# Customized Federated Learning for Multi-Source Decentralized Medical Image Classification

Jeffry Wicaksana, Zengqiang Yan, Xin Yang, *Member, IEEE*, Yang Liu, Lixin Fan, and Kwang-Ting Cheng, *Fellow, IEEE*

***Abstract*—The performance of deep networks for medical image analysis is often constrained by limited medical data, which is privacy-sensitive. Federated learning (FL) alleviates the constraint by allowing different institutions to collaboratively train a federated model without sharing data. However, the federated model is often suboptimal with respect to the characteristics of each client's local data. Instead of training a single global model, we propose Customized FL (CusFL), for which each client iteratively trains a client-specific/private model based on a federated global model aggregated from all private models trained in the immediate previous iteration. Two overarching strategies employed by CusFL lead to its superior performance: 1) the federated model is mainly for feature alignment and thus only consists of feature extraction layers; 2) the federated feature extractor is used to guide the training of each private model. In that way, CusFL allows each client to selectively learn useful knowledge from the federated model to improve its personalized model. We evaluated CusFL on multi-source medical image datasets for the identification of clinically significant prostate cancer and the classification of skin lesions.**

***Index Terms*—Federated learning, personalization, cancer diagnosis, skin cancer diagnosis.**

## I. Introduction

FEDERATED learning (FL) [1] allows multiple parties to collaboratively learn a federated model without data sharing. FL's privacy-preserving property encourages collaboration among different medical institutions to build a more generalized and accurate deep learning model for medical image analysis [2], [3]. In FL, each participating client downloads the federated model's parameters from a trusted server and updates them locally using its private data. Then, the local parameters updated from each client are aggregated to update the federated model. This process repeats until convergence.

The federated model, typically trained using federated averaging (FedAvg) [1], is in principle optimized for the average participants. For clients whose data distributions are different from the average, the federated model's performance might be inferior to the locally learned (LL) models (namely the models trained using the local data without any collaboration) [22]. In practice, this could impair the clients' desire to participate in the federation. Such a scenario is prevalent for FL's application to medical image analysis as inter-client variations [4], [5] are common due to variations in the data acquisition protocols, scanners, or calibration settings at different institutions. Instead of addressing the inter-client variations using a single federated model [25], [26], [27], which is restrictive, it is possible to train a personalized model for each client under a personalized FL (PFL) [6] framework.

Some PFL approaches require access to a public dataset with relevant tasks to perform knowledge transfer [7], [8], [9] which might not be feasible for medical data. In other PFL approaches, each client ends up with a personalized model which is a mixture of a federated and a private model, either in the output [10] or the parameter [11] space. As each client's personalized model contains a shared federated component, it may not be fully optimized for its local data. The shared federated component, trained with FL, targeting to optimize a shared model, would benefit those clients whose data shares similar distributions, while might be even detrimental to some other clients. Due to inter-client variations [4], [5], the federated component on clients whose data distributions deviate can hardly be beneficial or may even be detrimental [22].

We propose Customized FL (CusFL), a personalized federated learning (PFL) framework, where the availability of a public dataset and the intricate mixture between the federated model and the private models are not required. CusFL iteratively trains each client's personalized model instead of locally adapting [31] the federated model to each client's data post federated training, *e.g.* with fine-tuning, to avoid catastrophic forgetting. The main idea behind CusFL is that each client should train a fully customized model for itself through federated learning. With CusFL, each client trains a shared federated feature extractor and a fully personalized model, composed of both a feature

Manuscript received 5 July 2021; revised 28 February 2022 and 13 July 2022; accepted 4 August 2022. Date of publication 15 August 2022; date of current version 7 November 2022. This work was supported in part by Hong Kong General Research Fund under Grant 16203319, in part by a research grant from WeBank, and in part by the National Natural Science Foundation of China under Grants 61872417 and 62061160490. *(Corresponding author: Zengqiang Yan.)*

Jeffry Wicaksana and Kwang-Ting Cheng are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong (e-mail: jwicaksana@connect.ust.hk; timcheng@ust.hk).

Zengqiang Yan and Xin Yang are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: z_yan@hust.edu.cn; xinyang2014@hust.edu.cn).

Yang Liu is with the Institute for AI Industry Research (AIR), Tsinghua University, Beijing 100084, China (e-mail: liuy03@air.tsinghua.edu.cn).

Lixin Fan is with the WeBank, AI Lab, Shenzhen 518000, China (e-mail: lixinfan@webank.com).

Digital Object Identifier 10.1109/JBHI.2022.3198440

extractor and a classifier. Unlike the typical FL training process, each client's personalized model's parameters are never replaced by the federated model's parameters. Instead, each client's personalized model learns from its peers through feature alignment, analogous to knowledge-distillation (KD) in the feature space, with the federated feature extractor. The personalized classifier at each client guides the feature alignment to learn only the useful feature representations for its data distribution [14], [20]. The key contributions of our work are:

1) A new PFL framework: CusFL maps KD in the feature space for feature alignment between the federated model and each client-specific personalized model to effectively assist each client in training a fully customized model while selectively learning the useful knowledge from its peers.
2) The first PFL framework for medical image data: the effectiveness of the proposed CusFL framework is validated on two distinct tasks: (1) the identification of clinically significant prostate cancer [15], [16], [17] and (2) the classification of skin lesions [18].
3) Comprehensive analysis and comparison with the existing approaches: we demonstrate how CusFL manages to outperform existing frameworks.
4) The first attempt to combine PFL approaches with domain adaptation approaches: to evaluate the influence of inter-client variations, we integrate domain adaptation techniques for further performance improvement of PFL.

The paper is organized as follows. Existing approaches are summarized and discussed in Section II. Section III presents details of the proposed CusFL framework. In Section IV, we evaluate the effectiveness of the proposed CusFL framework through experiments followed by extensive comparison experiments. We provide various ablation studies in Section V and conclude the paper with Section VI.

## II. Related Work

There are two streams of approaches to handle inter-client variations in the federated setting: with a single federated model (*i.e.* classical federated learning) [25], [26], [27] and with multiple PFL models [10], [11], [21].

### A. Classical Federated Learning

The aim is to train a single federated model that is capable of handling inter-client variations. In medical image analysis, the target distributions between medical institutions may be highly different and thus a single model may not be sufficient [13]. FedProx [27], SCAFFOLD [25]. and MOON [26] attempted to address the variations by regularizing the local updates from each client with the federated model's parameters, among which MOON has been demonstrated as the most effective approach.

*MOON:* The underlying assumptions behind the design of MOON are: 1) a federated model extracts better representations compared to a locally learned model of each client, and 2) the local parameters updated from each client would worsen the learned representations. Therefore, MOON introduces a contrastive loss during the local update phase such that the representations learned are close to those of the federated model and far from those of the previous locally-updated model.

However, the assumption that the local parameters updated from each client would worsen the learned representations may not be always true. For such cases, pushing the learned features away from the previous locally-learned model's features could negatively impact the model performance for some clients.

### B. Personalized Federated Learning

PFL can be roughly divided into three categories [6]: user clustering [21], data interpolation [7], [8], [9], and model interpolation [10], [11]. User clustering and data interpolation approaches require additional private information sharing from each client which may not be an option for medical imaging scenarios. Model interpolation approaches combine models trained through FL and local learning for client-specific inference.

Typically, personalized models are trained through either locally adapting the federated model [13], [31] or optimizing the locally-learned client-specific models by referring to the federated model iteratively [10], [11], [12], [28], [29], [30]. The local adaptation techniques focus on customizing the trained federated model to each local client through post-federated training, analogous to transfer learning [32], [33], [34]. Instead of post-training, pFedMe [28], AL2SGD+ [29], and IAPGD [30] minimized the chance of counting on outliers during training using the mean-regularized approach. Specifically, they regularized the parameters of each client's personalized model to those of the global federated model. However, results have shown that these approaches may not necessarily lead to performance improvement. In addition, these approaches were only validated on multinomial logistic regression models and two-layer DNN models, and there is no evidence that they would be applicable to deep learning models.

We focus on iterative approaches, where the shared federated model and individual client's private model are mixed either in the output space [10] or the parameter space [11], [12]. We highlight two representative approaches in this category: mixture-of-experts (MoE) [10] and SplitNN [11] in detail.

*MoE:* MoE utilized FL to train a shared federated model $C_G$ and a unique private model $C_i$ for each participating client as shown in Fig. 1. For inference, the predictions of the two models of each client $i$ are ensembled using a learnable weight function $\alpha_i(x) \in [0, 1]$, defined as

$$\hat{y}_i = \alpha_i(x)C_G(x) + (1 - \alpha_i(x))C_i(x). \quad (1)$$

Ideally, $C_G$ is tasked to handle data with common characteristics across different clients, while $C_i$ is responsible for data in which the characteristics are unique to the client. Unfortunately, in practice, the gating function $\alpha_i$ may not be ideal and could be misleading. For instance, $\alpha_i$ may decide to trust $C_G$ more when performing prediction on data containing characteristics unique to the client.

*SplitNN:* SplitNN partitioned a network sequentially into private local layers and shared federated layers. One variant is to designate the first few layers as the private component and the rest of the network as the federated component, illustrated

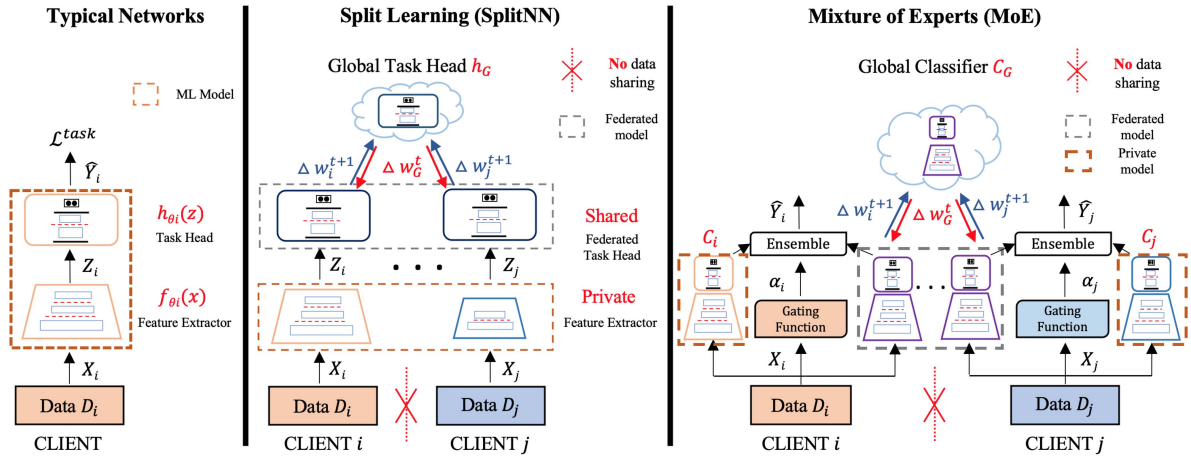


Fig. 1. Illustration of different PFL frameworks. Split Learning (SplitNN): each client has its private feature extractor followed by a federated task-specific head. Mixture of Experts (MoE): each client has its private model and a gating function as well as a shared federated model.

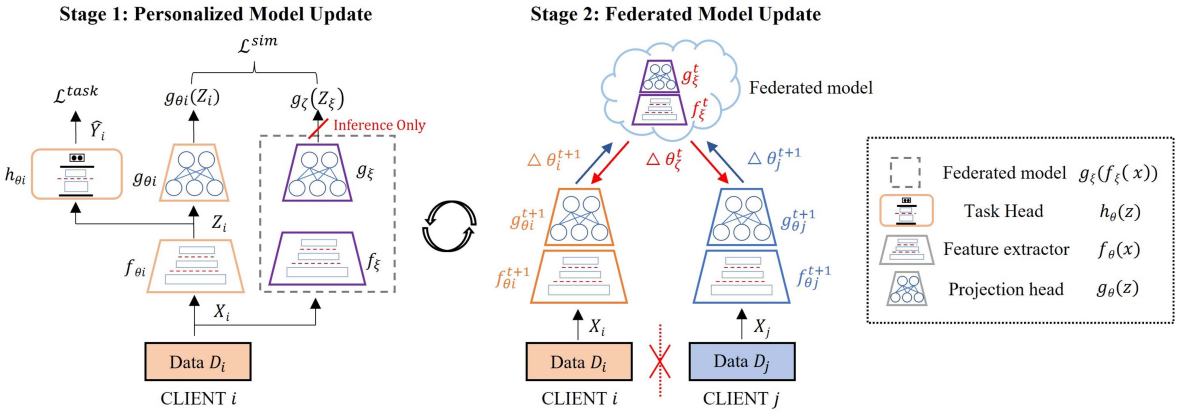


Fig. 2. Illustration of CusFL. Two-stage updates are employed during each training iteration: personalized model update (left) and federated model update (right). The federated model, which is responsible only for feature extraction, consists of a task head and a projection head. The personalized model is trained by referring to the federated model.

in the middle part of Fig. 1. Unlike personalized models trained using MoE where it is possible to trust only the private model, personalized models trained using SplitNN must use the shared federated layers due to its sequential nature. Therefore, the resulting personalized models also share the same limitation with the single-model approaches when attempting to address inter-client variations.

## III. Framework

In CusFL, each client first splits its network into two sequential blocks: a feature extractor and a task-specific head. Feature alignment is enforced at the output of the feature extractor block. Each client will then train its personalized network iteratively in two steps, personalized and federated model updates respectively, as depicted in Fig. 2. More details are presented in the following.

### A. Network Splitting

We split a deep network as a sequential block of a feature extractor $f(.)$ and a task-specific head $h(.)$. To facilitate better feature alignment, we project $Z$ to a lower-dimensional space using a 2-layer multi-layer-perceptron (MLP) $g(.)$. The output dimension of the projection head is set to 128 in this paper.

The split layer selection and the design of $g(.)$ are hyperparameters that can be tuned, depending on the network architecture and the application of interest. More details regarding the importance of $g(.)$ are presented in Section V-B.

As illustrated in the left part of Fig. 2, the personalized model of each client $i$'s consists of a feature extractor $f_{\theta i}$, a projection head $g_{\theta i}$, and a task-specific head $h_{\theta i}$, while the federated model consists of only a $f_\xi$ and a $g_\xi$.

### B. Personalized Model Update

Each client $i$ updates its personalized model, for which two key components are required: the client's private data $D_i$ and the latest federated model $f_\xi + g_\xi$. Before optimizing its personalized model, each client should first download the latest federated model parameters from the server.

In addition to the task-specific loss $L^{task}$, *i.e.* cross-entropy for classification, we add a similarity loss $L^{sim}$ to incorporate

relevant peers' collective knowledge from the federated model to the client's personalized model. Minimizing $L^{sim}$ allows $f_{\theta i}$ to learn a different set of potentially useful features [20] from $f_\xi$ for its task from $f_\xi$. In this stage, $f_\xi$ and $g_\xi$ are frozen as the goal is to help each client's personalized model learn useful features from the federated model, not the other way around.

The purpose of minimizing $L^{task}$ is to ensure that the client-specific model receive sufficient supervision for the task. $L^{task}$ is deployed to optimize $f_{\theta i}$ and $h_{\theta i}$ since the inference at each client $i$ relies only on $f_{\theta i}$ and $h_{\theta i}$, *i.e.* $\hat{Y} = h_{\theta i}(f_{\theta i}(X))$. In this study, we use cross-entropy loss (CE) as $L_\theta^{task}$ for classification, defined as

$$L_{\theta i}^{task} = \frac{1}{|D_i|} \sum_{(x,y)\in D_i} CE(h_{\theta i}(f_{\theta i}(x)), y). \tag{2}$$

As discussed above, the similarity loss is minimized to leverage the knowledge from the federated feature extractor to guide the local feature extractor, defined as

$$L_{\theta i,\xi}^{sim} = \frac{1}{|D_i|} \sum_{(x)\in D_i} 1 - \frac{< g_\xi(f_\xi(x)), g_{\theta i}(f_{\theta i}(x)) >}{\|g_\xi(f_\xi(x))\|_2 . \|g_{\theta i}(f_{\theta i}(x))\|_2}. \tag{3}$$

Then, the overall loss function for local model update in CusFL is defined as

$$L_{\theta i,\xi}^{cus} = \lambda_1 L_{\theta i}^{task} + \lambda_2 L_{\theta i,\xi}^{sim}, \tag{4}$$

where $\lambda_1$ and $\lambda_2$ tuning hyper-parameters. In our experiments, we empirically find that dynamically adjusting the relative importance to maintain $\lambda_1 L^{task} = \lambda_2 L^{sim}$ leads to the best results. The effect of tuning these hyper-parameters is described in Section V-D.

### C. Federated Model Update

After personalized model update, each client sends its updated parameters of $f(.)$ and $g(.)$, *i.e.* $\Delta\theta_i$ from client $i$, to the server to update $f_\xi$ and $g_\xi$. The server then deploys a standard FL algorithm, namely FedAvg [1] to update the federated model according to

$$\xi \leftarrow \xi + \frac{1}{N} \sum_{k=1}^{N} \Delta\theta k, \tag{5}$$

where $N$ is the total number of clients. In practice, the federated model can be updated using a wide range of existing aggregation techniques.

### D. Key Differences From Existing Approaches

Compared to existing PFL approaches, CusFL differs in how each client's personalized model benefits from the shared federated model. More specifically:

1) *MoE [10]:* model mixing in the output space is synonymous with ensembling. The federated model might generate incorrect predictions for some clients whose data distributions significantly deviate from the average. For such cases, it is better to rely more on each client's private model for prediction.
2) *SplitNN [11]:* in SplitNN, proper alignment between the federated model and each client's private model is crucial to ensure the effectiveness of each client's personalized model. Little parameter-level misalignment between the private and the federated models can lead to significant output divergence, impairing the personalized model's performance.
3) *CusFL:* CusFL utilizes feature alignment rather than a simple mixture of the federated and each client's private model in the output space (MoE [10]) or the parameter space (SplitNN [11]). Feature alignment is less restrictive than model mixing because it is implemented only as a regularization term. Therefore, it does not pursue exact parameter-level alignment like SplitNN nor compatible output-level alignment like MoE, which may not be possible to achieve. Each client in CusFL can then effectively capture more useful features from the federated model. With a better set of features, better client-customized, tasks-specific layers can be learned, leading to a better performing model for each client.

Compared to MOON [26], the differences lie in the number of resulting models, *i.e.* single vs. multiple, and the anchors used for training the feature extractor. The strong assumption that the local model updates would worsen the quality of learned representations has to be valid for MOON to work because the contrastive loss aims to push the learned features away from the features extracted from the previous local model. Comparatively, CusFL is designed by taking into account that each client may have a different set of optimal features for its data distribution. Moreover, the role of the federated model in CusFL is less restrictive compared to MOON since it only serves as guidance and is never used to directly replace the client-specific feature extractor. As a result, each client can own a client-customized model instead of conforming to the average data distribution. For comparison, an ablation study to compare the effectiveness of the contrastive loss in MOON and the similarity loss in CusFL is presented in Section V-A.

## IV. Evaluation

### A. Dataset and Preprocessing

We conducted case studies on two different multi-source medical applications: 1) clinically significant (CS) prostate cancer (PCa) classification and 2) malignant pigmented skin lesion classification. Here, we adopted multi-source datasets because they simulate realistic data silos between medical institutions. The consequences of data silos are as follows:

1) *Variations in patients' characteristics:* given geologically separate medical image datasets, patients' physical and physiological attributes might differ which potentially translates to varying symptoms for the same diseases.
2) *Variations in data characteristics:* different tools and methods used to acquire medical imaging data would affect the image characteristics and may create systematic variations across different datasets.

TABLE I
STATISTICS OF THE CS PCA DATASET

| Site | # Patients | # Images | # nonCS PCa / # CS PCa |
|---|---|---|---|
| LocalPCa | 135 | 825 | 547 / 278 |
| PROSTATEx | 188 | 1273 | 1041 / 232 |

TABLE II
STATISTICS OF THE HAM10000 DATASET

| Site | Source | # Patients | # Images | # Benign / # Malignant |
|---|---|---|---|---|
| Rosendahl | rosendahl | 1552 | 2259 | 1326 / 933 |
| Vidir | modern | 1695 | 3363 | 2440 / 923 |
| | old | 278 | 439 | 376 / 72 |
| | molemax | 3954 | 3954 | 3928 / 26 |

3) *Variations in data amount:* different datasets may contain varying amounts of data which affects the bias of the learned federated model.

*CS PCa Classification:* Accurate and timely identification of patients with CS PCa, *i.e.* patients whose Gleason scores are equal to or greater than 7, can significantly increase the survival rate. We perform CS PCa classification based on apparent diffusion coefficient (ADC) images. Two different image sources including a publicly available dataset PROSTATEx [15], [16], [17] and a privately collected dataset LocalPCa were used for evaluation. Specifically, we selected 188 patients' data, consisting of 64 CS PCa and 124 non-CS PCa patients, from PROSTATEx and used 135 patients' data from LocalPCa, consisting of 65 CS PCa and 70 nonCS PCa patients, all based on the quality of pixel-wise annotations by our experts. Statistics of the CS PCa dataset are presented in Table I.

The augmentation methods utilized in [22] were adopted to balance the ratio between CS PCa and non-CS PCa training images, including random non-rigid image deformation and random flip.

*Malignant pigmented skin lesion classification:* The publicly-available HAM10000 dataset [18], collected from four different sources located at two different sites was used for evaluation. Statistics of the dataset are presented in Table II. Following the class descriptions in HAM10000, we grouped the skin images categorized as actinic keratoses (akiec), melanoma (mel), and basal cell carcinoma (bcc) into the malignant class, and the skin images categorized as benign keratosis (bkl), dermatofibroma (df), melanocytic nevi (nv), and vascular skin lesions (vasc) into the benign class.

As skin lesion images are naturally captured from different views, with varying magnifications, and using different cameras, no additional image augmentation was adopted to validate the performance of different approaches under severe inter-client variations, *i.e.* Vidir-old has only 439 images in total while the other clients have around 3,000 images, and almost 99% of the images in Vidir-molemax are benign.

## B. Evaluation Metrics

We evaluated the models' performance using the balanced classification accuracy (ACC) with a prediction threshold of 0.5, and the area under a curve (AUC).

## C. Implementation Details

In our experiments, the federated training, performed using synchronous FL, is simulated on a single machine. The federated model update is performed after one training round. We also include centralized learning to provide a baseline performance when data sharing is feasible.

*1) Network Architectures:* Two network architectures are used for the two different studies:

1) The custom network in [22] for CS PCa classification. The network is composed of 4x convolutional-pooling blocks followed by 2× residual convolution blocks and fully connected layers. We split the network into a feature extractor and a task-specific head at the output of the 3rd convolutional-pooling block.
2) Resnet18 [24] for malignant pigmented skin lesion classification. Resnet18 is composed of 18 convolution layers. We split the network at the output of the 9th convolution layer, *i.e.* at the output of the 3rd convolution blocks.

Both SplitNN and CusFL follow the same network splitting way (as described above).

*2) Optimization:* All the networks were implemented within PyTorch and were trained for 200 rounds using ADAM optimizer with a learning rate of 1e-3 and a batch size of 32. By default, FedAvg [1] was selected as the aggregation function for federation.

## D. Experiment Setup

The experiment settings with 2, 4, 8, 16, 32, and 64 clients on the two application drivers are as follows:

1) *CS PCa classification:* We maintained the original split based on the data sources, *i.e.* LocalPCa and PROSTATEx for the 2-client setting. For the 4-, 8-, 16-, and 32-client settings, we randomly split the training set of the two data sources into 2, 4, 8, and 16 subsets respectively. For CS PCa classification, the 64-client setting was not implemented due to the limited data size.
2) *Skin lesion classification:* For the 2-client setting, we split the HAM10000 [18] dataset according to the hospital sources, *i.e.* Vidir and Rosendahl. For the 4-client setting, we divided the data according to the original data sources. For the 8-, 16-, 32-, and 64-client settings, we randomly split the training set of each of the four data sources into 2, 4, 8, and 16 subsets respectively.

We conducted 5-fold validation by randomly splitting each data source according to 80%/20% train-test split for each fold. To gain a better insight into different learning frameworks, evaluations were performed from two aspects: the private testing aspect and the public testing aspect. For private testing, we evaluated each client's model on the original test set of the corresponding data source, namely the training set and the testing set

TABLE III
COMPARISON RESULTS OF LOCALLY-LEARNED MODELS FOR CS PCA CLASSIFICATION. THE RESULTS ARE AVERAGED FROM 5-FOLD VALIDATION AND THE STANDARD DEVIATIONS ARE REPORTED IN BRACKETS

| LL | Metrics | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 |
|---|---|---|---|---|---|---|---|---|---|
| 2 clients | ACC(%) | 85.1(4.1) | 57.3(4.8) | | | | | | |
| | AUC(%) | 90.4(4.7) | 62.8(1.4) | | | | | | |
| 4 clients | ACC(%) | 81.2(5.2) | 55.6(5.4) | 84.7(2.4) | 61.3(7.4) | | | | |
| | AUC(%) | 89.6(5.2) | 55.3(2.6) | 90.2(0.3) | 61.3(7.6) | | | | |
| 8 clients | ACC(%) | 67.6(6.7) | 52.3(0.6) | 76.6(4.2) | 58.3(9.9) | 83.0(5.8) | 52.0(0.9) | 81.0(6.5) | 61.2(12) |
| | AUC(%) | 72.1(8.6) | 52.8(3.9) | 84.2(5.0) | 62.6(11) | 88.4(6.3) | 51.9(1.6) | 86.7(5.7) | 65.3(14) |

TABLE IV
COMPARISON RESULTS (*I.E.* AVG. ACC) OF LEARNING FRAMEWORKS FOR CS PCA CLASSIFICATION UNDER DIFFERENT SETTINGS: "PRIVATE" REPRESENTS PRIVATE TESTING (*I.E.*, TESTING ON EACH CLIENT'S DATA) AND "PUBLIC" IS PUBLIC TESTING (*I.E.*, TESTING ON THE SAME COLLECTION OF TESTING DATA FROM ALL CLIENTS)

| Frameworks | Avg. ACC (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 2 Clients | | 4 Clients | | 8 Clients | | 16 Clients | | 32 Clients | |
| | private | public | private | public | private | public | private | public | private | public |
| CL | 88.7(2.6) | | 89.2(3.9) | | 90.5(4.9) | | 91.2(3.4) | | 89.1(2.1) | |
| LL | 71.2(1.1) | 71.8(2.4) | 70.8(1.7) | 70.3(1.2) | 66.5(1.9) | 66.9(3.1) | 63.9(2.0) | 63.3(1.2) | 61.0(1.5) | 61.7(1.4) |
| FL [1] | 85.8(7.3) | 74.4(6.0) | 83.5(6.3) | 73.9(5.2) | 84.2(8.1) | 73.0(3.1) | 78.0(4.8) | 66.3(5.8) | 72.8(6.9) | 66.3(5.0) |
| MOON [26] | 86.5(3.5) | 77.1(2.4) | 85.6(4.1) | 74.3(6.9) | 84.5(4.2) | 74.7(4.4) | 79.0(8.5) | 69.9(7.0) | 72.1(1.5) | **69.5**(8.0) |
| MoE [10] | 86.0(5.4) | 75.0(**2.1**) | 88.0(**1.7**) | 73.6(2.1) | 84.2(4.4) | 73.4(2.0) | 77.7(5.0) | 68.3(5.1) | 72.3(2.9) | 65.0(9.0) |
| SplitNN [11] | 79.0(5.3) | 71.0(4.6) | 77.4(10) | 71.5(3.1) | 70.3(9.4) | 73.9(3.0) | 69.0(5.8) | 69.0(5.8) | 72.6(3.7) | 62.3(4.0) |
| CusFL (ours) | **87.5(3.0)** | **80.5**(3.7) | **88.3**(3.1) | **78.4(1.8)** | **86.1(1.6)** | **75.2(2.0)** | **80.8(2.2)** | **70.0(1.0)** | **73.8(1.5)** | 66.2(**0.9**) |

The standard deviations are reported in brackets and the best results are shown in bold.

TABLE V
COMPARISON RESULTS (*I.E.* AVG. AUC) OF LEARNING FRAMEWORKS FOR CS PCA CLASSIFICATION UNDER DIFFERENT SETTINGS: "PRIVATE" REPRESENTS PRIVATE TESTING (*I.E.*, TESTING ON EACH CLIENT'S DATA) AND "PUBLIC" IS PUBLIC TESTING (*I.E.*, TESTING ON THE SAME COLLECTION OF TESTING DATA FROM ALL CLIENTS)

| Frameworks | Avg. AUC (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 2 Clients | | 4 Clients | | 8 Clients | | 16 Clients | | 32 Clients | |
| | private | public | private | public | private | public | private | public | private | public |
| CL | 96.2(1.9) | | 95.5(3.1) | | 94.9(5.4) | | 93.7(3.5) | | 91.2(2.1) | |
| LL | 76.6(5.7) | 74.3(2.4) | 74.2(2.8) | 73.3(2.0) | 70.5(0.3) | 72.3(4.5) | 70.4(2.8) | 69.2(1.5) | 61.1(1.3) | 68.9(1.3) |
| FL [1] | 90.5(6.5) | 86.8(1.7) | 91.3(3.7) | 83.9(9.2) | 91.3(3.1) | 82.2(8.3) | 88.8(5.1) | 76.0(2.4) | 81.4(3.5) | 70.2(1.6) |
| MOON [26] | 95.0(2.0) | 87.7(**1.0**) | 87.9(7.9) | 83.1(9.2) | 88.0(4.2) | 80.3(4.2) | 86.6(3.6) | 74.4(3.4) | 81.2(2.9) | 71.2(6.5) |
| MoE [10] | 94.0(2.8) | 83.8(2.2) | 94.2(0.9) | 84.4(2.5) | 91.0(1.6) | 82.8(3.8) | 84.6(1.8) | 77.3(1.7) | 82.3(1.4) | 72.7(2.8) |
| SplitNN [11] | 87.4(3.5) | 84.7(7.6) | 87.6(4.4) | 84.2(7.2) | 86.9(4.5) | 80.2(5.3) | 84.4(9.6) | 75.6(8.4) | 80.7(5.5) | 72.0(1.7) |
| CusFL (ours) | **95.6(0.9)** | **88.7**(2.2) | **94.9(1.4)** | **88.5(1.4)** | **92.9(0.9)** | **84.3(1.6)** | **87.2(1.7)** | **78.6(1.7)** | **82.9(0.2)** | **73.4(0.4)** |

The standard deviations are reported in brackets and the best results are shown in bold.

coming from the same source. In this way, we can estimate how much benefit each client receives from the federation. For public testing, all clients' models were evaluated on the evaluation set consisting of the test sets coming from all data sources. Through this, we can assess how well each model performs on unseen data (*i.e.* generalizability).

### *E. Study on CS PCa Classification*

We assigned data from LocalPCa to the odd clients and PROSTATEx to the even clients. To better understand the characteristics of data on each client, we first evaluate the performance of the locally-learned models as summarized in Table III. The performance gaps between the odd clients and the even clients reflect the quality discrepancy between the two image sources, which commonly exists in clinical practice. With more clients and thus the amount of each client's local training data reduced, it becomes more difficult to train a high-quality local model. As a result, the average accuracy of C1 drops from 85.1% under the 2-client setting to 67.6% under the 8-client setting. Similar performance degradation can be observed for the 4-client setting. The increase in the standard deviations further validates this point. In addition, the model performances of different clients vary dramatically which becomes even more obvious with the increasing number of clients. In the 8-client setting, the maximum ACC gap is 31% while the corresponding AUC gap is 35.6%.

Comparison results of learning frameworks on Avg. ACC and Avg. AUC under different settings are presented in Tables IV and V respectively. In addition to LL and FL, centralized learning (CL) is implemented as a potential upper bound of FL frameworks for comparison. Comparing CL to LL, we find significant performance gaps, *e.g.* Avg. ACC from 71.2% to 88.7% and Avg. AUC from 76.6% to 96.2% under the 2-client setting. The observation that the performance gaps between LL and CL increase when the number of clients increases signifies the importance of privacy-preserving collaboration through FL.

According to Tables IV and V, FL frameworks consistently outperform LL under all settings. Comparing the results of different FL frameworks on private testing and public testing,

TABLE VI
COMPARISON RESULTS OF LOCALLY-LEARNED MODELS FOR SKIN LESION CLASSIFICATION

| LL | Metrics | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 |
|---|---|---|---|---|---|---|---|---|---|
| 2 clients | ACC(%) | 74.8(0.9) | 76.9(0.6) | | | | | | |
| | AUC(%) | 82.0(0.5) | 86.8(0.4) | | | | | | |
| 4 clients | ACC(%) | 74.8(0.9) | 64.3(2.1) | 52.2(0.8) | 53.5(1.6) | | | | |
| | AUC(%) | 82.0(0.5) | 72.8(6.4) | 54.7(4.5) | 62.5(3.2) | | | | |
| 8 clients | ACC(%) | 72.1(2.1) | 69.3(0.8) | 65.5(0.9) | 67.5(1.1) | 56.9(5.8) | 58.6(4.5) | 55.9(4.8) | 76.6(6.4) |
| | AUC(%) | 80.3(1.2) | 77.3(1.1) | 76.4(2.4) | 76.3(0.9) | 68.9(2.4) | 62.3(3.6) | 88.6(5.7) | 90.9(5.2) |

The results are averaged from 5-fold validation and the standard deviations are reported in brackets.

we observe considerable performance gaps. One interesting observation is that better performance on public testing does not necessarily lead to better performance on private testing, especially when the number of clients is relatively large. It demonstrates the need and value of PFL to improve individual clients' performance.

Next, we analyze the performances of different learning frameworks starting from the 2-client setting. In terms of private testing, MOON outperforms other PFL approaches in both Avg. ACC and Avg. AUC, validating its effectiveness in extracting useful information from each client for federation. Comparatively, data heterogeneity makes it counterproductive to pursue parameter-level alignment in SplitNN, degrading its performance. As output-level alignment is less sensitive to data heterogeneity, MoE performs better than FL and SplitNN but slightly worse than MOON. Compared to the above PFL approaches, CusFL achieves the best performance by training a fully personalized model for each client through feature-level alignment with the federated model.

The impact of inter-client variations is more apparent in public testing. In general, the performance gains compared to LL are much lower than those on private testing. In some cases, SplitNN even fails to outperform LL, highlighting the shortcoming of parameter-level alignment in alleviating inter-client variations. With an increasing number of clients, the inter-client variation problem becomes more severe. As a result, considerable performance degradation incurs for all learning frameworks. It should be pointed out that though MOON performs better than CusFL in public testing under the 32-client setting, its private testing performance is worse. It is due to that CusFL is designed primarily to target for private testing (*i.e.* personalization) while MOON is primarily for public testing (*i.e.* generalization). In general, CusFL stably outperforms other learning frameworks under all settings, validating its effectiveness.

### F. Study on Malignant Pigmented Skin Lesion Classification

For 8-, 16-, 32-, and 64-client settings, we split the 4 original data sources: Rosendahl, Vidir-modern, Vidir-old, and Vidir-molemax into 2, 4, 8, and 16 subsets respectively.

Quantitative results of LL under different settings are summarized in Table VI. The images of Vidir-old are digitized analog images with greater noise while Vidir-molemax has a severe class imbalance problem. As a result, the performances of those clients corresponding to the above two sources are much worse than others. Taking into account the standard deviations, client performances vary significantly under the 8-client setting, indicating large inter-client data variations in both quantity and quality.

Quantitative results of different learning frameworks on Avg. ACC and Avg. AUC are summarized in Tables VII and VIII respectively. For private testing, FL has the worst performance, even lower than LL, mainly due to the more severe inter-client variation problem in skin lesion classification compared to that in CS PCa classification. Consequently, training a single federated model may not be a winning strategy. For public testing, FL outperforms LL, which is consistent with the intuition that the federated model by FL should be more generalizable compared to the locally-learned models, especially with the existence of inter-client variations among the test sets in public testing. Under the 2-client setting, all learning frameworks outperform FL on private testing, while only CusFL achieves a better performance on public testing. It validates the effectiveness of feature alignment in CusFL compared to parameter aggregation in FL. With an increasing number of clients, SplitNN, MoE, and MOON gradually outperform FL, showing the limitations of FL in alleviating inter-client variations. For public testing, greater inter-client variations further degrade the performances of various federated approaches, making them quite similar.

Under all settings, CusFL stably outperforms other learning frameworks, highlighting the advantages of training a fully personalized model for each client in FL. In addition, CusFL manages to strike a better balance between the federated model and each client's personalized model, leading to lower standard deviations when the number of clients is high (*i.e.* 16-client, 32-client, and 64-client settings).

## V. DISCUSSION

We conducted a series of ablation studies to gain a better understanding of CusFL.

### A. Similarity Loss

We compare the average performances of the personalized models trained using the CusFL framework based on the proposed similarity loss in (3) and the model contrastive loss used in MOON [26]. According to the quantitative results in Table IX, pushing the extracted features away from the previous local model's features, *i.e.* using a model contrastive loss, is detrimental and leads to lower average performance.

### B. The Importance of Projection Layer

To validate the effectiveness of feature projection, we conduct additional experiments on the same backbone with and without using the projection layers as stated in Fig. 3. Based on the

TABLE VII

COMPARISON RESULTS (*I.E.* AVG. ACC) OF LEARNING FRAMEWORKS FOR SKIN LESION CLASSIFICATION UNDER DIFFERENT SETTINGS: "PRIVATE" REPRESENTS PRIVATE TESTING (*I.E.*, TESTING ON EACH CLIENT'S DATA) AND "PUBLIC" IS PUBLIC TESTING (*I.E.*, TESTING ON THE SAME COLLECTION OF TESTING DATA FROM ALL CLIENTS)

| Frameworks | Avg. ACC (%) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 2 Clients | | 4 Clients | | 8 Clients | | 16 Clients | | 32 Clients | | 64 Clients | |
| | private | public | private | public | private | public | private | public | private | public | private | public |
| CL | 83.9(1.4) | | 82.7(2.9) | | 82.3(2.9) | | 82.6(2.7) | | 81.4(3.6) | | 81.5(2.5) | |
| LL | 77.3(1.2) | 67.9(0.8) | 72.0(1.6) | 56.9(0.8) | 65.3(0.9) | 55.5(1.2) | 62.2(0.9) | 52.9(0.4) | 58.6(0.5) | 51.9(0.2) | 56.9(0.2) | 51.1(0.4) |
| FL [1] | 73.2(1.1) | 76.6(1.8) | 57.3(1.7) | 60.5(3.0) | 57.1(1.7) | 59.8(2.0) | 57.2(2.1) | 57.4(3.0) | 54.8(1.5) | 54.4(1.7) | 55.5(2.8) | 54.0(1.0) |
| MOON [26] | 76.7(1.6) | 72.4(**0.7**) | 70.5(**1.3**) | 61.0(1.3) | 67.8(1.6) | 59.1(1.8) | 63.5(1.1) | 56.0(2.0) | 59.4(1.6) | 53.2(1.7) | 55.0(0.6) | 51.9(0.4) |
| MoE [10] | 76.2(0.9) | 69.5(1.3) | 75.0(2.8) | 62.1(2.5) | 69.4(2.2) | 58.4(2.2) | 69.6(3.6) | 55.2(0.8) | 68.0(1.7) | 54.6(2.3) | 68.0(0.8) | 54.9(1.4) |
| SplitNN [11] | 75.8(**0.4**) | 70.0(1.0) | 68.4(1.5) | 57.7(1.5) | 66.1(**1.1**) | 54.6(**1.0**) | 64.3(0.8) | 52.0(1.2) | 62.0(1.0) | 50.2(0.3) | 62.7(0.7) | 50.2(**0.2**) |
| CusFL (ours) | **79.1**(1.2) | **77.7**(1.4) | **81.0**(3.0) | **64.4(1.2)** | **76.6**(2.3) | **62.8**(2.0) | **73.8(0.8)** | **60.7(0.8)** | **70.0(1.0)** | **59.6(0.3)** | **68.6(0.6)** | **60.4**(0.9) |

The standard deviations are reported in brackets and the best results are shown in bold.

TABLE VIII

COMPARISON RESULTS (*I.E.* AVG. AUC) OF LEARNING FRAMEWORKS FOR SKIN LESION CLASSIFICATION UNDER DIFFERENT SETTINGS: "PRIVATE" REPRESENTS PRIVATE TESTING (*I.E.*, TESTING ON EACH CLIENT'S DATA) AND "PUBLIC" IS PUBLIC TESTING (*I.E.*, TESTING ON THE SAME COLLECTION OF DATA FROM ALL CLIENTS)

| Frameworks | Avg. AUC (%) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 2 Clients | | 4 Clients | | 8 Clients | | 16 Clients | | 32 Clients | | 64 Clients | |
| | private | public | private | public | private | public | private | public | private | public | private | public |
| Cl | 89.6(3.2) | | 86.2(5.0) | | 85.8(2.5) | | 86.4(6.8) | | 87.2(7.9) | | 88.6(6.4) | |
| LL | 86.8(0.4) | 76.0(0.8) | 80.9(1.3) | 62.0(2.3) | 77.6(1.3) | 59.7(1.4) | 71.8(1.0) | 55.1(1.6) | 68.7(1.2) | 53.8(0.8) | 65.9(0.6) | 52.6(0.8) |
| FL [1] | 84.2(2.3) | **81.6**(2.7) | 70.8(3.2) | 73.5(3.1) | 70.1(2.9) | 70.2(2.6) | 68.6(4.9) | 65.3(1.5) | 68.1(2.3) | 63.8(0.6) | 65.9(3.2) | 62.8(1.2) |
| MOON [26] | 86.8(0.4) | 80.6(2.1) | 81.6(1.5) | 69.3(1.7) | 77.7(**0.4**) | 66.1(1.6) | 74.7(1.4) | 66.2(4.1) | 69.5(0.3) | 63.2(3.1) | 63.5(1.6) | 60.4(2.8) |
| MoE [10] | 86.5(2.6) | 79.8(2.8) | 80.7(1.2) | 68.1(3.7) | 74.7(2.5) | 65.2(4.4) | 73.7(3.7) | 60.8(1.6) | 71.5(1.7) | 61.6(1.7) | 72.7(1.7) | 63.4(1.8) |
| SplitNN [11] | 86.5(0.5) | 77.8(1.7) | 75.3(2.5) | 62.4(1.8) | 71.8(1.6) | 61.6(1.5) | 69.6(0.8) | 56.1(1.4) | 66.6(1.2) | 57.5(2.8) | 65.4(0.8) | 58.2(1.5) |
| CusFL (ours) | **87.7(0.4)** | 80.9(**0.6**) | **86.1(1.0)** | **74.0(1.7)** | **84.2**(1.1) | **70.7(0.2)** | **80.7(0.2)** | **68.3(0.8)** | **78.2(0.2)** | **67.6(0.4)** | **76.0(0.2)** | **66.7(0.4)** |

The standard deviations are reported in brackets and the best results are shown in bold.

TABLE IX

PERFORMANCE COMPARISON (*I.E.* AVG. ACC AND AVG. AUC) OF THE CUSFL FRAMEWORK WITH THE PROPOSED SIMILARITY LOSS VS. THE MODEL CONTRASTIVE LOSS IN MOON [26]

| CusFL | Classification Task | | | |
|---|---|---|---|---|
| | CS PCa | | Skin Lesion | |
| | ACC (%) | AUC (%) | ACC (%) | AUC (%) |
| contrastive loss | 87.92 | 91.12 | 77.41 | 86.74 |
| $L^{sim}$ | **90.37** | **92.59** | **81.34** | **87.86** |

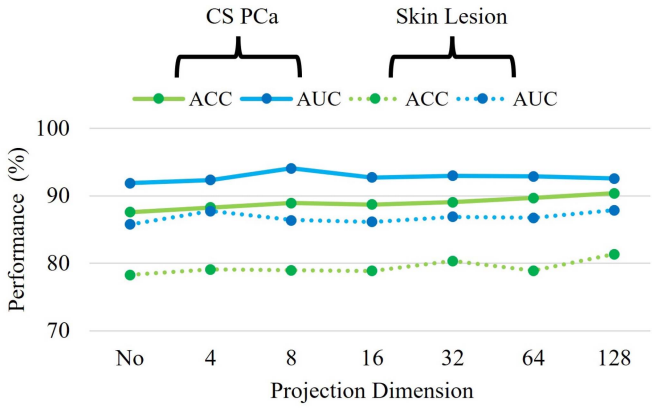


Fig. 3. The average ACC and AUC of the models learned using CusFL with varying projection dimensions for CS PCa classification and skin lesion classification.

quantitative results, using the projection layers stably improves the performance of the personalized models trained by CusFL (regardless of the projection dimensions) as it allows each client to retain its client-customized information at the output of the personalized feature extractor, $f_{\theta i}$. One of the repercussions of promoting similarity between the federated features and client-specific features is the need to minimize client-specific features if the other clients do not have them. As the projection layers extract the client-independent information for feature alignment while leaving the useful client-specific information intact in its features, greater model performance can be obtained with the projection layers.

TABLE X

PERFORMANCE COMPARISON (*I.E.* AVG. ACC AND AVG. AUC) OF THE CUSFL FRAMEWORK WHEN THE FEDERATED MODEL (FM) IS FROZEN VS. NON-FROZEN DURING TRAINING OF THE PERSONALIZED LOCAL UPDATE PHASE

| CusFL | Classification Task | | | |
|---|---|---|---|---|
| | CS PCa | | Skin Lesion | |
| | ACC (%) | AUC (%) | ACC (%) | AUC (%) |
| non-frozen FM | 87.52 | 92.55 | 80.50 | 87.33 |
| frozen FM | **90.37** | **92.59** | **81.34** | **87.86** |

### C. Does Freezing the Federated Model Helps Training in the Personalized Model Update Phase?

The performance of the personalized models learned through CusFL, as shown in Table X, would be better if we freeze the federated feature extractor during the training of the personalized model update phase. The reasons behind this are as follows:

1) It is harder to match moving targets than frozen targets.
2) Jointly training the federated model and the private model would force the federated feature extractor to match the client-customized feature extractor. Consequently, the learning process puts more emphasis on the local data, and largely ignores the training data from other clients, making it more similar to local learning.

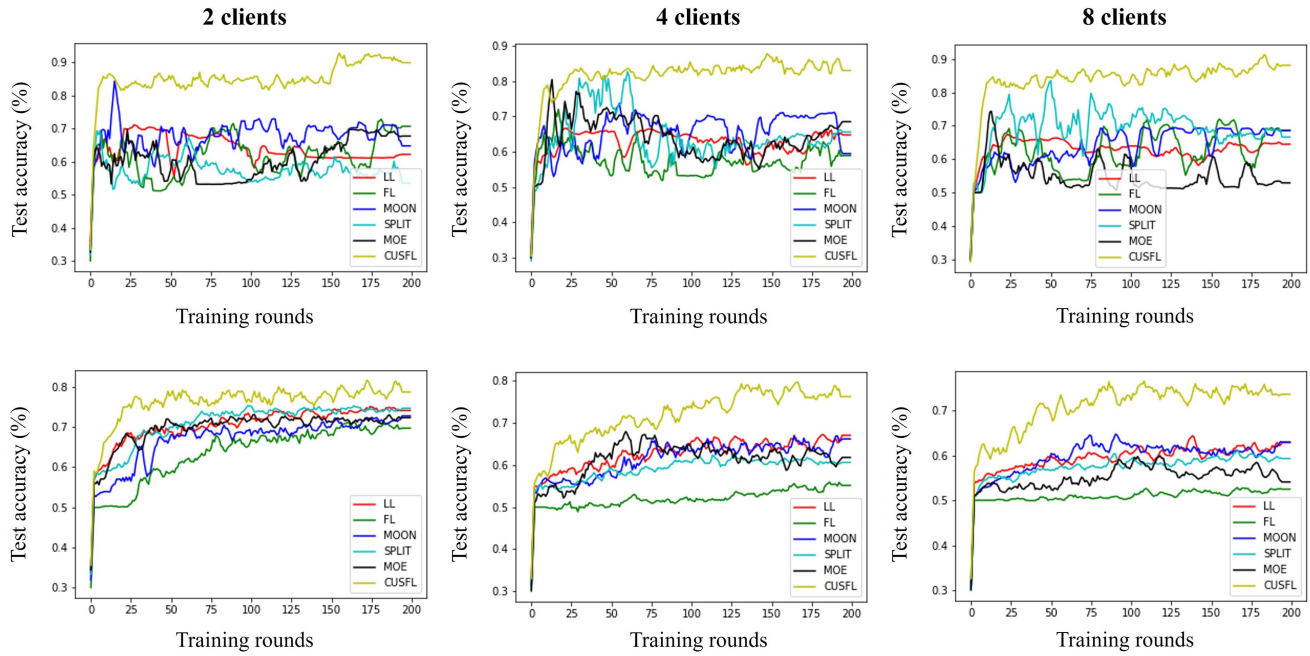


Fig. 4. Testing accuracy curves of different learning frameworks for CS PCa (Row 1) and skin lesion (Row 2) classification under 2-client, 4-client, and 8-client settings.

TABLE XI
PERFORMANCE COMPARISON (*I.E.* AVG. ACC AND AVG. AUC) OF THE CUSFL FRAMEWORK WITH A VARYING WEIGHT RATIO $r$ ACROSS CLIENTS

| r | Classification Task | | | |
|---|---|---|---|---|
| | CS PCa | | Skin Lesion | |
| | ACC (%) | AUC (%) | ACC (%) | AUC (%) |
| 0 | 86.17 | 90.99 | 78.95 | 86.63 |
| 0.3 | 85.49 | 92.55 | 80.88 | 87.62 |
| 0.4 | 87.71 | 91.51 | 81.02 | 87.01 |
| 0.5 | **90.37** | 92.59 | **81.34** | **87.86** |
| 0.6 | 88.01 | 92.02 | 79.52 | 86.88 |
| 0.7 | 87.86 | **92.76** | 78.96 | 87.68 |

### D. How Do $\lambda_1$ and $\lambda_2$ Affect CusFL?

According to the definition of the joint loss $L^{cus}$ in (4), $\lambda_1$ determines the importance of the task-specific loss such as cross-entropy loss, while $\lambda_2$ indicates the degree of influence of the federated model. Here, we study the effect of how the ratio between $\lambda_1 L^{task}$ and $\lambda_2 L^{sim}$ affects the performance of the learned models. As the ranges of $L^{task}$ and $L^{sim}$ might be different, we dynamically adjust the value of $\lambda_2$ to maintain the weight ratio $r = \lambda_2 L^{sim}/L^{cus}$ during training. Specially, setting $\lambda_2$ to 0 implies that each client will train its model locally, without any interaction from its peers.

According to the results in Table XI, for both CS PCa and skin lesion classifications, setting $r = 0.5$ strikes the best balance between the client-specific task loss $L^{task}$ and the feature alignment loss with the federated model $L^{sim}$, demonstrated by the highest Avg. ACC across clients. In practice, not all participating clients need to use the same value of $r$ during training. For instance, clients whose data distributions are different from the average may find it better to rely more on their local data rather than the federated model. Then, these clients can independently select a smaller $r$ without impacting the training of other clients' personalized models.

### E. How Does CusFL Converge Compared to Other Methods?

We follow the comparison method in [1] to gain a better understanding of CusFL's convergence compared to other methods. For both CS PCa and skin lesion classification, we plot the average test accuracy curves and the average training loss curves of different learning frameworks under 2-, 4-, and 8-client settings. Considering that the targeted convergence set by MOON is $-log(0.5)$ instead of 0, we further modify the training loss curves of MOON as $l_{moon} = l_{moon} + log(0.5)$ for fair comparison.

As shown in Fig. 4, CusFL's average testing accuracy is higher than other frameworks on both application drivers. In Fig. 4, the testing accuracy curves of other learning frameworks vary considerably due to data limitations for CS PCa classification. The trend is consistent across 2-, 4- and 8-client settings. The corresponding curves on skin lesion classification are more stable as shown in Fig. 4, due to the availability of more training data. Under all settings, the results show that FL performs the worst, as a result of high inter-client variations. MOON helps reduce the impact of inter-client variations, achieving greater testing accuracy while CusFL outperforms all other approaches.

The training loss curves of CusFL converge faster than those of the other learning frameworks for both CS PCa and skin lesion classification, as shown in Fig. 5. In contrast, the convergence performance of FL varies dramatically on the two application drivers. For CS PCa classification, FL converges better than other learning frameworks under 2- and 4-client settings. When the

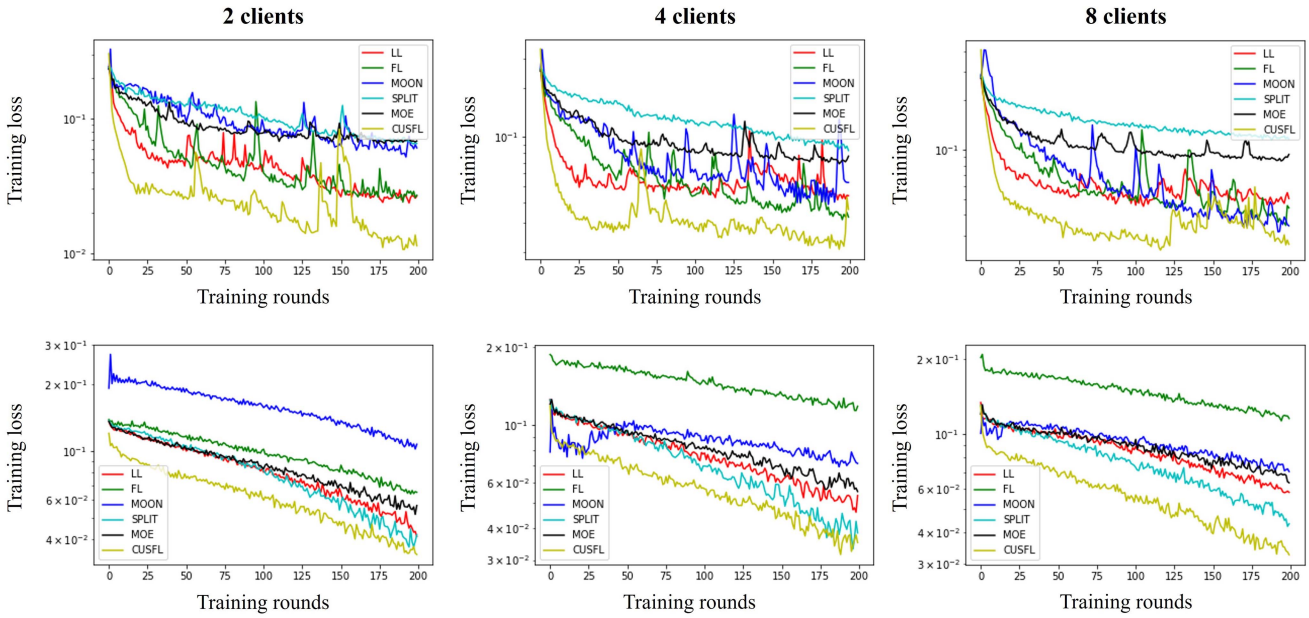


Fig. 5. Training loss curves of different learning frameworks for CS PCa (Row 1) and skin lesion (Row 2) classification under 2-client, 4-client, and 8-client settings. The loss is presented in *log* scale.

TABLE XII

COMPARISON RESULTS (*I.E.* AVG. ACC) OF LEARNING FRAMEWORKS COUPLED WITH THE VARIATION-AWARE MODULE TO REDUCE INTER-CLIENT VARIATIONS [22] FOR CS PCA CLASSIFICATION UNDER DIFFERENT SETTINGS: "PRIVATE" REPRESENTS PRIVATE TESTING (*I.E.*, TESTING ON EACH CLIENT'S DATA) AND "PUBLIC" IS PUBLIC TESTING (*I.E.*, TESTING ON THE SAME COLLECTION OF TESTING DATA FROM ALL CLIENTS)

| Frameworks (VA) | Avg. ACC(%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 2 Clients | | 4 Clients | | 8 Clients | | 16 Clients | | 32 Clients | |
| | private | public | private | public | private | public | private | public | private | public |
| FL [1] | 86.6(4.0) | 81.4(1.9) | 83.7(3.9) | 80.3(4.2) | 85.1(4.2) | 79.4(2.6) | 80.7(2.5) | 73.1(2.6) | 81.3(3.6) | 70.8(4.2) |
| MOON [26] | 90.2(3.4) | 82.3(5.8) | 90.3(**1.3**) | 81.2(2.2) | 90.2(2.4) | **80.3(1.2)** | 84.1(2.8) | 76.2(3.5) | 82.4(3.5) | 71.0(3.4) |
| MoE [10] | 89.5(**1.2**) | 82.5(3.6) | 89.8(1.9) | 80.4(3.8) | 90.3(1.4) | 80.1(1.7) | 82.3(**1.7**) | 76.9(2.0) | 83.3(1.6) | 70.9(5.8) |
| SplitNN [11] | 85.6(3.2) | 78.7(**1.2**) | 88.0(2.7) | 77.8(4.7) | 87.3(5.8) | 77.7(4.9) | 81.6(3.4) | 75.2(4.9) | 81.8(2.4) | 68.7(4.9) |
| CusFL (ours) | **90.6**(5.0) | **83.7**(2.1) | **90.8**(4.3) | **81.8(2.0)** | **91.0(1.0)** | 79.5(1.3) | **85.4**(2.1) | **77.2(0.8)** | **84.0(0.1)** | **71.5(1.0)** |

The standard deviations are reported in brackets and the best results are shown in bold.

number of clients increases, MOON gradually outperforms FL, leading to faster convergence. It is because more clients would incur greater inter-client variations, and thus would degrade FL's convergence performance. As shown in Fig. 5, the training loss curves of all learning frameworks on skin lesion classification are smoother than those curves on CS PCa classification, reflecting the fact of this dataset's higher data quantity and quality. For this case, the inter-client variation problem is the dominating issue for FL. As a result, FL performs the worst. One interesting observation is that MOON performs worse than most approaches, highlighting the limitations of single-model-based federated approaches.

The above analysis shows that CusFL achieves the best convergence compared to other frameworks. It validates experimentally the effectiveness of feature alignment in CusFL on extracting useful features from the federation and alleviating inter-client variations.

### *F. Can We Combine PFL With Domain Adaptation?*

We further assess the benefits of CusFL coupled with domain adaptation techniques on CS PCa classification. Here, the variation-aware (VA) module in [22] was introduced for variation reduction, as it can be used as a plug-n-play module without changing network architectures. Quantitative results of different learning frameworks with VA are summarized in Tables XII and XIII respectively. Compared to the results in Tables IV and V, alleviating inter-client variations effectively improves the performances of all learning frameworks, resulting in 0.8%-7% increase in Avg. ACC and 0.3%–8.7% increase in Avg. AUC. The improvement is more apparent under the 32-client setting, which exhibits the most severe inter-client variation problem. The more the clients, the greater the inter-client variations and the more performance gains achieved by introducing VA.

Compared to the experimental results without VA, similar performance trends among different learning frameworks are observed. MOON outperforms FL as it attempts to further minimize the inter-client variations. MoE achieves a comparable performance with MOON as it implicitly handles inter-client variations by utilizing both the federated model and each client's local model. For private testing, SplitNN outperforms FL in terms of both Avg. ACC and Avg. AUC under 4-, 8-, 16-, and 32-client settings, showing the value of variation reduction to ensure the efficiency of parameter-level alignment. However,

TABLE XIII
COMPARISON RESULTS (*I.E.* AVG. AUC) OF LEARNING FRAMEWORKS COUPLED WITH THE VARIATION-AWARE MODULE TO REDUCE INTER-CLIENT VARIATIONS [22] FOR CS PCA CLASSIFICATION UNDER DIFFERENT SETTINGS: "PRIVATE" REPRESENTS PRIVATE TESTING (*I.E.*, TESTING ON EACH CLIENT'S DATA) AND "PUBLIC" IS PUBLIC TESTING (*I.E.*, TESTING ON THE SAME COLLECTION OF TESTING DATA FROM ALL CLIENTS)

| Frameworks (VA) | Avg. AUC(%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 2 Clients | | 4 Clients | | 8 Clients | | 16 Clients | | 32 Clients | |
| | private | public | private | public | private | public | private | public | private | public |
| FL [1] | 92.9(4.3) | 88.7(0.9) | 90.9(2.6) | 88.1(2.4) | 90.9(2.0) | 87.3(2.1) | 86.3(6.7) | 83.1(2.6) | 84.3(8.6) | 78.9(5.6) |
| MOON [26] | 95.3(1.2) | 90.1(5.2) | 94.7(1.7) | 89.2(2.6) | 94.6(1.2) | **88.1**(2.0) | 91.4(2.6) | 83.2(3.4) | 86.7(1.8) | 79.7(2.1) |
| MoE [10] | 94.5(1.0) | 88.9(2.5) | 94.0(**1.0**) | 89.3(1.7) | 95.0(1.0) | 88.2(6.7) | 91.6(**0.8**) | 83.8(5.9) | **89.4**(2.4) | 79.0(5.8) |
| SplitNN [11] | 92.2(3.4) | 86.9(9.2) | 94.5(1.6) | 87.2(3.6) | 93.2(4.5) | 85.6(3.0) | 91.4(2.4) | 84.2(2.7) | 86.4(2.3) | 75.0(1.2) |
| CusFL (ours) | **96.0(0.8)** | **90.9**(2.4) | **96.7**(1.1) | **90.1**(2.5) | **95.6(0.7)** | 87.5(**0.8**) | **92.5**(1.2) | **84.4(1.0)** | 88.0(**0.5**) | **80.6(1.1)** |

The standard deviations are reported in brackets and the best results are shown in bold.

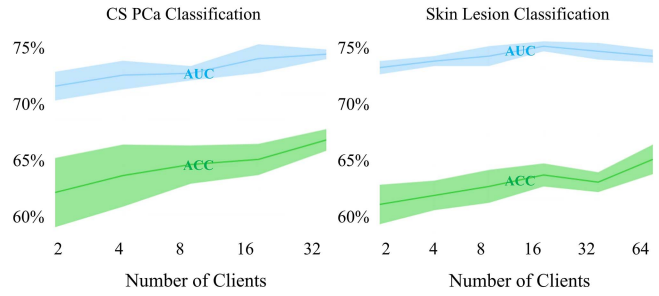


Fig. 6. The range and average ACC (%) and AUC (%) of the models learned by CusFL through 5-fold validation with the varying number of participating clients. The amount of data of each client is kept the same for this set of experiments.

for public testing, FL has a better performance than SplitNN, highlighting the limitation of parameter-level alignment in generalization.

CusFL achieves the best overall performance compared to other learning frameworks. Though MOON's can perform better than CusFL on certain clients for public testing, the performance gaps are quite limited. It should be noticed that MOON is designed to optimize a global federated model for all clients (*i.e.* primarily targeting for public testing) while CusFL is to optimize a personalized model for each client (*i.e.* primarily targeting for private testing). The above experimental results, with and without VA, validate that leveraging public knowledge through feature-level alignment with the federated model for personalization is beneficial.

### G. Can We Further Adapt CusFL Locally?

We utilized local adaptation techniques [31] to observe if the personalized models trained through FedAvg [1] and CusFL can be further adapted to local clients' data distributions. Therefore, additional experiments were conducted to evaluate local adaptation based on each client's private testing set through 5-fold validation using the largest number of clients, *i.e.* 32 clients for CS PCa classification and 64 clients for skin lesion classification.

As stated in Table XIV, local adaptation through fine-tuning (FT) [31] consistently improves the Avg. ACC (%) and Avg. AUC (%) performance of both FedAvg and CusFL. However, each client has to undergo another 100 training rounds on top of the 200 training rounds conducted through federated training.

TABLE XIV
PERFORMANCE COMPARISON (*I.E.* AVG. ACC (%) AND AVG. AUC (%)) OF FEDAVG AND CUSFL FRAMEWORKS WITH AND WITHOUT LOCAL ADAPTATION, *E.G.*, FINE-TUNING (FT) [31]

| Methods | Classification Task | | | |
|---|---|---|---|---|
| | CS PCa(32 clients) | | Skin Lesion(64 clients) | |
| | ACC (%) | AUC (%) | ACC (%) | AUC (%) |
| FedAvg[1] | 72.8(6.9) | 81.4(3.5) | 55.5(2.8) | 65.9(3.2) |
| FedAvg[1]+FT[31] | 73.5(7.0) | 82.5(4.4) | 56.1(3.0) | 66.3(3.4) |
| CusFL | 73.8(**1.5**) | 82.9(**0.2**) | 68.6(**0.6**) | 76.0(**0.2**) |
| CusFL+FT[31] | **74.2**(1.7) | **83.2**(0.4) | **69.3**(1.2) | **76.4**(1.3) |

The standard deviations are reported in brackets and the best results are shown in bold.

More importantly, both Avg. ACC (%) and Avg. AUC (%) of the adapted federated model through FedAvg are lower than those achieved by vanilla CusFL without local adaptation. According to our experiments, the main challenge of local adaptation is to ensure that the resulting federated model(s) can quickly adapt to each client's local data distribution without overfitting. With proper design, introducing FT as a post-processing step can be beneficial for CusFL as stated in Table XIV.

### H. Does Increasing Amount of Training Data Help CusFL?

In Sections IV-F and IV-G, we conducted various experiments by fixing the total amount of effective training data. Specifically, the total amount of training data used in the 2-client and 32-client settings are kept the same. To better validate the effectiveness of CusFL, we designed another set of experiments by progressively increasing the amount of training data. The dataset is first split into 32 clients and 64 clients for CS PCa and malignant skin lesion classification respectively. Then, we sampled subsets of the clients and used them for training. In this way, the total amount of training data gradually increases with the increasing number of participating clients. To better validate the generalization capability, we conducted 5-fold validation on the public testing set.

As shown in Fig. 6, as the amount of federated training data increases due to the increasing number of participating clients (since each client has a fixed amount of training data), the average performance in terms of both ACC (%) and AUC (%) steadily increases. For instance, the average accuracy of CusFL learned with 32 clients is higher than that with 2 clients (*i.e.* 66>62%) for CS PCa classification. Similarly, for skin lesion

classification, the average accuracy of CusFL learned with 64 clients is higher than that with 2 clients (*i.e.* 60>58%). Through validation, CusFL is proven to be more useful when having access to a larger set of training data. It further indicates that collaboration with other clients is beneficial especially when each client only has a limited amount of training data.

## VI. Conclusion

Personalization techniques are promising to handle inter-client variations compared to a single model approach due to its less restrictive nature. CusFL is a personalization technique that enables each client to leverage collective knowledge from its peers via federated learning to learn a fully customized client-optimized model. It introduces a new paradigm that utilizes the federated model only as a guide for feature alignment. In this way, each client can extract useful features from the federated model while avoiding negative influence from irrelevant model parameters. Extensive experiments on case studies of CS PCa classification and malignant skin lesion classification demonstrate the superiority of CusFL compared to other personalization frameworks and standard FL. Using CusFL, each client can train a personalized model with positive performance gains compared to its locally-learned model, which is important to incentivize clients into joining the federation. As the CusFL framework is not tied to any specific network architecture, we believe that it can be applied broadly for other applications. We also show that the proposed CusFL can be combined with domain adaptation techniques for handling inter-client variations in the federated setting, leading to better overall performance.

## 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.*, 2017, pp. 1273–1282.

[2] X. Li et al., "Multi-site fMRI analysis using privacy-preserving federated learning and domain adaptation: ABIDE results," *Med. Image Anal.*, vol. 65, 2020, Art. no. 101765.

[3] Q. Liu, C. Chen, J. Qin, Q. Dou, and P. A. Heng, "FedDG: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1013–1023.

[4] D. Wang, A. Haytham, J. Pottenburgh, O. Saeedi, and Y. Tao, "Hard attention net for automatic retinal vessel segmentation," *IEEE J. Biomed. Health Inform.*, vol. 24, no. 12, pp. 3384–3396, Dec. 2020.

[5] Z. Yan, X. Yang, and K.-T. Cheng, "A three-stage deep learning model for accurate retinal vessel segmentation," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 4, pp. 1427–1436, Jul. 2019.

[6] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," 2020, *arXiv:2002.10619*.

[7] 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.

[8] D. Li and J. Wang, "FedMD: Heterogeneous federated learning via model distillation," 2019, *arXiv:1910.03581*.

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

[10] D. Peterson, P. Kanani, and V. Marathe, "Private federated learning with domain adaptation," 2018, *arXiv:1912.06733*.

[11] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," 2019, *arXiv:1812.00564*.

[12] Y. Deng, M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," 2020, *arXiv:2003.13461v3*.

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

[14] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3320–3382.

[15] G. Litjens, O. Debats, J. Barentsz, N. Karssemeijer, and H. Huisman, "Computer-aided detection of prostate cancer in MRI," *IEEE Trans. Med. Imag.*, vol. 33, no. 5, pp. 1083–1092, May 2014.

[16] L. Geert, D. Oscar, B. Jelle, K. Nico, and H. Henkjan, "Prostatex challenge data. the cancer imaging archive", Feb. 2017. [Online]. Available: https://doi.org/10.7937/K9TCIA.2017.MURS5CL

[17] K. Clark et al., "The cancer imaging archive (TCIA): Maintaining and operating a public information repository," *J. Digit. Imag.*, vol. 26, pp. 1045–1057, 2013.

[18] P. Tschandl et al., "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Sci. Data*, vol. 5, 2018, Art. no. 180161.

[19] Y. Zhao et al., "Federated learning with non-iid data," 2018, *arXiv:1806.00582*.

[20] Z. Allen-Zhu and Y. Li, "Towards understanding ensemble, knowledge distillation and self-distillation in deep learning," 2020, *arXiv:2012.09816*.

[21] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez, "Personalized federated learning with first order model optimization," in *Proc. Int. Conf. Learn. Representations*, 2021.

[22] Z. Yan, J. Wicaksana, Z. Wang, X. Yang, and K. T. Cheng, "Variation-aware federated learning with multi-source decentralized medical data," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 7, pp. 2615–2628, Jul. 2021.

[23] C. Ju et al., "Federated transfer learning for EEG signal classification," in *Proc. 42nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2020, pp. 3040–3045.

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

[25] S. P. Karimireddy et al., "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.

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

[27] T. Li et al., "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, 2020, pp. 429–450.

[28] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with moreau envelopes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 21394–21405.

[29] F. Hanzely, S. Hanzely, S. Horvath, and P. Richtarik, "Lower bounds and optimal algorithms for personalized federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 2304–2315.

[30] F. Hanzely and P. Richtarik, "Federated learning of a mixture of global and local models," 2020, *arXiv:2002.05516*.

[31] T. Yu, E. Bagdasartyan, and V. Shmatikov, "Salvaging federated learning by local adaptation," 2020, *arXiv:2002.04758*.

[32] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," in *Proc. 31st Int. Neural Inf. Process. Syst.*, 2017, pp. 506–516.

[33] S-A. Rebuffi, H. Bilen, and A. Vedaldi, "Efficient parametrization of multi-domain deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8119–8127.

[34] N. Houlsby et al., "Parameter-efficient transfer learning for NLP," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2790–2799.