Decoupling Shared and Personalized Knowledge: A Dual-Branch Federated Learning Framework for Multi-Domain with Non-IID Data

Yiran Pang, Zhen Ni, Xiangnan Zhong*

Florida Atlantic University, Boca Raton, FL, USA
{ypang2022, zhenni, xzhong}@fau.edu

Abstract

Federated learning (FL) enables collaborative model training without centralizing data. In multi-domain scenarios with non-identically and independently distributed (non-IID) data, prediction performance is often hindered by catastrophic forgetting of specialized local knowledge and negative transfer from conflicting client updates. To address these challenges, we propose a personalized FL framework with dual-branch (pFedDB) structure and a two-phase training protocol. The dual-branch architecture separates the model into a shared branch for cross-client aggregation and a private branch that remains on each local client. The private branch is never overwritten by server updates, which prevents the catastrophic forgetting of domain-specific knowledge. This structure also significantly reduces communication overhead per round as only the shared branch is transmitted. To mitigate negative transfer, our two-phase protocol first establishes a personalized knowledge anchor by training a single-branch expert model on each client's local data. In the second phase, the locally trained model is cloned to initialize private and shared branches. Only the shared branch is aggregated in federated training. This process enables the shared branch to learn a general representation that complements the established local expertise. This design consistently improves the performance of every client over its single-domain baseline, overcoming the challenge of negative transfer among clients. Experiments on our new Chest-X-Ray-4 suite and three public benchmarks show that the proposed pFedDB method obtains 30% saving in communication overhead per round and competitive or better accuracy performance than recent FL methods.

Project Page — https://github.com/yiranpang/pFedDB

Introduction

Federated learning (FL) (McMahan et al. 2017) lets many clients build one model while keeping data local. This setup is attractive when privacy laws, limited storage, or high network costs block raw data exchange. However, FL must cope with data heterogeneity. Clients collect samples in different regions, devices, and under different services, so their data distribution drift apart. For example, cross-hospital healthcare (Ullah et al. 2024) and vehicle-to-vehicle systems (Do et al. 2024) expose large distribution gaps. They may use distinct sensors and run in varied contexts. Consequently, these non-identically and independently distributed (non-IID) gaps weaken the convergence guarantees and the observed accuracy of existing FL methods (Li et al. 2019; Pang, Ni, and Zhong 2024, 2025a,b; Li and Wai 2025).

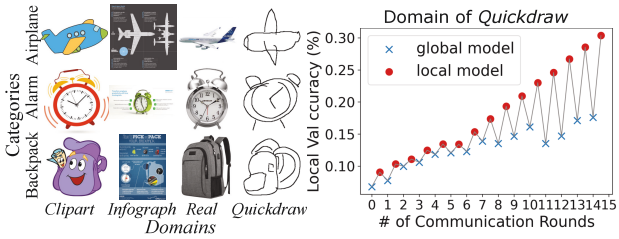


Figure 1: Left: Sample images from three classes across four domains in the DomainNet benchmark, illustrating significant domain shift. Right: FedAvg training dynamics on the *Quickdraw* domain. The periodic replacement of the local model with the global weights causes the global model's validation accuracy to fluctuate and lag behind the performance of the specialized local model.

The non-IID challenges are mostly manifested as domain shift in multi-domain FL. To make the issue concrete, we inspect the DomainNet benchmark (Peng et al. 2019a). This dataset keeps one common label set yet draws images from four separate collection platforms. Figure 1 (left) shows typical samples from the four domains—*Clipart*, *Infograph*, *Quickdraw*, and *Real*. Resolution, background, and object style vary widely. In multi-domain FL, each client owns images from only one domain, so their data distributions barely overlap. Many strategies try to patch the non-IID problem. FedProx (Li et al. 2020) adds an ℓ_2 term to the local loss so that updates stay close to the global model. MOON (Li, He, and Song 2021) adopts a contrastive loss that aligns local and global representations. FedCurv (Shoham et al. 2019), drawing from continual-learning ideas, penalizes changes to parameters that the global model deems important. Yet, these methods aim for a single global optimum. When each client faces a different domain shift, forcing a consensus can lead to negative transfer, where aggregating knowledge from disparate domains degrades model performance. As a result, one model cannot fit every detail (Zhuang and Lyu 2024).

*Corresponding author.

Additionally, weight replacement in FedAvg leads to catastrophic forgetting. Figure 1 right plots a FedAvg run on the Quickdraw client. At the start of every communication round, the server distributes the newest global weights. That download in the client overwrites the tuned local model, causing catastrophic forgetting of domain-specific cues. It leads to accuracy drops, the client must relearn, and then the cycle repeats. The result is a "saw-tooth" curve where the global model lags behind the local one (Karimireddy et al. 2020). As domain gaps grow, the drift between local and global optima widens, and convergence slows even more (Li et al. 2019; Khaled, Mishchenko, and Richtárik 2020; Li and Wai 2025).

Motivated by these observations, we propose a new personalized federated learning framework, named personalized federated dual-branch (pFedDB), to tackle both negative transfer and catastrophic forgetting by decoupling knowledge. We summaries the principal contributions of this work as follows:

- We propose the personalized federated dual-branch (pFedDB) framework, which features a dual-branch architecture to decouple shared and personalized knowledge. By aggregating only the shared branch, pFedDB protects each client's private knowledge from being overwritten, effectively mitigating catastrophic forgetting.
- We introduce a two-phase training protocol where clients first train a local expert model as a knowledge anchor before initiating federated collaboration. This enables that shared knowledge complements local expertise, allowing every client to surpass its single-domain baseline and overcoming the challenge of negative transfer.
- Our pFedDB framework reduces per-round communication overhead by about 30%, and it enhances user privacy by keeping the semantically-rich top layers local. Experiments on our new Chest-X-Ray-4 dataset and three public benchmarks show that pFedDB improves performance for every client, achieving an overall accuracy that is competitive with or superior to recent federated learning methods.

Related Work

Multi-Domain Federated Learning

When clients come from distinct domains, their feature distributions often differ, a problem known as domain shift caused by non-IID data in multi-domain (Bernecker et al. 2022). Many approaches attempt to mitigate this by enforcing alignment at the representation level. For instance, FADA (Peng et al. 2019b) uses adversarial training to learn domain-invariant representations, while AlignFed (Zhu et al. 2024b) pulls features of the same class closer using global class prototypes. These methods seek a shared representation space, which can suppress unique domain features essential for personalization. Another line of work focuses on normalization statistics or knowledge distillation. FedBN (Li et al. 2021b) tackles domain shift by keeping batch normalization (BN) layers local, preventing the aggregation of conflicting statistics. Distillation-based methods like FedMD (Li and Fedmd 2019) and FedDF (Shi et al. 2021) rely on a public dataset or ensemble logits to transfer knowledge, aiming to preserve global insights on local models. While effective, these methods address knowledge preservation indirectly—either at the statistical level or through a secondary knowledge source. They lack a structural mechanism to protect specialized parameters from being overwritten during aggregation.

Personalized Federated Learning

Personalized Federated Learning (PFL) aims to balance global collaboration with local adaptation. A popular strategy is structural partitioning, as seen in FedPer (Arivazhagan et al. 2019) and LG-FedAvg (Liang et al. 2020), which separate a model into a shared base and a private head. These methods prevent the classifier from being overwritten but still subject the entire feature extractor to conflicting gradients from diverse domains. Other methods like Ditto (Li et al. 2021a) and pFedMe (Hanzely and Richtárik 2020) enforce a soft coupling by regularizing local models to stay close to the global one. More recent multi-domain PFL methods explore decoupling. PartialFed (Sun et al. 2021) allows clients to selectively update a subset of global parameters, but the mechanism for choosing which layers to freeze is adaptive rather than structurally guarantee. The DualFed (Zhu et al. 2024a) framework separates generalized and personalized representations by inserting an additional projection network to sequentially transform features. Similarly, the pFedDR (Liu et al. 2025) framework decouples in-domain and global representations, but relies on a KL divergence objective to disentangle their influence, which complicates the optimization.

Proposed Method

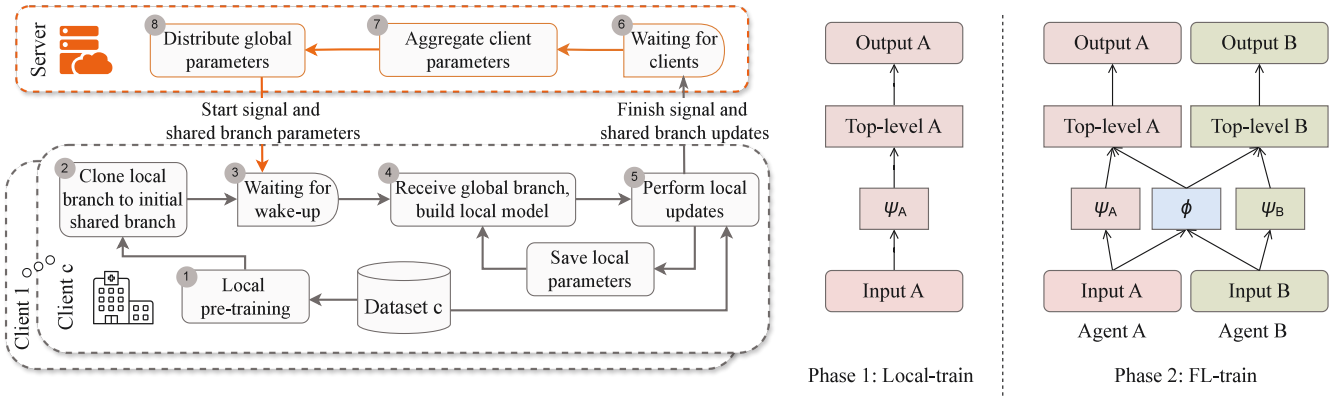
Let $\mathcal{S} = \{1, \ldots, N\}$ be the set of clients, indexed by $c \in \mathcal{S}$. Each client c owns a local dataset $\mathcal{D}_c = \{(\mathbf{x}_{c,i}, y_{c,i})\}_{i=1}^{n_c}$, where $\mathbf{x}_{c,i}$ represents the i-th data example for client c, $y_{c,i}$ is its associated label, and $n_c = |\mathcal{D}_c|$ denotes the number of samples for client c. Each dataset $\mathcal{D}_c$ is drawn from a client-specific data distribution $P_c(\mathbf{x}, y)$.

Let $\boldsymbol{\phi}$ denote the parameters of the shared network branch, which are common across all clients. For each client $c \in \mathcal{S}$, let $\boldsymbol{\psi}_c$ denote the parameters of its private domain-specific feature branch, and $\boldsymbol{\omega}_c$ denote the parameters of its private top-level layers. The personalized model for client c is denoted by M_c. This model is parameterized by the collective set of parameters $(\boldsymbol{\phi}, \boldsymbol{\psi}_c, \boldsymbol{\omega}_c)$, and its prediction for an input sample $\mathbf{x}$ is given by $M_c(\mathbf{x})$.

The primary objective is to learn the optimal shared parameters $\boldsymbol{\phi}$ along with the client-specific parameter sets $\{\boldsymbol{\psi}_c\}_{c\in\mathcal{S}}$ and $\{\boldsymbol{\omega}_c\}_{c\in\mathcal{S}}$. This is achieved by minimizing the sum of expected losses across all clients, as formulated below:

$$\min_{\boldsymbol{\phi}, \{\boldsymbol{\psi}_c, \boldsymbol{\omega}_c\}} \sum_{c\in\mathcal{S}} \mathbb{E}_{(\mathbf{x},y)\sim P_c} \left[\ell\left(M_c(\mathbf{x}), y\right)\right], \tag{1}$$

where $\ell(\cdot, \cdot)$ is the chosen loss function. The model output $M_c(\mathbf{x})$ implicitly depends on the parameters $\boldsymbol{\phi}$, $\boldsymbol{\psi}_c$, and $\boldsymbol{\omega}_c$.



(a) Two-phase federated learning workflow.

(b) Dual-branch architecture for knowledge decoupling.

Figure 2: An overview of the proposed pFedDB framework. (a) The client-server workflow is divided into two phases. In Phase 1 (Local Pre-training, Step 1), a client trains a model on its private data. In Phase 2 (Federated Collaboration, Steps 2-8), this model is cloned to initialize a private and a shared branch. The client then enters the collaborative loop, receiving the global shared branch, performing local updates on all its parameters, and uploading only the updated shared branch to the server for aggregation. (b) The corresponding model architecture transitions from a standard single-branch network (ψ_A) in Phase 1 to a dual-branch structure in Phase 2. This structure combines a private branch (ψ) for each client, which is never shared and preserves local expertise, with a shared branch (ϕ), which learns global patterns through federated aggregation.

Figure 2a shows the pFedDB workflow. Each client first learns a single-branch model on its own data. The client then copies the pre-trained local branch, creating a private branch $\boldsymbol{\psi}_c$ and a shared branch $\boldsymbol{\phi}$. In every round the client updates $\boldsymbol{\phi}$, $\boldsymbol{\psi}_c$, and the head $\boldsymbol{\omega}_c$ with local optimizer. It uploads only $\boldsymbol{\phi}$ to the server. The server averages the received $\boldsymbol{\phi}$ weights values and broadcasts the aggregated result. Private weights $\boldsymbol{\psi}_c$ and $\boldsymbol{\omega}_c$ never leave the device, so domain knowledge stays intact.

Personalized Federated Dual-Branch Structure

Learning from non-IID clients requires a model that both captures domain-invariant patterns and retains domain-specific nuances. A single-encoder design tends to collapse these objectives into an over-smoothed representation, erasing client-level distinctions. In this section, we introduce a dual-branch structure as shown in Figure 2b. The structure include a shared branch that extracts domain-agnostic features and participates in cross-client aggregation and a private branch that learns domain-specific cues and remains entirely local. This structural allows the model to generalize across clients while safeguarding each client's unique knowledge.

Architecturally, our method implements this separation by partitioning the model backbone. For any single-branch neural network that consists of L ordered layers. We introduce a single hyper-parameter, the cut depth l_c $(1 \leq l_c < L)$ to define the split point. The lower layers, from $1{:}l_c$, are cloned to form two parallel feature extractors: a shared extractor $f_s(\cdot\,;\boldsymbol{\phi})$ and a private extractor $f_p(\cdot\,;\boldsymbol{\psi}_c)$. The remaining top-level layers $l_c{+}1{:}L$ together with the classifier, constitute a local head $g_p(\cdot\,;\boldsymbol{\omega}_c)$.

For any input $\mathbf{x}$ on client c, the forward pass begins with the two extractors running in parallel to generate feature vectors:

$$\mathbf{z}_s = f_s(\mathbf{x};\boldsymbol{\phi}), \quad \mathbf{z}_p^c = f_p(\mathbf{x};\boldsymbol{\psi}_c). \tag{2}$$

These feature vectors are then fused through a computationally light element-wise addition:

$$\mathbf{z}^c = \mathbf{z}_s + \mathbf{z}_p^c. \tag{3}$$

Finally, the fused representation $\mathbf{z}^c$ is passed through the client's local head to produce the final prediction:

$$\hat{\mathbf{y}} = g_p(\mathbf{z}^c;\boldsymbol{\omega}_c). \tag{4}$$

In the federated phase, only the shared parameters $\boldsymbol{\phi}$ are aggregated, while the client-specific blocks $\boldsymbol{\psi}_c$ and $\boldsymbol{\omega}_c$ remain strictly local. This partition yields three appealing benefits. (i) Because the private branch is never uploaded, domain-specific knowledge is fully preserved and cannot be overwritten by global updates, preventing catastrophic forgetting. (ii) Only the shared parameters $\boldsymbol{\phi}$ are exchanged each round, which efficiently lowers the communication load compared with transmitting the entire model. (iii) All high-level semantic features and classifier weights remain on the user's device, providing an extra privacy protection.

Two-Phase Training for Knowledge Decoupling

We propose a two-phase training protocol for our dual-branch architecture to enable knowledge decoupling and to mitigate negative transfer. The protocol first establishes a personalized foundation for each client and then uses federated collaboration to learn shared knowledge. Our two-phase strategy proceeds as follows:

Phase 1 – Local Pre-training. Before federated collaboration begins, each client c first trains a standard, single-branch model exclusively on its own dataset D_c. The objective is to

establish a robust foundation of domain-specific knowledge by optimizing its private feature extractor f_p and head g_p:

$$\min_{\boldsymbol{\psi}_c, \boldsymbol{\omega}_c} \mathcal{L}_c^{(0)} = \frac{1}{n_c} \sum_{i=1}^{n_c} \ell\big(g_p(f_p(\mathbf{x}_{c,i}; \boldsymbol{\psi}_c); \boldsymbol{\omega}_c), \mathbf{y}_{c,i}\big). \quad (5)$$

After training for E_0 local epochs, the client possesses a pair of weights, $(\boldsymbol{\psi}_c^\star, \boldsymbol{\omega}_c^\star)$, that are highly specialized to its local data distribution.

Phase 2 – Federated Collaboration with Knowledge Preservation. This phase begins with a crucial initialization step. Each client c creates its dual-branch model by cloning its pre-trained private extractor $\boldsymbol{\psi}_c^\star$ to serve as the initial weights for the shared branch. To form the initial global model, the server aggregates these weights from all clients, typically via averaging:

$$\boldsymbol{\phi}^{(0)} = \sum_{c \in \mathcal{S}} \frac{n_c}{\sum_{j \in \mathcal{S}} n_j} \boldsymbol{\psi}_c^\star. \quad (6)$$

With the dual-branch model $\{\boldsymbol{\phi}^{(0)}, \boldsymbol{\psi}_c^*, \boldsymbol{\omega}_c^*\}$ now established on each client, the iterative federated training proceeds for T rounds. In each round $t = 0, 1, \ldots, T-1$:

Firstly, the server broadcasts the current global shared parameters $\boldsymbol{\phi}^{(t)}$ to a subset of clients $\mathcal{S}_t$.

Then, each client $c \in \mathcal{S}_t$ receives $\phi^{(t)}$ and performs E local epochs of training. It updates all three parameter blocks $\{\boldsymbol{\phi}, \boldsymbol{\psi}_c, \boldsymbol{\omega}_c\}$ by minimizing the loss on its local data:

$$\mathcal{L}_c^{(t)} = \frac{1}{n_c} \sum_{i=1}^{n_c} \ell\big(g_p(f_s(\mathbf{x}_{c,i}; \boldsymbol{\phi}) + f_p(\mathbf{x}_{c,i}; \boldsymbol{\psi}_c); \boldsymbol{\omega}_c), \mathbf{y}_{c,i}\big). \quad (7)$$

After local training, each client c uploads only its updated shared branch parameters $\boldsymbol{\phi}_c^{(t+1)}$ to the server.

Finally, the server aggregates the received parameters to form the global model for the next round:

$$\boldsymbol{\phi}^{(t+1)} = \sum_{c \in \mathcal{S}_t} \frac{n_c}{\sum_{j \in \mathcal{S}_t} n_j} \boldsymbol{\phi}_c^{(t+1)}. \quad (8)$$

This two-phase process provides a principled approach to knowledge decoupling. By first establishing a local model in Phase 1, which can leverage a client's pre-existing legacy model in practice, each client creates a powerful personalized knowledge anchor. During the subsequent federated collaboration, this anchor guides the optimization process through an implicit gradient gating mechanism. Because only the shared branch is synchronized and the two branches fuse additively (Eq. 3), the local gradient used to update the shared parameters $\boldsymbol{\phi}$ on client c is scaled by the upstream error signal:

$$\nabla_\phi \mathcal{L}_c = \mathbb{E}_{(\mathbf{x},\mathbf{y}) \in D_c} \left[\nabla_{\mathbf{z}} \ell\big(g_p(\mathbf{z}^c, \boldsymbol{\omega}_c); \mathbf{y}\big) \frac{\partial f_s(\mathbf{x}; \boldsymbol{\phi})}{\partial \boldsymbol{\phi}} \right]. \quad (9)$$

The upstream error term $\nabla_{\mathbf{z}} \ell(\cdot)$ serves as a coefficient: if the private branch already explains a local feature well, the loss is small, muting the corresponding update to $\boldsymbol{\phi}$. Conversely, unexplained patterns produce larger errors and stronger updates. By prompting the shared branch learns a generalized representation that complements the client's established expertise, this two phase training protocol mitigates negative transfer.

Algorithm 1: pFedDB Training Process

Input: Number of clients N, communication rounds T, local epochs E_0 and E, client datasets $\{\mathcal{D}_c\}_{c=1}^N$.

 # Phase 1: Local pre-training on each client
for each client $c \in \{1, \ldots, N\}$ **in parallel do**
  Initialize local model parameters $(\boldsymbol{\psi}_c, \boldsymbol{\omega}_c)$ randomly.
  for $e = 1, \ldots, E_0$ **do**
    Update $(\boldsymbol{\psi}_c, \boldsymbol{\omega}_c)$ by minimizing Eq. (5) on its local dataset $\mathcal{D}_c$.
  end for
  Store final pre-trained parameters $(\boldsymbol{\psi}_c^\star, \boldsymbol{\omega}_c^\star)$.
end for

 # Phase 2: Federated Collaboration
Server Executes:
Receive all pre-trained extractors $\{\boldsymbol{\psi}_c^\star\}_{c=1}^N$ from clients.
Initialize global model by Eq. (6).
for round $t = 0, \ldots, T-1$ **do**
  Select a subset of clients $\mathcal{S}_t \subseteq \{1, \ldots, N\}$.
  Broadcast the global model $\boldsymbol{\phi}^{(t)}$ to all clients in $\mathcal{S}_t$.
  for each client $c \in \mathcal{S}_t$ **in parallel do**
    $\boldsymbol{\phi}_c^{(t+1)} \leftarrow \text{ClientUpdate}(c, \boldsymbol{\phi}^{(t)})$.
  end for
  Aggregate updated weights to form the new global model using Eq. (8).
end for

function ClientUpdate(c, $\boldsymbol{\phi}$)
 # Local parameters $(\boldsymbol{\psi}_c, \boldsymbol{\omega}_c)$ are persistent on the client.
for $e = 1, \ldots, E$ **do**
  Update all three parameter blocks $\{\boldsymbol{\phi}, \boldsymbol{\psi}_c, \boldsymbol{\omega}_c\}$ by minimizing Eq. (7) on $\mathcal{D}_c$.
end for
return updated shared parameters $\boldsymbol{\phi}$ to the server.
end function

Experiments

We conducted a comprehensive study of the proposed pFedDB method, including its performance on multiple benchmarks, empirical validations of its knowledge decoupling, and analyses of its communication efficiency and backbone generality.

We assume that each domain is treated as one client under a realistic multi-institution scenario with non-IID multi-domain data. Training setup is the same as the typical setup to mimic the data silo in practice (McMahan et al. 2017; Li et al. 2020; Hsu, Qi, and Brown 2019; Li et al. 2021b). Every client is limited to only 10^2 to 10^3 labeled images. All remaining data are reserved as a test split. For com-

Method	Venue	Chest-X-Ray-4					Digits-Five					
		C	O	R	V	Avg.	Mn	Sv	Up	Sy	MM	Avg.
Single domain†	–	82.3	68.1	83.6	88.9	80.7	94.4	65.3	95.2	80.3	77.8	82.6
FedAvg (McMahan et al. 2017)	PMLR	82.2	67.6	84.1	84.7	79.6	95.9	62.9	95.6	82.3	76.9	82.7
FedPer (Arivazhagan et al. 2019)	Arixv	83.3	66.7	83.3	84.1	79.4	92.9	59.7	95.1	83.1	71.6	80.5
FedProx (Li et al. 2020)	MLSys	79.3	65.5	85.9	85.5	79.0	95.8	63.1	95.6	82.3	76.6	82.7
FedBN (Li et al. 2021b)	ICLR	82.9	67.2	**86.5**	87.6	81.0	96.6	71.0	**97.0**	83.2	78.3	85.2
Ditto (Li et al. 2021a)	ICML	80.5	62.6	84.4	82.4	77.5	95.8	64.4	95.6	81.7	76.0	82.7
pFedSD (Jin et al. 2023)	TPDS	81.1	72.3	**86.5**	84.5	81.1	95.7	64.8	95.4	81.6	80.7	83.6
FedSSD (He et al. 2024)	TBDATA	84.2	72.4	83.2	84.9	81.2	**96.9**	61.9	96.7	82.2	79.1	83.4
FedLGF (Yan and Guo 2024)	ECCV	78.7	65.5	82.7	79.6	76.6	95.9	64.2	95.6	81.6	76.2	82.7
FedWon (Zhuang and Lyu 2024)	ICLR	84.5	72.3	85.2	83.8	81.5	96.5	**71.3**	96.3	85.4	77.7	85.4
pFedDR (Liu et al. 2025)	ESWA	79.4	62.5	84.0	78.9	76.2	96.7	**71.3**	95.8	83.3	80.0	85.4
pFedDB (Ours)	AAAI	**85.4**	**73.3**	86.0	**89.1**	**83.4**	96.7	68.6	**97.0**	**86.4**	**85.2**	**86.8**

Table 1: The table shows the mean classification accuracy (%) averaged over 5 runs. The best result in each column is shown in bold, and the average columns are shaded for emphasis. † "Single domain" uses DenseNet-121 (Huang et al. 2017) on Chest-X-Ray-4 and a Simple-CNN (Li et al. 2021b) baseline on Digits-Five.

Method	Office-Caltech-10					DomainNet						
	A	C	D	W	Avg.	C	I	P	Q	R	S	Avg.
Single‡	54.9	40.2	78.7	86.4	65.1	41.0	23.8	36.2	73.1	48.5	34.0	42.8
FedAvg	54.1	44.8	66.9	85.1	62.7	48.8	24.9	36.0	56.1	46.3	36.6	41.5
FedProx	54.2	44.5	65.0	84.4	62.0	48.9	24.9	36.6	54.4	47.8	36.9	41.6
FedBN	63.0	45.3	83.1	90.5	70.5	51.2	26.8	41.5	71.3	54.8	42.1	48.0
pFedSD	64.1	45.7	90.6	90.4	72.7	49.4	**26.9**	39.9	70.5	53.3	37.4	46.2
FedSSD	59.8	46.0	87.5	87.1	70.1	51.0	26.6	37.2	69.5	50.8	36.8	45.3
FedWon	**65.1**	49.2	**90.6**	83.7	72.2	**54.7**	**26.9**	40.0	68.3	54.0	48.9	48.8
pFedDB	62.5	**50.1**	**90.6**	**90.5**	**73.4**	48.5	25.0	**44.1**	**74.0**	**60.2**	**49.9**	**50.3**

Table 2: The table shows the mean classification accuracy (%) averaged over 5 runs. The best result in each column is shown in bold, and the average columns are shaded for emphasis. ‡ "Single" uses AlexNet (Krizhevsky, Sutskever, and Hinton 2012) on both dataset. For citations of the other methods, please refer to Table 1.

parison model training, we use the cross-entropy loss and SGD optimizer with a learning rate of 10^2. Except for FedWon, we maintained the original implementation, using SGD_AGC (Brock et al. 2021) as optimizer. For all comparison algorithms, the communication round is 300. Our method includes 150 rounds of local training and 150 rounds of communication. Thus, our method has the same local training load and lower communication costs.

Comparison with State-of-the-Art Methods

Settings: We evaluate our method on four diverse multi-domain benchmarks, including our newly proposed Chest-X-Ray-4 with real-world non-IID data. It unifies CheXpert (Irvin et al. 2019), OpenI (Demner-Fushman et al. 2016), RSNA (Shih et al. 2019), and VinBigdata (Nguyen et al. 2022) into a shared binary task, Lung Opacity vs. Normal. Please refer to the project page for details of the dataset. And three standard benchmarks: Digits-Five (Li et al. 2021b), Office-Caltech-10 (Gong et al. 2012), and DomainNet (Peng et al. 2019a). Consistent with prior work, we adopt DenseNet-121 for Chest-X-Ray-4, a Simple-CNN for Digits-Five, and AlexNet for the other two datasets. We compare pFedDB with state-of-the-art methods for non-IID FL, including training on a single domain as baselines. We perform a 5-trial repeating experiment with different random seeds. A detailed standard deviation of the accuracy is shown in the project page.

Result and Analysis: As shown in Figure 3, we randomly select two domain pairs from the Chest-X-Ray-4 dataset and project their average accuracies onto the x- and y-axes, respectively. The intersection of the dashed lines marks the reference performance of single-domain training. Only pFedDB is located in the upper right quadrant, surpassing the baseline for both domain groups. In contrast, comparison methods benefit one domain while degrading another or lag overall. This result shows that pFedDB can enable all clients to benefit from multi-domain federated collaboration without sacrificing local performance. The results in Tables 1 and 2 further confirm this benefit, where our method consistently outperforms the single baseline in every indi-

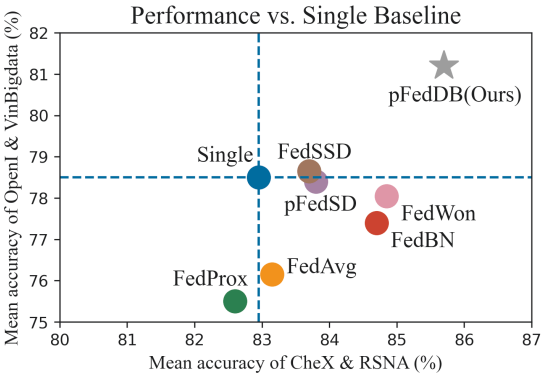


Figure 3: Scatter plot of mean accuracy on CheXpert + RSNA (x-axis) and OpenI + VinBigdata (y-axis) for all methods on Chest-X-Ray-4 dataset. The optimal value is in the top right corner. The dashed line marks the single domain reference. Only our pFedDB occupies the upper-right region, indicating simultaneous gains for every participating client.

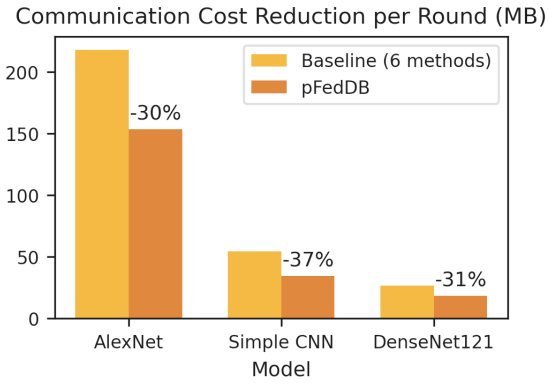


Figure 4: Per-round communication cost on the three backbones. Baseline methods—FedAvg, FedProx, FedBN, pFedSD, FedSSD, and FedWon—differ by $\leq 0.5\%$, so they are aggregated for clarity.

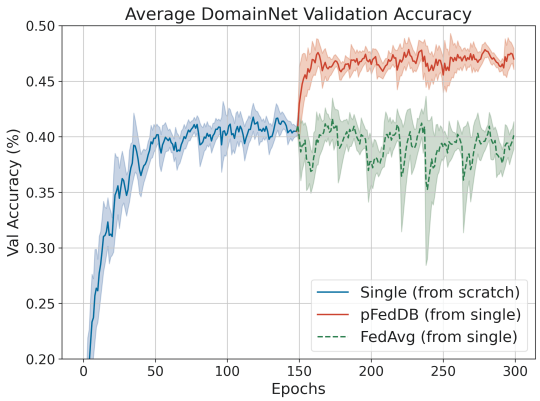


Figure 5: Comparison of average validation accuracy on DomainNet, federated methods start from models pre-trained on local data for 150 epochs.

vidual domain. Compared with state-of-the-art methods, our method attains or approaches optimal performance across most domains and achieves the highest overall average accuracy.

Communication Efficiency

We verify that the dual-branch structure is architecture-agnostic by repeating all experiments on a lightweight 7-layer Simple CNN, the classic AlexNet, and DenseNet-121. Across these very different parameter budgets, pFedDB reproduces the same accuracy gains, while its communication load is reduced because only the shared branch is transmitted. Figure 4 shows that this structure trims 30–37% of the bytes sent per round. The competing algorithms in Table 1 consume nearly identical bandwidth per round. FedProx and pFedSD preserve the standard FL protocol as FedAvg. FedSSD transmits a small auxiliary matrix, FedBN stops sharing BN statistics, and FedWon removes BN layers but adds per-channel adaptive weight-normalization coefficients inside convolutional blocks. However, these added or removed tensors constitute less than 0.5% of the total parameter count for their backbones. Accordingly, we aggregate them into a single baseline bar in Figure 4. Compared with these baselines, the saving of our method is amplified in practice: our two-phase schedule spends the first 150 of 300 rounds purely on local training. It communicates during only the second half, so the total traffic is well below half that of full-model baselines.

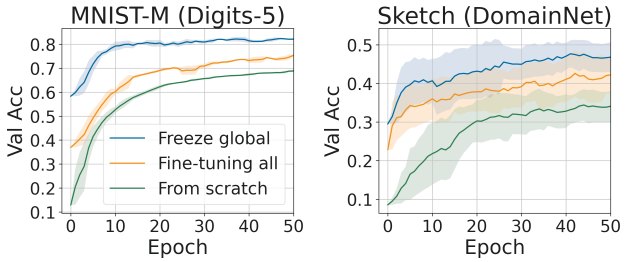
Decoupling Knowledge in Two-Phase Training

Figure 5 presents the typical training dynamics of pFedDB. For comparison, we add the two-phase training to the FedAvg method, giving every method the same 300-round training budget. Each client trains a single-branch network solely on its local data during Phase 1 (rounds 0–150). Phase 2 (rounds 151–300) then enables FL communication. The plot contains three curves: the local-only baseline (Single), FedAvg running under the same two-phase protocol, and our dual-branch pFedDB.

The warm-up procedure is identical, all variants start Phase 2 from the same pre-trained checkpoint. Once the server broadcasts begin, however, their trajectories diverge sharply. The FedAvg oscillates markedly because every download overwrites the client's tuned weights, triggering repeated forgetting and relearning. By contrast, the pFedDB rises smoothly and remains consistently above the local baseline. The private branch of pFedDB preserves domain-specific knowledge from Phase 1, while the shared branch aggregates domain-agnostic features across clients without harming what has already been learned.

Validating Knowledge Decouple through New-Client Onboarding

In real deployments, new institutions may join an established federation at any time. This scenario provides a di-



(a) Adaptation on MNIST-M (b) Adaptation on Sketch

Figure 6: Validation accuracy for a new client joining an existing federation on two domains. We compare three strategies. Freeze global: A new client downloads the shared branch, freezes it, and trains only its private branch with top layers. Finetune all: The client fine-tunes the entire network. From scratch: The client trains its model from random initialization without federated knowledge.

rect test of our central hypothesis: if the shared branch ϕ has captured domain-agnostic features, a previously unseen client can reuse it as a high-quality extractor with minimal adaptation. We pre-train shared branch ϕ on all but one domain in each benchmark and hold the remaining domain out as a late-joining client (domain of *Sketch* for DomainNet dataset and domain of *MNIST-M* for Digits-Five dataset). The newcomer download the current global shared branch ϕ and integrate it with their own private model ψ_c and top-layers, which based on a pre-existing local model. Then the newcomer tries three adaptation strategies: (i) training from scratch; (ii) fine-tuning the entire network; (iii) freezing ϕ and updating only its private branch ψ_c and classifier.

Figure 6 shows that freezing ϕ yields the highest starting accuracy and converges speed. Allowing gradients to flow into ϕ (fine-tuning all) erodes its first round performance and slows learning, while training from scratch lags throughout. The ability to freeze ϕ and achieve superior performance confirms that pFedDB achieves knowledge decoupling, allowing global insights to be seamlessly combined with new client's specialized local adaptation without conflict.

Effect of the Cut Depth l_c

Table 3 summarizes an ablation in which we vary the cut depth l_c, defined as the layer index at which the backbone is split into a shared and a private branch. A consistent pattern emerges: the best mean accuracy on both AlexNet and DenseNet-121 backbones is obtained when roughly 70% of the total parameters are assigned to the shared branch.

The ideal cut depth l_c can be identified as a breakpoint that does not alter the network topology. For DenseNet-121, parameters are grouped by dense blocks. The first three blocks comprise 68% of the weights, so cutting after Block 3 naturally realises the desired 70% split. For AlexNet, whose parameters are heavily concentrated in the fully connected layers, including CNN layers with the first FC layer, achieves a comparable 71% ratio and yields the highest average accuracy on Office-Caltech-10. We recommend setting the cut

Office-Caltech-10				**Chest-X-Ray-4**			
l_c	Param	Ratio	Acc	l_c	Param	Ratio	Acc
5	9.4	0.04	66.8	14	1.4	0.05	80.9
6	153.5	<u>0.71</u>	**73.4**	39	5.4	0.20	81.2
7	217.5	0.99	72.1	88	18.2	<u>0.68</u>	**83.4**
8	217.7	1.00	70.3	120	26.5	0.99	82.7

Table 3: Cut-depth l_c study on two datasets. *Param* is the number of parameters (in MB) uploaded per round. *Ratio* is the fraction of the full network that is communicated in FL. *Acc* is the mean test accuracy. Office-Caltech-10 uses an AlexNet backbone, whereas Chest-X-Ray-4 uses DenseNet-121.

depth near the 70% parameter for better accuarcy. When stricter bandwidth limits apply, the dual-branch design allows shallower cuts to trade a small amount of accuracy for additional communication savings.

Ablation Study on Two-phase Training

Office-Caltech-10					
Two-phase	A	C	D	W	Avg.
✗	57.8	46.2	90.6	89.8	71.1
✓	**62.5**	**50.1**	**90.6**	**90.5**	**73.4**
Chest-X-Ray-4					
Two-phase	C	O	R	V	Avg.
✗	83.5	73.3	85.4	83.0	81.3
✓	**85.4**	**73.3**	**86.0**	**89.1**	**83.4**

Table 4: Ablation on the two-phase training protocol. Rows without two-phase represent models directly trained from beginning under the federated setting.

We conduct an ablation study to analyze the impact of the two-phase training protocol. The two-phase schedule aims to create a local knowledge anchor before clients begin collaboration. Without this anchor (first row in Table 4), domains that are already easy to optimize, ***Dslr*** and ***Webcam*** in Office-Caltech-10, show little change. However, the more challenging domains ***Amazon*** and ***Caltech10*** suffer noticeable degradation due to interference from other domains. A similar pattern appears on Chest-X-Ray-4 dataset. The hardest domain ***VinBigdata*** loses average accuracy when the two-phase training is removed. These observations demonstrate that Phase 1 equips each client with a robust, domain-specific starting point. Phase 2 can learn with others without erasing local expertise, yielding consistent gains across all domains.

Ablation Study on Feature Fusion

We selected element-wise addition (Eq. 3) for its computational simplicity and parameter-free nature. To validate this

design choice, we compared it against several common fusion alternatives: (i) *concat*, which concatenates features and uses a linear layer to adapt dimensionality; (ii) *attention*, which applies a lightweight gating mechanism over the two streams; and (iii) a *weighted sum* $\mathbf{z}^c = \alpha\,\mathbf{z}_s + (1-\alpha)\,\mathbf{z}_p^c$ with fixed coefficients $\alpha \in \{0.3, 0.5, 0.7\}$.

As shown in Table 5, our *add* operation achieves the best average accuracy. The *concat* and *attention* methods not only underperformed but also introduced extra parameters and computational overhead. The *weighted sum* variant proved sensitive to the hyperparameter α and, on average, failed to surpass the performance of our parameter-free addition. This confirms that simple addition is the most effective and efficient fusion strategy for our framework.

Office-Caltech-10					
Fusion method	A	C	D	W	Avg.
attention	56.3	46.0	90.1	89.8	70.5
concat	57.8	46.2	90.6	89.8	71.1
weighted (α=0.3)	59.9	47.3	90.6	90.5	71.6
weighted (α=0.5)	61.5	48.3	90.6	90.5	72.7
weighted (α=0.7)	59.9	47.6	90.6	90.5	72.1
add (ours)	**62.5**	**50.1**	<u>90.6</u>	<u>90.5</u>	**73.4**

Table 5: Ablation of fusion strategies on Office-Caltech-10. We report mean accuracy (%) over 5 runs.

Backbone Study with Vision Transformers

The use of AlexNet in our main experiments, while standard for comparing against prior work, yields low absolute baselines. To verify our framework's external validity, we revisited the Office-Caltech-10 experiment using a lightweight Vision Transformer (ViT) backbone (Wu et al. 2022).

The results shown in Table 6 confirm that pFedDB's benefits generalize to modern architectures. Our method again improved accuracy over the ViT-based baselines of *Single* (73.5%) and *FedAvg* (72.4%). Furthermore, we analyzed the cut depth l_c. The findings were consistent with our CNN-based experiments: varying l_c confirmed that the $\sim 70\%$ shared-parameter rule balancing accuracy and per-round traffic. This study validates that pFedDB is backbone-agnostic and that its core principles generalize effectively to Transformer-based models.

Conclusion

In this paper, we proposed the personalized federated dual-branch (pFedDB) framework, a novel solution for multi-domain, non-IID federated learning. Our approach features a dual-branch architecture that decouples shared knowledge from personalized expertise by maintaining a private branch that is never shared, thus preserving local domain knowledge. This structure is guided by a two-phase protocol where a local expert model is first established, ensuring that subsequent federated collaboration complements rather than overwrites local learning. Consequently, pFedDB consistently improves performance for every participating client beyond their individual baselines and achieves competitive or superior accuracy. The framework delivers significant practical advantages, including a 30% reduction in per-round communication overhead and enhanced user privacy.

Office-Caltech-10 (ViT)			
l_c	Param	Ratio	Acc
3	6.7	0.24	71.6
6	13.6	0.49	74.5
9	20.3	<u>0.73</u>	**75.9**
12	27.2	0.98	75.7

Table 6: Backbone study with a lightweight ViT on Office-Caltech-10. The l_c is cut-depth. *Param* is the number of communicated parameters (MB) per round for the shared branch. *Ratio* is the fraction of the full ViT communicated in FL. *Acc* is the mean test accuracy (%). The $\sim 70\%$ split achieves the best average while offering a favorable accuracy with traffic balance.

Acknowledgments

This work was supported by the National Science Foundation, United States under Grant 2047010, 2047064, 2205205, and 1947419. Ni was also partially supported by the Department of Transportation, United States under Grant 69A3552348304.

References

Arivazhagan, M. G.; Aggarwal, V.; Singh, A. K.; and Choudhary, S. 2019. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*.

Bernecker, T.; Peters, A.; Schlett, C. L.; Bamberg, F.; Theis, F.; Rueckert, D.; Weiß, J.; and Albarqouni, S. 2022. Fednorm: Modality-based normalization in federated learning for multi-modal liver segmentation. *arXiv preprint arXiv:2205.11096*.

Brock, A.; De, S.; Smith, S. L.; and Simonyan, K. 2021. High-performance large-scale image recognition without normalization. In *International conference on machine learning*, 1059–1071. PMLR.

Demner-Fushman, D.; Kohli, M. D.; Rosenman, M. B.; Shooshan, S. E.; Rodriguez, L.; Antani, S.; Thoma, G. R.; and McDonald, C. J. 2016. Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association*, 23(2): 304–310.

Do, T.; Nguyen, B. X.; Tran, Q. D.; Nguyen, H.; Tjiputra, E.; Chiu, T.-C.; and Nguyen, A. 2024. Reducing Non-IID Effects in Federated Autonomous Driving with Contrastive Divergence Loss. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2190–2196. IEEE.

Gong, B.; Shi, Y.; Sha, F.; and Grauman, K. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *2012*

IEEE conference on computer vision and pattern recognition, 2066–2073. IEEE.

Hanzely, F.; and Richtárik, P. 2020. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*.

He, Y.; Chen, Y.; Yang, X.; Yu, H.; Huang, Y.-H.; and Gu, Y. 2024. Learning Critically: Selective Self-Distillation in Federated Learning on Non-IID Data. *IEEE Transactions on Big Data*, 10(6): 789–800.

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

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Irvin, J.; Rajpurkar, P.; Ko, M.; Yu, Y.; Ciurea-Ilcus, S.; Chute, C.; Marklund, H.; Haghgoo, B.; Ball, R.; Shpanskaya, K.; et al. 2019. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 590–597.

Jin, H.; Bai, D.; Yao, D.; Dai, Y.; Gu, L.; Yu, C.; and Sun, L. 2023. Personalized Edge Intelligence via Federated Self-Knowledge Distillation. *IEEE Transactions on Parallel and Distributed Systems*, 34(2): 567–580.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, 5132–5143. PMLR.

Khaled, A.; Mishchenko, K.; and Richtárik, P. 2020. Tighter theory for local SGD on identical and heterogeneous data. In *International conference on artificial intelligence and statistics*, 4519–4529. PMLR.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Li, D.; and Fedmd, W. J. 2019. Heterogenous federated learning via model distillation. In *Proceedings of the NeurIPS Workshop Feder. Learn. Data Privacy Confidentiality*, volume 1, 8.

Li, Q.; He, B.; and Song, D. 2021. Model-Contrastive Federated Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Li, Q.; and Wai, H.-T. 2025. Tighter Analysis for Decentralized Stochastic Gradient Method: Impact of Data Homogeneity. *IEEE Transactions on Automatic Control*.

Li, T.; Hu, S.; Beirami, A.; and Smith, V. 2021a. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, 6357–6368. PMLR.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2: 429–450.

Li, X.; Huang, K.; Yang, W.; Wang, S.; and Zhang, Z. 2019. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*.

Li, X.; Jiang, M.; Zhang, X.; Kamp, M.; and Dou, Q. 2021b. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*.

Liang, P. P.; Liu, T.; Ziyin, L.; Allen, N. B.; Auerbach, R. P.; Brent, D.; Salakhutdinov, R.; and Morency, L.-P. 2020. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*.

Liu, Y.; Qu, Z.; Wang, S.; Shen, C.; Liang, Y.; and Wang, J. 2025. A unified Personalized Federated Learning framework ensuring Domain Generalization. *Expert Systems with Applications*, 263: 125700.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Nguyen, H. Q.; Lam, K.; Le, L. T.; Pham, H. H.; Tran, D. Q.; Nguyen, D. B.; Le, D. D.; Pham, C. M.; Tong, H. T.; Dinh, D. H.; et al. 2022. VinDr-CXR: An open dataset of chest X-rays with radiologist's annotations. *Scientific Data*, 9(1): 429.

Pang, Y.; Ni, Z.; and Zhong, X. 2024. Federated Learning for Crowd Counting in Smart Surveillance Systems. *IEEE Internet of Things Journal*, 11(3): 5200–5209.

Pang, Y.; Ni, Z.; and Zhong, X. 2025a. Integration of a new layer normalization process into federated reinforcement learning for environments with heterogeneous attribute spaces. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications VII*, volume 13473, 246–253. SPIE.

Pang, Y.; Ni, Z.; and Zhong, X. 2025b. Personalized Observation Normalization for Federated Reinforcement Learning in Simulation Environments with Heterogeneity. In *2025 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019a. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1406–1415.

Peng, X.; Huang, Z.; Zhu, Y.; and Saenko, K. 2019b. Federated adversarial domain adaptation. *arXiv preprint arXiv:1911.02054*.

Shi, N.; Lai, F.; Kontar, R. A.; and Chowdhury, M. 2021. Fed-ensemble: Improving generalization through model ensembling in federated learning. *arXiv preprint arXiv:2107.10663*.

Shih, G.; Wu, C. C.; Halabi, S. S.; Kohli, M. D.; Prevedello, L. M.; Cook, T. S.; Sharma, A.; Amorosa, J. K.; Arteaga, V.; Galperin-Aizenberg, M.; et al. 2019. Augmenting the national institutes of health chest radiograph dataset with expert annotations of possible pneumonia. *Radiology: Artificial Intelligence*, 1(1): e180041.

Shoham, N.; Avidor, T.; Keren, A.; Israel, N.; Benditkis, D.; Mor-Yosef, L.; and Zeitak, I. 2019. Overcoming forgetting in federated learning on non-iid data. *arXiv preprint arXiv:1910.07796*.

Sun, B.; Huo, H.; YANG, Y.; and Bai, B. 2021. PartialFed: Cross-Domain Personalized Federated Learning via Partial Initialization. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 23309–23320. Curran Associates, Inc.

Ullah, F.; Srivastava, G.; Xiao, H.; Ullah, S.; Lin, J. C.-W.; and Zhao, Y. 2024. A Scalable Federated Learning Approach for Collaborative Smart Healthcare Systems With Intermittent Clients Using Medical Imaging. *IEEE Journal of Biomedical and Health Informatics*, 28(6): 3293–3304.

Wu, K.; Zhang, J.; Peng, H.; Liu, M.; Xiao, B.; Fu, J.; and Yuan, L. 2022. Tinyvit: Fast pretraining distillation for small vision transformers. In *European conference on computer vision*, 68–85. Springer.

Yan, H.; and Guo, Y. 2024. Local and global flatness for federated domain generalization. In *European Conference on Computer Vision*, 71–87. Springer.

Zhu, G.; Liu, X.; Niu, J.; Tang, S.; Wu, X.; and Zhang, J. 2024a. DualFed: Enjoying both Generalization and Personalization in Federated Learning via Hierachical Representations. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, 11060–11069. New York, NY, USA: Association for Computing Machinery. ISBN 9798400706868.

Zhu, G.; Liu, X.; Tang, S.; and Niu, J. 2024b. Aligning Before Aggregating: Enabling Communication Efficient Cross-Domain Federated Learning via Consistent Feature Extraction. *IEEE Transactions on Mobile Computing*, 23(5): 5880–5896.

Zhuang, W.; and Lyu, L. 2024. FedWon: Triumphing Multi-domain Federated Learning Without Normalization. In *The Twelfth International Conference on Learning Representations*.