FedDTR: Leveraging intra-domain global priors via Domain-Invariant Text Representation for Personalized Federated Learning

Zhenhu Zhang, Ruofeng Tong*, Lanfen Lin
Zhejiang University
Hangzhou, China
{12251005, trf, llf}@zju.edu.cn

Yen-Wei Chen
Ritsumeikan University
Osaka, Japan
chen@is.ritsumei.ac.jp

Abstract

Federated learning has emerged as a promising paradigm for privacy-preserving collaborative learning. Recently, personalized federated learning has attracted growing interest due to its ability to handle statistical heterogeneity across clients—such as hospitals or mobile devices. However, existing approaches often align local models with a global representation that is both domain-variant and biased (e.g., toward dominant clients), thereby compromising fairness and generalization. To address these limitations, we propose FedDTR, a novel personalized federated learning framework that leverages textual descriptions as noise-free, unbiased, and domain-invariant global representations. These text-based representations serve as fair convergence targets and provide intra-domain global priors. Specifically, FedDTR pulls sample embeddings toward their corresponding class text embeddings while pushing them away from those of other classes, thereby enhancing intra-class compactness and inter-class separability. Furthermore, an Intra-domain Prior Module exploits local client data to estimate domain-specific global priors, enabling better modeling of the underlying data distribution. We evaluate FedDTR on eight benchmark datasets spanning computer vision and natural language processing, and demonstrate its consistent superiority over state-of-the-art methods.

***Keywords:** federated learning, deep learning, text representation, personalization*

1. Introduction

Federated learning is increasingly popular in the field of distributed machine learning due to its unique privacy-preserving mechanisms. It allows multiple devices or organizations to collaboratively train models using local data without sharing the original data. This method is particularly effective in scenarios where data is abundant but difficult to centralize [25, 11, 16, 38]. The traditional federated learning method, FedAvg [27], centers on an averaging aggregation mechanism. It uses a weighted average to combine the local models into a new global model, which is then distributed back to each local client for further training. This mechanism enables federated learning to achieve collaborative machine learning without the need to share raw data. Although FedAvg ensures privacy, its performance is severely constrained by data heterogeneity [1, 15, 33]. Specifically, because the data on local clients is not independent and identically distributed (non-IID), the models trained on this data belong to different distributions. Simple averaging aggregation can lead to significant degradation in model performance. Moreover, the presence of noisy data on local clients exacerbates this issue. Recently, personalized federated learning [40, 34, 41] has gained popularity

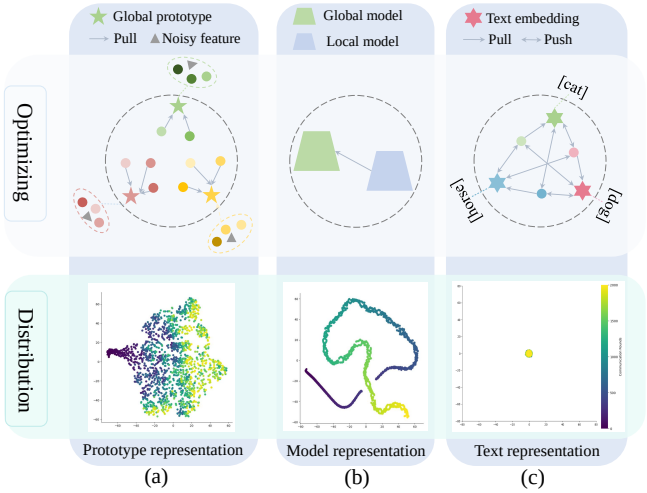


Figure 1: Optimization and distribution of different global representations: (a) Global prototype representation is susceptible to noisy data and undergoes significant changes in distribution during collaborative training. (b) Global model representation also results in continuously changing distributions during collaborative learning. (c) Text embeddings as global representations are unaffected by noisy data and maintain a stable distribution during training

*Corresponding author.

as it addresses the issue of statistical heterogeneity by learning personalized model parameters. For each client, the integration of federated learning provides additional global information from server, such as a global model [21] or global prototypes [34], to address the problem of limited local data and enhance the performance of their own models. To obtain high-quality server information that encompasses knowledge from diverse distributions, each client also needs to upload their learned information for server aggregation. Therefore, the objective of personalized federated learning is to leverage global information to train effective personalized models.

Currently, existing personalized federated learning methods tend to bias local models towards global representations, including global prototypes [34, 23, 35] and global models [21, 32]. **For global prototypes** (as shown in Figure 1(a)), methods like FedProto [34] and FedPHP [23] utilize global prototypes to guide the learning of personalized features, causing local features to align with global prototypes. **For global models** (as shown in Figure 1(b)), methods like pFedMe [32] and Ditto [21] employ additional proximal terms to regularize the differences between personalized local model parameters and frozen global parameters, thus biasing local model parameters towards global model parameters.

However, global models/prototypes aggregated from local models/prototypes derived from different distributions present several challenges: (1) The aggregated parameters tend to favor dominant client parameters, leading to an unfair convergence target. (2) The quality of global models and prototypes depends on local model quality, and poor global models can mislead local training. (3) As shown in Figure 1, the distribution of global models/prototypes continuously changes during collaborative learning, biasing local models towards a shifting target and hindering optimization.

To address these issues, we propose FedDTR, a personalized federated learning method that utilizes Domain-Invariant Text Representation (DTR). In this approach, data labels are converted into text, and text embeddings are used as global representations. These embeddings are unbiased and do not favor any specific client. Local features and text embeddings guide each other to reduce intra-class distance and increase inter-class distance, effectively mitigating data heterogeneity. As shown in Figure 1 (c), the distribution of text embeddings remains stable during collaborative learning, providing a consistent target for optimization. This stability, along with domain-invariant nature of text embeddings, makes them ideal for global representation. Additionally, we propose an intra-domain prior module that aggregates observable data embeddings into global sample priors, helping local models better understand global data distribution.

To evaluate the performance of FedDTR in terms of effectiveness, scalability, and stability, we compared it with 13 state-of-the-art (SOTA) methods across 7 datasets in the fields of computer vision (CV) and natural language processing (NLP). Experimental results show that FedDTR achieves the best performance on each dataset. In summary, our main contributions are:

- We propose the FedDTR framework, which introduces text as a domain-invariant global representation to guide local sample features, enhancing intra-class similarity and increasing inter-class distance, thereby addressing the issue of data heterogeneity.
- We introduce the Intra-domain Prior module to provide global sample priors, aiding the model in better understanding the overall data distribution and alleviating overfitting of personalized models to local data.
- We conduct extensive experiments in the CV, NLP domains. The results show that our FedDTR out performs the SOTA methods in terms of effectiveness, scalability and stability.

2. Related Work

Federated learning (FL) enables collaborative model training across distributed clients while preserving data privacy by aggregating model parameters instead of raw data. The seminal FedAvg algorithm [27] addresses limited local data through iterative model averaging. However, its performance degrades significantly under non-IID (statistically heterogeneous) data distributions, where client data exhibit divergent feature or label distributions. To tackle this challenge, personalized federated learning (PFL) methods have been developed, which can be broadly grouped into three categories: (1) decoupling global and local model components, (2) using global models as regularization references, and (3) leveraging global prototypes as semantic anchors.

Decoupled Global–Local Architectures. These methods partition the model into shared (global) and client-specific (local) components, communicating only the former with the server. FedPer [2] trains a global feature extractor while keeping a personalized classification head locally. FedRep [5] alternates between local head fine-tuning and global extractor updates. FedRoD [3] further decouples global and personalized objectives, applying a balanced softmax loss [30] to the global branch while retaining a local head for client-specific adaptation. Despite their simplicity, such approaches may suffer from gradient misalignment due to insufficient global supervision during local training.

Global Model as Regularization Reference. In this paradigm, the global model serves as a stable reference to

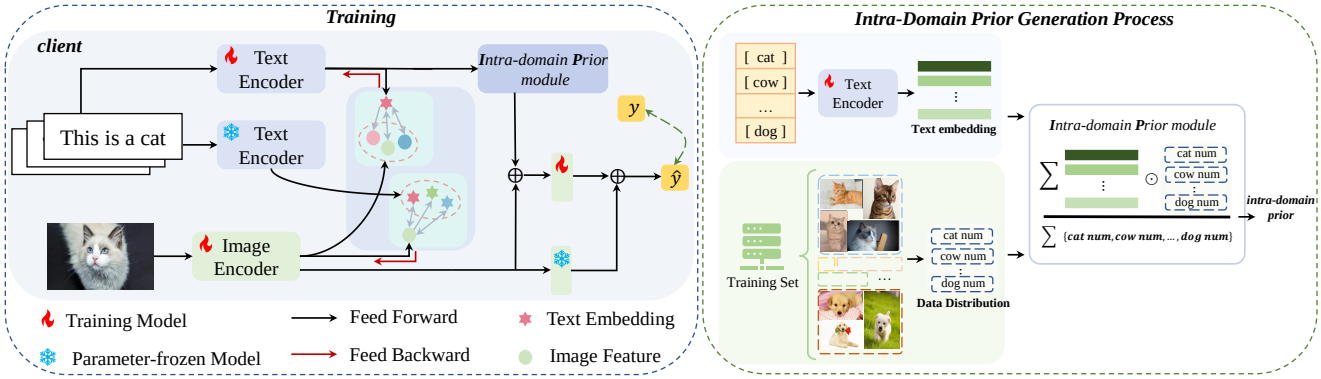


Figure 2: **Left: The structure of FedDTR**. The input consists of a image and text set. Image are extracted by the image encoder ($f(\omega_i)$) to obtain features F_i (belongs to class c). The text set is processed by both the text encoder ($f(\theta_i)$) and the frozen text encoder ($f(\theta_f)$), extracting features $\mathcal{T}_i = \{T_i^c\}_{c=1}^C$ and $\mathcal{T} = \{T^c\}_{c=1}^C$. We aim to make F_i close to T^c (reducing intra-class distance) and T_i^c close to F_i (mapping text space to image space). The Intra-domain Prior module (IP) calculates the intra-domain global prior from T_i to help the model better understand the overall data distribution. **Right: Intra-Domain Prior Generation Process**. The text embedding and the data distribution derived from the training set serve as inputs to the Intra-domain Prior module to obtain the intra-domain prior. [cat] = "This is a cat".

regularize local updates. FedProx [22] introduces a proximal term to constrain local models toward the global one. SCAFFOLD [17] corrects client drift using control variates derived from global and local gradients. pFedMe [32] employs Moreau envelopes to decouple personalized optimization from global model learning, while Ditto [21] explicitly optimizes a personalized model under a proximal constraint to the global model. However, these methods remain vulnerable to bias toward dominant clients and suffer from the shifting distribution of the global model during training, which can destabilize convergence.

Prototype-Based Global Guidance. Rather than relying on full models, these methods use class-wise feature prototypes as global semantic references. FedProto [34] and FedPCL [35] exchange high-dimensional prototypes to align local features with global class centroids. FPL [12] introduces cluster-level prototypes to mitigate bias, while FedPHP [23] preserves historical personalized knowledge via Maximum Mean Discrepancy (MMD) [9, 29]. Although effective, prototype-based methods critically depend on well-trained feature extractors; moreover, the dynamic and potentially noisy nature of aggregated prototypes can lead to unstable optimization, especially under severe data heterogeneity or limited local samples.

Other Personalization Strategies. Beyond the above categories, several alternative approaches have been explored. Per-FedAvg [6] adopts meta-learning to enable rapid personalization with minimal local updates. FedFomo [41] computes client-specific aggregation weights based on other clients' personalized models. FedAMP [13] uses attention mechanisms to guide personalized model aggregation, and FedALA [40] adaptively blends global and local models before each training round to align with individual client objectives.

In summary, while existing PFL methods improve performance under heterogeneity, most rely on *learned* global representations—whether models or prototypes—that are inherently domain-variant, biased, and unstable. In contrast, our method, FedDTR, leverages *fixed*, domain-invariant text embeddings as global references, circumventing these fundamental limitations.

3. Method

In this section, we present FedDTR (Federated Learning with Domain-invariant Text Representation), as illustrated in Figure 2. To overcome the limitations of existing personalized federated learning methods—such as biased global references and unstable prototype dynamics—FedDTR introduces two key components: (1) *Domain-Invariant Text Representation* and (2) the *Intra-domain Prior* module.

The core idea is to leverage fixed, semantically grounded text embeddings as domain-invariant anchors to guide local feature learning, thereby enhancing intra-class compactness and inter-class separability. Simultaneously, these text embeddings are adaptively refined using local image features to align them with the image embedding space. The Intra-domain Prior module further incorporates client-specific class statistics into the training process, providing domain-aware global context that mitigates overfitting to local data. Finally, inspired by prior work [2] showing that classification heads better capture personalization than feature extractors, we maintain a dual-head architecture: a trainable personalized head h_i and a frozen global head h_f, the latter of which is synchronized with the server to preserve global

knowledge.

3.1. Domain-Invariant Text Representation

We aim to learn a set of unbiased, domain-invariant global representations—derived from natural language descriptions—to serve as stable convergence targets across heterogeneous clients. FedDTR takes two inputs: an image I_i from client i, and a shared text template set $\mathbb{T} = \{[\text{cat}], [\text{dog}], \ldots, [\text{horse}]\}$, where each [class] is a prompt such as "This is a cat".

The image I_i is encoded by a local image feature extractor $f(\omega_i; I_i)$ to produce a feature vector $\mathcal{F}_i \in \mathbb{R}^d$. The text set $\mathbb{T}$ is tokenized and embedded into $\mathfrak{T} \in \mathbb{R}^{C\times d}$, where C is the total number of classes. A local text encoder $f(\theta_i; \cdot)$ then maps each tokenized prompt to a text embedding, yielding $\mathcal{T}_i = \{T_i^c\}_{c=1}^C$.

To align image features with semantic text representations, we design a contrastive local objective that simultaneously:

1. pulls image features closer to their corresponding class text embeddings while pushing them away from others, and
2. updates the text encoder so that text embeddings better reflect the local image distribution and reside in the same semantic space as image features.

This yields the following loss:

$$\mathcal{L}_{cl}^{(1)} = -\log \frac{e^{\varphi(\mathcal{F}_i, T_i^c)}}{e^{\varphi(\mathcal{F}_i, T_i^c)} + \sum_{\tilde{c}\neq c} e^{\varphi(\mathcal{F}_i, T_i^{\tilde{c}})}}, \tag{1}$$

where $\varphi(\mathbf{a}, \mathbf{b}) = \mathbf{a}^\top \mathbf{b}$ denotes cosine similarity (assuming normalized embeddings), and c is the ground-truth class of I_i.

However, updating both the image and text encoders within the same batch leads to inconsistent alignment targets: as θ_i evolves during backpropagation, the text embeddings guiding $\mathcal{F}_i$ shift mid-iteration, destabilizing training. To address this, we introduce a *frozen global text encoder* $f(\theta_f)$ on each client. The parameters θ_f are downloaded from the server at the beginning of each round and kept fixed during local training. This ensures that all clients share the same set of *global text embeddings* $\mathcal{T} = \{T^c\}_{c=1}^C = f(\theta_f; \mathbb{T})$, which serve as stable, domain-invariant references.

We thus refine the local objective to jointly leverage both the dynamic local text embeddings $\mathcal{T}_i$ and the static global ones $\mathcal{T}$:

$$\begin{aligned}\mathcal{L}_{cl} = &-\log \frac{e^{\varphi(\mathcal{F}_i, T^c)}}{e^{\varphi(\mathcal{F}_i, T^c)} + \sum_{\tilde{c}\neq c} e^{\varphi(\mathcal{F}_i, T^{\tilde{c}})}} \\ &-\log \frac{e^{\varphi(\hat{\mathcal{F}}_i, T_i^c)}}{e^{\varphi(\hat{\mathcal{F}}_i, T_i^c)} + \sum_{\tilde{c}\neq c} e^{\varphi(\hat{\mathcal{F}}_i, T_i^{\tilde{c}})}},\end{aligned} \tag{2}$$

where $\hat{\mathcal{F}}_i = \mathcal{F}_i$ in value but is detached from the computational graph (i.e., $\hat{\mathcal{F}}_i$ does not propagate gradients to the image encoder). This ensures that:

- The first term guides the *image encoder* using stable global text embeddings $\mathcal{T}$.
- The second term updates the *text encoder* using fixed image features $\hat{\mathcal{F}}_i$, preventing co-adaptation instability.

Minimizing $\mathcal{L}_{cl}$ is equivalent to minimizing:

$$\log\left(\frac{\sum_{\tilde{c}\neq c} e^{-\varphi(\mathcal{F}_i, T^{\tilde{c}})}}{e^{-\varphi(\mathcal{F}_i, T^c)}}\right) + \log\left(\frac{\sum_{\tilde{c}\neq c} e^{-\varphi(\hat{\mathcal{F}}_i, T_i^{\tilde{c}})}}{e^{-\varphi(\hat{\mathcal{F}}_i, T_i^c)}}\right), \tag{3}$$

which explicitly encourages (1) small intra-class distances and (2) large inter-class margins in the joint embedding space.

3.2. Intra-domain Prior Module

In non-IID federated settings, each client i observes only a subset of classes $C_i \subseteq C$. Directly training a C-way classifier on such partial data can distort the decision boundary and degrade generalization. To mitigate this, we introduce an *Intra-domain Prior* P_i that encodes client-specific class prevalence as a global contextual cue.

Formally, let D_i denote the local dataset of client i. We compute the empirical class distribution over C (padding unseen classes with zero) and construct a weighted average of local text embeddings:

$$P_i = \sum_{c\in C} \left(T_i^c \cdot \frac{1}{|D_i|} \sum_{(x,y)\in D_i} \mathbb{I}\{y = c\} \right), \tag{4}$$

where $\mathbb{I}(\cdot)$ is the indicator function. Note that P_i aggregates semantic prototypes of observed classes, scaled by their local frequencies, thereby capturing domain-specific prior knowledge.

During training, P_i is added to the image feature before classification, enriching the representation with intra-domain context. To balance personalization and global consistency, we adopt a dual-head design:

- A *personalized head* h_i, trained locally and never updated from the server, captures client-specific decision logic.
- A *frozen global head* h_f, initialized with the server's aggregated classifier and kept fixed during local training, preserves global semantic structure.

The final prediction combines both perspectives:

$$\mathcal{L}_{ce} = -\mathbf{1}_{y_i}^\top \log \sigma\big(f(h_i; \mathcal{F}_i + P_i) + f(h_f; \mathcal{F}_i)\big), \tag{5}$$

where σ denotes the softmax function, and $\mathbf{1}_{y_i}$ is the one-hot label vector.

The overall local objective is:

$$\mathcal{L} = \mathcal{L}_{cl} + \mathcal{L}_{ce}. \tag{6}$$

At each communication round, clients upload updated image encoder parameters ω_i and text encoder parameters θ_i to the server, which aggregates them (e.g., via FedAvg) to produce global models ω and θ. The server then broadcasts ω and θ to all clients, where θ is used to update the frozen global text encoder ($\theta_f \leftarrow \theta$) for the next round.

The complete algorithm is summarized in Algorithm 1.

4. Convergence Analysis

To analyze the convergence of FedDTR, we introduce the following notations. Let $t \in \{0, 1, \ldots, T-1\}$ denote the communication round, and $e \in \{0, 1, \ldots, E-1\}$ the local iteration index within each round, where E is the number of local updates per round. The global iteration counter is thus $tE + e$, corresponding to the e-th local step in round t. At the beginning of round t (i.e., at iteration tE), each client k replaces its local classification head with the global head received from the server. The local model of client k is denoted by $W_k^{t,e}$, which includes both the image encoder and the text encoder. The learning rate is denoted by η.

We make the following standard assumptions commonly adopted in federated optimization literature.

Assumption 1 (Lipschitz Smoothness). The local loss function $\mathcal{L}_k$ of client k is L_1-smooth, i.e., for any model parameters W, W',

$$\|\nabla\mathcal{L}_k(W) - \nabla\mathcal{L}_k(W')\|_2 \leq L_1\|W - W'\|_2.$$

Equivalently, for all W, W',

$$\mathcal{L}_k(W) \leq \mathcal{L}_k(W') + \langle\nabla\mathcal{L}_k(W'), W - W'\rangle + \frac{L_1}{2}\|W - W'\|_2^2.$$

Assumption 2 (Unbiased Gradient and Bounded Variance). Let $g_k^{t,e} = \nabla\mathcal{L}_k(W_k^{t,e}; B_k^{t,e})$ be the stochastic gradient computed on a mini-batch $B_k^{t,e} \subseteq D_k$. We assume:

$$\mathbb{E}[g_k^{t,e}] = \nabla\mathcal{L}_k(W_k^{t,e}), \quad \mathbb{E}\big[\|g_k^{t,e} - \nabla\mathcal{L}_k(W_k^{t,e})\|_2^2\big] \leq \sigma^2,$$

where the expectation is taken over the randomness of the mini-batch.

Assumption 3 (Bounded Client Drift). Let θ^t denote the global feature extractor (e.g., image encoder) broadcast by the server at round t, and $\theta_k^{t,E}$ the corresponding local parameter after E local updates on client k. We assume the deviation between local and global models is bounded:

$$\|\theta_k^{t,E} - \theta^t\|_2^2 \leq \delta^2, \quad \forall k, t.$$

Algorithm 1 The Learning Process in FedDTR

Input: client dataset D_n, communication rounds R, clients N, initial server parameters ω^0, h^0, θ^0, learning rate η, client joining ratio J.
Output: Personalized model parameters $\{W_i = (\omega_i, \theta_i, h_i, h_f)\}_{i=1}^N$.

Client i, $\forall i \in [N]$, initializes h_i^0.
for $t = 0, \ldots, R$ **do**
  Server samples a client subset $\mathcal{I}^t$ based on J.
  Server sends $\{\omega^t, h^t, \theta^t\}$ to $\mathcal{I}^t$.
  for all Client $i \in \mathcal{I}^t$ **in parallel do**
    ▷**Local initialization**
    Initialize ω_i^t, θ_i^t with $\{\omega^t, \theta^t\}$.
    Initialize h_f^t, θ_f^t with $\{h^t, \theta^t\}$.
    ▷**Local training**
    Update ω_i, θ_i, h_i simultaneously:
      $\omega_i^{t+1} \leftarrow \omega_i^t - \eta(\nabla L_{cl}(\omega_i^t, \theta_f^t; D_i) + \nabla L_{ce}(\omega_i^t, h_i^t; D_i))$.
      $h_i^{t+1} \leftarrow h_i^t - \eta\nabla L_{ce}(h_i^t, h_f^t; D_i)$.
      $\theta_i^{t+1} \leftarrow \theta_i^t - \eta\nabla L_{cl}(\omega_i^t, \theta_i^t; D_i)$.
    Upload $\{\omega_i^{t+1}, h_i^{t+1}\, \theta_i^{t+1}\}$ to the server.
  end for
  ▷**Local training**
  Server calculates $n^t = \sum_{i\in\mathcal{I}^t} n_i$ and obtains:
    $\omega^{t+1} = \sum_{i\in\mathcal{I}^t} \frac{n_i}{n^t}\omega_i^{t+1}$.
    $h^{t+1} = \sum_{i\in\mathcal{I}^t} \frac{n_i}{n^t}h_i^{t+1}$.
    $\theta^{t+1} = \sum_{i\in\mathcal{I}^t} \frac{n_i}{n^t}\theta_i^{t+1}$.
end for
return $\{W_1, \ldots, W_N\}$.

Based on these assumptions, we establish the following lemmas and theorems.

Lemma 1 (Local Training Progress). Under Assumptions 1 and 2, the expected decrease in local loss over E local steps in round t satisfies:

$$\mathbb{E}[\mathcal{L}_k(W_k^{t,E})] \leq \mathcal{L}_k(W_k^{t,0}) - \left(\eta - \frac{L_1\eta^2}{2}\right)\sum_{e=0}^{E-1}\mathbb{E}[\|\nabla\mathcal{L}_k(W_k^{t,e})\|_2^2] + \frac{L_1 E\eta^2\sigma^2}{2}.$$

Lemma 2 (Effect of Model Aggregation). Under Assumption 3, the loss after model aggregation (i.e., at the start of the next round) is bounded by:

$$\mathbb{E}\big[\mathcal{L}_k(W_k^{t+1,0})\big] \leq \mathbb{E}\big[\mathcal{L}_k(W_k^{t,E})\big] + \eta\delta^2.$$

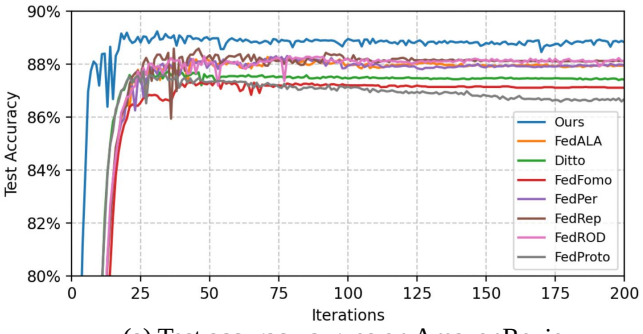
Combining Lemma 1 and Lemma 2 yields the progress over one full federated round.

Theorem 1 (One-Round Progress). For any client k, under Assumptions 1–3, the expected loss at the beginning of round $t+1$ satisfies:

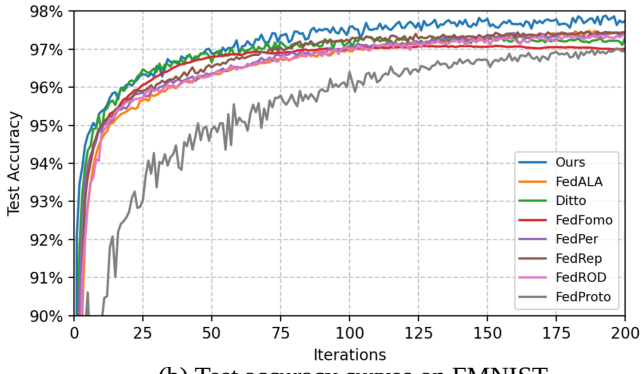
$$\mathbb{E}[\mathcal{L}_k(W_k^{t+1,0})] \leq \mathcal{L}_k(W_k^{t,0}) - \left(\eta - \frac{L_1\eta^2}{2}\right)\sum_{e=0}^{E-1}\mathbb{E}[\|\nabla\mathcal{L}_k(W_k^{t,e})\|_2^2] + \frac{L_1 E\eta^2\sigma^2}{2} + \eta\delta^2.$$

Table 1: The test accuracy (%) on the CV task in the pathological setting and the CV/NLP task in the practical setting. T-I represents using a 4-layer CNN on Tiny-ImageNet, while T-I* represents using ResNet-18 on Tiny-ImageNet. Bold indicates the highest accuracy results.

Settings	**Pathological Setting**			**Practical Setting**					
	MNIST	Cifar10	Cifar100	MNIST	Cifar10	Cifar100	T-I	T-I*	AG News
FedAvg	97.93	55.09	25.98	98.81	59.16	31.89	19.46	19.45	79.57
FedProx	98.01	55.06	25.94	98.82	59.21	31.99	19.37	19.27	79.35
Per-FedAvg	99.63	89.63	56.80	98.90	87.74	44.28	25.07	21.81	93.27
pFedMe	99.75	90.11	58.20	99.52	88.09	47.34	26.93	33.44	91.41
FedAMP	99.76	90.79	64.34	99.47	88.70	47.69	27.99	29.11	94.18
FedPer	99.70	91.15	63.53	99.47	89.22	49.63	33.84	38.45	95.54
FedRep	99.77	91.93	67.56	99.48	90.40	52.39	37.27	39.95	96.28
Ditto	99.81	92.39	67.23	99.64	90.59	52.87	32.15	35.92	95.45
FedFomo	99.83	91.85	62.49	99.33	88.06	45.39	26.33	26.84	95.84
FedPHP	99.73	90.01	63.09	99.58	88.92	50.52	35.69	29.90	94.38
FedRoD	99.90	91.98	62.30	99.66	89.93	50.94	36.43	37.99	95.99
FedProto	99.86	90.18	69.18	99.53	90.59	52.70	31.21	26.38	96.34
FedALA	99.88	92.44	67.83	99.64	90.67	55.92	40.54	41.94	96.52
Ours	**99.91**	**93.19**	**71.61**	**99.77**	**92.11**	**59.67**	**43.80**	**48.52**	**96.89**



(a) Test accuracy curves on AmazonReview.



(b) Test accuracy curves on FMNIST.

Figure 3: The test accuracy curves on the Amazon Review and FMNIST datasets.

Summing Theorem 1 over T rounds and rearranging terms gives the convergence rate.

Theorem 2 (Non-Convex Convergence Rate). Suppose the learning rate satisfies $\eta < 2/L_1$. Then, the average squared gradient norm over all local iterations satisfies:

$$\frac{1}{TE}\sum_{t=0}^{T-1}\sum_{e=0}^{E-1}\mathbb{E}[\|\nabla\mathcal{L}_k(W_k^{t,e})\|_2^2] \leq \frac{2(\mathcal{L}_k(W_k^{0,0})-\mathcal{L}_k^*)}{TE\left(2\eta-L_1\eta^2\right)}+\frac{L_1\eta\sigma^2}{2\left(2-L_1\eta\right)}+\frac{2\delta^2}{2-L_1\eta},$$

where $\mathcal{L}_k^*$ is the infimum of $\mathcal{L}_k$. Consequently, FedDTR converges to a stationary point at a rate of $\mathcal{O}\left(\frac{1}{T}+\frac{\eta\sigma^2}{2}+\delta^2\right)$. In particular, when $\eta = \mathcal{O}(1/\sqrt{T})$, the convergence rate is $\mathcal{O}(1/\sqrt{T})$; if data heterogeneity is mild ($\delta \to 0$) and noise is small ($\sigma \to 0$), the rate improves to $\mathcal{O}(1/T)$.

Thus, under standard assumptions, FedDTR guarantees convergence for non-convex objectives, with the rate explicitly depending on local update steps E, gradient noise σ^2, and client drift δ^2.

5. Experiment

In this section, we validate the performance of FedDTR in terms of effectiveness, scalability, stability, and convergence time. We also examine the effectiveness of each proposed module. Specifically, we compare FedDTR with 13 state-of-the-art (SOTA) federated learning methods across CV and NLP tasks, including:

- **Traditional Methods**: FedAvg, FedProx;
- **Methods with Separation of Global and Local Components**: FedPer, FedRoD, FedRep;
- **Methods Using Models as Global References**: Ditto, pFedMe;

Table 2: The test accuracy (%) on the CV and NLP task regarding Heterogeneity and the CV task regarding scalability

Settings	**Heterogeneity**					**Scalability**			
Dataset	TINY			AG News		Cifar-100			
	β=0.01	β=0.1	β=0.5	β=0.1	β=1	N=10	N=20	N=30	N=100
FedAvg	15.70	19.46	21.14	79.57	87.12	31.47	31.89	31.15	31.95
FedProx	15.66	19.37	21.22	79.35	87.21	31.24	31.99	31.21	31.97
Per-FedAvg	39.39	25.07	16.36	93.27	87.08	37.24	44.28	41.57	36.07
pFedMe	41.45	26.93	17.48	91.41	87.08	44.06	47.34	47.04	46.45
FedAMP	48.42	27.99	12.48	94.18	83.35	49.23	47.69	45.33	40.43
FedPer	51.83	33.84	17.31	95.54	91.85	50.31	49.63	44.98	40.37
FedRep	55.43	37.27	16.74	96.28	92.25	52.89	52.39	50.24	44.61
Ditto	50.62	32.15	18.98	95.45	91.89	52.32	52.87	52.53	52.89
FedFomo	46.36	26.33	11.59	95.84	91.20	46.71	45.39	43.20	38.91
FedPHP	48.63	35.69	21.09	94.38	90.52	49.32	50.52	49.28	49.70
FedRoD	49.17	36.43	23.23	95.99	92.16	49.83	50.94	50.11	46.65
FedProto	52.04	31.21	16.99	96.34	81.16	49.13	52.70	52.32	47.11
FedALA	55.75	40.54	27.85	96.52	92.45	57.01	55.92	55.52	54.68
Ours	**59.93**	**43.80**	**29.62**	**96.89**	**94.37**	**60.01**	**59.67**	**58.71**	**56.51**

Table 3: The accuracy (%) on Cifar100 ($N = 50, \beta = 0.1$) when clients accidentally drop out.

	$J = 1$	$J \in [0.5, 1]$	$J \in [0.1, 1]$
Per-FedAvg	44.31	43.66	43.63
pFedMe	48.36	43.28	41.71
FedAMP	44.39	42.91	42.92
Ditto	50.59	49.78	48.33
FedPer	44.22	44.12	44.07
FedRep	47.41	46.93	46.61
FedRoD	49.38	49.07	47.80
FedFomo	42.56	40.96	40.93
FedPHP	50.23	45.19	44.43
FedProto	50.29	49.45	46.05
FedALA	55.64	54.83	53.09
Ours	**57.68**	**57.50**	**57.25**

- **Methods Using Prototypes as Global References**: FedProto, FedPHP;
- **Other Personalized Methods**: Per-FedAvg, FedFomo, FedAMP, FedALA.

5.1. Setup

Datasets. For CV tasks, we use five public datasets: MNIST [19], Fashion-MNIST (FMNIST) [39], CIFAR-10 [18], CIFAR-100 [18], and Tiny-ImageNet [4]. For NLP tasks, we employ AG News [42] and Amazon Review [7].

Backbone. Following prior work [27, 26, 8], we adopt a 4-layer CNN for MNIST, FMNIST, CIFAR-10, CIFAR-100, and Tiny-ImageNet. To evaluate FedDTR on a more expressive architecture, we also conduct experiments on Tiny-ImageNet using ResNet-18 [10]. For AG News and Amazon Review, we use fastText [14] and a 3-layer MLP [28], respectively. Notably, we use a lightweight 1-layer MLP as the text encoder to map class labels into embedding space.

Hyperparameter Settings. Regarding the local learning rate η, we set $\eta = 0.005$ for the 4-layer CNN and the 3-layer MLP, and $\eta = 0.1$ for ResNet-18 and fastText. Following the protocol in pFedMe [32], we simulate a system with $N = 20$ clients under full participation ($J = 1.0$), unless otherwise specified. For local datasets, each client allocates 75% of its data for training and the remaining 25% for evaluation. Consistent with FedAvg [27], we adopt a batch size of 10 and perform $E = 1$ local epoch per communication round.

Statistical Heterogeneity Settings. We simulate two widely adopted non-IID settings: the *pathological setting* [27, 31] and the *practical setting* [20, 24]. In the pathological setting, each client is assigned data from only 2/2/10 classes for MNIST/CIFAR-10/CIFAR-100, respectively, with non-overlapping label sets and balanced per-class sample counts. In the practical setting, we partition data using a Dirichlet distribution $\text{Dir}(\beta)$ with $\beta = 0.1$, where the proportion of samples with label c allocated to client i is $Q_{c,i} \sim \text{Dir}(\beta)$.

Implementation Details. All methods are trained for 2000 communication rounds. Our implementation is based on PyTorch 1.7 and runs on a server with two In-

Table 4: Training time per round and GPU/CPU memory usage of federated learning methods on the AmazonReview dataset.

	Ours	FedALA	Ditto	FedFomo	FedPer	FedProto	FedRep	FedROD
Time cost	14.74s	15.98s	23.95s	16.03s	13.63s	21.15s	16.39s	15.87s
Memory	223.42M	618.02M	396.06M	485.41M	221.99M	221.99M	221.99M	222.00M

Table 5: The Convergence Times and iterations on Tiny-ImageNet using ResNet-18

	Total Time	Iterations	Average time
FedAvg	365 min	230	1.59 min
FedProx	325 min	163	1.99 min
Per-FedAvg	121 min	34	3.56 min
pFedMe	1157 min	113	10.24 min
FedAMP	92 min	60	1.53 min
Ditto	318 min	27	11.78 min
FedPer	83 min	43	1.92 min
FedRep	471 min	115	4.09 min
FedRoD	87 min	50	1.74 min
FedFomo	193 min	71	2.72 min
FedPHP	264 min	65	4.06 min
FedProto	416 min	72	5.78 min
FedALA	123 min	60	2.05 min
Ours	186 min	55	3.38 min

tel(R) Xeon(R) Silver 4210 CPUs, 256GB RAM, and eight NVIDIA RTX 2080 Ti GPUs under Ubuntu 16.04.

5.2. Comparison Experiment

Effectiveness. We evaluate FedDTR and baselines under both pathological (Pa) and practical (Pr) settings with $N = 20$ clients and $\beta = 0.1$. As shown in Table 1, FedDTR consistently achieves the highest accuracy across all datasets. On CIFAR-100, FedDTR outperforms the best baseline as follows:

- **vs. Global-Local Separation Methods (e.g., FedRep)**: +4.05% (Pa) and +7.28% (Pr). FedDTR mitigates local gradient misalignment by guiding local training with domain-invariant textual representations and in-domain global priors.
- **vs. Model-Based Global References (e.g., Ditto)**: +4.38% (Pa) and +6.80% (Pr). Unlike Ditto's dynamically evolving global model, FedDTR leverages stable, unbiased text embeddings as fixed global references.
- **vs. Prototype-Based Methods (e.g., FedProto)**: +2.43% (Pa) and +3.75% (Pr). FedProto's prototypes are sensitive to feature extractor noise and shift during training, whereas FedDTR's textual representations remain noise-free and invariant.

Table 6: Study on nonsensical labels

	origin	Digit	Alphanum
MNIST	99.77	99.76 (-0.01)	99.74 (-0.03)
CIFAR-100	59.67	59.63 (-0.04)	59.68 (+0.01)
AGNews	96.89	96.91 (+0.02)	96.86 (-0.03)

- **vs. Other SOTA Methods (e.g., FedALA)**: +3.78% (Pa) and +3.75% (Pr). The advantage of FedDTR becomes more pronounced with larger backbones: on Tiny-ImageNet with ResNet-18, FedDTR surpasses FedALA by **6.58%** (48.52% vs. 41.94%).

Additional results on FMNIST and Amazon Review (see Figure 3 and supplementary materials) further confirm FedDTR's superior convergence speed and final accuracy.

Heterogeneity. We vary $\beta \in \{0.01, 0.1, 0.5\}$ on Tiny-ImageNet and AG News to control heterogeneity severity (smaller β = higher heterogeneity). As shown in Table 2, most personalized methods degrade significantly as β increases (e.g., FedAMP drops by 38.95% on Tiny-ImageNet when β rises from 0.1 to 0.5), due to over-reliance on local data without robust global guidance. In contrast, FedDTR maintains consistent gains across all β values and achieves the best performance in every setting.

Scalability. We scale the number of clients $N \in \{10, 20, 30, 100\}$ on CIFAR-100 ($\beta = 0.1$). As N increases, local data per client decreases, causing significant accuracy drops for methods like FedPer (-9.26%) and FedAMP (-7.26%). FedDTR, however, exhibits the smallest performance degradation (from 60.01% at $N = 10$ to 56.51% at $N = 100$), demonstrating superior data efficiency and scalability.

Stability. We simulate unstable client participation by sampling the participation ratio J from $[0.5, 1]$ or $[0.1, 1]$ per round (vs. fixed $J = 1$). As shown in Table 3, most methods suffer severe accuracy drops under high volatility (e.g., pFedMe: -6.65%). FedDTR, by contrast, maintains robust performance with only a **0.43%** drop when $J \in [0.1, 1]$, thanks to its stable textual global priors.

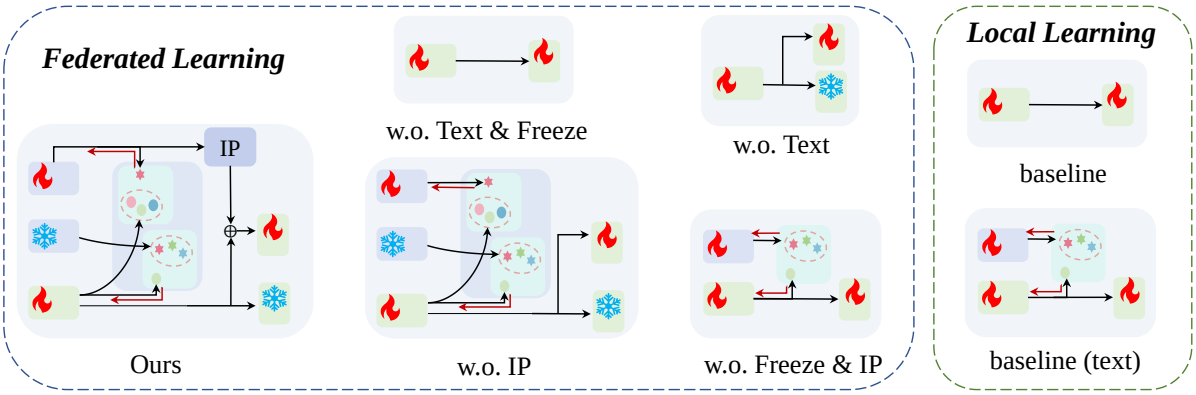


Figure 4: Illustration of variants for ablation study. "w.o." is short for "without"

Table 7: The accuracy (%) of FedDTR and its variants on Cifar-10

	Federated learning					Local learning	
	Ours	w.o. IP	w.o. Freeze & IP	w.o. Text	w.o. Text & Freeze	baseline (text)	baseline
Cifar-10	**92.11**	91.51(-0.60)	90.86(-0.65)	91.07(-1.04)	89.22(-1.85)	89.64	89.43
PACS	**64.25**	63.25(-1.00)	63.03(-1.22)	60.71(-3.29)	60.13(-4.12)	-	-
Digits	**83.32**	82.52(-0.80)	82.41(-0.91)	81.39(-1.93)	81.05(-2.27)	-	-

Table 8: The accuracy (%) on CUB-200-2011 and ISIC datasets.

Dataset	**CUB-200-2011**				**ISIC**			
Methods	FedAvg	Ditto	FedProto	**Ours**	FedAvg	Ditto	FedProto	**Ours**
Accuracy	62.30	73.81	70.16	**80.74**	68.27	74.41	74.33	**77.58**

Table 9: Parameter of different text encoders and their impact on FedDTR accuracy(%).

Text Encoder	**CLIP**	**Transformer-tiny**	**10-layer MLP**	**5-layer MLP**	**1-layer MLP**
Parameters	89 M	9.3 M	2.63 M	1.31 M	**0.26 M**
Accuracy	92.11	92.12	92.15	92.09	92.11

5.3. Convergence Time and Computational Cost

Table 5 reports convergence statistics on Tiny-ImageNet (ResNet-18). Although FedDTR incurs a moderate per-iteration time (3.38 min), it converges in only 55 rounds—far fewer than pFedMe (113), FedRep (115), or Ditto (27 but with 11.78 min/round). Overall, FedDTR achieves competitive total training time (186 min) while delivering the highest accuracy.

Moreover, as shown in Table 4, FedDTR has the lower per-round time (14.74s) and memory usage (223.42MB) among recent SOTA methods on AmazonReview, making it suitable for resource-constrained devices.

5.4. Robustness to Label Semantics

To assess FedDTR's applicability in domains without semantic labels, we replace original class descriptions (e.g., "This is a cat") with nonsensical labels: random 3–5 digit strings ("This is a 8420") or alphanumeric codes ("This is a h2t56"). Results in Table 6 show that performance changes by less than **0.05%** across MNIST, CIFAR-100, and AG News. This confirms that as long as labels are distinguishable, their semantic meaning has negligible impact—enabling FedDTR to work with arbitrary or structured textual labels.

5.5. Performance on Other Tasks

To validate the effectiveness of FedDTR on diverse tasks(N=20, $\beta = 0.1$), we conducted comparative experiments on the fine-grained CUB-200-2011 dataset [37] and the large-scale ISIC skin lesion segmentation dataset [36], benchmarking against FedAvg, Ditto, and FedProto. As shown in Table 8, FedDTR consistently achieves superior performance across both scenarios. Notably, in the challenging CUB classification task, our method attains an accuracy of 80.74%, significantly outperforming the second-best baseline (Ditto) by a margin of 6.93%, which highlights its ability to capture granular features despite high inter-class similarity. Furthermore, on the ISIC segmentation task, FedDTR reaches 77.58% accuracy compared to 68.27% for FedAvg, demonstrating its robustness and strong generalization capability in complex, pixel-level dense prediction tasks under non-IID conditions.

5.6. Impact of Text Encoder Architecture

To investigate the influence of the text encoder component within FedDTR, we evaluated five distinct text encoder variants on the CIFAR-10 dataset under a non-IID setting ($N = 20, \beta = 0.1$). The comparative results are presented in Table 9.We observed that the choice of text encoder has a negligible impact on the final global model performance, with accuracy fluctuations restricted to a narrow margin of merely 0.04%. Remarkably, the lightweight 1-layer MLP achieves an accuracy of 92.11%, matching the performance of the heavy-weight CLIP encoder (which possesses 89M parameters), despite containing significantly fewer parameters (0.26M). This result suggests that a simple linear projection is sufficient to capture the necessary semantic features for this task, rendering complex, heavy models unnecessary. Consequently, to minimize communication overhead and computational costs on client devices without compromising model accuracy, we adopt the parameter-efficient 1-layer MLP as the default text encoder in our framework.

5.7. Ablation Study

We analyze FedDTR's components on CIFAR-10 ($N = 20, \beta = 0.1$); see Table 7 and Figure 4. Removing the in-domain prior (IP) module reduces accuracy by 0.60%. Further removing the frozen text encoder causes an additional 0.65% drop. Eliminating all text-related components leads to a 1.04% decrease, and removing both text and frozen modules (i.e., reverting to a standard baseline) results in a 2.89% drop. These results highlight the critical role of textual guidance and frozen representations.

We also evaluate on multi-domain datasets (PACS and Digits), where each client holds data from a single domain. FedDTR achieves significant gains (64.25% on PACS, 83.32% on Digits), confirming that textual embeddings effectively bridge domain gaps.

Notably, even in local training (no federation), adding text embeddings yields only a 0.24% improvement. In contrast, within federated learning, the same text module boosts accuracy by 1.64%—**6.8× larger gain**—demonstrating that textual priors are especially valuable for mitigating client drift in FL.

6. Limitation

While FedDTR demonstrates superior performance and strong generalization capabilities across discriminative tasks—specifically classification and semantic segmentation—it is currently designed to operate within discrete label spaces. Consequently, the framework is not yet adapted for regression tasks (e.g., depth estimation or object detection bounding box regression), which involve predicting continuous target variables. Extending the FedDTR framework to support continuous numerical outputs remains a challenging but promising direction for our future work.

7. Conclusion

Personalized federated learning is gaining traction for addressing data heterogeneity, but existing methods often rely on "domain-variant" global representations. We propose FedDTR, which uses domain-invariant, unbiased, and noise-free text embeddings to improve intra-class similarity and inter-class separation, effectively tackling data heterogeneity. Additionally, the Intra-domain Prior module offers global priors to better model overall data distribution and reduce overfitting to local data. FedDTR outperforms 13 state-of-the-art methods in effectiveness, scalability, and stability.

References

[1] Z. Alamgir, F. K. Khan, and S. Karim. Federated recommenders: methods, challenges and future. *Cluster Computing*, 25(6):4075–4096, 2022. 1

[2] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019. 2, 3

[3] H. Chen and W. Chao. On bridging generic and personalized federated learning for image classification. In *ICLR*. OpenReview.net, 2022. 2

[4] P. Chrabaszcz, I. Loshchilov, and F. Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017. 7

[5] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pages 2089–2099. PMLR, 2021. 2

[6] 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*, 33:3557–3568, 2020. 3

[7] H. Feng, Z. You, M. Chen, T. Zhang, M. Zhu, F. Wu, C. Wu, and W. Chen. Kd3a: Unsupervised multi-source decentralized domain adaptation via knowledge distillation. In *ICML*, volume 4, page 5, 2021. 7

[8] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller. Inverting gradients - how easy is it to break privacy in federated learning? In *NeurIPS*, 2020. 7

[9] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006. 3

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[11] T.-M. H. Hsu, H. Qi, and M. Brown. Federated visual classification with real-world data distribution. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow,*

UK, August 23–28, 2020, Proceedings, Part X 16, pages 76–92. Springer, 2020. 1

[12] 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. 3

[13] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 7865–7873, 2021. 3

[14] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *EACL (2)*, pages 427–431. Association for Computational Linguistics, 2017. 7

[15] 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*, 14(1–2):1–210, 2021. 1

[16] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2(6):305–311, 2020. 1

[17] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020. 3

[18] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4), 2009. 7

[19] 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. 7

[20] 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. 7

[21] T. Li, S. Hu, A. Beirami, and V. Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021. 2, 3

[22] 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. 3

[23] X.-C. Li, D.-C. Zhan, Y. Shao, B. Li, and S. Song. Fedphp: Federated personalization with inherited private models. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part I 21*, pages 587–602. Springer, 2021. 2, 3

[24] T. Lin, L. Kong, S. U. Stich, and M. Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020. 7

[25] X. Liu, W. Xi, W. Li, D. Xu, G. Bai, and J. Zhao. Comda: Federated multi-source domain adaptation on black-box models. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 1

[26] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. In *NeurIPS*, pages 5972–5984, 2021. 7

[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. 1, 2, 7

[28] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019. 7

[29] C. Qin, H. You, L. Wang, C.-C. J. Kuo, and Y. Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[30] J. Ren, C. Yu, X. Ma, H. Zhao, S. Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020. 2

[31] A. Shamsian, A. Navon, E. Fetaya, and G. Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pages 9489–9502. PMLR, 2021. 7

[32] C. T Dinh, N. Tran, and J. Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020. 2, 3, 7

[33] A. Z. Tan, H. Yu, L. Cui, and Q. Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 1

[34] 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. 1, 2, 3

[35] 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*, 35:19332–19344, 2022. 2, 3

[36] P. Tschandl, C. Rosendahl, B. N. Akay, G. Argenziano, A. Blum, R. P. Braun, H. Cabo, J.-Y. Gourhant, J. Kreusch, A. Lallas, et al. Human-computer collaboration for skin cancer recognition. In *Nature Medicine*, volume 26, pages 1229–1234. Nature Publishing Group, 2020. 9

[37] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 9

[38] Y. Wei, L. Yang, Y. Han, and Q. Hu. Multi-source collaborative contrastive learning for decentralized domain adaptation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022. 1

[39] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 7

[40] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, and H. Guan. Fedala: Adaptive local aggregation for personalized federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11237–11244, 2023. 1, 3

[41] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez. Personalized federated learning with first order model optimization. *arXiv preprint arXiv:2012.08565*, 2020. 1, 3

[42] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015. 7