DISTILLPFL: DISTILLING FOUNDATION MODEL KNOWLEDGE FOR PERSONALIZED FEDERATED LEARNING VIA SEMANTIC PROXIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Personalized federated learning (PFL) aims to learn models adapted to heterogeneous data distributions across clients. In practical settings, substantial data heterogeneity and limited local samples can cause personalized models to overfit scarce local data and generalize poorly. Foundation models offer external knowledge that can enhance PFL. However, existing approaches that leverage foundation models for PFL often rely on a shared public dataset or require clients to execute the foundation model, limiting their applicability when suitable public data are unavailable or client resources are constrained. To tackle these challenges, we propose DistillPFL, a PFL framework that uses semantic proxies to construct predictive anchors from a server-side foundation model to distill its knowledge to client models. DistillPFL supports effective personalization under data heterogeneity across clients and limited local data, without requiring a shared public dataset or client-side execution of the foundation model. Extensive experiments across three benchmark datasets show consistent improvements over representative baselines under heterogeneous data distributions and limited local data. White-box reconstruction attacks further show that the transmitted semantic proxies resist recovery of raw samples even under a strong adversary.

1 INTRODUCTION

Personalized federated learning (PFL) aims to learn models adapted to heterogeneous data distributions across clients while benefiting from collaborative training. Achieving this goal becomes particularly challenging when data heterogeneity is high, local data are limited, and client resources are constrained. In practical deployments, a client may have access to only a small number of training samples concentrated in a subset of classes, while local label distributions can differ substantially across clients (Xia et al., 2021; Shanmugarasa et al., 2023). When local data are limited and highly non-IID, knowledge shared across clients may not adequately reflect a given client's local distribution, while the small number of local samples can make personalized models prone to overfitting and poor generalization. Client-side computation and storage constraints can further limit the expressive capacity of models that can be trained and deployed locally (Lim et al., 2020; Nguyen et al., 2021).

Existing PFL methods aim to improve personalization by accounting for differences in local data distributions. Regularization-based methods use collaboratively learned models to guide client-specific learning (Dinh et al., 2020; Li et al., 2021), but their effectiveness depends on the quality of the shared reference and sufficient local data for personalization. Representation-based methods decompose models into shared and personalized components, such as a shared feature extractor with client-specific prediction heads (Collins et al., 2021; McLaughlin & Su, 2024). This structure allows representation learning to benefit from collaboration, but each personalized head is still learned from the local samples available to that client, which can limit local predictive performance when only a small number of samples are available. Other methods share class-level knowledge, such as prototypes and soft predictions (Tan et al., 2022; Chen et al., 2023b; Zhang et al., 2024a; Zhou et al., 2025), or use client models to train generators that produce auxiliary representations or samples (Zhu et al., 2021; Zhang et al., 2022). However, both forms of knowledge are ultimately derived from participating clients and can therefore be constrained by what has been learned from scarce and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

heterogeneous local data. These limitations motivate a richer source of predictive knowledge that can complement information obtained through federated collaboration.

Foundation models offer a rich source of external knowledge through large-scale pretraining, with representations that can generalize across a broad range of downstream tasks and domains (Bommasani et al., 2021; Radford et al., 2021; Oquab et al., 2024). Recent studies have explored leveraging this knowledge to improve client models. Existing approaches, however, often rely on a shared public dataset to establish knowledge transfer between the server and clients (Caligiuri et al., 2026), or require clients to store and execute the foundation model during local training (Wu et al., 2023; Shi et al., 2024). The former assumes access to suitable public data for the target problem, while the latter introduces substantial computation and storage requirements on participating clients. Deploying the foundation model on the server avoids client-side execution, but raises a different challenge: without raw client samples or a public dataset, the server lacks task-relevant inputs through which the foundation model can provide knowledge. Moreover, under high data heterogeneity across clients, the transferred knowledge should capture diverse predictive patterns that can support adaptation across different local distributions. These considerations lead to a central question: *how can PFL leverage knowledge from a server-side foundation model under limited and heterogeneous client data, without relying on public datasets or client-side execution of the foundation model?*

To address these challenges, we propose DistillPFL, a PFL framework that distills predictive knowledge from a server-side foundation model into personalized client models through semantic proxies and multiple predictive anchors. Keeping the foundation model entirely on the server avoids the computation and storage requirements of executing it on resource-constrained clients. To obtain task-relevant foundation model knowledge without relying on a shared public dataset, clients construct semantic proxies from their local data and transmit the proxies to the server. These proxies provide task-relevant inputs to the foundation model, enabling its rich pretrained representations to be used for the target task. A server classifier maps these pretrained representations to task-specific logits, extracting predictive knowledge from the foundation model for the target task. This provides an external source of predictive guidance that complements the information learned from participating clients through federated collaboration. To accommodate data heterogeneity, the resulting logits are further organized into multiple predictive anchors for each class, preserving diverse predictive patterns rather than reducing each class to a single target. During personalized federated training, these anchors serve as distillation targets while clients collaboratively learn a shared feature extractor and retain personalized prediction heads. Through these designs, DistillPFL leverages foundation model knowledge to strengthen personalized learning under limited and heterogeneous data while avoiding reliance on public datasets and client-side execution of the foundation model.

We evaluate DistillPFL extensively across three benchmark datasets and two data heterogeneity settings using MobileNetV2 and ResNet-8 as client models. DistillPFL achieves the highest mean test accuracy across all evaluated settings. Compared with the strongest baseline in each setting, DistillPFL improves test accuracy by $1.41\%-4.03\%$ with MobileNetV2 and $1.33\%-4.54\%$ with ResNet-8. We further examine the privacy of the transmitted semantic proxies through white-box reconstruction attacks, where the reconstructed images exhibit low visual similarity to both individual private images and their group-level appearance, providing empirical evidence against successful recovery of the underlying private samples. We summarize our main contributions as follows:

- We introduce DistillPFL, a PFL framework that incorporates predictive knowledge from a server-side foundation model into personalized federated training, providing an external source of knowledge to support learning under limited local data and data heterogeneity across clients. In contrast to existing approaches that rely on shared public data or require clients to execute the foundation model, DistillPFL keeps the foundation model entirely on the server and distills its knowledge into client models without either requirement.
- We develop a heterogeneity-aware knowledge distillation mechanism that distills foundation model knowledge from the server to clients via semantic proxies and multiple predictive anchors. Semantic proxies provide informative inputs that connect private client data with the server-side foundation model without sharing raw samples or relying on public data, while multiple predictive anchors capture diverse predictive patterns under data heterogeneity across clients.
- We conduct extensive experiments across three benchmark datasets and two client architectures under limited local data and high data heterogeneity. DistillPFL consistently outper-

forms representative PFL baselines across all evaluated settings. White-box reconstruction attacks further show that the transmitted semantic proxies resist successful recovery of the underlying private samples.

2 RELATED WORK

Personalized federated learning. Personalized federated learning (PFL) seeks to learn models adapted to different local data distributions while benefiting from collaborative training. Existing methods achieve personalization through regularized local objectives (Dinh et al., 2020; Li et al., 2021), decomposition into shared and personalized model components (Collins et al., 2021; Oh et al., 2021), and adaptive sharing across clients (Chen et al., 2023a; Lee et al., 2023; Ghari & Shen, 2024). Recent approaches further personalize feature distributions (McLaughlin & Su, 2024) or account for uncertainty during personalization (Makhija et al., 2024). Beyond adapting model parameters or components, another line of work transfers class level knowledge across clients. These methods exchange local or trainable global prototypes (Tan et al., 2022; Zhang et al., 2024a), construct semantic anchors (Zhou et al., 2025), aggregate representations and soft predictions (Chen et al., 2023b), or train generators using knowledge extracted from client models (Zhu et al., 2021; Zhang et al., 2022). Although these strategies enrich the information available during local training, the resulting knowledge is still largely derived from participating clients. When local data are limited and highly heterogeneous, both local personalization and transferred knowledge can be constrained by the information available from scarce client data, motivating further study of PFL under these challenging conditions.

Foundation models in federated learning. Foundation models have recently been incorporated into federated learning in several ways. One line of work focuses on collaboratively adapting foundation models using decentralized data, including federated model construction (Yu et al., 2024), parameter efficient tuning (Sun et al., 2024), instruction tuning (Zhang et al., 2024b; Qin et al., 2025), and domain specific adaptation (Chen et al., 2025). A second line uses pretrained foundation models to improve smaller client models. Some approaches execute the foundation model locally to provide semantic or distillation guidance (Wu et al., 2023; Shi et al., 2024), while others transfer foundation model knowledge through shared data available to the server and clients (Fan et al., 2025). These studies demonstrate the potential of foundation models in federated learning, but existing designs commonly require either client side access to the foundation model or suitable shared data. Our work considers the complementary setting in which the foundation model is available only on the server and no public dataset is assumed.

3 METHOD

3.1 PROBLEM FORMULATION

We consider a federated learning system consisting of a central server and N clients. Each client $i \in [N]$ holds a local dataset $\mathcal{D}_i$. The local data distributions are non-IID across clients. Let w denote the model parameters. For client i, the local objective is defined as

$$f_i(w) = \frac{1}{|\mathcal{D}_i|} \sum_{(x,y)\in\mathcal{D}_i} \ell(w; x, y), \tag{1}$$

where $\ell(w; x, y)$ denotes the loss evaluated on sample (x, y). The standard FL objective is

$$\min_w f(w) := \sum_{i=1}^{N} p_i f_i(w), \text{ where } p_i = \frac{|\mathcal{D}_i|}{\sum_{j=1}^{N} |\mathcal{D}_j|}. \tag{2}$$

This formulation learns a single global model shared by all clients. In PFL, the objective is instead to learn a collection of client-specific models $\{w_i\}_{i=1}^{N}$, where each w_i is adapted to the local data distribution of client i while still leveraging information shared across clients.

3.2 OUR FRAMEWORK

In DistillPFL, each client-specific model w_i is instantiated as a composition of a feature extractor ϕ_i and a personalized head h_i, whose parameters are denoted by θ_i^{ϕ} and θ_i^{h}, respectively. Given

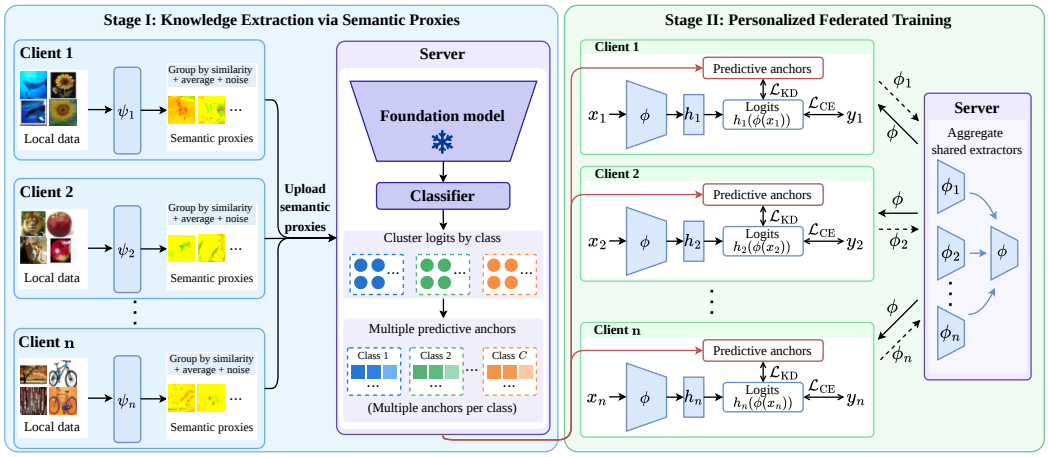


Figure 1: Overview of DistillPFL. Stage I constructs semantic proxies locally and uses a frozen server-side foundation model to derive multiple predictive anchors, while Stage II distills these anchors into personalized client models during federated training.

an input x, its prediction is $h_i\big(\phi_i(x;\theta_i^{\phi});\theta_i^{h}\big)$. When no ambiguity arises, we omit the explicit parameter arguments and write the prediction as $h_i(\phi_i(x))$. DistillPFL follows a two-stage learning process, as illustrated in Fig. 1, with the complete training procedure summarized in Algorithm 1. In the first stage, clients construct semantic proxies from their local samples, and the server processes these proxies with a foundation model to produce class-wise predictive anchors. In the second stage, the resulting anchors are used for personalized federated training. This design enables knowledge distillation from the server-side foundation model to client models without sharing raw local samples or deploying the foundation model on clients.

Stage I: knowledge extraction via semantic proxies. The first stage extracts predictive knowledge from the server-side foundation model through semantic proxies constructed locally from client data. Each client constructs semantic proxies using a learnable transformation module ψ_i, which transforms local samples while preserving information relevant to the target task. Training ψ_i requires a meaningful local representation space to evaluate the transformed samples and preserve their semantic characteristics. Therefore, each client first pretrains a local model for a small number of epochs and then keeps it fixed while optimizing ψ_i. The pretrained feature extractor is subsequently used to measure semantic similarity among local samples during proxy construction.

Let θ_i^{ψ} denote the parameters of ψ_i, and let $\psi_i(x)$ denote the transformed sample for a local input x. We optimize ψ_i using a classification loss and a contrastive loss while keeping the pretrained local model fixed. Specifically, for client i, let $\mathcal{B} = (x_j, y_j)_{j=1}^{|\mathcal{B}|} \subseteq \mathcal{D}_i$ denote a local mini-batch. The classification loss encourages the transformed samples to preserve information relevant to class discrimination and is defined as

$$\mathcal{L}_i^{\mathrm{cls}} := \frac{1}{|\mathcal{B}|} \sum_{j=1}^{|\mathcal{B}|} \mathrm{CE}\left(h_i\big(\phi_i(\psi_i(x_j))\big), y_j\right), \tag{3}$$

where $\mathrm{CE}(\cdot,\cdot)$ denotes the cross-entropy loss.

Beyond class-level discriminability, the transformed samples should retain sample-specific semantic characteristics of the local data, so that the resulting semantic proxies provide informative inputs to the server-side foundation model. We therefore introduce a contrastive loss that encourages each transformed sample to remain semantically related to its original sample. Let $\mathbf{z}_j = \frac{\phi_i(x_j)}{\|\phi_i(x_j)\|_2}$ and $\tilde{\mathbf{z}}_j = \frac{\phi_i(\psi_i(x_j))}{\|\phi_i(\psi_i(x_j))\|_2}$ denote the normalized features of the raw and transformed samples, respectively. For $\tilde{\mathbf{z}}_j$, the corresponding raw feature $\mathbf{z}_j$ forms the positive pair, while raw features from different classes are used as negatives. Let $\mathcal{N}_j = \big\{k \in \{1,\dots,|\mathcal{B}|\} | y_k \neq y_j\big\}$ denote the indices of the

negative samples. The contrastive loss is then defined as

$$\mathcal{L}_i^{\text{con}} = -\frac{1}{|\mathcal{B}|}\sum_{j=1}^{|B|}\log\frac{\exp(\tilde{\mathbf{z}}_j^\top \mathbf{z}_j/\tau_{\text{con}})}{\exp(\tilde{\mathbf{z}}_j^\top \mathbf{z}_j/\tau_{\text{con}}) + \sum_{k\in\mathcal{N}_j}\exp(\tilde{\mathbf{z}}_j^\top \mathbf{z}_k/\tau_{\text{con}})}, \tag{4}$$

where τ_{con} is the temperature parameter. This contrastive loss pulls the transformed feature toward the feature of its corresponding raw sample while separating it from raw features of different classes. The classification and contrastive losses are combined into the following training objective for the transformation module:

$$\min_{\theta_i^\psi} \mathcal{L}_i^{\text{cls}} + \lambda_{\text{con}}\mathcal{L}_i^{\text{con}}, \tag{5}$$

where λ_{con} is the weighting parameter for the contrastive loss.

After training ψ_i, each client constructs semantic proxies from its transformed samples. To avoid mixing transformed samples with substantially different semantic characteristics, local samples are first grouped within each class according to the similarity of their original features extracted by ϕ_i. The transformed samples within each group are then averaged, followed by Gaussian perturbation to form a semantic proxy. Let $\mathcal{G}_{i,c,k}$ denote the k-th group of class c at client i. The corresponding proxy is constructed and perturbed as

$$\hat{x}_{i,c,k} = \frac{1}{|\mathcal{G}_{i,c,k}|}\sum_{x\in\mathcal{G}_{i,c,k}}\psi_i(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}). \tag{6}$$

To ensure scalability to large client populations, we impose a predefined collection budget for each class, so that the number of semantic proxies processed by the server remains bounded regardless of the number of clients. Each client reports the number of locally available proxies, and the server allocates collection quotas accordingly. The server then processes the collected proxies with a fixed foundation model and trains a classifier to produce logits. Proxies from the same class can yield different predictive patterns under data heterogeneity across clients. We therefore retain multiple predictive anchors for each class by clustering their logits into M groups. For class c, the cluster centers form the predictive anchor set $\mathcal{A}_c = \{\mathbf{a}_{c,1}, \ldots, \mathbf{a}_{c,M}\}$, where $\mathbf{a}_{c,m}$ denotes the m-th predictive anchor of class c. These predictive anchors provide class-wise guidance for personalized federated training in Stage II.

Stage II: personalized federated training. Before federated training begins, the server distributes the predictive anchor sets to clients, which cache them locally throughout Stage II. Since each class is represented by multiple predictive anchors, each sample is guided by the anchor that is most compatible with the client's current prediction. For a sample (x, y), let $m^*(x)$ denote the index of the anchor in $\mathcal{A}_y$ with the smallest Kullback–Leibler divergence to the current prediction of client i. Let $D_{\text{KL}}(\cdot\|\cdot)$ denote the Kullback–Leibler divergence and τ_{KD} the distillation temperature. For a mini-batch $\mathcal{B}$, the distillation loss is defined as

$$\mathcal{L}_i^{\text{KD}} := \frac{\tau_{\text{KD}}^2}{|\mathcal{B}|}\sum_{(x,y)\in\mathcal{B}} D_{\text{KL}}\Big(\text{softmax}\Big(\frac{\mathbf{a}_{y,m^*}}{\tau_{\text{KD}}}\Big)\Big\|\,\text{softmax}\Big(\frac{h_i(\phi_i(x))}{\tau_{\text{KD}}}\Big)\Big), \tag{7}$$

The local objective combines the predictive guidance from the foundation model with supervision from the local labels:

$$\mathcal{L}_i^{\text{local}} = \frac{1}{|\mathcal{B}|}\sum_{(x,y)\in\mathcal{B}}\text{CE}(h_i(\phi_i(x)), y) + \lambda_{\text{KD}}\mathcal{L}_i^{\text{KD}}, \tag{8}$$

where λ_{KD} is the weighting parameter for the distillation loss. This objective is optimized by first updating the personalized head while keeping the shared feature extractor fixed, followed by updating the feature extractor while keeping the personalized head fixed. After local training, the updated feature extractors are uploaded to the server for aggregation, while each client's personalized head remains local. This decomposition allows the feature representation to benefit from collaborative learning across clients, while the locally retained prediction head preserves adaptation to each client's data distribution.

Algorithm 1 DistillPFL Training Procedure

Input: Local datasets $\{\mathcal{D}_i\}_{i=1}^N$; a fixed server foundation model; class-wise proxy collection budget; number of predictive anchors per class M; number of classes C; number of communication rounds T.

Stage I: Knowledge extraction via semantic proxies

for each client $i \in [N]$ **in parallel do**
  Initialize the local model (ϕ_i, h_i) and transformation module ψ_i.
  Pretrain the local model on $\mathcal{D}_i$.
  Optimize θ_i^{ψ} using Eq. (5) with the pretrained local model fixed.
  Group samples within each class by the similarity of their original features $\phi_i(x)$.
  Construct semantic proxies $\hat{x}_{i,c,k}$ using Eq. (6).
  Send the number of available proxies per class to the server.
end for
Server allocates client collection quotas using the reported proxy counts and the class-wise budget.
Server collects the requested semantic proxies from clients.
Server extracts proxy features using the fixed foundation model and trains the server classifier.
Server computes proxy logits and clusters them into M groups within each class.
Server retains the M cluster centers for each class c as the predictive anchor set $\mathcal{A}_c$.
Server broadcasts the predictive anchor sets $\{\mathcal{A}_c\}_{c=1}^C$ to clients, which cache them locally.

Stage II: Personalized federated training

for $t = 1, \ldots, T$ **do**
  Server samples a subset of participating clients $\mathcal{S}_t \subseteq [N]$.
  Server sends the shared feature extractor parameters θ^{ϕ} to clients in $\mathcal{S}_t$.
  for each client $i \in \mathcal{S}_t$ **in parallel do**
    Set $\theta_i^{\phi} \leftarrow \theta^{\phi}$.
    Update θ_i^h using Eq. (8), with θ_i^{ϕ} fixed.
    Update θ_i^{ϕ} using Eq. (8), with θ_i^h fixed.
    Communicate the updated feature extractor parameters θ_i^{ϕ} to server.
  end for
  Server aggregates the uploaded feature extractors to update θ^{ϕ}.
end for

4 Experiments

4.1 Experimental setup

Datasets and models. We conduct experiments on CIFAR-100 (Krizhevsky et al., 2009), GTSRB (Stallkamp et al., 2011), and PatternNet (Zhou et al., 2018) using MobileNetV2 (Sandler et al., 2018) and ResNet-8 (He et al., 2016) as lightweight client models. We consider two forms of data heterogeneity: label-per-client (LPC) heterogeneity and Dirichlet heterogeneity. Under LPC heterogeneity, each client is assigned five classes. Under Dirichlet heterogeneity, each client independently samples a class-proportion vector from a Dirichlet distribution with concentration parameter $\alpha = 0.05$, and its local samples are allocated according to the resulting class proportions. We refer to these two settings as LPC-5 and $\mathrm{Dir}(0.05)$, respectively. To model a more practical low-data setting, we assign each client an exact training budget following a long-tailed client budget distribution. The budgets range from 100 to 300 samples for CIFAR-100 and from 50 to 150 samples for GTSRB and PatternNet, with most clients receiving relatively few training samples and only a small fraction receiving more.

Baselines. We consider FedAvg (McMahan et al., 2017), which learns a single global model, together with six representative methods related to personalized learning or knowledge transfer in FL. FedRep (Collins et al., 2021) learns a shared feature representation while maintaining a personalized classifier for each client, providing a direct architectural comparison with our personalized federated training stage. pFedFDA (McLaughlin & Su, 2024) further separates representation learning from classifier personalization by modeling class-conditional feature distributions and adapting the resulting classifier to each client. FedProto (Tan et al., 2022) transfers class prototypes across clients,

while FedTGP (Zhang et al., 2024a) learns trainable global prototypes through adaptive-margin contrastive learning. FedSA (Zhou et al., 2025) uses semantic anchors to guide local representation learning and classifier calibration. FedHKD (Chen et al., 2023b) aggregates class-level representations and soft predictions to provide feature and prediction supervision to local models.

Implementation. We use 100 clients and randomly select 10% of them to participate in each communication round. DINOv2 ViT-L/14 (Oquab et al., 2024) is used as the server-side foundation model. Each experiment is repeated three times, and we report mean test accuracy with standard deviation. Further details are shown in Appendix B.

Table 1: Test accuracy (%) with MobileNetV2 under different datasets and data heterogeneity settings. Results are reported as mean ± standard deviation over three runs.

Method	CIFAR-100		GTSRB		PatternNet	
	LPC-5	Dir(0.05)	LPC-5	Dir(0.05)	LPC-5	Dir(0.05)
FedAvg	15.91 ± 0.38	16.66 ± 0.55	20.42 ± 1.22	19.05 ± 0.93	13.94 ± 0.09	14.33 ± 0.04
FedRep	55.56 ± 0.67	52.44 ± 0.53	72.79 ± 0.21	75.16 ± 0.36	49.87 ± 0.50	59.19 ± 0.55
pFedFDA	57.04 ± 0.25	53.89 ± 0.42	68.20 ± 1.20	65.43 ± 0.39	46.51 ± 0.82	52.93 ± 1.21
FedProto	25.11 ± 0.52	18.82 ± 0.85	33.99 ± 0.20	28.85 ± 0.65	20.24 ± 0.45	20.09 ± 0.37
FedTGP	46.05 ± 0.24	42.82 ± 0.83	56.05 ± 0.64	54.82 ± 0.73	43.31 ± 1.15	50.65 ± 1.37
FedSA	47.47 ± 0.41	46.44 ± 0.26	63.58 ± 1.38	63.43 ± 0.78	45.61 ± 1.66	54.05 ± 0.58
FedHKD	56.08 ± 0.64	53.01 ± 0.20	61.44 ± 0.47	64.09 ± 1.11	52.66 ± 0.22	54.67 ± 0.86
DistillPFL	$\mathbf{58.45 \pm 0.26}$	$\mathbf{55.32 \pm 0.65}$	$\mathbf{76.82 \pm 0.41}$	$\mathbf{78.31 \pm 0.46}$	$\mathbf{55.13 \pm 0.88}$	$\mathbf{63.21 \pm 0.42}$

Table 2: Test accuracy (%) with ResNet-8 under different datasets and data heterogeneity settings. Results are reported as mean ± standard deviation over three runs.

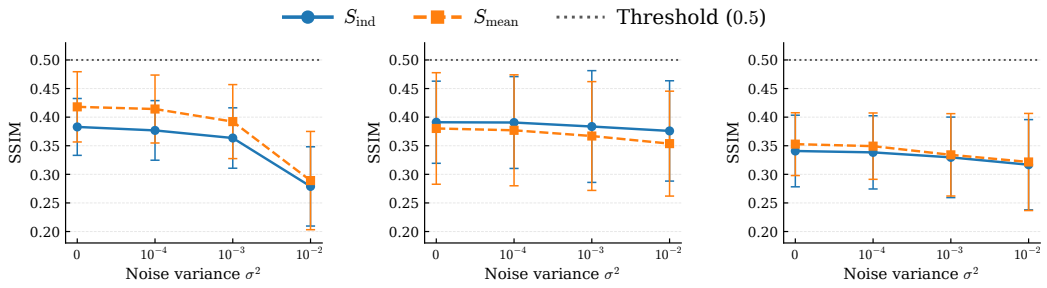
Method	CIFAR-100		GTSRB		PatternNet	
	LPC-5	Dir(0.05)	LPC-5	Dir(0.05)	LPC-5	Dir(0.05)
FedAvg	21.71 ± 0.66	22.10 ± 0.89	29.51 ± 1.59	27.90 ± 1.72	24.29 ± 0.31	23.42 ± 0.46
FedRep	70.12 ± 0.28	65.01 ± 0.35	82.19 ± 0.79	81.60 ± 0.92	74.56 ± 0.85	76.44 ± 0.57
pFedFDA	71.75 ± 0.41	66.88 ± 0.34	84.25 ± 0.15	82.72 ± 0.91	76.47 ± 0.39	77.94 ± 0.29
FedProto	35.07 ± 0.24	48.19 ± 0.63	53.20 ± 0.71	54.84 ± 0.37	52.04 ± 0.92	54.41 ± 0.88
FedTGP	55.73 ± 0.30	52.82 ± 0.56	67.59 ± 0.83	66.28 ± 0.67	59.23 ± 1.17	64.87 ± 1.03
FedSA	62.49 ± 0.12	59.19 ± 0.21	78.43 ± 0.27	79.58 ± 1.07	65.63 ± 0.53	71.82 ± 1.89
FedHKD	65.15 ± 0.44	62.34 ± 0.56	83.13 ± 1.74	82.40 ± 1.18	66.79 ± 0.54	72.48 ± 1.45
DistillPFL	$\mathbf{73.09 \pm 0.22}$	$\mathbf{68.21 \pm 0.15}$	$\mathbf{86.37 \pm 0.52}$	$\mathbf{84.61 \pm 0.83}$	$\mathbf{80.63 \pm 0.84}$	$\mathbf{82.48 \pm 0.76}$

4.2 Main results

Table 1 and Table 2 report the test accuracy of MobileNetV2 and ResNet-8, respectively. DistillPFL consistently achieves the highest mean accuracy across all evaluated settings. Compared with the strongest baseline in each setting, DistillPFL improves accuracy by $1.41\%-4.03\%$ with MobileNetV2 and $1.33\%-4.54\%$ with ResNet-8. Representation-based PFL methods such as FedRep and pFedFDA generally outperform FedAvg and methods relying primarily on client-derived class-level knowledge, highlighting both the importance of personalization and the limitations of knowledge extracted solely from scarce local data. DistillPFL addresses this limitation by introducing predictive knowledge from a server-side foundation model as an additional source of guidance for personalized learning. Semantic proxies enable the server to utilize representations acquired through large-scale pretraining for the target task, while multiple predictive anchors preserve diverse predictive patterns for subsequent distillation. The consistent improvements over both personalization methods and methods that transfer client-derived knowledge indicate that foundation model knowledge can effectively complement the information learned through federated collaboration under limited and heterogeneous local data.

4.3 Privacy analysis

We further examine whether the semantic proxies transmitted by clients expose recoverable information about their source samples. Under our protocol, the transformation module ψ_i is trained



(a) CIFAR-100 (b) GTSRB (c) PatternNet

Figure 2: Reconstruction SSIM under different Gaussian noise variances. Each panel reports the SSIM for individual source images (S_{ind}) and the pixel-wise group mean (S_{mean}). Error bars indicate standard deviations across clients, and the dotted line denotes the reconstruction threshold.

and retained locally, and the resulting semantic proxies are transmitted to the server. To stress-test privacy against a deliberately strong adversary, we grant the attacker information that is not available under the actual protocol: full white-box access to the target client's transformation module, including both its architecture and trained parameters. This assumption allows the attacker to directly optimize through ψ_i and removes uncertainty about the transformation itself. Following the optimization-based reconstruction paradigm used in prior attacks (Geiping et al., 2020; Yin et al., 2021), the attacker keeps ψ_i fixed and optimizes a randomly initialized candidate image u for each observed proxy $\hat{x}_{i,c,k}$ by minimizing

$$\mathcal{L}_{inv}(u) = \text{MSE}\left(\psi_i(u), \hat{x}_{i,c,k}\right) + \lambda_{\text{TV}}\,\text{TV}(u_{\text{unit}}), \tag{9}$$

where u_{unit} denotes the candidate represented in $[0,1]$, and the total variation (TV) term provides an image prior. A sigmoid parameterization constrains the candidate to the valid input range throughout optimization. We use Adam for $20{,}000$ steps with a learning rate of 10^{-2}.

We measure reconstruction fidelity using SSIM (Wang et al., 2004). Because each semantic proxy is constructed from a group of transformed samples, a reconstruction may reveal information about either an individual source image or visual structure shared across the group. We therefore evaluate both forms of reconstruction fidelity. Let $\tilde{x}_{i,c,k}$ denote the reconstruction and $\bar{x}_{i,c,k} = |\mathcal{G}_{i,c,k}|^{-1} \sum_{x \in \mathcal{G}_{i,c,k}} x$ denote the pixel-wise mean of the original images. We evaluate

$$S_{\text{ind}} = \max_{x \in \mathcal{G}_{i,c,k}} \text{SSIM}(\tilde{x}_{i,c,k}, x), \quad S_{\text{mean}} = \text{SSIM}(\tilde{x}_{i,c,k}, \bar{x}_{i,c,k}). \tag{10}$$

By taking the maximum, S_{ind} measures the highest structural similarity between the reconstruction and any source image contributing to the proxy, while S_{mean} measures the structural similarity between the reconstruction and the pixel-wise group mean. Following prior reconstruction studies (Zhao et al., 2024; Scheliga et al., 2025; Xu et al., 2025), we use an SSIM of 0.5 as a reference threshold for successful reconstruction.

We conduct the privacy evaluation using MobileNetV2 under $\text{Dir}(0.05)$ on all three datasets. For each dataset, we randomly select five of the 100 clients and attack all semantic proxies constructed from their local data. We vary the Gaussian noise variance over $\sigma^2 \in \{0, 10^{-4}, 10^{-3}, 10^{-2}\}$, while keeping the trained transformation module and source groups fixed across noise levels for each client. For each client and variance, we select $\lambda_{\text{TV}} \in \{0, 10^{-3}, 10^{-2}, 10^{-1}\}$ to maximize the average S_{ind} over that client's proxies. All images are rescaled to $[0,1]$ for SSIM evaluation.

Figure 2 shows the reconstruction SSIM for individual source images and the pixel-wise group mean under different Gaussian noise variances. Both S_{ind} and S_{mean} remain below the reconstruction threshold of 0.5 across all evaluated datasets. Reconstruction SSIM also generally decreases as the noise variance increases, indicating that Gaussian perturbation further reduces the visual information recoverable from the semantic proxies. Even when the attacker is granted full access to the locally retained transformation module ψ_i, the reconstructed images exhibit only limited structural similarity to individual source images and their pixel-wise group means.

4.4 ABLATION STUDY

We conduct component ablations to examine three key designs of DistillPFL: knowledge distillation from the server-side foundation model, the contrastive objective used for semantic proxy construction, and the use of multiple predictive anchors. All ablation experiments use MobileNetV2. For each configuration, we disable only the corresponding component while keeping the remaining model components and training settings unchanged. The results are summarized in Table 3.

Table 3: Component ablation of DistillPFL using MobileNetV2. We respectively disable knowledge distillation by setting $\lambda_{\mathrm{KD}} = 0$, disable the contrastive loss by setting $\lambda_{\mathrm{con}} = 0$, or retain a single predictive anchor per class by setting $M = 1$. Results are reported as mean $\pm$ standard deviation over three runs.

Configuration	CIFAR-100		GTSRB		PatternNet	
	LPC-5	Dir(0.05)	LPC-5	Dir(0.05)	LPC-5	Dir(0.05)
w/o KD	55.87 ± 0.73	53.09 ± 0.54	72.26 ± 0.30	75.12 ± 0.58	49.24 ± 0.67	59.25 ± 0.60
w/o contrastive loss	57.08 ± 0.44	54.35 ± 0.39	75.48 ± 0.37	76.77 ± 0.80	53.39 ± 0.58	62.15 ± 0.32
Single anchor ($M = 1$)	56.89 ± 0.32	54.22 ± 0.49	74.67 ± 0.53	76.90 ± 0.36	52.77 ± 0.59	62.10 ± 0.79
DistillPFL	$\mathbf{58.45 \pm 0.26}$	$\mathbf{55.32 \pm 0.65}$	$\mathbf{76.82 \pm 0.41}$	$\mathbf{78.31 \pm 0.46}$	$\mathbf{55.13 \pm 0.88}$	$\mathbf{63.21 \pm 0.42}$

Contribution of knowledge distillation from the foundation model. We examine whether knowledge distilled from the server-side foundation model contributes to personalized learning by setting $\lambda_{\mathrm{KD}} = 0$. As shown in Table 3, removing the distillation objective reduces test accuracy across all datasets and both data heterogeneity settings, with decreases ranging from 2.23% to 5.89%. The degradation is particularly pronounced on GTSRB and PatternNet. These results show that the predictive knowledge extracted from the server-side foundation model provides useful guidance beyond local labels and collaboratively learned representations, improving personalization when local data are limited and heterogeneous across clients.

Contribution of the contrastive objective. We examine the contrastive loss used during semantic proxy construction by setting $\lambda_{\mathrm{con}} = 0$, such that the transformation module is trained only with the classification objective. Removing the contrastive loss consistently reduces test accuracy, with decreases ranging from 0.97% to 1.74%. While the classification objective preserves class-level discriminability, the contrastive objective additionally encourages transformed samples to remain aligned with their corresponding local representations. The resulting improvements indicate that preserving this sample-specific semantic information helps construct more informative semantic proxies and, consequently, supports more effective extraction of predictive knowledge from the foundation model.

Effect of multiple predictive anchors. We examine the contribution of representing each class with multiple predictive anchors by replacing them with a single anchor per class ($M = 1$). The full DistillPFL consistently outperforms the single-anchor configuration, with improvements ranging from 1.10% to 2.36%. A single class-level anchor compresses the predictions obtained from different semantic proxies into one target and can therefore discard useful predictive variation within a class. In contrast, multiple predictive anchors preserve diverse predictive patterns derived from the foundation model and provide more suitable distillation targets for clients with different local data distributions.

5 CONCLUSION

We propose DistillPFL, a PFL framework that distills predictive knowledge from a server-side foundation model through semantic proxies and multiple predictive anchors. By keeping the foundation model on the server, DistillPFL avoids reliance on public data and client-side execution while providing additional knowledge for personalized learning under limited and heterogeneous local data. Extensive experiments demonstrate consistent improvements over representative baselines. Reconstruction analysis further shows limited recoverability of information from the transmitted semantic proxies. Overall, these results highlight the potential of server-side foundation models to provide complementary knowledge for personalized federated learning.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing and improving the clarity of the manuscript. We reviewed all AI-assisted content and revised it as necessary. We take responsibility for the final content of this work, including any text produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, personally identifiable information, or sensitive data. All experiments are conducted using publicly available benchmark datasets. We are not aware of any specific ethical concerns arising from this work.

REPRODUCIBILITY STATEMENT

We provide the complete training procedure of DistillPFL in the main paper. Detailed dataset partitioning and training configurations, including client sampling, optimization settings, communication rounds, and model architectures, are provided in the experimental settings. We further report the hyperparameter configurations used for all datasets, heterogeneity settings, and client architectures to facilitate reproduction. An anonymous implementation of DistillPFL is available at `https://anonymous.4open.science/r/review-code-112F/`.

REFERENCES

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Matteo Caligiuri, Francesco Barbato, Donald Shenaj, Umberto Michieli, and Pietro Zanuttigh. FedPromo: Federated lightweight proxy models at the edge for fine-grained image classification with foundation models. *IEEE Internet of Things Journal*, 2026.

Daoyuan Chen, Liuyi Yao, Dawei Gao, Bolin Ding, and Yaliang Li. Efficient personalized federated learning via sparse model-adaptation. In *International Conference on Machine Learning*, pp. 5234–5256. PMLR, 2023a.

Huancheng Chen, Chianing Wang, and Haris Vikalo. The Best of Both Worlds: Accurate global and personalized models through federated learning with data-free hyper-knowledge distillation. In *The Eleventh International Conference on Learning Representations*, 2023b.

Shengchao Chen, Guodong Long, Jing Jiang, and Chengqi Zhang. Federated foundation models on heterogeneous time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15839–15847, 2025.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, pp. 2089–2099. PMLR, 2021.

Canh T. Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020.

Tao Fan, Guoqiang Ma, Yan Kang, Hanlin Gu, Yuanfeng Song, Lixin Fan, Kai Chen, and Qiang Yang. Fedmkt: Federated mutual knowledge transfer for large and small language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 243–255, 2025.

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.

Pouya M. Ghari and Yanning Shen. Personalized federated learning with mixture of models for adaptive prediction and model fine-tuning. *Advances in Neural Information Processing Systems*, 37:92155–92183, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Royson Lee, Minyoung Kim, Da Li, Xinchi Qiu, Timothy Hospedales, Ferenc Huszár, and Nicholas Lane. Fedl2p: Federated learning to personalize. *Advances in Neural Information Processing Systems*, 36:14818–14836, 2023.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pp. 6357–6368. PMLR, 2021.

Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE communications surveys & tutorials*, 22(3):2031–2063, 2020.

Disha Makhija, Joydeep Ghosh, and Nhat Ho. A bayesian approach for personalized federated learning in heterogeneous settings. *Advances in Neural Information Processing Systems*, 37: 102428–102455, 2024.

Connor J McLaughlin and Lili Su. Personalized federated learning via feature distribution adaptation. *Advances in Neural Information Processing Systems*, 37:77038–77059, 2024.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282. Pmlr, 2017.

Dinh C Nguyen, Ming Ding, Pubudu N Pathirana, Aruna Seneviratne, Jun Li, and H Vincent Poor. Federated learning for internet of things: A comprehensive survey. *IEEE communications surveys & tutorials*, 23(3):1622–1658, 2021.

Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Towards enhanced representation for federated image classification. *arXiv preprint arXiv:2106.06042*, 2021.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Zhen Qin, Zhaomin Wu, Bingsheng He, and Shuiguang Deng. Federated data-efficient instruction tuning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15550–15568, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PmLR, 2021.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520. Ieee, 2018.

Daniel Scheliga, Patrick Mäder, and Marco Seeland. Privacy preserving federated learning with convolutional variational bottlenecks. *Cybersecurity*, 8(1):33, 2025.

Yashothara Shanmugarasa, Hye-young Paik, Salil S Kanhere, and Liming Zhu. A systematic review of federated learning from clients' perspective: challenges and solutions: Y. shanmugarasa et al. *Artificial Intelligence Review*, 56(Suppl 2):1773–1827, 2023.

Jiangming Shi, Shanshan Zheng, Xiangbo Yin, Yang Lu, Yuan Xie, and Yanyun Qu. Clip-guided federated learning on heterogeneity and long-tailed data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14955–14963, 2024.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 International Joint Conference on Neural Networks*, pp. 1453–1460. IEEE, 2011.

Guangyu Sun, Umar Khalid, Matias Mendieta, Pu Wang, and Chen Chen. Exploring parameter-efficient fine-tuning to enable foundation models in federated learning. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 8015–8024. IEEE, 2024.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8432–8440, 2022.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Xidong Wu, Wan-Yi Lin, Devin Willmott, Filipe Condessa, Yufei Huang, Zhenzhen Li, and Madan Ravi Ganesh. Leveraging foundation models to improve lightweight clients in federated learning. *arXiv preprint arXiv:2311.08479*, 2023.

Qi Xia, Winson Ye, Zeyi Tao, Jindi Wu, and Qun Li. A survey of federated learning for edge computing: Research problems and solutions. *High-Confidence Computing*, 1(1):100008, 2021.

Xiangrui Xu, Zhize Li, Yufei Han, Bin Wang, Jiqiang Liu, and Wei Wang. From risk to resilience: Towards assessing and mitigating the risk of data reconstruction attacks in federated learning. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 3141–3160, 2025.

Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16332–16341. IEEE, 2021.

Sixing Yu, Juan Pablo Munoz, and Ali Jannesari. Federated foundation models: Privacy-preserving and collaborative learning for large models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 7174–7184, 2024.

Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 16768–16776, 2024a.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP 2024-2024 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 6915–6919. IEEE, 2024b.

Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10164–10173. IEEE, 2022.

Joshua C Zhao, Atul Sharma, Ahmed Roushdy Elkordy, Yahya H Ezzeldin, Salman Avestimehr, and Saurabh Bagchi. Loki: Large-scale data reconstruction attack against federated learning through model manipulation. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 1287–1305. IEEE, 2024.

Weixun Zhou, Shawn Newsam, Congmin Li, and Zhenfeng Shao. PatternNet: A benchmark dataset for performance evaluation of remote sensing image retrieval. *ISPRS journal of photogrammetry and remote sensing*, 145:197–209, 2018.

Yanbing Zhou, Xiangmou Qu, Chenlong You, Jiyang Zhou, Jingyue Tang, Xin Zheng, Chunmao Cai, and Yingbo Wu. Fedsa: A unified representation learning via semantic anchors for prototype-based federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23009–23017, 2025.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning*, pp. 12878–12889. PMLR, 2021.

A Additional privacy experiments

To complement the quantitative privacy evaluation in the main text, we provide qualitative examples of the white-box reconstruction attack. For each example, we show the three source images used to construct a semantic proxy, their pixel-wise group mean, and the reconstructed image. All examples use a Gaussian noise variance of $\sigma^2 = 10^{-4}$ and the attacker-favorable λ_{TV} selected in the main privacy evaluation.

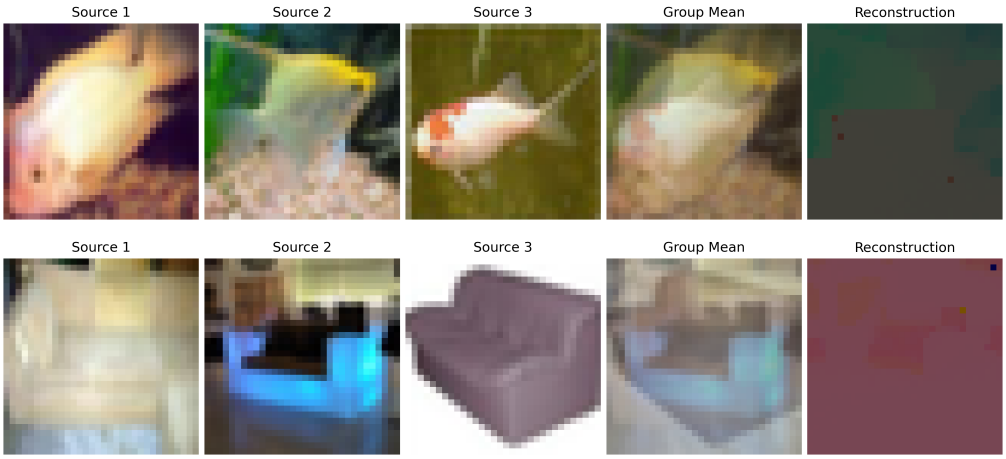


Figure 3: Qualitative reconstruction examples on CIFAR-100. Each example shows three source images, their pixel-wise group mean, and the reconstructed image.

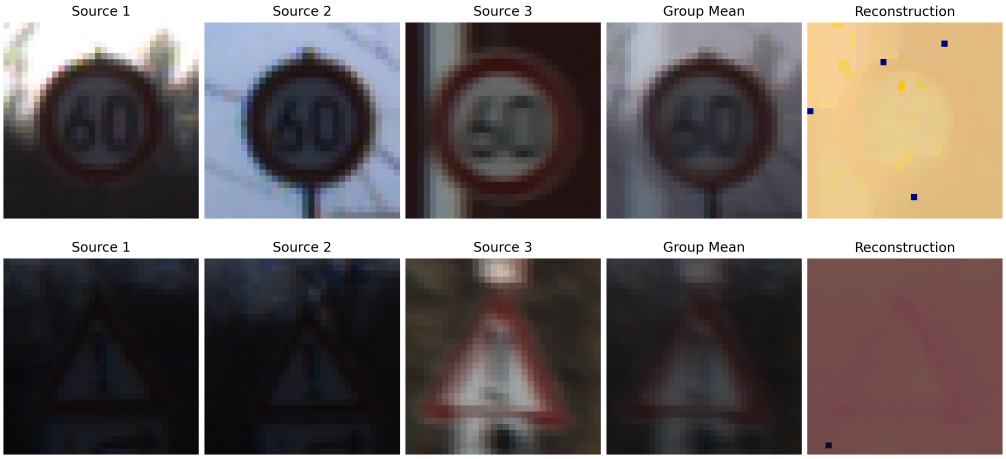


Figure 4: Qualitative reconstruction examples on GTSRB. Each example shows three source images, their pixel-wise group mean, and the reconstructed image.

Figures 3–5 provide a visual interpretation of the quantitative reconstruction results reported in the main text. The reconstructed images contain little visually recognizable source-specific information

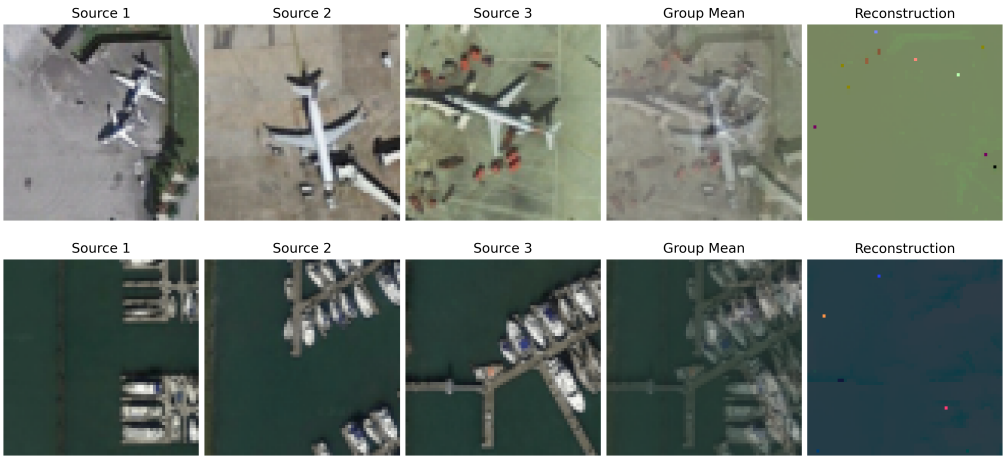


Figure 5: Qualitative reconstruction examples on PatternNet. Each example shows three source images, their pixel-wise group mean, and the reconstructed image.

and fail to recover identifiable object content or fine-grained visual details from the source images. These qualitative results are consistent with the low reconstruction SSIM observed across all three datasets and further demonstrate the limited recoverability of the underlying private samples from the transmitted semantic proxies.

B Experimental Settings

B.1 Dataset Partitioning

We construct federated data partitions using the same procedure for CIFAR-100, GTSRB, and PatternNet. Each dataset is divided among $N = 100$ clients. We consider two forms of data heterogeneity. Under LPC-5, each client is assigned exactly five classes, and its local samples are drawn only from these assigned classes. Under Dir(0.05), each client independently samples a class-proportion vector from a Dirichlet distribution with concentration parameter $\alpha = 0.05$, and its local samples are allocated according to the resulting proportions. To create a limited-local-data setting, we do not distribute the entire available training set across clients. Instead, each client is restricted to an exact training budget, so that only a subset of the original training samples is used for federated training. The client training budgets follow a predefined long-tailed profile. For $i \in \{0, \ldots, N-1\}$, we generate

$$B_i = \text{round}\left(B_{\min} + (B_{\max} - B_{\min}) \frac{\exp\left(-\beta i/(N-1)\right) - \exp(-\beta)}{1 - \exp(-\beta)} \right),$$

where $B_{\min}$ and $B_{\max}$ denote the minimum and maximum local training budgets, respectively, and β controls the degree of variation in local data sizes. We set $\beta = 3$ for all datasets. The resulting budget values are randomly permuted across clients before data allocation. We use $B_{\min} = 100$ and $B_{\max} = 300$ for CIFAR-100, and $B_{\min} = 50$ and $B_{\max} = 150$ for GTSRB and PatternNet. All training samples are allocated without replacement.

After the training partitions are determined, personalized test sets are constructed from the original test split. For each class, its test samples are distributed among clients in proportion to the corresponding class counts in their local training sets. Consequently, each client's test distribution follows its realized training distribution, while each test sample is assigned to only one client.

B.2 Training Configuration

All experiments are conducted using NVIDIA L40 GPUs. All experiments use 100 clients, with 10% of the clients randomly selected to participate in each communication round. Each selected client uses a batch size of 10, and Adam is used for local optimization. CIFAR-100 is trained for

200 communication rounds and PatternNet for 50 rounds. For GTSRB, we use 100 communication rounds with MobileNetV2 and 30 rounds with ResNet-8. All client models are initialized from scratch. For methods that decompose the client model into shared and personalized components, the final linear classification layer is treated as the personalized prediction head, while all preceding layers constitute the feature extractor. For DistillPFL, the transformation module ψ_i is implemented as a lightweight two-layer MLP, using a ReLU activation between the two linear layers and a Tanh activation at the output. For all baselines, we follow the recommended hyperparameter settings from the original papers or official implementations, with method-specific parameters adjusted when necessary.

C HYPERPARAMETER CONFIGURATIONS OF DistillPFL

The main DistillPFL hyperparameters are selected through grid search for each dataset, heterogeneity setting, and client architecture. Specifically, we tune the knowledge distillation weight λ_{KD} and temperature τ_{KD}, the contrastive loss weight λ_{con} and temperature τ_{con}, and the number of predictive anchors per class M. The selected configurations used in the main experiments are reported in Table 4.

Table 4: Hyperparameter configurations of DistillPFL used in the main experiments.

Model	Dataset	Setting	M	λ_{KD}	τ_{KD}	λ_{con}	τ_{con}
MobileNetV2	CIFAR-100	LPC-5	5	0.1	2.0	0.1	0.1
		Dir(0.05)	3	0.3	4.0	0.5	0.1
	GTSRB	LPC-5	5	0.5	1.0	0.1	0.05
		Dir(0.05)	5	0.7	2.0	0.3	1.0
	PatternNet	LPC-5	5	1.0	4.0	0.5	0.5
		Dir(0.05)	3	0.3	2.0	1.0	1.0
ResNet-8	CIFAR-100	LPC-5	5	0.5	4.0	0.5	0.1
		Dir(0.05)	3	0.5	4.0	1.0	0.1
	GTSRB	LPC-5	7	0.3	4.0	0.5	0.1
		Dir(0.05)	5	0.1	2.0	1.0	0.05
	PatternNet	LPC-5	3	0.5	1.0	0.1	0.1
		Dir(0.05)	5	0.3	2.0	0.5	0.05

The remaining configurations are fixed across experiments. Semantic proxies are constructed from disjoint groups of $g = 3$ samples within each class. Using ℓ_2-normalized features extracted by the local model, we greedily group semantically similar samples. We use a class-wise proxy collection budget of $B = 20$. Each client pretrains its local model for 10 epochs before optimizing the transformation module, which is subsequently trained for 3 epochs with the pretrained local model fixed. The Gaussian perturbation variance is fixed to $\sigma^2 = 10^{-2}$. The server classifier is a linear classifier trained with AdamW using a learning rate of 10^{-3}, a weight decay of 10^{-4}, and a batch size of 32. Its training duration is selected from a maximum of 50 epochs using three-fold cross-validation. Predictive anchors are constructed using class-wise k-means clustering on the server logits. The clustering procedure uses M clusters per class and runs for 20 iterations.