

Take Your Pick: Enabling Effective Personalized Federated Learning within Low-dimensional Feature Space

Guogang Zhu[✉], Xuefeng Liu[✉], Shaojie Tang, Jianwei Niu[✉], *Senior Member, IEEE*, Xinghao Wu[✉], and Jiaxing Shen[✉]

Abstract—Personalized federated learning (PFL) is a popular framework that allows clients to have different models to address application scenarios where clients’ data are in different domains. The typical model of a client in PFL features a global encoder trained by all clients to extract universal features from the raw data and personalized layers (e.g., a classifier) trained using the client’s local data. Nonetheless, due to the differences between the data distributions of different clients (aka, domain gaps), the universal features produced by the global encoder largely encompass numerous components irrelevant to a certain client’s local task. Some recent PFL methods address the above problem by personalizing specific parameters within the encoder. However, these methods encounter substantial challenges attributed to the high dimensionality and non-linearity of neural network parameter space. In contrast, the feature space exhibits a lower dimensionality, providing greater intuitiveness and interpretability as compared to the parameter space. To this end, we propose a novel PFL framework named FedPick. FedPick achieves PFL in the low-dimensional feature space by selecting task-relevant features adaptively for each client from the features generated by the global encoder based on its local data distribution. It presents a more accessible and interpretable implementation of PFL compared to those methods working in the parameter space. Extensive experimental results show that FedPick could effectively select task-relevant features for each client and improve model performance in cross-domain FL.

Index Terms—Personalized Federated Learning, Feature Selection, Low-dimensional Feature Space.

I. INTRODUCTION

IN recent years, the rapid progress of big data has significantly accelerated the unprecedented growth of deep learning. Nonetheless, in real-world scenarios, the data typically originate from geographically dispersed clients, such as mobile phones or wireless sensors [1], [2]. Due to concerns regarding privacy or communication limitations, centralizing these scattered data for model training is commonly infeasible. To address the above challenges, federated learning (FL) emerges as a promising solution that enables multiple clients to collaboratively train the model without sharing their raw

data. Currently, FL has shown broad prospects for applications in various fields such as mobile edge computing [3], [4], healthcare [5], [6], [7], and finance [8], [9]. However, in practical applications, the data distributions across clients are commonly heterogeneous, which brings significant performance degradation to the FL model [10], [11].

Cross-domain FL, where the raw data on different clients come from various domains, is a prevalent source of statistical heterogeneity that appears in practical FL applications. In cross-domain FL, the raw data on different clients are distributed across various spaces, namely $\mathcal{X}_1 \neq \mathcal{X}_2 \neq \dots \neq \mathcal{X}_N$. However, it is assumed that the labels on different clients share the same space, namely $\mathcal{Y}_1 = \mathcal{Y}_2 = \dots = \mathcal{Y}_N$. For instance, when multiple hospitals collaborate to train a deep model for detecting pneumonia (e.g., COVID-19) [12], the diagnostic images (e.g., CT or MRI) from these hospitals can exhibit significant variations due to differences in sensor parameters, scanning protocols, and subject populations [13], [14], [15]. Another cross-domain example is autonomous driving [16], [17], where the data acquired from different automobiles encompass a wide range of weather conditions, lighting conditions, and geographical locations. The above cross-domain cases introduce domain gaps across clients, leading to a so-called feature shift [18] in the feature space. The feature shift can subsequently degrade the model performance of standard FL methods, such as FedAvg [19].

Personalized FL (PFL) [20], [21], [22] is a widely known means of mitigating the performance degradation in cross-domain FL. The fundamental concept behind PFL involves training a personalized model for each client that can adapt to this client’s own data distribution with the collaborative assistance of other clients. Currently, most PFL methods necessitate the sharing of a global encoder among all clients to extract high-level semantics from raw data, while personalizing other components of the model, such as the classifier [23], [24], to adapt the models to diverse domains. It is worth noting that the concept of employing a shared encoder across diverse domains stems from centralized cross-domain learning [25]. The rationale behind this approach lies in the belief that the shallow parameters in the encoder are less sensitive to data heterogeneity and thus can be applied to various domains [25].

However, due to the domain gaps across clients in cross-domain FL, the global encoder tends to extract universal features that are applicable for different domains simultane-

G. Zhu, X. Liu, J. Niu, and X. Wu are with the School of Computer Science and Engineering, Beihang University, China. E-mail: buaa_zgg@buaa.edu.cn, liu_xuefeng@buaa.edu.cn, niujianwei@buaa.edu.cn, wuxinghao@buaa.edu.cn.

S. Tang is with the Naveen Jindal School of Management, The University of Texas at Dallas. E-mail: tangshaojie@gmail.com.

J. Shen is with the Department of Computing and Decision Sciences, Lingnan University. E-mail: jiaxingshen@LN.edu.hk.

Corresponding author: Xuefeng Liu.

ously. These universal features often encompass numerous components that are irrelevant to the local task of a certain client, thereby potentially impairing the performance of the FL model. Although some recent PFL methods can tackle the above issue by personalizing specific parameters within the encoder [18], [26], [27], [28], it is important to note that these methods are commonly conducted within the parameter space. Given the complex nature of neural networks, characterized by high dimensionality and non-linearity, identifying an appropriate personalization strategy within the parameter space often proves challenging. Therefore, we wonder whether it is possible to implement PFL in a low-dimensional feature space, which would offer a more straightforward and interpretable alternative to PFL methods conducted in the parameter space.

We conduct several experiment to answer the above question. In these experiments, we employ the Fisher Score [29] to assess the importance of the features. The Fisher Score, which captures both intra-class consistency and inter-class discrimination, assigns higher scores to components of greater importance in features. We first arrange the features in descending order based on their Fisher Score, and then select a predetermined proportion of top-ranked features to retrain a classifier to probe the quality of features [30], [31], [32]. Fig. 1 illustrates the test accuracy on several commonly used cross-domain datasets using various feature subsets to retrain the classifier. The highest accuracy occurs when a feature subset with high Fisher Score is selected, surpassing even the accuracy obtained when utilizing all available features. The above results indicate that even using a simple metric to conduct PFL in the low-dimensional feature space, it is still feasible to achieve relatively good performance.

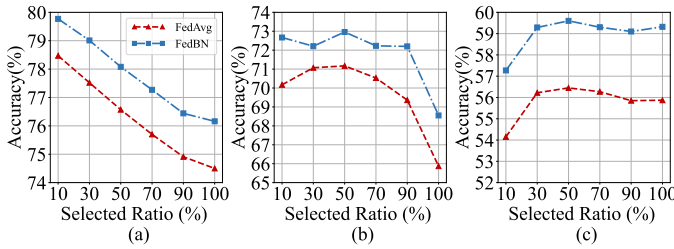


Fig. 1. Test accuracy when different feature subsets are selected to retrain a classifier. (a) Digits-Five, (b) Office-Caltech-10, (c) DomainNet.

However, the above vanilla feature selection approach ignores the correlations among different dimensions of features. In this paper, we propose FedPick, an approach that can adaptively select a subset of task-relevant features for each client. FedPick evaluates the overall quality of the feature subset rather than individually assessing its components, thereby alleviating the drawback of the vanilla feature selection approach. In FedPick, the global encoder and classifier are retained to extract universal features from raw data. It incorporates a personalized feature selection module (PFSM) for each client, enabling the selection of task-relevant features from the universal features. Since feature selection is a discretized operation that cannot be directly optimized by commonly used gradient-based algorithms, PFSM leverages Gumbel-Sigmoid [33] reparameterization to make feature selection differen-

tiable. Consequently, PFSM can be optimized simultaneously with the backbone model in an end-to-end manner by local data on individual client. Once training is completed, the PFSM can directly output the subset of task-relevant features given the universal features produced by the global encoder.

We conduct comprehensive experiments on multiple commonly used cross domain datasets. The experimental results demonstrate that FedPick can effectively select task-relevant features for each client and subsequently enhance the performance of the FL model. Our contributions of this paper are summarized as follows:

- We unveil a critical limitation in cross-domain FL, wherein the features generated by the global encoder are frequently redundant and cannot be directly adapted to the local task due to the domain gaps across clients.
- We propose FedPick, a cross-domain FL method that empowers individual clients to adaptively select task-relevant features based on their local data distribution, thereby enhancing the performance of the FL model.
- We conduct comprehensive experiments to validate the effectiveness of FedPick. The results demonstrate that FedPick significantly improves the model's performance in cross-domain FL scenarios.

II. RELATED WORK

Statistical Heterogeneity in Federated Learning. In recent years, FL [19], [10], [11] has emerged as a promising machine learning paradigm that enables model training without sharing the raw data on local clients. In a conventional setting of FL, there is a central server coordinates multiple distributed clients for model training. The training procedure involves iterative local training on the clients and model aggregation on the server. Nonetheless, practical applications often exhibit significant statistical heterogeneity across clients [10], [11]. Such statistical heterogeneity leads to divergence among locally trained models and subsequently hampers the performance of the aggregated FL model [34]. Statistical heterogeneity can manifest in various forms, with cross-domain FL [18] being one of the most common scenarios. Cross-domain FL refers to the situation where the distribution of raw input data varies across different clients, which is commonly observed in practical applications such as autonomous driving [16], [17], video surveillance [35], [36], [37], and medical imaging [13], [14], [15]. Nowadays, several methods are proposed to address the performance degradation in cross-domain FL [18], [28], [38], [39], [40]. Among the aforementioned studies, PFL has effectively showcased its capability to facilitate model adaptation to the local distribution. The subsequent paragraph provides a comprehensive description of PFL methods.

Personalized Federated Learning. The primary objective of PFL is to train personalized models for individual clients by leveraging the collaborative efforts of other clients, enabling them to better align with their respective local data distributions. FPE [41] directly utilizes local data to fine-tune the global model, thereby enhancing its ability to adapt to the local data distribution. Meta-learning is integrated into PFL to explore an effective initial model capable of achieving

high performance on local clients following a limited number of updates [42], [43]. Parameter decoupling enables PFL by separating personalized parameters from the global model. FedPer [23] and FedRep [24] both share shallow parameters (e.g., the encoder) while personalizing deep parameters (e.g., the classifier). Nevertheless, in these methods, the universal features produced by the global encoder commonly exhibit limited adaptability to local tasks. Therefore, recent studies try to personalize specific parameters within the encoder to extract features that align local data distribution. LG-FedAvg [26] employs a contrasting approach to FedPer and FedRep, which establishes local representations and a global head over them. FedBN [18] and SiloBN [27] address the domain shift in cross-domain FL by localizing BN layers while sharing other parameters. PartialFed [28] adaptively loads partial rather than entire global parameters at the initialization of local training in cross-domain FL. However, the aforementioned studies primarily concentrate on integrating global and local knowledge within the parameter space, where determining an appropriate personalization strategy becomes challenging due to the complexities of high dimensionality and non-linearity. In this paper, we realize PFL within the low-dimensional feature space, which offers the benefits of simplified implementation and enhanced interpretability compared with aforementioned PFL methods that operate in the parameter space.

III. PROBLEM FORMULATION OF CROSS-DOMAIN FL

In this section, we outline the problem formulation of cross-domain FL. The key notations utilized in this paper are listed in Table I.

FL commonly involves a central server that coordinates N distributed clients to perform model training without sharing their private data. Suppose that each client consists of M_i samples that are generated from \mathcal{D}_i , which are denoted as (x_i^j, y_i^j) , $j = 1, 2, \dots, M_i$. Specifically, $x_i^j \in \mathcal{X}_i \subseteq \mathbb{R}^n$ denotes the raw input, and $y_i^j \in \mathcal{Y}_i$ denotes the corresponding label. In cross-domain FL, the raw data on different clients come from various domains, that is, $\mathcal{X}_1 \neq \mathcal{X}_2 \neq \dots \neq \mathcal{X}_N$. However, the labels on different clients are assumed to be uniformly distributed in the same space, that is, $\mathcal{Y}_1 = \mathcal{Y}_2 = \dots = \mathcal{Y}_N$. Moreover, we only consider the classification task in this paper, so $\mathcal{Y}_1 = \mathcal{Y}_2 = \dots = \mathcal{Y}_N = \{1, 2, \dots, C\}$, where C is the total number of classes.

We follow the training paradigm of PFL, whose core idea is to train a personalized model θ_i for each client that can adapt to the client's data distribution, as shown below:

$$\argmin_{\theta_1, \theta_2, \dots, \theta_N} \frac{1}{N} \sum_{i=1}^N \mathcal{L}_i(\theta_i; \mathcal{D}_i), \quad (1)$$

where \mathcal{L}_i denotes the expected risk on client i . In practice, the expected risk \mathcal{L}_i is often inaccessible due to the unavailability of the underlying data distribution \mathcal{D}_i . Therefore, the empirical risk $\hat{\mathcal{L}}_i(\theta_i)$ on empirical data distribution $\hat{\mathcal{D}}_i$ is frequently employed as an approximation for the expected risk \mathcal{L}_i , which is formulated as follows:

$$\mathcal{L}_i(\theta_i) \approx \hat{\mathcal{L}}_i(\theta_i) = \frac{1}{M_i} \sum_{j=1}^{M_i} \ell(y_i^j, \hat{y}_i^j), \quad (2)$$

where $\hat{y}_i^j = f_i(x_i^j; \theta_i)$ is the predicted label, and $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ denotes the loss function that measures the prediction error.

Generally, θ_i can be decomposed into two parts: an encoder ϕ_i (typically composed of stacked convolutional layers) and a classifier h_i (typically composed of one or more fully connected layers). The encoder $\phi_i : \mathbb{R}^n \rightarrow \mathbb{R}^k$ maps raw inputs $\mathcal{X}_i \subseteq \mathbb{R}^n$ to a lower-dimensional feature space $\mathcal{Z}_i \subseteq \mathbb{R}^k$, which is denoted as $z_i = \phi_i(x_i)$ and typically $k \ll n$ in practice. The classifier $h_i : \mathbb{R}^k \rightarrow \mathcal{Y}$ gives the final prediction \hat{y}_i based on the feature z_i , which is denoted as $\hat{y}_i = h_i(z_i)$. As discussed in following subsections, we empirically observe that z_i is redundant for h_i to accomplish the local task in cross-domain FL. Specifically, some components in z_i are irrelevant or even harmful to the local task on each client. Therefore, we propose FedPick, whose main purpose is to select a subset of task-relevant features from z_i to adapt to the local data distribution of each client.

TABLE I
LIST OF KEY NOTATIONS.

Symbol	Description
Federated Learning System	
N	number of clients participating the FL training
M_i	number of samples on client i
C	total number of classes of samples
(x_i^j, y_i^j)	the j_{th} training sample on client i
Model Architecture	
θ_i	personalized model on client i
ϕ^g	global encoder
$h^g / h_i^p / h_i^u$	classifier for global / task-relevant / irrelevant features
Feature Selection	
S_b / S_w	inter-class / intra-class variance of features
G', G''	noise for Gumbel sampling
z_i^l	unbounded logits
$z_i^g / z_i^p / z_i^u$	global / task-relevant / irrelevant features
m_i^s / m_i	soft / hard feature mask
$\hat{y}_i^g / \hat{y}_i^p / \hat{y}_i^u$	predictions of global / task-relevant / irrelevant features
Loss Function	
\mathcal{L}_{ent}	entropy of predictions from task irrelevant features
\mathcal{L}_{lce}	cross entropy of predictions from personalized features
\mathcal{L}_{dis}	distillation between personalized and global predictions
\mathcal{L}_{gce}	cross entropy of predictions from global features

IV. MOTIVATION OF FEATURE SELECTION IN CROSS-DOMAIN FL

In this section, we discuss the motivation of personalized feature selection in cross-domain FL. At first, we illustrate the observation that the universal features generated by the global encoder in FL often exhibit a higher degree of redundancy compared to the features generated from models trained through centralized learning (CL). Then we demonstrate that the performance of the model can be improved by selecting an appropriate subset of features tailored to the local task.

1) *Feature Redundancy*: We employ the sparsity ratio as quantitative metric to assess the feature redundancy. The sparsity ratio is defined as the percentage of components that

are closed to zero in the features. We apply a threshold value ε to each individual component of L_2 -normalized features to determine whether it is closed to zero. The formulation for computing the sparsity ratio $S(z, \varepsilon)$ is as follows:

$$S(z, \varepsilon) = \frac{\|M(\bar{z}, \varepsilon)\|_1}{|z|}, \quad (3)$$

$$M(\bar{z}, \varepsilon)[i] = \begin{cases} 1, & \text{if } \bar{z}^i \leq \varepsilon \\ 0, & \text{if } \bar{z}^i > \varepsilon, \end{cases} \quad (4)$$

where ε is the threshold value, \bar{z} is the L_2 -normalized features, $\|\cdot\|_1$ is the L_1 norm of vector, $|z|$ is the dimension of z .

We conduct experiments to measure the sparsity ratio of features generated by different methods on Digits-Five (a commonly used cross-domain dataset), including FedAvg [19], FedBN [18] and SingleSet (training an individual model for each domain). To mitigate mutual interference between multiple domains, we train a separate model for each domain (i.e., SingleSet) for CL training paradigm, instead of training a model using the data from all domains. The threshold value ε in Eq. (3) and Eq. (4) is set to 10^{-5} .

Fig. 2 (a) presents the average sparsity ratio of features generated by different methods on various domains in Digits-Five. As expected, the features exhibit increased sparsity as the training progresses. This observation can be attributed to the fact that the ground truth is represented by an one-hot vector, which inherently possesses extreme sparsity. Consequently, the features near the classifier should strive for maximum sparsity to minimize the training loss. Moreover, it can be observed that the sparsity ratio of FL methods is notably lower than that of SingleSet. These findings suggest that the features generated by FL methods exhibit greater redundancy compared to those produced by the CL method, which deviates from the training objective. One plausible explanation for this disparity is that FL methods aim to generate universal features that are versatile enough to be shared across various domains, whereas SingleSet models only need to meet the requirements of the local data distribution within that specific domain.

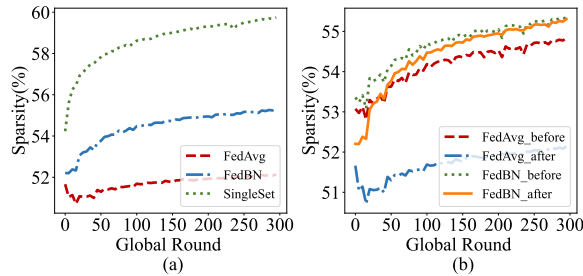


Fig. 2. Analysis of feature redundancy, (a) sparsity ratio of different methods, (b) sparsity ratio before and after model aggregation.

Additionally, we demonstrate that the increase in feature redundancy is a consequence of model aggregation. Fig. 2 (b) presents a comparative analysis of the sparsity ratios of features both before and after the model aggregation. The results clearly indicate a substantial rise in redundancy among the features generated by the model after aggregation, in contrast to those originating from the models prior to aggregation.

2) *Model Performance with Different Feature Subsets:* Inspired by the above observation, we posit that improving feature redundancy can enhance the generalization ability of the learned model to all clients. Nonetheless, this enhancement comes at the cost of diminishing the model's generalization ability when confronted with the local distribution of individual clients, i.e., the personalization ability. Therefore, we pose the following question: **Can model performance be improved by selecting an approximated feature subset from the universal features?** To answer the above question, we utilize Fisher Score [44] to assess the feature importance and perform a vanilla feature selection approach, as shown in Fig. 3. Fisher Score is a widely known metric that can evaluate the importance of an individual component in features, which is defined as follows:

$$F_i = \frac{S_b}{S_w}, \quad (5)$$

where S_b is the inter-class variance, S_w is intra-class variance. The inter-class variance S_b is defined as:

$$S_b = \sum_{j=1}^C m_j (\mu_{ij} - \mu_i)^2, \quad (6)$$

where m_j is the number of samples in class j , μ_i is the mean of feature z_i , μ_{ij} is the mean of feature z_i for samples in class j . The intra-class variance S_w is defined as:

$$S_w = \sum_{j=1}^C m_j \sigma_{ij}^2, \quad (7)$$

where m_j is the number of samples in class j , σ_{ij} is the variance of feature z_i for samples in class j . A higher Fisher Score is achieved when the component exhibits greater similarity within the same class and dissimilarity across different classes, which implies that this component is more discriminative and is more relevant to the local task.

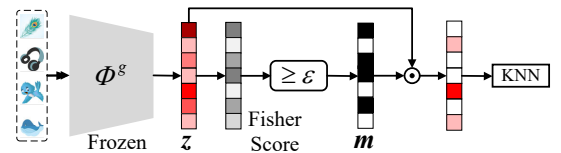


Fig. 3. Procedure of vanilla feature selection.

In the vanilla feature selection, we first pre-train a model using existed FL methods until it converges. Then we calculate the Fisher Score of features on the training dataset and constitute the feature mask m based on the Fisher Score. m is a binary vector, wherein positions exceeding a specified threshold based on the Fisher Score are assigned a value of 1, while the remaining positions are assigned a value of 0. In the experiments, the threshold is defined as the percentile obtained from the sorted Fisher Score vector based on the selected ratio. At last, we freeze the encoder and implement a probe [30], [31], [32] on the masked features, denoted as $z \odot m$, to quantitatively assess their quality. Specifically, we discard the previously trained classifier and employ the masked features to retrain a new classifier.

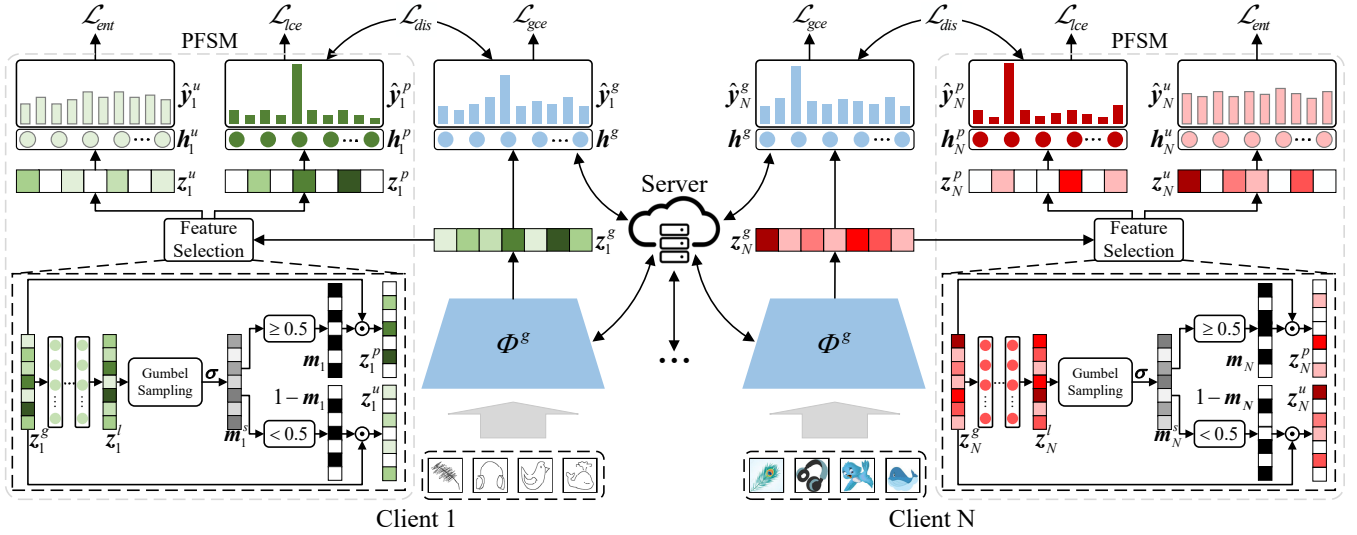


Fig. 4. Framework of FedPick. It mainly consists of the following steps: (1) The raw data are passed through a global encoder to generate universal features. (2) The universal features are subsequently fed into a PFSM to select personalized task-relevant features. (3) Both the universal features and the personalized features are then utilized as inputs for a global classifier and a personalized classifier, respectively, to generate the prediction.

We employ FedAvg and FedBN to pre-train the FL models. During the feature selection, the component indexes in features are sorted in descending order based on the Fisher Score. The selection ratio ranges from 10% – 100%. We employ a K Nearest Neighbor (KNN) classifier (with $K = 10$) to probe the feature subsets. Fig. 1 presents the test accuracy achieved with different feature subsets. It can be observed that by adaptively choosing task-relevant features (those with high Fisher Scores) for the downstream task, the model can outperform the performance achieved using all available features.

However, the aforementioned vanilla feature selection method encounters several challenges. Firstly, although the Fisher Score has been proved to be effective for the classification task, designing a practical and applicable feature evaluation metric remains a challenge in real-world scenarios. Secondly, the above approach evaluates features independently, thereby overlooking the interrelationships among different dimensions in the feature. Lastly, determining the optimal proportion of selected components poses a challenging problem. To address these issues, we further propose a novel approach called FedPick. FedPick facilitates the automatic selection of task-relevant features based on the local data distribution of each client. By leveraging this method, we aim to overcome the limitations of vanilla feature selection method and enhance the performance of FL models.

V. FRAMEWORK OF FEDPICK

In this section, we first provide an overview of FedPick. Second, we introduce personalized feature selection module (PFSM), the core part of FedPick. Then, we discuss the knowledge transfer mechanism between global and personalized features. At last, we provide the training procedure of FedPick.

A. Framework Overview

The framework of FedPick is depicted in Fig. 4, which mainly consists of three steps. First, the raw data are passed

through a global encoder to generate universal features. Then, the universal features are fed into a PFSM for feature selection, separating the features to a task-relevant feature subset and a task-irrelevant feature subset. At last, the universal features and personalized task-relevant features (for brevity, we sometimes refer to these features as ‘personalized features’ or ‘task-relevant features’ in the subsequent sections) are passed into a global and a personalized classifier for prediction, respectively. During inference, the global and personalized predictions are integrated to derive the final prediction. For simplification, we omit the index of clients and samples in this section.

B. Personalized Feature Selection Module

In FedPick, the encoder ϕ^g is shared across multiple clients to extract universal features. As previously mentioned, the universal features generated by the global encoder exhibit a higher level of generalization for a wider distribution, offering potential advantages in subsequent feature selection processes. To preserve the universality of global features, a global classifier h^g (also shared across clients) is maintained within FedPick. The features generated by the global encoder are denoted as z^g , and the corresponding predictions are represented as $\hat{y}^g = h^g(z^g)$. The features generated by the global encoder are required to possess sufficient discriminative power to successfully carry out the classification task. This objective is achieved by cross entropy loss, as demonstrated in Eq. (8).

$$L_{gce} = \sum_{c=1}^C y_c \log(\hat{y}_c^g) \quad (8)$$

After generating universal features from the global encoder, FedPick introduces a PFSM for each client to adaptively select each client’s task-relevant features based on its local data distribution. In PFSM, the global features z^g are first fed into a FC network, denoted as ϕ , to generate the unbounded logits z^l . Subsequently, the Gumbel-Sigmoid reparameterization

technique, as described in [29], is employed to generate a soft mask denoted as \mathbf{m}^s . Specifically, the calculation of the i_{th} component's corresponding mask z^l is as follows:

$$m_i^s = \sigma((z_i^l + G' - G'')/\tau) = \frac{e^{(z_i^l + G')/\tau}}{e^{(z_i^l + G')/\tau} + e^{G''/\tau}}, \quad (9)$$

where G' and G'' are two independent Gumbel noises sampled from uniform distribution $U[0, 1]$, $\tau \in [0, +\infty]$ is the temperature scale that controls the distribution tendency of sampling, $\sigma(\cdot)$ is the sigmoid function. It should note that the noises G' and G'' are activated during training to facilitate the exploration of various feature masks. However, during inference, these noises are deactivated to ensure consistent and reliable results. Similar to vanilla feature selection approach, the soft mask \mathbf{m}^s is then discretized into a binary vector \mathbf{m} by a threshold ε , that is, m_i is set to 1 if $m_i^s \geq \varepsilon$ else 0, as depicted in Eq. (4). In FedPick, unless otherwise specified, the value of ε is set to 0.5.

However, the previously mentioned hard masking operation is discrete, which cannot be directly optimized by commonly used gradient-based algorithms. To make the hard masking differentiable, FedPick adopts sigmoid during the backward process and hard masking during the forward process [45]. This design enables simultaneous optimization of PFSM with the backbone model using the training data and can be easily implemented within popular deep learning frameworks such as PyTorch [46].

PFSM outputs task-relevant features \mathbf{z}^p and task-irrelevant features \mathbf{z}^u by Hadamard product between the \mathbf{z}^g and \mathbf{m} , $1 - \mathbf{m}$, respectively, as shown in Eq. (10).

$$\mathbf{z}^p = \mathbf{z}^g \odot \mathbf{m}, \quad \mathbf{z}^u = \mathbf{z}^g \odot (1 - \mathbf{m}). \quad (10)$$

Similar to \mathbf{z}^g , the task-relevant features \mathbf{z}^p also shall be discriminative enough to accomplish the local task. Therefore, PFSM implements a personalized classifier \mathbf{h}^p for \mathbf{z}^p and enforces it to accomplish the classification task by minimizing the cross entropy loss, as shown in Eq. (11).

$$L_{lce} = \sum_{c=1}^C y_c \log(\hat{y}_c^p) \quad (11)$$

In contrast, the task-irrelevant features \mathbf{z}^u are expected to exhibit lower levels of discriminative capability towards the local task. As a result, the PFSM employs a personalized classifier \mathbf{h}^u to \mathbf{z}^u and encourages its prediction to be uncertain by maximizing its prediction, as illustrated in Eq. (12).

$$L_{ent} = \sum_{c=1}^C \hat{y}_c^u \log(\hat{y}_c^u) \quad (12)$$

C. Transferring Global and Personalized Knowledge

In FedPick, the global features and personalized features offer distinct benefits. Global features possess a higher level of generalization on different data distribution, allowing for their transferability across various distributions. However, this high transferability may result in performance degradation when dealing with local data distributions on individual clients. Conversely, personalized features are tailored to local distributions,

thereby enhancing performance for specific distributions on local clients (personalization). Nonetheless, these features may lack generalization capabilities when confronted with unseen data distributions, such as test datasets. Hence, it is preferable to combine the advantages of both global and personalized features to enhance the overall model performance. Motivated by knowledge distillation [47], FedPick incorporates cyclic distillation between the predictions from global and personalized features to facilitate mutual knowledge transfer. The cyclic distillation of global and personalized features ensures the balance between the model's generalization and personalization abilities across different data distributions. This cyclic distillation is achieved by minimizing the loss function presented in Eq. (13), where $KL(\cdot)$ denotes the Kullback-Leibler (KL) divergence, which quantifies the dissimilarity between two distributions.

$$L_{dis} = KL(\hat{\mathbf{y}}^p || \hat{\mathbf{y}}^g) + KL(\hat{\mathbf{y}}^g || \hat{\mathbf{y}}^p) \quad (13)$$

To leverage the comprehensive knowledge provided by both global and personalized features, FedPick combines the predictions of the global and personalized classifiers through an ensemble approach, resulting in a more robust prediction. This ensemble process is illustrated in Eq. (14).

$$\hat{\mathbf{y}} = \text{softmax}(\hat{\mathbf{y}}^g + \hat{\mathbf{y}}^p) \quad (14)$$

D. Training Procedure of FedPick

The total loss adopted by FedPick is shown in Eq. (15), where λ_{lce} , λ_{ent} , and λ_{dis} are hyper-parameters to adjust the effect of different loss terms. During the training, all parameters in the model are optimized by the loss in Eq. (15) simultaneously.

$$L_{total} = L_{gce} + \lambda_{lce} L_{lce} + \lambda_{ent} L_{ent} + \lambda_{dis} L_{dis} \quad (15)$$

After training, the global encoder and classifier are uploaded to the server for aggregation, as shown in Eq. (16), where ϕ_i^g and \mathbf{h}_i^g denote the global encoder and classifier updated on client i , $\tilde{\phi}^g$ and $\tilde{\mathbf{h}}^g$ the aggregated global encoder and classifier, respectively. $\tilde{\phi}^g$ and $\tilde{\mathbf{h}}^g$ are then broadcast to each client for next round of local updates. The parameters in PFSM are localized on each client to select its own task-relevant features. Moreover, motivated by previous studies in [18], [27], FedPick also keeps BN layers personalized on each client to avoid the performance degradation caused by aggregating them.

$$\tilde{\mathbf{h}}^g = \sum_{i=1}^N \frac{M_i}{\sum_{j=1}^N M_j} \mathbf{h}_i^g, \quad \tilde{\phi}^g = \sum_{i=1}^N \frac{M_i}{\sum_{j=1}^N M_j} \phi_i^g. \quad (16)$$

The entire process of FedPick is summarized in Algorithm 1. For brevity, we denote the parameters in PFSM as ψ_i , that is, $\psi_i \equiv \{\mathbf{h}_i^p, \mathbf{h}_i^u, \varphi_i\}$.

VI. THEORETICAL FOUNDATION OF FEDPICK

FedPick selects appropriate feature subset relevant to the local task from the global features, thereby enhancing their compatibility with the local distribution on each client. Such

Algorithm 1 FedPick

Notations: T : global update rounds, E : local update epochs, B : local minibatch size, η : learning rate, λ_{lce} , λ_{ent} , and λ_{dis} : hyperparameters used to balance loss terms.

Sever Executes:

- 1: initialize and broadcast $\phi^{g,1}, h^{g,1}, \psi_1^1, \dots, \psi_N^1$ to clients
- 2: **for** $t = 1, 2, 3, \dots, T$ **do**
- 3: **for** each client i **in parallel do**
- 4: $\phi_i^{g,t+1}, h_i^{g,t+1} \leftarrow \text{ClientUpdate}(i, t, \tilde{\phi}^{g,t}, \tilde{h}^{g,t}, \psi_i^t)$
- 5: **end for**
- 6: $\tilde{h}^{g,t+1} = \sum_{i=1}^N \frac{M_i}{\sum_{j=1}^N M_j} h_i^{g,t}$
- 7: $\tilde{\phi}^{g,t+1} = \sum_{i=1}^N \frac{M_i}{\sum_{j=1}^N M_j} \phi_i^{g,t}$
- 8: broadcast $\tilde{\phi}^{g,t+1}, \tilde{h}^{g,t+1}$ to clients
- 9: **end for**
- 10: **ClientUpdate**(i, t, ϕ, h, ψ):
- 11: $\mathcal{B} \leftarrow$ (split local dataset into batches of size B)
- 12: **for** $j = 1, 2, 3, \dots, E$ **do**
- 13: **for** batch $b \in \mathcal{B}$ **do**
- 14: $z^g = \phi(b), \{z^p, z^u\} = \psi(z^g)$
- 15: $\hat{y}^g = h^g(z^g), \hat{y}^p = h^p(z^p), \hat{y}^u = h^u(z^u)$
- 16: $L_{total} = L_{gce}(\hat{y}^g) + \lambda_{lce} L_{lce}(\hat{y}^p)$
 $\quad + \lambda_{ent} L_{ent}(\hat{y}^u) + \lambda_{dis} L_{dis}(\hat{y}^p, \hat{y}^u)$
- 17: $\{\phi, h, \psi\} \leftarrow \{\phi, h, \psi\} - \eta \nabla_{\{\phi, h, \psi\}} L_{total}$
- 18: **end for**
- 19: $\psi_i^{t+1} \leftarrow \psi$
- 20: **return** ϕ, h to server

a feature selection operation can be theoretically supported by Vapnik-Chervonenkis (VC) theory [48]. The VC dimension serves as a quantifiable measure of the capacity of a hypothesis class, which represents the set of possible functions that a learning algorithm can learn. Specifically, it establishes an upper bound on the number of training samples that can be perfectly classified by the hypothesis class. When a hypothesis class exhibits a high VC dimension, it indicates a larger capacity, enabling the class to potentially capture a wide range of complex patterns present in the training data.

As described in Section III, each client's model consists of an encoder and a classifier. The encoder $\phi_i : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is responsible for mapping the raw inputs $X_i \subseteq \mathbb{R}^n$ to a lower-dimensional feature space $\mathcal{Z}_i \subseteq \mathbb{R}^k$, denoted as $z_i = \phi_i(x_i)$. The classifier $h_i : \mathbb{R}^k \rightarrow \mathcal{Y}$ generates the final prediction \hat{y}_i based on the extracted feature z_i , expressed as $\hat{y}_i = h_i(z_i)$. The classifier h_i is sampled from the hypothesis space \mathcal{H} . When considering only the stage after feature extraction, the optimization objective of the FL system can be reformulated as follows:

$$\argmin_{h_1, h_2, \dots, h_N} \frac{1}{N} \sum_{i=1}^N \mathcal{L}_i(h_i; \phi_i(\mathcal{D}_i)), \quad (17)$$

For client i , it is proved by VC dimension theory that with a probability of at least $1 - \delta$, the expected risk $\mathcal{L}_i(h_i)$ is upper

bounded by:

$$\mathcal{L}_i(h_i) \leq \hat{\mathcal{L}}_i(h_i) + \sqrt{\frac{8\gamma \ln(2M_i) + 8 \ln \frac{4}{\delta}}{M_i}}, \quad (18)$$

where $\hat{\mathcal{L}}_i(h_i)$ is the empirical risk, M_i is the number of training samples on client i , h_i is the classification model distributed in hypothesis space \mathcal{H} , γ is the VC dimension of \mathcal{H} .

Denoting the global data distribution as $\mathcal{D}_g = \sum_{i=1}^N \lambda_i \mathcal{D}_i$, where $\lambda_i = \frac{M_i}{M}$ and $M = \sum_{i=1}^N M_i$. We use the notation $d_{\mathcal{H}}(\mathcal{D}_1, \mathcal{D}_2)$ to represent the divergence between two distributions. Based on the global model generalization proposed by previous studies [49], [50], with a probability of at least $1 - \delta$, the risk of client i in the FL system is bounded by:

$$\mathcal{L}_i(h_i) \leq \hat{\mathcal{L}}_i(h_g) + \sqrt{\frac{8\gamma \ln(2M) + 8 \ln \frac{4}{\delta}}{M}} + d_{\mathcal{H}}(\mathcal{D}_i, \mathcal{D}_g), \quad (19)$$

where h_g denotes the global classifier obtained from \mathcal{D}_g .

Combining the global optimization objective in Eq. (17) with Eq. (19) yields the generalization bound of FL system as:

$$\begin{aligned} \frac{1}{N} \sum_{i=1}^N \argmin_{h_i \in \mathcal{H}} \mathcal{L}_i(h_i) &\leq \frac{1}{N} \sum_{i=1}^N \hat{\mathcal{L}}_i(h_g) + \frac{1}{N} \sum_{i=1}^N d_{\mathcal{H}}(\mathcal{D}_i, \mathcal{D}_g) \\ &\quad + \sqrt{\frac{8\gamma \ln(2M) + 8 \ln \frac{4}{\delta}}{M}}. \end{aligned} \quad (20)$$

To enhance the generalization of the model, the primary objective is to reduce the upper bound, which can be achieved by increasing the number of training samples. By comparing Eq. (18) and Eq. (20), it can be observed that clients can augment the quantity of training samples by FL, thereby bolstering the generalization capability of the local model in contrast to local training. Another approach to tightening the generalization bound is to reduce the VC dimension γ . For a linear classifier, γ is upper bounded by the dimension of the features [51]. Consequently, it is viable to further enhance the generalization capability of the FL model to local distributions (i.e., personalization) by selecting task-relevant features from z_i to construct sparse features.

VII. EXPERIMENT

To demonstrate the effectiveness of FedPick, we conduct experiments on several cross-domain datasets and compare its results with those of several benchmark methods. The experimental details are discussed as follows.

A. Dataset Description

Experiments are conducted on three commonly used cross-domain datasets: Digits-Five, Office-Caltech-10 and Domain-Net. These datasets are all used for classification tasks. Fig. 5 presents some example images in these datasets. The specific details of each dataset are provided below.

Digits-Five. The Digits-Five dataset contains five datasets for handwritten character recognition with different background, namely MNIST-M (MM) [52], MNIST (MT) [53], USPS (UP) [54], SynthDigits (SY) [52], and SVHN (SV) [55]. Each dataset consists of images from a single domain, categorized into 10 classes. To construct the training dataset for each client, we sample 1000 images from each dataset, resulting in a total of five clients participating in the training process. All images from the test datasets are utilized for evaluating the model. Prior to being fed into the model, all images are converted into RGB images of size 32×32 .

Office-Caltech-10. The Office-Caltech-10 [56] dataset is composed of images captured by cameras with diverse imaging parameters. It is categorized into four domains: Amazon (A), Caltech (C), DSLR (D), and Webcam (W). Each domain encompasses 10 classes of images. For our experiments, we select 125 training images from each domain and assign them to a single client. The test dataset is exclusively reserved for evaluation purposes. To standardize the input, the images are transformed into RGB format with dimensions of 256×256 pixels. Additionally, random flipping and rotation techniques are applied to augment the dataset.

DomainNet. The DomainNet [57] dataset contains six domains: Clipart (C), Infograph (I), Painting (P), Quickdraw (Q), Real (R), and Sketch (S). These domains consist of images with various artistic styles, such as painting and sketching. Originally, each domain contains 345 classes. But for our experiments, we select 10 commonly used classes to construct our dataset. For each domain, we sample 500 training images to create the training dataset for a single client. Similar to the Office-Caltech-10 dataset, all testing images are reserved exclusively for evaluation. The images are converted into 256×256 RGB images and augmented by randomly flipping and rotating before being fed into the model.

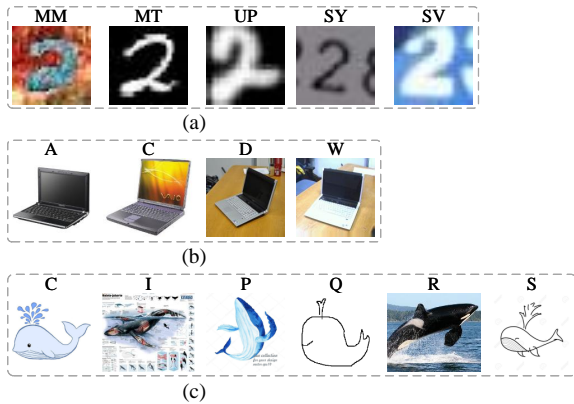


Fig. 5. Example images in different cross-domain datasets, (a) Digits-Five, (b) Office-Caltech-10, (c) DomainNet.

B. Compared Methods

To evaluate the effectiveness of FedPick, we perform a comparative analysis against the following methods:

SingleSet: This method trains and tests independent models for each client, utilizing only its own data. Despite not

involving collaboration with other clients, it serves as a robust benchmark, especially as the quantity of local data increases.

FedAvg: FedAvg is a classical FL method that aggregates all local model parameters after several rounds of local updates on clients [19].

FedProx: FedProx introduces a regularization term between the local and global models, which constrains the direction of local updates [58].

FedPer: This method utilizes a personalized classifier for each client, allowing adaptation of client-specific features extracted from the shared encoder [23].

FedRep: Similar to FedPer, FedRep employs personalized classifiers for each client to adapt their features from a shared encoder [24]. However, it distinguishes itself by performing multiple local updates with respect to the personalized classifier for every update of the global encoder.

LG-FedAvg: In contrast to FedPer and FedRep, LG-FedAvg maintains personalized encoders on each client while sharing a global classifier across all clients [26].

FedBN: This method localizes BN layer parameters on each client while sharing the remaining parameters [18].

C. Implementation Details

For Digits-Five, we employ a CNN consisting of multiple convolutional and FC layers. For Office-Caltech-10 and DomainNet, we make use of a modified version of AlexNet [59]. This modification involves integrating a BN layer after each convolutional and FC layer.

The PFSM is plugged on the features generated by the global encoder. The feature dimension of Digits-Five is 8192, while the feature dimensions of Office-Caltech-10 and DomainNet are both 4096. Gumbel sampling is carried out using a compact FC network with the following architecture: $[\text{Linear}(d, \frac{d}{2}) - \text{ReLU} - \text{Linear}(d, \frac{d}{2}) - \text{Gumbel Sigmoid}]$, where d represents the feature dimension.

All experiments are implemented using the PyTorch [46] framework and executed on a four-card Nvidia V100 cluster. We use SGD with momentum to update the model. The learning rate and momentum are set to 0.01 and 0.5, respectively, in all experiments. Across all methods, a batch size of 64 is employed during the local updating process. The local epoch is set to 1 for all methods except for FedRep. In the case of FedRep, the local update consists of 5 epochs, where the first 4 epochs are dedicated to optimizing the classifier, and the final epoch focuses on optimizing the encoder. The total number of global communication rounds is set to 300. To mitigate the randomness of the experimental results, we repeat all experiments five times. The mean and standard deviation values of the best test accuracy during the FL training are presented in the following sections.

The hyperparameter μ adopted to balance the loss terms in FedProx is set to 0.01 in the experiments. For LG-FedAvg, a pre-training phase comprising 20 global rounds is performed by aggregating all parameters on Office-Caltech-10 and DomainNet. The hyperparameter combinations of FedPick are detailed in Table III.

TABLE II
TEST ACCURACY ON DIGITS-FIVE, OFFICE-CALTECH-10 AND DOMAINNET.

Method	MM	MT	UP	SY	SV	A	C	D	W	C	I	P	Q	R	S
SingleSet	82.45 (0.42)	95.89 (0.17)	97.94 (0.13)	84.21 (0.17)	70.63 (0.85)	71.35 (1.14)	47.91 (0.44)	98.75 (1.53)	93.22 (1.07)	64.83 (0.21)	35.28 (0.56)	53.86 (0.36)	82.44 (0.33)	69.20 (0.45)	56.39 (1.01)
FedAvg	80.93 (0.49)	96.88 (0.15)	97.37 (0.22)	83.98 (0.45)	73.19 (0.61)	62.92 (1.21)	53.69 (0.99)	82.50 (2.50)	94.58 (1.27)	71.06 (0.72)	33.24 (0.98)	56.16 (0.74)	68.24 (1.18)	63.78 (0.56)	60.76 (1.24)
FedProx	81.71 (0.20)	96.92 (0.24)	97.29 (0.33)	83.79 (0.22)	71.78 (0.79)	61.35 (1.79)	53.78 (0.74)	82.50 (3.19)	95.59 (1.34)	70.61 (0.80)	32.88 (0.10)	55.22 (1.25)	67.02 (0.96)	63.47 (0.52)	58.16 (1.28)
FedPer	82.21 (0.35)	96.87 (0.25)	97.63 (0.44)	84.26 (0.40)	70.57 (0.61)	63.65 (0.83)	54.31 (0.52)	83.75 (3.64)	94.58 (1.27)	68.90 (0.26)	35.34 (0.85)	54.25 (0.62)	75.28 (0.89)	66.77 (0.77)	54.84 (0.52)
FedRep	80.89 (0.67)	96.30 (0.21)	97.65 (0.18)	83.08 (0.52)	66.07 (1.48)	63.44 (2.24)	49.96 (1.81)	80.63 (5.00)	92.20 (0.83)	65.86 (0.98)	34.31 (0.81)	50.92 (0.40)	69.66 (2.08)	63.04 (0.74)	52.24 (1.13)
LG-FedAvg	82.42 (0.23)	95.87 (0.05)	97.78 (0.09)	84.34 (0.42)	70.84 (0.49)	71.88 (0.57)	52.36 (1.10)	96.88 (1.98)	97.97 (0.68)	69.77 (0.67)	35.83 (0.80)	59.39 (1.15)	82.42 (0.45)	73.05 (0.68)	61.77 (0.21)
FedBN	83.53 (0.26)	97.24 (0.08)	98.45 (0.09)	85.58 (0.31)	78.22 (0.30)	72.40 (0.87)	54.22 (1.41)	97.50 (1.25)	98.64 (0.68)	71.22 (1.00)	34.82 (0.42)	59.55 (1.02)	80.80 (0.39)	70.29 (0.56)	63.03 (0.63)
FedPick	89.60 (0.34)	97.81 (0.02)	98.81 (0.04)	91.34 (0.25)	82.87 (0.19)	74.06 (1.06)	57.51 (0.36)	100.00 (0.00)	98.64 (0.68)	74.07 (0.49)	37.05 (0.92)	63.26 (0.82)	81.76 (0.42)	74.38 (0.50)	66.75 (1.00)

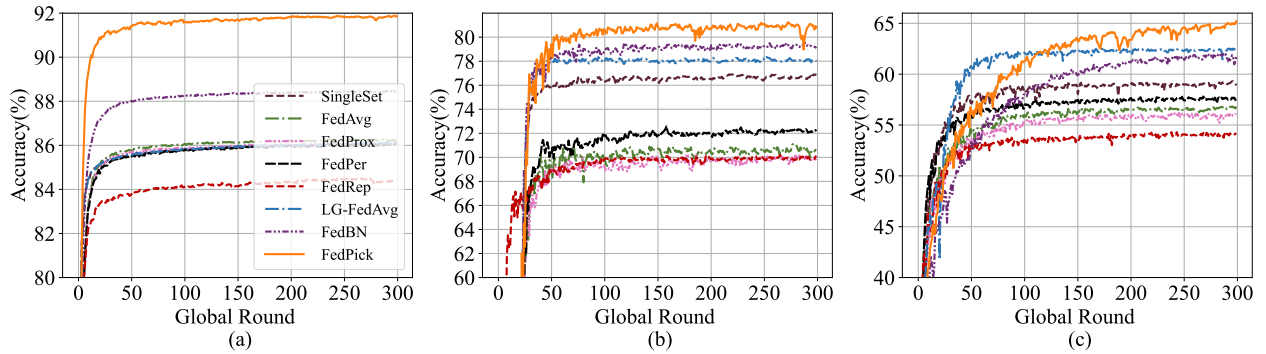


Fig. 6. Variations in test accuracy throughout the training process: (a) Digits-Five, (b) Office-Caltech-10, (c) DomainNet.

TABLE III
HYPERPARAMETERS OF FEDPICK.

	τ	λ_{lce}	λ_{ent}	λ_{dis}
Digits-Five	10.0	10.0	0.001	10.0
Office-Caltech-10	1.0	1.0	0.001	1.0
DomainNet	1.0	1.0	0.001	1.0

D. Experimental Results

Tables II presents the basic experimental results on Digits-Five, Office-Caltech-10, and DomainNet, respectively. These findings highlight that SingleSet, despite being trained on data from a single domain, performs remarkably well and serves as a strong benchmark method. However, traditional FL methods like FedAvg suffer from significant performance degradation due to the domain gaps between different clients, often failing to surpass the performance of SingleSet. On the other hand, FedPer, FedRep, LG-FedAvg, and FedBN achieve higher model accuracy by allowing the model to adapt to each domain through personalized parameters such as classifiers or BN layers in the encoder. Notably, our proposed method, FedPick, outperforms the compared methods on all datasets in most time, demonstrating its effectiveness in cross-domain FL. Fig. 6 presents the variations in test accuracy during the training. FedPick can converge to higher accuracy stably compared with other FL methods.

VIII. ADDITIONAL ANALYSIS

In this section, we provide additional analysis of FedPick. First, we conduct several ablation studies to demonstrate the efficacy of the components employed in FedPick. Second, we delve into the feature analysis for FedPick, highlighting the effectiveness of feature selection operation. At last, we discuss the experiments conducted on the hyper-parameters employed in FedPick.

A. Ablation Study

In this subsection, we conduct ablation studies to illustrate the efficacy of the components introduced in FedPick. The results of the ablation study are presented in Table IV. In the majority of cases, when the components specifically designed for FedPick are removed, there is a noticeable decline in model performance, thereby substantiating the effectiveness of these components. The detailed experimental results are discussed as follows.

Without PFSM. FedPick leverages PFSM to select task-relevant features from the universal features. When PFSM is excluded from the FedPick framework, FedPick essentially reduces to FedBN. From the results presented in Table IV, it is evident that the removal of PFSM leads to a substantial decline in the performance of FedPick. This observation underscores the critical role of personalized feature selection in enhancing the FL model performance.

TABLE IV
ABLATION STUDY RESULTS ON DIGITS-FIVE, OFFICE-CALTECH-10 AND DOMAINNET.

Setting	MM	MT	UP	SY	SV	A	C	D	W	C	I	P	Q	R	S
FedPick	89.60 (0.34)	97.81 (0.02)	98.81 (0.04)	91.34 (0.25)	82.87 (0.19)	74.06 (1.06)	57.51 (0.36)	100.00 (0.00)	98.64 (0.68)	74.07 (0.49)	37.05 (0.92)	63.26 (0.82)	81.76 (0.42)	74.38 (0.50)	66.75 (1.00)
w/o PFSM	83.53 (0.26)	97.24 (0.08)	98.45 (0.09)	85.58 (0.31)	78.22 (0.30)	72.40 (0.87)	54.22 (1.41)	97.50 (1.25)	98.64 (0.68)	71.22 (1.00)	34.82 (0.42)	59.55 (1.02)	80.80 (0.39)	70.29 (0.56)	63.03 (0.63)
w/o L_{lce}	88.76 (0.14)	97.61 (0.05)	98.81 (0.09)	89.55 (0.05)	81.24 (0.25)	74.06 (1.29)	57.07 (1.37)	99.38 (1.25)	98.31 (0.00)	71.33 (1.14)	36.56 (0.38)	62.10 (0.63)	81.08 (0.30)	72.49 (0.90)	64.19 (1.25)
w/o L_{ent}	89.55 (0.22)	97.79 (0.04)	98.70 (0.08)	91.18 (0.20)	82.77 (0.14)	72.19 (0.53)	57.33 (1.57)	98.75 (1.53)	98.31 (0.00)	73.16 (0.58)	37.02 (0.70)	62.78 (0.39)	81.62 (0.29)	74.38 (0.27)	66.43 (0.68)
w/o L_{dis}	88.84 (0.17)	97.66 (0.10)	98.65 (0.02)	90.93 (0.16)	81.88 (0.23)	73.44 (1.23)	56.44 (0.63)	98.13 (1.53)	98.31 (0.00)	73.16 (1.01)	37.14 (0.62)	63.36 (0.60)	81.60 (0.32)	74.18 (1.04)	66.17 (0.69)
Soft Mask	88.62 (0.16)	97.78 (0.04)	98.84 (0.05)	90.26 (0.16)	82.16 (0.26)	71.77 (1.06)	56.36 (0.95)	100.00 (0.00)	98.31 (0.00)	73.73 (0.62)	37.29 (0.63)	61.78 (0.60)	81.48 (0.47)	73.54 (0.49)	65.45 (1.15)
Share BN	88.83 (0.10)	97.62 (0.09)	98.71 (0.13)	90.78 (0.18)	80.37 (0.28)	64.69 (1.21)	53.87 (0.65)	90.00 (2.34)	95.93 (0.83)	74.18 (0.99)	35.86 (0.83)	57.77 (0.45)	71.80 (0.30)	67.05 (0.84)	63.43 (0.71)
Share PFSM	90.04 (0.30)	97.93 (0.08)	98.67 (0.09)	90.92 (0.05)	84.28 (0.24)	73.85 (0.83)	56.18 (1.15)	98.75 (1.53)	98.31 (0.00)	73.38 (0.71)	35.59 (1.22)	61.23 (0.57)	81.32 (0.47)	72.56 (0.41)	65.99 (1.13)
w/o Ensemble	87.69 (0.19)	97.87 (0.05)	98.69 (0.09)	89.64 (0.17)	83.78 (0.21)	72.81 (1.52)	56.27 (0.45)	97.50 (1.25)	98.64 (0.68)	72.55 (1.12)	35.56 (1.27)	60.84 (0.88)	80.38 (0.37)	71.59 (0.48)	65.56 (0.49)

Without L_{lce} , L_{ent} , and L_{dis} . In Eq. (15), the loss terms L_{lce} , L_{ent} , and L_{dis} are specifically designed to serve different purposes within the FedPick. These terms aim to minimize the classification loss of personalized features, maximize the entropy of predictions derived from task-irrelevant features, and facilitate knowledge distillation between global and personalized predictions, respectively. The purpose of this experiment is to evaluate the impact of these loss terms on the performance of FedPick. The results presented in Table IV indicate that removing these loss terms leads to a drop in model accuracy, suggesting that these loss terms play a crucial role in enhancing the model performance of FedPick.

Using Soft Mask. In FedPick, we employ a hard binary mask, denoted as m , to discretely select features (either selecting or not). However, an alternative approach involves employing a soft mask to re-weight the features in a continuous manner. In this experiment, we exclude the binary operation during the sampling and directly employ a soft mask, denoted as m_s , to re-weight the features. The re-weighting process is shown in Eq. (21), where z^g represents the global features, m_s denotes the soft mask, z_s^p and z_s^u represents the task-relevant and task-irrelevant features, respectively. The results depicted in Table IV indicate that although the soft feature selection mechanism is effective in the context of FedPick, it falls short of surpassing the performance achieved by the hard feature selection mechanism adopted in this paper.

$$z_s^p = z^g \odot m_s, \quad z_s^u = z^g \odot (1 - m_s). \quad (21)$$

Sharing BN layers. BN is commonly employed by default in most state-of-the-art (SOTA) CNNs. Nevertheless, earlier studies have demonstrated that aggregating BN layers can lead to a notable performance degradation [18], [27]. Therefore, in FedPick, we adopt a personalized BN layer strategy to mitigate the mutual inference among multiple domains, following the approach taken by numerous previous studies [18], [27], [28], [20], [32]. In this experiment, we aim to demonstrate the effectiveness of personalizing the BN layers. As shown in Table IV, when the BN layers are shared across clients, there is a noticeable decrease in accuracy, consistent with the findings

reported in prior research. Given the observed benefits of personalizing BN layers in a cross-domain scenario and the ease of integration into existing frameworks, we decide to incorporate personalized BN layers into FedPick, aiming to further enhance its performance.

Sharing PFSM. In this paper, the PFSM is localized on each client to specially select the personalized features based on each client's data distribution. This experiment aims to showcase the efficacy of personalizing the PFSM. The results presented in Table IV demonstrate that sharing the PFSM parameters leads to improved model performance for simpler datasets like Digits-Five. However, as the complexity of the dataset increases, as observed in Office-Caltech-10 and DomainNet, sharing the PFSM parameters negatively affects the model performance.

Without Ensemble. In FedPick, the final predictions are obtained by ensembling the predictions derived from both global features and personalized features. The purpose of this experiment is to showcase the effectiveness of this ensemble strategy. As depicted in Table IV, when using only a global classifier for prediction, there is a significant decrease in accuracy. Nevertheless, the accuracy still remains higher compared to FedBN. This higher accuracy can be attributed to the knowledge transfer from the personalized classifier to the global classifier through a process known as distillation.

B. Analysis of Features

Selection Ratios for Different Domains. Fig. 7 presents the ratios of selected task-relevant features to the total number of features for different domains. As the complexity of the dataset increases, the local task becomes more challenging, leading to a higher proportion of selected features to accomplish the task. For instance, the average selection ratio of DomainNet is significantly higher compared to the other two datasets. Similarly, within the same dataset, a more intricate domain tends to require a larger number of features. For instance, SVHN necessitates more features compared to other domains within the Digits-Five dataset. The complexity of the datasets

can be discerned from the example images presented in Fig. 5. These findings demonstrate the adaptive feature selection capability of FedPick, allowing it to tailor the feature selection process based on the unique data distribution of each client.

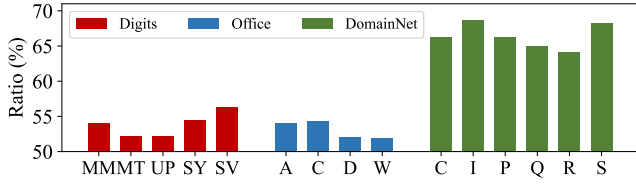


Fig. 7. Feature selection ratios for different domains.

Feature Discrimination. Fig. 8 presents the T-SNE visualization [60] of task-relevant and task-irrelevant features on the DomainNet dataset. The visualization demonstrates that task-relevant features exhibit higher discriminative characteristics, as they are more consistent within classes and scattered between classes compared to task-irrelevant features. This observation highlights the effective feature selection capability of FedPick in selecting important features for the local task. To provide a quantitative evaluation of feature discrimination, we further utilize the Fisher Score, as depicted in Fig. 9. The figure reveals that task-relevant features possess higher Fisher Scores in comparison to task-irrelevant features, which aligns with the findings from Fig. 8. Additionally, it is worth noting that the disparity in Fisher Scores between task-relevant and task-irrelevant features increases as the dataset complexity rises. This finding underscores the greater usefulness of feature selection for more complex datasets.

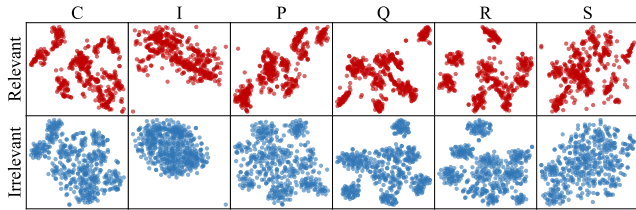


Fig. 8. T-SNE [60] visualization of task-relevant and task-irrelevant features on the DomainNet dataset.

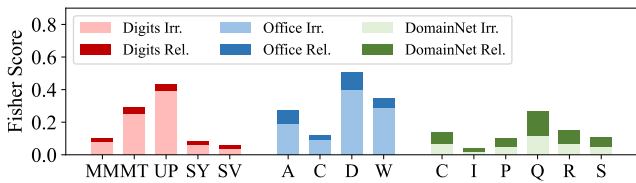


Fig. 9. Fisher Scores of task-relevant and task-irrelevant features on different domains.

Overlapped Ratio of Important Features. In our analysis, we consider the top 50% of the most frequently selected features on a test dataset as important features for the entire domain. To calculate the overlapped ratio of these important features between different domains, we determine how many of these features are common or shared across the domains under study. The overlapped ratio between two domains is

defined as the ratio of the cardinal of the intersection to the union of important features. The overlapped ratios of important features between different domains are shown in Fig. 10. It can be seen that there are a large proportion (about 20% to 30%) of important features that do not overlap between different domains, indicating that each domain possesses its own set of significant features. This result further supports the motivation behind FedPick. Additionally, within the same dataset, domains with a higher degree of similarity exhibit a greater overlapped ratio of important features. For example, MNIST and MNIST-M in the Digits-Five domain demonstrate a higher overlapped ratio, indicating that similar domains tend to share important features.

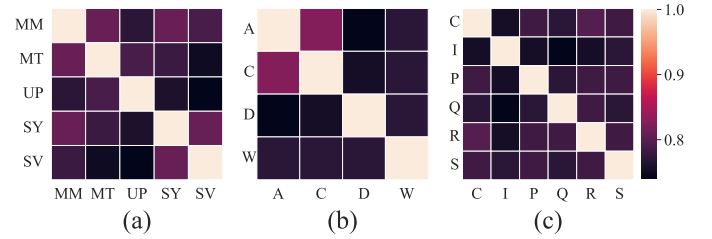


Fig. 10. Overlapped ratios of important features among different domains, (a) Digits-Five, (b) Office-Caltech-10, (c) DomainNet.

C. Effect of Hyperparameters

Hyperparameters in Loss Function. The total loss function in Eq. (15) incorporates four hyperparameters, that is τ , λ_{lce} , λ_{ent} , and λ_{dis} . The accuracy obtained with different hyperparameter settings is depicted in Fig. 11. It can be observed that the model's performance remains stable with respect to changes in τ . However, the performance is more sensitive to variations in λ_{lce} and λ_{dis} . Although the curve representing λ_{ent} in Figure 15 appears stable, we encountered gradient explosion issues when increasing its value during training. Consequently, in our experiments, we only report the results for smaller values of λ_{ent} ranging from 10^{-7} to 10^{-1} .

Hyperparameters for FL System. We consider two hyperparameters that are important to FL, that are the local update epoch and local batch size. Fig. 12 illustrates the model accuracy with varying local epochs. It can be observed that in cross-domain FL, the models of all FL methods exhibit a certain level of robustness to changes in local epochs. Particularly for simpler datasets like Digits-Five, increasing local epochs can even lead to a slight improvement in model accuracy. Fig. 13 displays the accuracy with different batch sizes. While the model accuracy decreases as the local batch size increases for all FL methods, FedPick demonstrates greater robustness to batch size variations and outperforms other FL methods across different batch sizes.

IX. CONCLUSION

In this paper, we propose FedPick, a novel PFL framework for cross-domain scenario that works within the low-dimensional feature space. We are motivated by the observation that existing FL methods often extract redundant

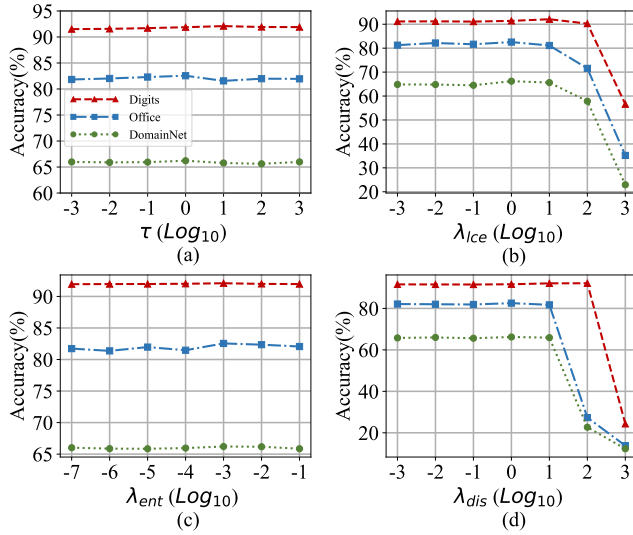


Fig. 11. Test accuracy with different hyperparameters in Eq. (15): (a) τ , (b) λ_{lce} , (c) λ_{ent} , (d) λ_{dis} .

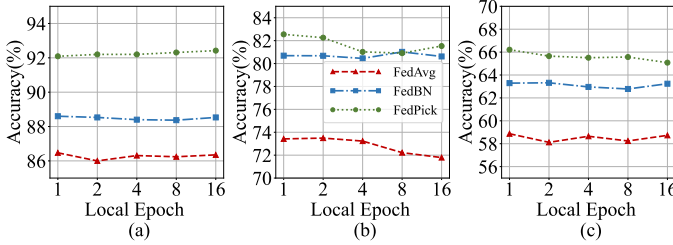


Fig. 12. Test accuracy with different local epochs: (a) Digits-Five, (b) Office-Caltech-10, (c) DomainNet.

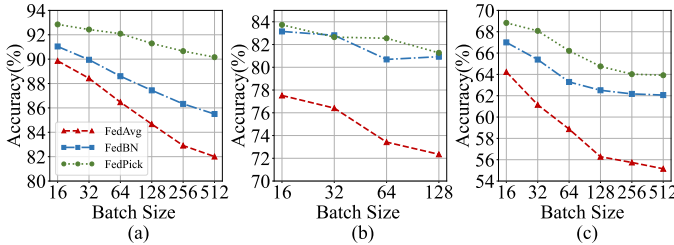


Fig. 13. Test accuracy with different batch sizes: (a) Digits-Five, (b) Office-Caltech-10, (c) DomainNet.

features, which can limit model performance. To address this issue, FedPick enhances model performance by adaptively selecting task-relevant features for each domain, leveraging its specific data distribution. To achieve feature selection, FedPick introduces a PFSM for each client. The PFSM employs reparameterization techniques to make the discretized feature selection process differentiable. This enables easy integration of the PFSM into the backbone model, allowing it to be trained end-to-end using local data on each client. Our experimental results demonstrate that FedPick successfully mitigates feature redundancy by selecting task-relevant features based on the data distribution of each client. Consequently, the proposed framework significantly improves model performance. FedPick provides an effective approach to achieving PFL within

a low-dimensional feature space. This not only simplifies implementation but also enhances interpretability, thereby showcasing its potential for practical applications in cross-domain FL.

ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation of China under Grant No. 61976012.

REFERENCES

- [1] C. Wang, Y. Yang, and P. Zhou, "Towards efficient scheduling of federated mobile devices under computational and statistical heterogeneity," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 394–410, 2020.
- [2] M. Duan, D. Liu, X. Chen, R. Liu, Y. Tan, and L. Liang, "Self-balancing federated learning with global imbalanced data in mobile systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 59–71, 2020.
- [3] W. Wu, L. He, W. Lin, and R. Mao, "Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 7, pp. 1539–1551, 2020.
- [4] R. Lu, W. Zhang, Y. Wang, Q. Li, X. Zhong, H. Yang, and D. Wang, "Auction-based cluster federated learning in mobile edge computing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 4, pp. 1145–1158, 2023.
- [5] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.
- [6] I. Dayan, H. R. Roth, A. Zhong, A. Harouni, A. Gentili, A. Z. Abidin, A. Liu, A. B. Costa, B. J. Wood, C.-S. Tsai *et al.*, "Federated learning for predicting clinical outcomes in patients with covid-19," *Nature medicine*, vol. 27, no. 10, pp. 1735–1743, 2021.
- [7] R. Yan, L. Qu, Q. Wei, S.-C. Huang, L. Shen, D. Rubin, L. Xing, and Y. Zhou, "Label-efficient self-supervised federated learning for tackling data heterogeneity in medical imaging," *IEEE Transactions on Medical Imaging*, 2023.
- [8] G. Long, Y. Tan, J. Jiang, and C. Zhang, "Federated learning for open banking," in *Federated Learning: Privacy and Incentive*. Springer, 2020, pp. 240–254.
- [9] D. Byrd and A. Polychroniadou, "Differentially private secure multi-party computation for federated learning in financial applications," in *Proceedings of the First ACM International Conference on AI in Finance*, 2020, pp. 1–9.
- [10] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE signal processing magazine*, vol. 37, no. 3, pp. 50–60, 2020.
- [11] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.
- [12] Q. Dou, T. Y. So, M. Jiang, Q. Liu, V. Vardhanabuthi, G. Kaissis, Z. Li, W. Si, H. H. Lee, K. Yu *et al.*, "Federated deep learning for detecting covid-19 lung abnormalities in ct: a privacy-preserving multinational validation study," *NPJ digital medicine*, vol. 4, no. 1, pp. 1–11, 2021.
- [13] Y. Zhang, Y. Wei, Q. Wu, P. Zhao, S. Niu, J. Huang, and M. Tan, "Collaborative unsupervised domain adaptation for medical image diagnosis," *IEEE Transactions on Image Processing*, vol. 29, pp. 7834–7844, 2020.
- [14] H. Guan and M. Liu, "Domain adaptation for medical image analysis: a survey," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 3, pp. 1173–1185, 2021.
- [15] R. Wang, P. Chaudhari, and C. Davatzikos, "Embracing the disharmony in medical imaging: A simple and effective framework for domain adaptation," *Medical image analysis*, vol. 76, p. 102309, 2022.
- [16] V. Muşat, I. Fursa, P. Newman, F. Cuzzolin, and A. Bradley, "Multi-weather city: Adverse weather stacking for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2906–2915.
- [17] X. Xiong, Y. Liu, T. Yuan, Y. Wang, Y. Wang, and H. Zhao, "Neural map prior for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 535–17 544.

- [18] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," *arXiv preprint arXiv:2102.07623*, 2021.
- [19] 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*. PMLR, 2017, pp. 1273–1282.
- [20] J. Mills, J. Hu, and G. Min, "Multi-task federated learning for personalised deep neural networks in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 630–641, 2021.
- [21] H. Jin, D. Bai, D. Yao, Y. Dai, L. Gu, C. Yu, and L. Sun, "Personalized edge intelligence via federated self-knowledge distillation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 2, pp. 567–580, 2022.
- [22] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [23] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.
- [24] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 2089–2099.
- [25] Y. Cai and X. Wan, "Multi-domain sentiment classification based on domain-aware embedding and attention," in *IJCAI*, 2019, pp. 4904–4910.
- [26] P. P. Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Federated learning with local and global representations," *arXiv preprint arXiv:2001.01523*, 2020.
- [27] M. Andreux, J. O. d. Terrail, C. Beguier, and E. W. Tramel, "Siloted federated learning for multi-centric histopathology datasets," in *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*. Springer, 2020, pp. 129–139.
- [28] B. Sun, H. Huo, Y. Yang, and B. Bai, "Partialfed: Cross-domain personalized federated learning via partial initialization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 309–23 320, 2021.
- [29] X. Geng, L. Wang, X. Wang, B. Qin, T. Liu, and Z. Tu, "How does selective mechanism improve self-attention networks?" *arXiv preprint arXiv:2005.00979*, 2020.
- [30] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.
- [31] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.
- [32] L. Wang, K. Zhang, Y. Li, Y. Tian, and R. Tedrake, "Does learning from decentralized non-iid unlabeled data benefit from self supervision?" in *The Eleventh International Conference on Learning Representations*, 2023.
- [33] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.
- [34] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.
- [35] C. H. Bahnsen and T. B. Moeslund, "Rain removal in traffic surveillance: Does it matter?" *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 2802–2819, 2018.
- [36] M. A. Kenk and M. Hassaballah, "Dawn: vehicle detection in adverse weather nature dataset," *arXiv preprint arXiv:2008.05402*, 2020.
- [37] S. Leroux, B. Li, and P. Simoons, "Multi-branch neural networks for video anomaly detection in adverse lighting and weather conditions," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2358–2366.
- [38] G. Zhu, X. Liu, S. Tang, and J. Niu, "Aligning before aggregating: Enabling cross-domain federated learning via consistent feature extraction," in *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2022, pp. 809–819.
- [39] Z. Luo, Y. Wang, Z. Wang, Z. Sun, and T. Tan, "Disentangled federated learning for tackling attributes skew via invariant aggregation and diversity transferring," *arXiv preprint arXiv:2206.06818*, 2022.
- [40] Y. Tan, G. Long, J. Ma, L. Liu, T. Zhou, and J. Jiang, "Federated learning from pre-trained models: A contrastive learning approach," *Advances in Neural Information Processing Systems*, vol. 35, pp. 19 332–19 344, 2022.
- [41] K. Wang, R. Mathews, C. Kiddon, H. Eichner, F. Beaufays, and D. Ramage, "Federated evaluation of on-device personalization," *arXiv preprint arXiv:1910.10252*, 2019.
- [42] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," *arXiv preprint arXiv:1909.12488*, 2019.
- [43] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3557–3568, 2020.
- [44] Q. Gu, Z. Li, and J. Han, "Generalized fisher score for feature selection," *arXiv preprint arXiv:1202.3725*, 2012.
- [45] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.
- [46] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.
- [47] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.
- [48] N. Cristianini, J. Shawe-Taylor *et al.*, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.
- [49] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4615–4625.
- [50] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," *arXiv preprint arXiv:2002.10619*, 2020.
- [51] D. Das, S. Yun, and F. Porikli, "Confess: A framework for single source cross-domain few-shot learning," in *International Conference on Learning Representations*, 2021.
- [52] Y. Ganin and V. S. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Int. Conf. Machin. Learn.*, 2015, pp. 1180–1189.
- [53] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE (USA)*, pp. 2278–2324, 1998.
- [54] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, 1994.
- [55] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.
- [56] 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, Providence, RI, USA, June 16-21, 2012*. IEEE Computer Society, 2012, pp. 2066–2073.
- [57] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 1406–1415.
- [58] 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*, vol. 2, pp. 429–450, 2020.
- [59] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [60] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. 11, 2008.