ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240731

FedSeProto: Learning Semantic Prototype in Federated Learning

Yanyi Lai[a], Lele Fu[b], Tianchi Liao[c], Chuan Chen[a,*] and Zibin Zheng[c]

[a]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[b]School of Systems Science and Engineering, Sun Yat-sen University, Guangzhou, China
[c]School of Software Engineering, Sun Yat-sen University, Zhuhai, China
ORCID (Yanyi Lai): https://orcid.org/0009-0001-0636-2569, ORCID (Lele Fu): https://orcid.org/0000-0001-5304-0434, ORCID (Tianchi Liao): https://orcid.org/0000-0002-2265-9194, ORCID (Chuan Chen): https://orcid.org/0000-0002-7048-3445, ORCID (Zibin Zheng): https://orcid.org/0000-0002-7878-4330

Abstract. Federated learning enables multiple clients to collaboratively train a global model without revealing their local data. However, conventional federated learning often overlooks the fact that data stored on different clients may originate from diverse domains, and the resulting domain shift problem can significantly impair the performance of the global model. In this paper, we introduce **Fed**erated **Se**mantic **Proto**type Learning (FedSeProto), a semantic prototype-based approach designed to address the domain shift issue in federated learning. The proposed method comprises two components: feature decoupling and feature alignment. Feature decoupling aims to learn semantic prototypes that can represent semantic information associated with specific categories, while feature alignment utilizes these semantic prototypes to facilitate learning of cross-client consistent features. Two key techniques are employed to achieve feature decoupling. On one hand, feature separation is achieved through the minimization of mutual information between semantic and domain features. On the other hand, the knowledge distillation is leveraged to ensure that both semantic and domain features carry the correct information. For feature alignment, intra-class semantic features are used to generate the local prototypes, which are further aggregated to the global prototypes. These global prototypes serve as guides during the local training process. Specifically, the local intra-class semantic features are driven to close to the corresponding global prototypes, thereby encouraging all clients to learn the globally consistent semantic features. Comprehensive experiments conducted on four challenging multi-domain datasets demonstrate the effectiveness of the proposed method compared with existing federated learning algorithms.

1 Introduction

Federated Learning (FL) is a distributed machine learning paradigm that enables collaborative training across multiple clients while safeguarding privacy. Within the FL framework, clients are not required to share their local data with the server. Instead, they conduct model training locally and then upload either the update gradients or model parameters to the server. The server subsequently aggregates the information received from all clients to construct a global model [27, 39, 5]. Despite its ability to achieve notable results compared with local training, FL is confronted with the challenge of data heterogeneity. Specifically, data heterogeneity refers to the dissimilarity in local data sources across different clients, resulting in non-independent and identically distributed (non-IID) data. Numerous studies have been conducted to address this issue, yet the majority have focused on label heterogeneity [25, 9], meaning the local data across different clients is class-imbalanced. Recently, there has been a growing interest in exploring feature heterogeneity issue [15, 30], acknowledging that local data may originate from distinct domains, a phenomenon referred to as domain shift. Such occurrences are prevalent in practical training scenarios. For instance, an image of a dog might originate from a real-world photograph or a piece of artwork. However, research on FL that addresses the problem of domain shift is relatively limited.

In the context of FL with domain shift, each client possesses data from a certain domain, leading to local models overfitting to their respective domain data. Existing research has explored various approaches to mitigate this challenge. Some methods employ adversarial training to capture domain-invariant information [3, 14]. However, these methods require domain labels to aid in the learning of domain-invariant features, which are unavailable in the FL scenario due to limited data access permissions. To unify the cross-client feature space, prototype-based methods gain wide attention. For example, FedProto [32] guides the local representation learning on each client by introducing the prototypes, which average the features of samples within the same category. Prototypes can represent the informative characteristics of a category, serving as a regularization term to enhance the intra-class compactness of feature representations. However, in the presence of data heterogeneity, the intra-class representations across different clients are inconsistent, resulting in the generation of biased prototypes. To address this, FedCD [26] employs a hierarchical prototype contrastive learning strategy to learn fine-grained prototypes, while utilizing global fine-grained prototypes to augment the local dataset, thereby mitigating the problems of data heterogeneity. FPL [15] enhances the diversity of prototypes through clustering, thereby balancing the proportion of prototypes from different domains. These methods represent effective

* Corresponding Author. Email: chenchuan@mail.sysu.edu.cn

improvements over conventional prototype learning approaches but still possess inherent limitations. The fundamental cause of performance degradation in domain shift scenarios is the lack of data from other domains at the client level, leading local models to exhibit bias towards their own datasets. While existing federated prototype learning methods alleviate the negative effects of domain shifts by enriching prototype diversity, they still learn biased prototypes and do not fundamentally resolve the issue of local dataset bias, which introduces certain limitations when addressing the issue of domain shift.

Since prototypes essentially serve as abstract representations of particular data categories, using prototypes as a guide during model training can help ensure that representations are learned consistently across clients. Accordingly, we propose FedSeProto, which employs semantic features to aggregate the unbiased local and global prototypes, mitigating the impact of domain-specific information in FL. Broadly speaking, general features extracted by a general encoder incorporate a blend of semantic and domain information, which are separately encoded by two distinct encoders at the client side. Mutual information minimization and knowledge distillation are employed to disentangle the components. Mutual information is a metric that quantifies the extent of shared information between two variables. Minimizing the mutual information between semantic features and domain features can reduce the amount of information each contains about the other, thereby facilitating the separation of features. Moreover, it is essential to ensure that both semantic and domain features retain relevant information rather than merely unrelated to each other. To achieve this, we train a basic model with general features, employing its outputs as soft labels. Semantic features are guided to preserve the knowledge learned by the basic model to accomplish downstream tasks leveraging semantic information, while domain features are directed to discard this semantic information, thereby retaining domain information. Following this process, the semantic features, which are not affected by any particular domain, maintain consistent across various domains. Based on the semantic features, the local and global unbiased prototypes are obtained and circumvent the issue of client shift caused by cross-domain aggregation. Notably, FedSeProto is also applicable in scenarios with model heterogeneity, requiring only that the local prototypes uploaded by all clients have the same dimensions. Overall, our main contributions can be summarized as follows:

- To address the issue of biased representations learned by different clients in FL, we propose feature decoupling to learn semantic prototypes of the data. Mutual information minimization and knowledge distillation are utilized to ensure that semantic and domain features are distinctly separated and carry the appropriate information.
- To address the issue of intra-class feature inconsistency across clients, we propose aggregating semantic features to derive local and global semantic prototypes. The constraints imposed by global prototypes are utilized to facilitate the learning of unified intra-class representations across different clients.
- We conduct extensive experiments on four multi-domain datasets: Digits, Office Caltech, PACS, and VLCS. The results demonstrate that our proposed FedSeProto addresses the domain shift problem in FL more effectively compared to other methods.

2 Related work

2.1 Federated learning with data heterogeneity

Federated learning enables multiple clients to collaboratively train a global model without sharing their data. However, the heterogeneity of data across different clients can significantly impair the effectiveness of FL. A common approach to address this issue is enforcing consistency between the global model and the local models. FedProx [24] mitigates the deviation of the local models from the global model by incorporating a proximal term. FedDyn [1] introduces a dynamic regularizer for each device. FedDC [10] tracks the disparity between the local and global models through learned local drift variables. MOON [23] adopts contrastive learning to align the local representations closely with the global representations while increasing the distance from the local representations of the previous round. Another strategy focuses on server-side aggregation policy. FedMA [33] matches and averages hidden elements with similar feature signatures, while FedNova [34] normalizes the local updates before model aggregation to reduce inconsistencies. In addition to class imbalance, another important scenario of data heterogeneity is domain shift, where data originate from different domains. GA [40] dynamically adjusts the weight distribution of each round's aggregation based on the generalization gap of the global model across different client datasets, allowing the global model to adapt to data from various domains. FedSR [30] employs regularization to learn a simple data representation, facilitating better generalization across diverse domains. Methods like FSFL [14] employ adversarial training to develop a domain discriminator to assist in the knowledge distillation process. These approaches require additional models and datasets, consuming substantial resources.

2.2 Prototype learning

A prototype refers to the average representation of data within the same category [31, 36]. Employing prototype loss as a regularization term can enhance the intra-class compactness of feature representations. Prototype learning has been demonstrated to exhibit exceptional performance in zero-shot and few-shot learning [35, 43]. Furthermore, [42] has shown that prototypes help address catastrophic forgetting in incremental learning. For source-free domain adaptation, [41] uses prototypes for clustering data and introduces prototype regularization to align the distribution between target samples and prototypes. Due to its broad range of applications, prototype learning has also attracted attention in the field of FL. FedProto [32] proposes the prototype aggregation to facilitate information sharing between clients, supporting model heterogeneity while significantly reducing communication overhead. FedProc [28] introduces prototype contrastive learning to address issues of data heterogeneity. FedCD [26] proposes a hierarchical prototype contrastive learning strategy, introducing fine-grained prototypes to balance class distributions across clients. FPL [15] introduces a cluster prototype strategy to alleviate the issue of domain imbalance among clients. However, these methods learn prototypes that are biased towards specific domains, which limits their effectiveness in addressing domain shift issue.

3 Methodology

3.1 Preliminaries

Problem formulation. We consider a FL setting involving N clients, where the n-th client possesses a local dataset $D_n = \{(x_n^i, y_n^i)\}_{i=1}^{|D_n|}$, with $|D_n|$ denoting the number of samples held by the n-th client, and the i-th sample x_n^i is accompanied by a corresponding label y_n^i. Additionally, each client n possesses a local model $f_n(w_n; x) : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ that takes a d_x-dimensional sample x as input and outputs a d_y-dimensional prediction vector, where

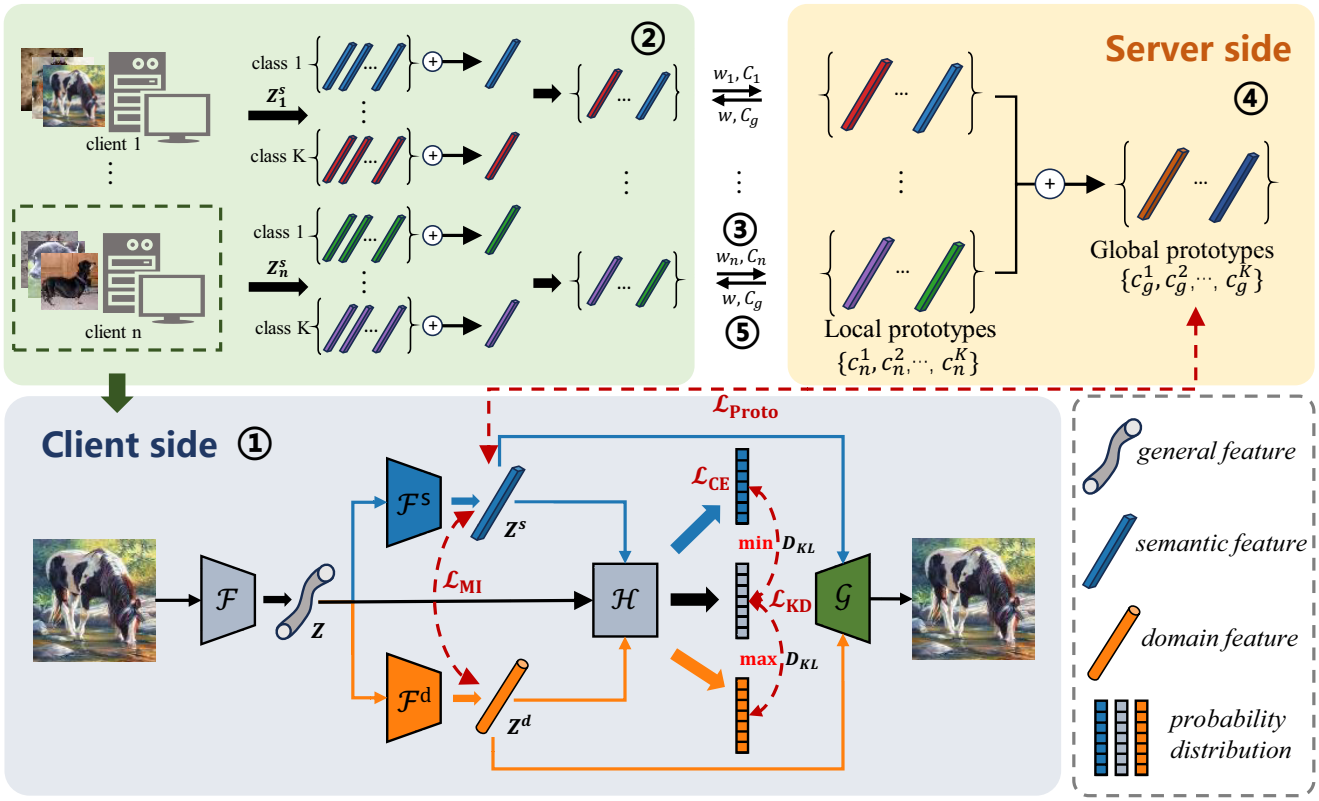


Figure 1: Overview of the proposed FedSeProto approach, where different types of features are represented by distinct shapes, and the same shapes are differentiated by color to denote different domains. In each communication round: ① Clients conduct local training, achieve feature decoupling, and align semantic features with global prototypes from the previous round. ② Local model output semantic features, which are averaged by category to generate local prototypes. ③ Clients upload the local model and prototypes to the server. ④ The server aggregates all clients' information into a global model and global prototypes. ⑤ The server distributes the global model and prototypes back to the clients. Steps ①-⑤ are repeated until the global model converges.

w_n denotes the parameters of the local model. In this setting, although the data across different participants may share the same label space, they originate from distinct underlying distributions. This leads to different conditional feature distributions $p(x|y)$, manifesting as domain shift in FL. Our objective is to construct a global model $w = \sum_{n=1}^{N} \frac{|D_n|}{|D|} w_n$, which performs well across all domains, where $|D|$ is the number of samples held by all clients. The problem can be formalized as follows:

$$\min_{w} \sum_{n=1}^{N} \frac{|D_n|}{|D|} \mathcal{L}_n(w; D_n), \tag{1}$$

where $\mathcal{L}_n(w; D_n)$ is the loss of the global model w on the n-th client's local dataset D_n.

Federated prototype learning. Federated prototype learning in scenarios with domain shift aims to address the issue of inconsistent cross-domain data distributions among multiple clients. In current federated prototype learning approaches, each prototype represents the average of the feature vectors corresponding to the same category, thereby capturing the key characteristics of the data and encompassing category-specific information. Specifically, the local model of the n-th client is composed of two components, denoted as $f_n(w_n) = \mathcal{F}_n(\delta_n) \circ \mathcal{H}_n(\theta_n)$. Here, $\mathcal{F}_n(\delta_n; x) : \mathbb{R}^{d_x} \to \mathbb{R}^{d_z}$ is a feature encoder that encodes a sample x into a feature vector z with δ_n being the parameters of $\mathcal{F}_n$, $\mathcal{H}_n(\theta_n; z) : \mathbb{R}^{d_z} \to \mathbb{R}^{d_y}$ is the prediction head with parameters θ_n, and $\circ$ denotes concatenation.

Thus, the prototype for the k-th category on client n can be defined as follows:

$$c_n^k = \frac{1}{|D_n^k|} \sum_{x_i \in D_n^k} \mathcal{F}_n\left(\delta_n; x_i\right), \tag{2}$$

where D_n^k denotes the subset of samples from the local dataset of client n that belong to the k-th category. Under the FL framework, the local prototypes from all clients are aggregated to form a global set of prototypes. A straightforward approach is averaging the prototypes of the same category across different clients. Consequently, the global prototype for the k-th category can be computed as follows:

$$c_g^k = \frac{1}{N^k} \sum_{n=1}^{N^k} c_n^k, \tag{3}$$

where N^k denotes the number of clients that contain samples of the k-th category. Upon obtaining the global prototypes set, it can be utilized to impose constraints on the local training process of the clients. For instance, regularization terms such as L_1 or L_2 can be incorporated during the local model training, which enables the local models of different clients to acquire knowledge from other domains.

3.2 *Overview of proposed method*

The overall framework of FedSeProto is depicted in Figure 1. Following the setting of conventional FL, the proposed FedSeProto in-

corporates N clients and a server. Each client n possesses a local model that includes a general feature encoder $\mathcal{F}_n$, a semantic feature encoder $\mathcal{F}_n^s(\varphi_n) : \mathbb{R}^{d_z} \to \mathbb{R}^{d_z}$ with parameters φ_n, and a domain feature encoder $\mathcal{F}_n^d(\phi_n) : \mathbb{R}^{d_z} \to \mathbb{R}^{d_z}$ with parameters ϕ_n, each designed to extract semantic features z^s and domain features z^d from the general features z, respectively. Specifically, at the start of the t-th round, each client n calculates the semantic features $Z_n^s = \mathcal{F}_n^s(\varphi_n; \mathcal{F}_n(\delta_n; D_n))$ and the domain features $Z_n^d = \mathcal{F}_n^d(\phi_n; \mathcal{F}_n(\delta_n; D_n))$, performing feature decoupling through mutual information minimization and knowledge distillation. Subsequently, Z_n^s is utilized for downstream prediction tasks and is aligned with the global prototypes $\{c_t^k\}_{k=1}^K$, where K is the number of categories. After local training, Z_n^s is averaged by category to generate the local prototypes $\{c_{t,n}^k\}_{k=1}^K$, and both the local model w_n^t and the local prototypes $\{c_{t,n}^k\}_{k=1}^K$ are uploaded to the server. The server aggregates all the local models and prototypes into the global model w and the global prototypes $\{c_g^k\}_{k=1}^K$, which are then distributed back to the clients.

In the following section, we will elaborate on how to achieve feature decoupling during the local training process, isolating domain-invariant semantic features, and how prototype-based feature alignment can be utilized to address the issue of domain shift.

3.3 Semantic prototype learning

The proposed method employs feature decoupling and feature alignment to facilitate semantic prototype learning. The features obtained by general encoders $\mathcal{F}(\delta)$ contain semantic information and domain information. To disentangle these components and aggregate the extracted semantic features as prototypes, Mutual Information (MI)-based loss and Knowledge Distillation (KD)-based loss are utilized to ensure the decoupling of semantic and domain information, while prototype-based loss is employed for feature alignment: 1) MI-based loss guarantees that the semantic encoder acquires information unrelated to the domain content, facilitating feature separation. 2) KD-based loss ensures that the semantic encoder captures category-relevant semantic information, while the domain encoder learns semantically irrelevant domain information. 3) Prototype-based loss encourages the features learned by the client to be consistent with global prototypes, eliminating the effects of domain shift. In this section, we will focus on the analysis of these loss functions.

3.3.1 MI-based loss

To avoid the domain encoder capturing semantic information, we employ MI during the training process. Specifically, feature separation is performed by minimizing the MI between the outputs of the semantic encoder and the domain encoder. Our training objective can be expressed as follows:

$$\min I(\mathbf{Z^s}; \mathbf{Z^d}). \tag{4}$$

It is widely acknowledged that direct calculation of MI is not feasible [38], leading to the common practice of minimizing the variational upper bound of MI as an alternative approach [4, 6]. In our approach, $\mathbf{Z^s}$ and $\mathbf{Z^d}$ further complicating the scenario due to their the latent nature. Following the hypothesis in [17], by introducing a data variable X, the MI between two latent variables can be transformed into the MI between a data variable and a latent variable, we obtain:

$$I(\mathbf{Z^s}; \mathbf{Z^d}) = I(\mathbf{X}; \mathbf{Z^s}) + I(\mathbf{X}; \mathbf{Z^d}) - I(\mathbf{X}; \mathbf{Z^s}, \mathbf{Z^d}). \tag{5}$$

Therefore, the problem of minimizing the upper bound of $I(\mathbf{Z^s}; \mathbf{Z^d})$ is transformed into minimizing the upper bounds of $I(\mathbf{X}; \mathbf{Z^s})$ and $I(\mathbf{X}; \mathbf{Z^d})$, as well as maximizing the lower bound of $I(\mathbf{X}; \mathbf{Z^s}, \mathbf{Z^d})$. Let $q(\mathbf{z^s})$ be the variational approximation of $p(\mathbf{z^s})$. Theoretically, we can derive the variational upper bound of $I(\mathbf{X}; \mathbf{Z^s})$ as:

$$\begin{aligned} I(\mathbf{X}; \mathbf{Z^s}) &= \int\int p(\mathbf{x}, \mathbf{z^s}) \log\left(\frac{p(\mathbf{z^s}|\mathbf{x}) q(\mathbf{z^s})}{q(\mathbf{z^s}) p(\mathbf{z^s})}\right) d\mathbf{x} d\mathbf{z^s} \\ &= \int\int p(\mathbf{x}, \mathbf{z^s}) \log\left(\frac{p(\mathbf{z^s}|\mathbf{x})}{q(\mathbf{z^s})}\right) d\mathbf{x} d\mathbf{z^s} \\ &\quad - D_{KL}[p(\mathbf{z^s}) \parallel q(\mathbf{z^s})] \\ &\leq \int p(\mathbf{x}) d\mathbf{x} \int p(\mathbf{z^s}|\mathbf{x}) \log\left(\frac{p(\mathbf{z^s}|\mathbf{x})}{q(\mathbf{z^s})}\right) d\mathbf{z^s} \\ &= \frac{1}{|D_n|} \sum_{i=1}^{|D_n|} D_{KL}[p(\mathbf{z_i^s}|\mathbf{x}_i) \parallel q(\mathbf{z_i^s})], \end{aligned} \tag{6}$$

where D_{KL} denotes the KL divergence. Similarly, let $r(\mathbf{z^d})$ be the variational approximation of $p(\mathbf{z^d})$, we obtain:

$$I(\mathbf{X}; \mathbf{Z^d}) \leq \frac{1}{|D_n|} \sum_{i=1}^{|D_n|} D_{KL}[p(\mathbf{z_i^d}|\mathbf{x}_i) \parallel r(\mathbf{z_i^d})]. \tag{7}$$

Letting $t(\mathbf{x}|\mathbf{z^s}, \mathbf{z^d})$ be the variational approximation of $p(\mathbf{x}|\mathbf{z^s}, \mathbf{z^d})$, we can derive the variational lower bound of $I(\mathbf{X}; \mathbf{Z^s}, \mathbf{Z^d})$:

$$\begin{aligned} &I(\mathbf{X}; \mathbf{Z^s}, \mathbf{Z^d}) \\ &= \iiint p(\mathbf{x}, \mathbf{z^s}, \mathbf{z^d}) \log\left(\frac{p(\mathbf{x}|\mathbf{z^s}, \mathbf{z^d}) t(\mathbf{x}|\mathbf{z^s}, \mathbf{z^d})}{t(\mathbf{x}|\mathbf{z^s}, \mathbf{z^d}) p(\mathbf{x})}\right) d\mathbf{x} d\mathbf{z^s} d\mathbf{z^d} \\ &= \iiint p(\mathbf{x}, \mathbf{z^s}, \mathbf{z^d}) \log\left(\frac{t(\mathbf{x}|\mathbf{z^s}, \mathbf{z^d})}{p(\mathbf{x})}\right) d\mathbf{x} d\mathbf{z^s} d\mathbf{z^d} \\ &\quad + \mathbb{E}_{p(\mathbf{z^s}, \mathbf{z^d})}[D_{KL}[p(\mathbf{x}|\mathbf{z^s}, \mathbf{z^d}) \parallel t(\mathbf{x}|\mathbf{z^s}, \mathbf{z^d})]] \\ &\geq \int p(\mathbf{x}) d\mathbf{x} \int\int p(\mathbf{z^s}, \mathbf{z^d}|\mathbf{x}) \log(t(\mathbf{x}|\mathbf{z^s}, \mathbf{z^d})) d\mathbf{z^s} d\mathbf{z^d} \\ &= \frac{1}{|D_n|} \sum_{i=1}^{|D_n|} \log(t(\mathbf{x_i}|\mathbf{z_i^s}, \mathbf{z_i^d})). \end{aligned} \tag{8}$$

Finally, we can derive the variational upper bound of $I(\mathbf{Z^s}; \mathbf{Z^d})$:

$$\begin{aligned} I(\mathbf{Z^s}; \mathbf{Z^d}) \leq \frac{1}{|D_n|} \sum_{i=1}^{|D_n|} \Big(&D_{KL}\left[p(\mathbf{z_i^s}|\mathbf{x_i}) \parallel q(\mathbf{z_i^s})\right] \\ &+ D_{KL}\left[p(\mathbf{z_i^d}|\mathbf{x_i}) \parallel r(\mathbf{z_i^d})\right] + \log\left(-t(\mathbf{x_i}|\mathbf{z_i^s}, \mathbf{z_i^d})\right)\Big). \end{aligned} \tag{9}$$

Following the assumption in [18], $q(\mathbf{z^s})$ and $r(\mathbf{z^d})$ can be considered as standard normal distributions $\mathcal{N}(0, 1)$. The last term can be optimized by minimizing the reconstruction error of $\mathbf{X}$ from $\mathbf{Z^s}$ and $\mathbf{Z^d}$. The MI-based loss of the n-th client can be formulated as follows:

$$\begin{aligned} \mathcal{L}_{\text{MI}} = \frac{1}{|D_n|} \sum_{i=1}^{|D_n|} \Big(&D_{KL}\left(z_{n,i}^s, \mathcal{N}(0, 1)\right) + D_{KL}\left(z_{n,i}^d, \mathcal{N}(0, 1)\right) \\ &+ \ell_{MSE}\left(x_i, \mathcal{G}_n(\psi_n; z_{n,i}^s \circ z_{n,i}^d)\right)\Big), \end{aligned} \tag{10}$$

where $\mathcal{G}_n(\psi_n) : \mathbb{R}^{2d_z} \to \mathbb{R}^{d_x}$ represents the decoder that reconstructs the original data from z^s and z^d with parameters ψ_n, ℓ_{MSE} denotes the mean squared error.

3.3.2 KD-based loss

Existing work on feature disentangling typically employs adversarial learning techniques [3, 14], however, these methods often require domain labels of the data, which poses a challenge in FL settings where each client can only utilize data from its local domain. Therefore, it is necessary to design a method that facilitates the separation of semantic features and domain features under constrained conditions.

We divide the local training process into two distinct phases. In the first phase, a general encoder $\mathcal{F}(\delta)$ is employed as the feature extractor to train a basic model $\mathcal{F}(\delta) \circ \mathcal{H}(\theta)$ that possesses preliminary prediction capabilities. Subsequently, in the second phase, while fixing the parameters of the prediction head θ, inspired by *Knowledge Distillation*, we use the output logits of the basic model as soft labels. The general features z are further input into the semantic encoder $\mathcal{F}^s(\varphi)$ and domain encoder $\mathcal{F}^d(\phi)$. Intuitively, the features extracted by the semantic encoder z^s should retain the semantically relevant aspects of the general features, thereby preserving as much as possible the prediction knowledge learned by the basic model. In contrast, the domain features z^d should capture the semantically irrelevant aspects of the general features, hence discarding the prediction knowledge learned by the basic model. Based on the above understanding, we use KL divergence to measure the similarity between the final output logits and soft labels. Then, we reconcile these aspects into a KD-based loss:

$$\begin{aligned} \mathcal{L}_{\text{KD}} &= \frac{1}{|D_n|} \sum_{i=1}^{|D_n|} \left(-\log \left(\frac{exp(s_1)}{exp(s_1) + exp(s_2)} \right) \right), \\ \text{where} \quad s_1 &= D_{KL} \left(\mathcal{H}_n(\theta_n; z_{n,i}), \mathcal{H}_n(\theta_n; \mathcal{F}_n^s(\varphi_n; z_{n,i})) \right), \\ s_2 &= D_{KL}(\mathcal{H}_n(\theta_n; z_{n,i}), \mathcal{H}_n(\theta_n; \mathcal{F}_n^d(\phi_n; z_{n,i}))). \end{aligned} \tag{11}$$

Here, $\mathcal{H}_n(\theta_n)$ is the prediction head of the n-th client, s_1 and s_2 are intermediate variables, each representing the KL divergence between the soft labels and the output logits predicted using features obtained from the semantic encoder and the domain encoder, respectively.

3.3.3 Prototype-based loss

Given the process of feature decoupling at the local training phase, the computation of local prototypes is defined as follows:

$$c_n^k = \frac{1}{|D_n^k|} \sum_{i=1}^{|D_n^k|} \mathcal{F}_n^s(\varphi_n; \mathcal{F}_n(\delta_n; x_i)). \tag{12}$$

The local prototypes are uploaded to the server and aggregated into the global prototypes via Equation (3). The prototype-based loss can be integrated as a regularization term into the overall loss function, serving to measure the distance between the local semantic feature and the corresponding global prototype c_g^k. Here, we employ the L_2 distance:

$$\mathcal{L}_{\text{Proto}} = \frac{1}{K} \sum_{k=1}^{K} \left(\frac{1}{|D_n^k|} \sum_{i=1}^{|D_n^k|} \| \mathcal{F}_n^s(\varphi_n; \mathcal{F}_n(\delta_n; x_i)) - c_g^k \|^2 \right). \tag{13}$$

Additionally, the classification loss can be calculated as follows:

$$\mathcal{L}_{\text{CE}} = \frac{1}{|D_n|} \sum_{i=1}^{|D_n|} -y_i \log(\mathcal{H}_n(\theta_n; z_{n,i}^s)). \tag{14}$$

Finally, we can derive the total loss function:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \alpha \mathcal{L}_{\text{KD}} + \beta \mathcal{L}_{\text{MI}} + \lambda \mathcal{L}_{\text{Proto}}, \tag{15}$$

where α, β, λ are the trade-off parameters.

3.3.4 Global aggregation

Upon completion of local training, clients upload their trained local model parameters w_n to the server. The server then aggregates the local models from all clients into a global model, calculated as follows:

$$w = \sum_{n=1}^{N} \frac{|D_n|}{|D|} w_n, \tag{16}$$

where D represents the dataset collectively constituted by all clients. It is noteworthy that the domain encoder $\mathcal{F}_n^d(\phi_n)$ and the decoder $\mathcal{G}_n(\psi_n)$ of the local model should be tailored to the local domain and do not require global aggregation. Consequently, the model parameters $w_n = (\delta_n, \varphi_n, \theta_n)$ are uploaded by the clients.

3.4 Extending to model heterogeneity scenarios

The previous description pertains to FL under the scenario of model homogeneity, which requires all clients to share the same model architecture to facilitate global aggregation. However, in practical FL, clients may possess varying computational resources. Clients with limited resources can hinder the overall training process of FL [7, 2]. This highlights the necessity of allowing model heterogeneity in FL, whereby clients are not required to use a uniform model architecture. Instead, each client can independently choose a model that aligns with their local resource capabilities. The proposed FedSeProto can be naturally extended to scenarios involving model heterogeneity. Although global model aggregation becomes unfeasible due to model heterogeneity, global prototype aggregation remains viable. Through global prototypes, effective information sharing can be achieved among clients. Specifically, for the local general encoder $\mathcal{F}_n(\delta_n; x) : \mathbb{R}^{d_x} \to \mathbb{R}^{d_z}$ and semantic encoder $\mathcal{F}_n^s(\varphi_n) : \mathbb{R}^{d_z} \to \mathbb{R}^{d_z}$ of client n, communication can be ensured as long as the dimensionality d_z is consistent across different clients, while the specific model architecture can be determined by each client. The efficacy of our method in the scenario of model heterogeneity will also be validated in the experimental section. The overall workflow of FedSeProto is detailed in Algorithm 1.

4 Experiments

4.1 Experiments setup

Baselines. In the scenario of model homogeneity, our method is compared against various FL algorithms, including three classical methods: FedAvg [27], FedProx [24], MOON [23], three prototype-based methods: FedProto [32], FedMLP [13] and FPL [15], and two state-of-the-art methods developed in recent years: FedSR [30] and GA [40]. In the model heterogeneous scenario, certain aforementioned methods are not applicable. Thus, we additionally include methods that support model heterogeneity, namely FedMD [21] and FedGH [37], for comparison. Furthermore, the Local method denotes the strategy where clients utilize only their local data for model training, without engaging in knowledge sharing between clients.

Datasets. We conducted experiments on 4 cross-domain datasets: Digits [16, 20, 29], Office Caltech [11, 12], PACS [22], VLCS [8].

Algorithm 1 The proposed FedSeProto

Input: Number of clients N, communication rounds T, local training epochs E
Output: Global model w^T
for $t = 1, 2, ..., T$ **do**
  for $n = 1, 2, ..., N$ **in parallel do**
    $w_n^t, c_n^t \leftarrow ClientUpdate(w^t, c_g^t)$
  end for
  Aggregate global prototypes c_g^{t+1} via Equation (3) (and global model w^{t+1} via Equation (16) if model homogeneous)
end for
ClientUpdate(w^t, c_g^t):
$w_n^t \leftarrow w^t$ if model homogeneous
for $e = 1, 2, ..., E$ **do**
  Compute the total loss $\mathcal{L}$ via Equation (15)
  Update local model $w_n^t \leftarrow w_n^t - \eta \nabla \mathcal{L}$
end for
Update local prototypes c_n^t via Equation (12)
Upload c_n^t (and w_n^t if model homogeneous) to the server

Among these, Digits is a digit recognition dataset, and the remaining 3 datasets are object recognition datasets. For all the datasets, we established 10 clients. To simulate realistic domain shift scenarios, we assumed the number of clients belonging to different domains varies. For instance, the distribution for the PACS dataset is configured as: Photo: 2, Art Painting: 3, Cartoon: 4, Sketch: 1.

Backbone. A simple multi-layer CNN architecture was employed as the backbone. In the model-homogeneous scenario, the general feature extractor comprises three convolution layers followed by a fully connected layer, outputting 512-dimensional features. Both the semantic encoder and the domain encoder consist of a single fully connected layer, ultimately connected to a classification layer. In the model-heterogeneous scenario, the general feature extractor for some clients is set to two convolution layers, with the remaining settings identical to those in the model homogeneous scenario. For additional details on the experimental datasets, models, and implementation, please refer to the supplementary material [19].

4.2 Main experiments results

4.2.1 Model homogeneous scenarios

In scenarios of model homogeneity, experiments are conducted on four challenging multi-domain datasets. Table 1 and Table 2 provide the accuracy for individual domains as well as the average accuracy across all domains, where $\triangle$ indicates the improvement compared to FedAvg. The best results are highlighted in bold, while the next best results are indicated with an underline. The results demonstrate that all prototype-based methods, namely FedProto, FedMLP, FPL, and FedSeProto, exhibited outstanding performance in most cases, highlighting the significant role of prototype learning. However, the two methods designed for federated domain generalization, FedSR and GA, sometimes failed to adequately address the issue of domain shift. Notably, GA exhibited a significant bias towards particular data domains on the Digits dataset. The proposed FedSeProto outperformed all other comparison methods, especially on the Office Caltech dataset, where it achieved a 6.85% increase in accuracy compared to the baseline method and a 3.56% improvement over the next best method, FPL. This demonstrates the effectiveness of FedSeProto in addressing domain shift issue. Figure 2 presents the comparison of t-SNE visualization of the embedding representations on the MNIST dataset between FedSeProto and FedAvg. The results indicate that FedSeProto is able to learn clearer decision boundaries. Due to space limitation, more extensive experiments comparing FedSeProto with other methods are provided in the supplementary material [19].

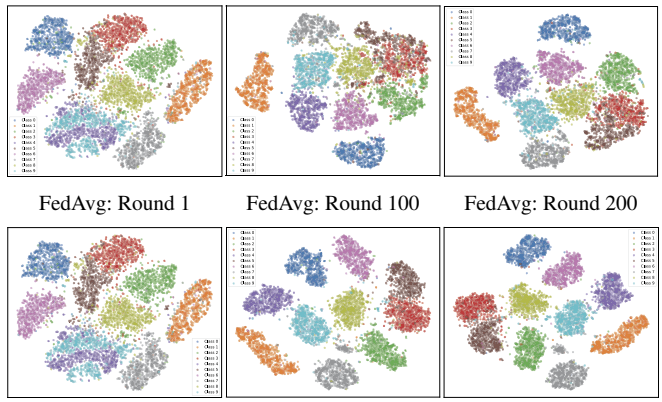


Figure 2: t-SNE visualization of FedAvg (top) and FedSeProto (bottom) on MNIST dataset at different communication rounds.

4.2.2 Model heterogeneous scenarios

The proposed method is also effective in scenarios of model heterogeneity. As model heterogeneity is not the central focus of this paper, experimental validation in this section is limited to Digits dataset, with results detailed in Table 3. Performance assessments reveal that all comparison methods underperform relative to scenarios with model homogeneity. This discrepancy is linked to the procedure of model testing, where the test set comprises data from all domains, while the model only learns from single-domain knowledge during training. Additionally, the constraints imposed by model heterogeneity, which preclude straightforward model aggregation, hinder the ability of models to assimilate knowledge from different domains across clients. Despite these challenges, it is noteworthy that FedSeProto consistently outperforms other methods, achieving a substantial improvement of 11.32% over the Local method and 1.23% over the next best method, FedProto.

4.3 Ablation study

The proposed FedSeProto comprises four components, reflected in the four terms of the loss function: $\mathcal{L}_{\text{CE}}$, $\mathcal{L}_{\text{KD}}$, $\mathcal{L}_{\text{MI}}$, and $\mathcal{L}_{\text{Proto}}$. To assess the contribution of each component, we designed a series of experiments, with results presented in Figure 3. With only $\mathcal{L}_{\text{CE}}$, FedSeProto degenerates to FedAvg, exhibiting the poorest performance. In the absence of $\mathcal{L}_{\text{MI}}$, the model can separate semantic and domain features to some extent but fails to fully decouple the two. Lacking $\mathcal{L}_{\text{KD}}$ prevents ensuring that the semantic and domain encoders learn the intended information. Without $\mathcal{L}_{\text{Proto}}$, there is no global consistent goal to guide the learning process. Optimal results are achieved when all components are employed simultaneously, indicating that each component of the proposed method is indispensable.

4.4 Parameter sensitivity analysis

To investigate the impact of hyperparameters on the performance of FedSeProto, we conducted several parameter sensitivity experiments for the three trade-off parameters, α, β, and λ, on Office Caltech and

Table 1: Comparison of performance (%) on Digits and Office Caltech datasets.

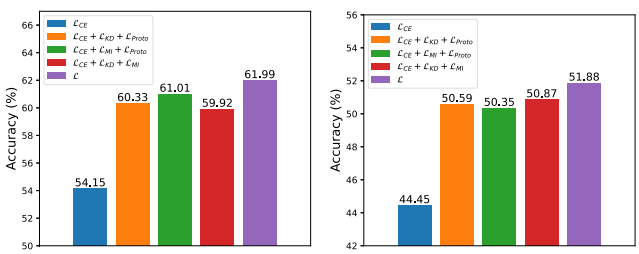
Methods	Digits						Office Caltech					
	MNIST	USPS	SVHN	SYN	Avg.	△	Caltech	Amazon	Webcam	DSLR	Avg.	△
FedAvg [27]	80.18	72.67	**71.92**	55.82	70.15 ± 0.28	-	54.31	63.51	48.27	54.44	55.14 ± 1.07	-
FedProx [24]	79.09	72.12	71.12	55.57	69.48 ± 0.48	-0.67	54.76	61.23	45.40	57.78	54.79 ± 0.22	-0.35
MOON [23]	81.30	71.78	71.84	55.85	70.19 ± 0.07	+0.04	54.91	62.10	52.30	54.45	55.94 ± 0.72	+0.80
FedProto [32]	79.09	72.12	71.12	55.57	69.48 ± 0.48	-0.67	55.35	65.79	48.28	**61.11**	57.63 ± 0.60	+2.49
FedMLP [13]	80.21	75.15	70.63	55.82	70.45 ± 0.56	+0.30	54.91	63.45	45.40	54.45	54.55 ± 0.30	-0.59
FedSR [30]	80.29	74.64	68.54	52.69	69.04 ± 0.43	-0.49	55.66	66.49	48.27	48.89	54.83 ± 0.69	-0.31
FedAvg+GA [40]	80.75	**83.21**	59.50	49.40	68.22 ± 0.19	-1.93	46.58	46.84	**63.79**	58.89	54.03 ± 0.54	-1.11
FPL [15]	77.47	73.17	71.67	**61.30**	70.90 ± 0.31	+0.75	56.25	**66.84**	51.72	58.89	58.43 ± 0.26	+3.29
FedSeProto	**81.41**	76.29	71.18	58.36	**71.81 ± 0.34**	**+1.66**	**63.10**	66.14	62.07	56.67	**61.99 ± 1.19**	**+6.85**

Table 2: Comparison of performance (%) on PACS and VLCS datasets.

Methods	PACS						VLCS					
	Photo	Art painting	Cartoon	Sketch	Avg.	△	Caltech	Labelme	Pascal	Sun	Avg.	△
FedAvg [27]	50.10	36.36	51.71	39.83	44.50 ± 0.06	-	72.56	55.67	46.89	56.18	57.83 ± 0.08	-
FedProx [24]	57.98	37.50	54.06	45.99	48.88 ± 0.12	+4.38	72.64	55.83	46.89	56.45	57.95 ± 0.09	+0.12
MOON [23]	55.69	36.27	54.27	45.22	47.87 ± 0.24	+3.37	71.70	55.63	46.79	56.99	57.78 ± 0.02	-0.05
FedProto [32]	57.49	37.42	54.91	46.23	49.01 ± 0.18	+4.51	72.80	55.92	47.25	57.29	58.32 ± 0.21	+0.49
FedMLP [13]	57.20	37.51	57.06	46.27	49.51 ± 0.32	+5.01	69.25	55.48	46.32	56.82	56.97 ± 0.22	-0.86
FedSR [30]	57.58	34.80	55.20	45.99	48.39 ± 0.18	+3.89	68.95	56.00	44.59	54.72	56.06 ± 0.10	-1.77
FedAvg+GA [40]	52.59	37.83	52.64	36.73	44.95 ± 0.14	+0.45	68.32	55.04	44.75	54.45	55.64 ± 0.07	-2.19
FPL [15]	61.38	35.29	57.19	46.50	50.09 ± 0.83	+5.59	72.25	56.46	48.44	56.72	58.47 ± 0.38	+0.64
FedSeProto	**61.98**	**38.08**	**60.11**	**47.35**	**51.88 ± 0.44**	**+7.38**	**73.82**	**57.51**	**48.96**	**60.04**	**60.08 ± 0.24**	**+2.25**

Table 3: Comparison of performance (%) on Digits dataset under model heterogeneous scenarios.

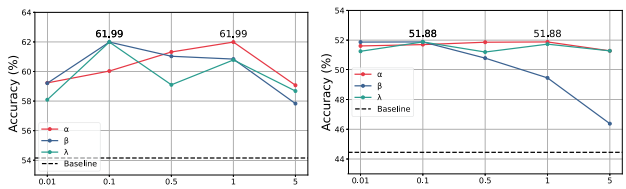
Methods	Digits				
	MNIST	USPS	SVHN	SYN	Avg.
Local	29.61	42.99	21.75	20.57	28.73 ± 0.03
FedMD	39.90	51.36	29.20	23.10	35.89 ± 0.03
FedProto	39.82	54.61	33.44	27.41	38.82 ± 1.76
FedGH	37.12	51.05	33.39	25.59	36.79 ± 0.20
FedSeProto	**40.75**	**56.44**	**34.53**	**28.46**	**40.05 ± 0.28**



(a) Office Caltech (b) PACS

Figure 3: Ablation study on Office Caltech and PACS datasets.

PACS datasets. The results are depicted in Figure 4, where the *Baseline* performance of FedAvg is illustrated with a black dashed line. The results reveal that the choice of the three hyperparameters has a noticeable influence on the performance of the proposed method. Parameter choices that are too small fail to impose sufficient constraints on the model, while excessively large parameters cause the model to overly prioritize feature decoupling and alignment, detracting from primary task learning. The model is most sensitive to variations in β. Particularly, a larger value of β results in a substantial performance decline for FedSeProto. This occurs because while optimizing $\mathcal{L}_{\text{MI}}$ decouples the features, it does not guarantee that they represent semantic and domain information effectively. If β is overly large, the model learns two meaningless features, failing to achieve the objective of learning semantic features. However, it is noteworthy that FedSeProto still significantly outperforms the Baseline within a considerable range of hyperparameter fluctuations.



(a) Office Caltech (b) PACS

Figure 4: Parameter sensitivity analysis of the trade-off hyperparameters.

5 Conclusion

In this paper, we propose a Federated Semantic Prototype Learning (FedSeProto) approach to address the issue of domain shift. The proposed method facilitates the learning of unbiased prototypes through feature decoupling and alignment, effectively resolving biases of local models towards local datasets. Overall, FedSeProto incorporates three key components: mutual information minimization to separate semantic and domain features, a KD-based component to ensure accurate representation of semantic and domain features, and the alignment of local semantic features with global prototypes, which is critical for all clients to learn global consistent features. The effectiveness of FedSeProto has been thoroughly validated across four challenging multi-domain datasets. In future work, we will explore further the potential applications of semantic prototype learning in model heterogeneity scenarios.

Acknowledgements

The research is supported by the National Key Research and Development Program of China (2023YFB2703700), the National Natural Science Foundation of China (62176269), the Guangzhou Science and Technology Program (2023A04J0314).

References

[1] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

[2] S. Alam, L. Liu, M. Yan, and M. Zhang. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. *Advances in neural information processing systems*, 35:29677–29690, 2022.

[3] H. Bai, R. Sun, L. Hong, F. Zhou, N. Ye, H.-J. Ye, S.-H. G. Chan, and Z. Li. Decaug: Out-of-distribution generalization via decomposed feature representation and semantic augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6705–6713, 2021.

[4] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540. PMLR, 2018.

[5] C. Chen, T. Liao, X. Deng, Z. Wu, S. Huang, and Z. Zheng. Advances in robust federated learning: Heterogeneity considerations. *arXiv preprint arXiv:2405.09839*, 2024.

[6] P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin. Club: A contrastive log-ratio upper bound of mutual information. In *International conference on machine learning*, pages 1779–1788. PMLR, 2020.

[7] E. Diao, J. Ding, and V. Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. *arXiv preprint arXiv:2010.01264*, 2020.

[8] C. Fang, Y. Xu, and D. N. Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664, 2013.

[9] C.-M. Feng, K. Yu, N. Liu, X. Xu, S. Khan, and W. Zuo. Towards instance-adaptive inference for federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23287–23296, 2023.

[10] L. Gao, H. Fu, L. Li, Y. Chen, M. Xu, and C.-Z. Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10112–10121, 2022.

[11] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012.

[12] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007.

[13] S. Guo, H. Wang, and X. Geng. Dynamic heterogeneous federated learning with multi-level prototypes. *Pattern Recognition*, 153:110542, 2024.

[14] W. Huang, M. Ye, B. Du, and X. Gao. Few-shot model agnostic federated learning. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 7309–7316, 2022.

[15] W. Huang, M. Ye, Z. Shi, H. Li, and B. Du. Rethinking federated learning with domain shift: A prototype view. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16312–16322. IEEE, 2023.

[16] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[17] H. Hwang, G.-H. Kim, S. Hong, and K.-E. Kim. Variational interaction information maximization for cross-domain disentanglement. *Advances in Neural Information Processing Systems*, 33:22479–22491, 2020.

[18] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[19] Y. Lai, L. Fu, T. Liao, C. Chen, and Z. Zheng. Supplementary Material for "FedSeProto: Learning Semantic Prototype in Federated Learning", Aug. 2024. URL https://doi.org/10.5281/zenodo.13312814.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324, 1998.

[21] D. Li and J. Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

[22] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[23] Q. Li, B. He, and D. Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021.

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

[25] Z. Li, X. Shang, R. He, T. Lin, and C. Wu. No fear of classifier biases: Neural collapse inspired federated learning with synthetic and fixed classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5319–5329, 2023.

[26] Y. Long, Z. Xue, L. Chu, T. Zhang, J. Wu, Y. Zang, and J. Du. Fedcd: A classifier debiased federated learning framework for non-iid data. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8994–9002, 2023.

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

[28] X. Mu, Y. Shen, K. Cheng, X. Geng, J. Fu, T. Zhang, and Z. Zhang. Fedproc: Prototypical contrastive federated learning on non-iid data. *Future Generation Computer Systems*, 143:93–104, 2023.

[29] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 7. Granada, Spain, 2011.

[30] A. T. Nguyen, P. Torr, and S. N. Lim. Fedsr: A simple and effective domain generalization method for federated learning. *Advances in Neural Information Processing Systems*, 35:38831–38843, 2022.

[31] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[32] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8432–8440, 2022.

[33] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

[34] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

[35] W. Xu, Y. Xian, J. Wang, B. Schiele, and Z. Akata. Attribute prototype network for zero-shot learning. *Advances in Neural Information Processing Systems*, 33:21969–21980, 2020.

[36] H.-M. Yang, X.-Y. Zhang, F. Yin, and C.-L. Liu. Robust classification with convolutional prototype learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3474–3482, 2018.

[37] L. Yi, G. Wang, X. Liu, Z. Shi, and H. Yu. Fedgh: Heterogeneous federated learning with generalized global header. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8686–8696, 2023.

[38] L. Zhang, L. Fu, T. Wang, C. Chen, and C. Zhang. Mutual information-driven multi-view clustering. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3268–3277, 2023.

[39] L. Zhang, L. Fu, C. Liu, Z. Yang, J. Yang, Z. Zheng, and C. Chen. Towards few-label vertical federated learning. *ACM Transactions on Knowledge Discovery from Data*, 2024.

[40] R. Zhang, Q. Xu, J. Yao, Y. Zhang, Q. Tian, and Y. Wang. Federated domain generalization with generalization adjustment. pages 3954–3963, 2023.

[41] L. Zhou, N. Li, M. Ye, X. Zhu, and S. Tang. Source-free domain adaptation with class prototype discovery. *Pattern recognition*, 145:109974, 2024.

[42] F. Zhu, X.-Y. Zhang, C. Wang, F. Yin, and C.-L. Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, 2021.

[43] K. Zhu, Y. Cao, W. Zhai, J. Cheng, and Z.-J. Zha. Self-promoted prototype refinement for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6801–6810, 2021.