced local SGD with lower communication complexity," 2019, *arXiv:1912.12844*.

[27] T. Zhou, J. Zhang, and D. Tsang, "FedFA: Federated learning with feature anchors to align feature and classifier for heterogeneous data," 2022, *arXiv:2211.09299*.

[28] Z. Li, T. Lin, X. Shang, and C. Wu, "Revisiting weighted aggregation in federated learning with neural networks," in *Proc. 40th Int. Conf. Mach. Learn*. PMLR, 2023, pp. 19767–19788.

[29] R. Ye, M. Xu, J. Wang, C. Xu, S. Chen, and Y. Wang, "FedDisco: Federated learning with discrepancy-aware collaboration," in *Proc. 40th Int. Conf. Mach. Learn*. PMLR, 2023, pp. 39879–39902.

[30] S. J. Reddi et al., "Adaptive federated optimization," in *Proc. Int. Conf. Learn. Representations*, 2020.

[31] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proc. Int. Conf. Mach. Learn*. PMLR, 2021, pp. 12878–12889.

[32] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 2351–2363.

[33] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan, "Fine-tuning global model via data-free knowledge distillation for non-IID federated learning," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2022, pp. 10174–10183.

[34] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *Proc. Int. Conf. Mach. Learn*. PMLR, 2021, pp. 2089–2099.

[35] Y. Huang et al., "Personalized cross-silo federated learning on non-IID data," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 9, pp. 7865–7873.

[36] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with Moreau envelopes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 21394–21405.

[37] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 3557–3568.

[38] Y. Tan et al., "FedProto: Federated prototype learning across heterogeneous clients," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 8, pp. 8432–8440.

[39] J. Xu, X. Tong, and S.-L. Huang, "Personalized federated learning with feature alignment and classifier collaboration," in *Proc. 11th Int. Conf. Learn. Representations*, 2022.

[40] H. Jamali-Rad, M. Abdizadeh, and A. Singh, "Federated learning with taskonomy for nonIID data," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, 2022.

[41] D. K. Dennis, T. Li, and V. Smith, "Heterogeneity for the win: One-shot federated clustering," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2021, pp. 2611–2620.

[42] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.* (CVPR), vol. 1. Piscataway, NJ, USA: IEEE, 2005, pp. 539–546.

[43] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 815–823.

[44] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 4004–4012.

[45] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, vol. 29.

[46] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. 13th Int. Conf. Artif. Intell. Statist*. JMLR Workshop Conf. Proc., 2010, pp. 297–304.

[47] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[48] R. Salakhutdinov and G. Hinton, "Learning a nonlinear embedding by preserving class neighbourhood structure," in *Proc. Artif. Intell. Statist.* PMLR, 2007, pp. 412–419.

[49] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn*. PMLR, 2020, pp. 1597–1607.

[50] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *Proc. Int. Conf. Mach. Learn*. PMLR, 2021, pp. 12310–12320.

[51] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2020, pp. 9729–9738.

[52] P. Khosla et al., "Supervised contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 18661–18673.

[53] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn*. PMLR, 2021, pp. 8748–8763.

[54] J. Li, R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, and S. C. H. Hoi, "Align before fuse: Vision and language representation learning with momentum distillation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 9694–9705.

[55] X. Mu et al., "FedProc: Prototypical contrastive federated learning on non-IID data," *Future Gener. Comput. Syst.*, vol. 143, pp. 93–104, 2023.

[56] Y. Tan, G. Long, J. Ma, L. Liu, T. Zhou, and J. Jiang, "Federated learning from pre-trained models: A contrastive learning approach," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 19332–19344.

[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770–778.

[58] N. Zhao, Z. Wu, R. W. Lau, and S. Lin, "What makes instance discrimination good for transfer learning?" in *Proc. Int. Conf. Learn. Representations*, 2020.

[59] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng, "No fear of heterogeneity: Classifier calibration for federated learning with non-IID data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 5972–5984.

[60] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1175–1191.

[61] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *Proc. Int. Conf. Learn. Representations*, 2019.

[62] J. Zhang et al., "FedCP: Separating feature information for personalized federated learning via conditional policy," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2023, pp. 3249–3261.

[63] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32.

[64] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[65] F. Pedregosa et al., "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[66] S. Caldas et al., "Leaf: A benchmark for federated settings," 2018, *arXiv:1812.01097*.

Rui Ye received the B.E. degree in information engineering from Shanghai Jiao Tong University, Shanghai, China, in 2022. Since 2022, he has been working toward the Ph.D. degree with the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University. He was a Research Intern with Microsoft Research Asia, in 2022. His research interests include federated learning, multi-agent collaboration, and responsible AI.

Zhenyang Ni received the B.E. degree in information engineering from Shanghai Jiao Tong University, Shanghai, China, in 2022. Since 2022, he has been working toward the master's degree with the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University. His research interests include federated learning, multi-agent prediction, and collaborative perception.

Chenxin Xu received the B.E. degree in information engineering from Shanghai Jiao Tong University, Shanghai, China, in 2019. Since 2019, he has been working toward the joint Ph.D. degree with the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University and in electrical and computer engineering with the National University of Singapore. His research interests include computer vision, machine learning, graph neural network, and multi-agent prediction. He was the Reviewer of some prestigious international journals and conferences, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, Conference on Computer Vision and Pattern Recognition (CVPR), Conference on Neural Information Processing Systems (NeurIPS), International Conference on Computer Vision (ICCV), and Association for the Advancement of Artificial Intelligence (AAAI).

Jianyu Wang received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, in 2017, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2022. He is currently a Research Scientist with Apple. He was a Research Interns with Google Research, in 2020 and 2021, and Facebook AI Research, in 2019. His research interests include federated learning, distributed optimization, and systems for large-scale machine learning. His research has been supported by Qualcomm Ph.D. Fellowship in 2019.

Siheng Chen (Senior Member, IEEE) received the bachelor's degree in electronics engineering from Beijing Institute of Technology, China, in 2011, two master's degrees in electrical and computer engineering from the College of Engineering and machine learning from the School of Computer Science, and the doctorate degree in electrical and computer engineering from Carnegie Mellon University, in 2016. He is a Tenure-Track Associate Professor with Shanghai Jiao Tong University. Before joining Shanghai Jiao Tong University, he was a Research Scientist with Mitsubishi Electric Research Laboratories (MERL), and an autonomy engineer at Uber Advanced Technologies Group (ATG), working on the perception and prediction systems of self-driving cars. Before joining the industry, he was a Postdoctoral Research Associate with Carnegie Mellon University. His work on the sampling theory of graph data received the 2018 IEEE Signal Processing Society Young Author Best Paper Award. He coauthored a paper on structural health monitoring received ASME SHM/NDE 2020 Best Journal Paper Runner-Up Award and another paper on 3D point cloud processing received the Best Student Paper Award at the 2018 IEEE Global Conference on Signal and Information Processing. He contributed to the project of scene-aware interaction, winning MERL President's Award. His research interests include graph neural networks, autonomous driving, and collective intelligence.

Yonina C. Eldar (Fellow, IEEE) received the B.Sc. degree in physics, in 1995, and electrical engineering, in 1996, from Tel-Aviv University (TAU), Tel-Aviv, Israel, and the Ph.D. degree in electrical engineering and computer science, in 2002, from the Massachusetts Institute of Technology (MIT), Cambridge. She is currently a Professor with the Department of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel, where she holds the Dorothy and Patrick Gorman Professorial Chair and Heads the Center for Biomedical Engineering. She was previously a Professor with the Department of Electrical Engineering at the Technion, where she held the Edwards Chair in Engineering. She is also a Visiting Professor with MIT, a Visiting Scientist at the Broad Institute, a Visiting Research Collaborator at Princeton, an Adjunct Professor at Duke University, an Advisory Professor of Fudan University, a Distinguished Visiting Professor of Tsinghua University, and was a Visiting Professor at Stanford. She is a member of the Israel Academy of Sciences and Humanities (elected 2017) and of the Academia Europaea (elected 2023), an IEEE Fellow, a EURASIP Fellow, a Fellow of the Asia-Pacific Artificial Intelligence Association, and a Fellow of the 8400 Health Network. Her research interests are in the broad areas of statistical signal processing, sampling theory and compressed sensing, learning and optimization methods, and their applications to biology, medical imaging and optics. She has received many awards for excellence in research and teaching, including the IEEE Signal Processing Society Technical Achievement Award (2013), the IEEE/AESS Fred Nathanson Memorial Radar Award (2014), and the IEEE Kiyo Tomiyasu Award (2016). She was a Horev Fellow of the Leaders in science and technology program at the Technion and an Alon Fellow. She received the Michael Bruno Memorial Award from the Rothschild Foundation, the Weizmann Prize for Exact Sciences, the Wolf Foundation Krill Prize for Excellence in Scientific Research, the Henry Taub Prize for Excellence in Research (twice), the Hershel Rich Innovation Award (three times), the Award for Women with Distinguished Contributions, the Andre and Bella Meyer Lectureship, the Career Development Chair at the Technion, the Muriel & David Jacknow Award for Excellence in Teaching, and the Technion's Award for Excellence in Teaching (twice). She received several best paper awards and best demo awards together with her research students and colleagues including the SIAM outstanding Paper Prize, the UFFC Outstanding Paper Award, the Signal Processing Society Best Paper Award and the IET Circuits, Devices and Systems Premium Award, was selected as one of the 50 most influential women in Israel and in Asia, and is a highly cited researcher. She was a member of the Young Israel Academy of Science and Humanities and the Israel Committee for Higher Education. She was the Editor-in-Chief of *Foundations and Trends in Signal Processing*, a member of the IEEE Sensor Array and Multichannel Technical Committee and serves on several other IEEE committees. In the past, she was a Signal Processing Society Distinguished Lecturer, member of the IEEE Signal Processing Theory and Methods and Bio Imaging Signal Processing technical committees, and served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the *EURASIP Journal of Signal Processing*, the *SIAM Journal on Matrix Analysis and Applications*, and the *SIAM Journal on Imaging Sciences*. She was Co-Chair and Technical Co-Chair of several international conferences and workshops. She is Author of the book "Sampling Theory: Beyond Bandlimited Systems" and coauthor of seven other books.