FedPAGR: Federated Prototype Alignment via Geometric Refinement for Heterogeneous Architectures

Kris Prasad
Kennesaw State University
kprasad@students.kennesaw.edu

Md Abdullah Al Hafiz Khan
Kennesaw State University
mkhan74@kennesaw.edu

Abstract

Federated learning with heterogeneous client architectures cannot rely on parameter aggregation. Prototype-based methods address architectural heterogeneity by exchanging class-level representations, but naively averaging prototypes across non-IID clients leads to semantic drift and poor inter-class separation. We propose FedPAGR, a framework where heterogeneous clients project their features into a shared consensus space and exchange class prototypes with a central server. The server refines aggregated prototypes through a geometric regularization objective that enforces agreement with client submissions and inter-class angular separation. Clients anchor their classifiers to the refined prototypes and train with a composite objective combining classification, prototype alignment, and entropy regularization. We evaluate FedPAGR across multiple domains, including four image benchmarks and a clinical NLP task using heterogeneous ClinicalBERT variants, with five architectures per federation under severe label heterogeneity (α=0.1). FedPAGR achieves the highest ensemble accuracy across all four image datasets and the highest local test accuracy on low-class and clinical tasks, including a 4.99-point improvement over the strongest baseline on MIMIC-IV, while remaining competitive on high-class benchmarks.

1 Introduction

Federated learning (FL) enables distributed model training without sharing raw data (McMahan et al., 2017). Most FL methods, including FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), and SCAFFOLD (Karimireddy et al., 2020), aggregate model parameters across clients. This requires all clients to share an identical model architecture. In practice, clients often differ in architecture due to device constraints, legacy systems, or task-specific requirements (Kairouz and McMahan, 2021; Ye et al., 2023).

This heterogeneity is particularly pronounced in NLP, where institutions may use different language models (e.g., BERT variants with different fine-tuning depths) and handle sensitive text data such as clinical records or legal documents that cannot leave institutional boundaries. Federated learning is a natural fit for these settings, but standard parameter aggregation is inapplicable when clients run structurally different models.

Prototype-based methods draw on the idea of representing classes by their mean embeddings (Snell et al., 2017) and apply it to federated settings. FedProto (Tan et al., 2022a) enables heterogeneous FL by exchanging class prototypes instead of parameters. Each client computes prototypes by averaging features per class and sends them to a server, which aggregates them into global prototypes. This allows heterogeneous architectures to collaborate without sharing parameters. However, simple prototype averaging is sensitive to non-IID data distributions (Zhao et al., 2018): when clients observe different subsets of classes with skewed frequencies, the averaged prototypes can drift semantically and lose inter-class discriminability.

FedTGP (Zhang et al., 2024) addresses this by processing prototypes through a trainable server-side network. Other approaches use contrastive learning over prototypes (Tan et al., 2022b; Mu et al., 2023), but were primarily designed for homogeneous architectures where parameter averaging provides the primary alignment signal, or require frozen pre-trained encoders.

We propose FedPAGR (Federated Prototype Alignment via Geometric Refinement), which refines aggregated prototypes through direct geometric optimization rather than a learned network. The server minimizes a regularization objective with two components: (1) an agreement loss that keeps refined prototypes close to client submissions, and (2) a separation loss that enforces an angular margin between different class prototypes. Clients an-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 420–427

chor their classifier weights to the global prototypes and train with cross-entropy, prototype-based alignment, and entropy regularization.

FedPAGR requires no public data, no shared model, and no frozen encoders. Each client trains its full model end-to-end while communicating only d-dimensional prototype vectors per class.

Contributions

- A geometric prototype refinement procedure on the server that enforces inter-class separation through angular margin constraints.
- A client-side training objective combining classifier anchoring, temperature-scaled prototype alignment, and entropy regularization.
- Cross-domain evaluation across image benchmarks and a clinical NLP task with heterogeneous transformer-based models, showing effectiveness for heterogeneous transformer-based NLP models without shared architectures or auxiliary data.

2 Related Work

Prototype-based FL FedProto (Tan et al., 2022a) is the foundational method in this category. It uses an L2 alignment loss to encourage each client's local features to stay close to the global prototypes. The alignment is purely local: the server performs no optimization on the aggregated prototypes, so the quality of the global prototypes depends entirely on the client submissions. Under severe non-IID conditions, where each client observes only a few classes, the aggregated prototypes can become noisy or poorly separated.

FedNH (Dai et al., 2023) enforces uniform prototype distributions in the embedding space to mitigate class imbalance effects. FedHP (Fonio et al., 2024) applies hyperspherical prototypical regularization, constraining prototypes to lie on a hypersphere and aligning local prototypes with global ones through a regularization term.

FedTGP (Zhang et al., 2024) processes aggregated prototypes through a trainable network on the server, with adaptive-margin contrastive learning to enforce inter-class separation. Clients align their representations to the resulting prototypes via a supervised distance loss.

FedPAGR directly optimizes the aggregated prototypes through a geometric objective that balances fidelity to client submissions with inter-class angular separation, without requiring a server-side model.

Contrastive FL FedProc (Mu et al., 2023) uses supervised contrastive learning over global prototypes to reduce representation drift between rounds. Each client treats the global prototype of its class as a positive anchor and prototypes of other classes as negatives. However, FedProc was designed for homogeneous architectures where FedAvg provides the primary alignment mechanism. The contrastive loss acts as a regularizer on top of shared parameters, not as the sole alignment signal.

FedPCL (Tan et al., 2022b) uses contrastive learning with per-client prototypes from multiple pre-trained frozen encoders. This approach requires frozen pre-trained backbones and does not support end-to-end training of diverse architectures.

Distillation and auxiliary-data FL FedMD (Li and Wang, 2019) aligns heterogeneous models via logit matching on shared public data. FedDF (Lin et al., 2020) extends this with ensemble distillation, where the server trains a central model by distilling knowledge from the ensemble of client models. FedDistill (Song et al., 2024) decomposes global models into feature extractors and classifiers, applying group distillation to de-bias local models under non-IID data. These approaches rely on access to external data or a shared global model for knowledge transfer (Li et al., 2024). FedAux (Sattler et al., 2021) reduces this dependency through self-supervised pre-training on unlabeled auxiliary data and certainty-weighted ensemble distillation, but still requires access to auxiliary data.

FedPAGR avoids any dependency on external data. Knowledge transfer occurs exclusively through prototype vectors.

Model-heterogeneous FL Recent surveys (Fan et al., 2024; Ye et al., 2023) identify prototype exchange, knowledge distillation, and mutual learning as the three main strategies for model-heterogeneous FL. A separate line of work addresses heterogeneity through sub-model extraction. HeteroFL (Diao et al., 2021) allows clients with different computational budgets to train subnetworks of varying widths extracted from a single global model. FedRolex (Alam et al., 2022) extends this with a rolling extraction scheme that ensures all parts of the global model receive train-

ing across rounds. These methods accommodate computational heterogeneity but require all clients to share a common base architecture.

FedPAGR supports fully distinct architectures, communicating only through prototype vectors in a shared consensus space.

3 Method

3.1 Problem Setting

We consider M clients, each with a private dataset $D_k = \{(x_i, y_i)\}$ and a model with architecture-specific feature extractor f_k. Architectures differ across clients. The server holds no data and no model. Communication is restricted to d-dimensional prototype vectors.

3.2 Consensus Projection

Different architectures produce features in different spaces with different dimensions. We write $\text{Normalize}(v) = v/\|v\|_2$ throughout. To enable meaningful prototype exchange, each client augments its model with a projection head g_k that maps architecture-specific features into a shared d-dimensional consensus space:

$$z = g_k(f_k(x)), \quad \hat{z} = \text{Normalize}(z) \tag{1}$$

The L_2 normalization ensures that all features lie on the unit hypersphere, so that dot products correspond to cosine similarities. This prevents any single architecture from dominating the prototype space through larger feature magnitudes.

The projection head consists of Linear($d \to 2d$), LayerNorm, ReLU, Dropout, Linear($2d \to d$), and LayerNorm. Each architecture may include an additional linear layer before g_k to map its backbone output dimension to d. The projection head is trained locally and remains private.

3.3 Local Prototype Computation

For each class c in its local data, client k computes a prototype by averaging normalized consensus features and re-normalizing to the unit sphere:

$$p_{k,c} = \text{Normalize}\left(\frac{1}{|D_{k,c}|}\sum_{i:y_i=c}\hat{z}_i\right) \tag{2}$$

These prototypes are sent to the server.

3.4 Server Aggregation

The server aggregates client prototypes via uniform averaging:

$$\bar{p}_c = \text{Normalize}\left(\frac{1}{|\mathcal{K}_c|}\sum_{k\in\mathcal{K}_c} p_{k,c}\right) \tag{3}$$

where $\mathcal{K}_c$ is the set of clients that reported class c. Under Dirichlet-based label heterogeneity, not all clients observe all classes, so $\mathcal{K}_c$ may be a strict subset of the participating clients. If no client reported a particular class in a given round, the server retains the prototype from the previous round.

3.5 Geometric Prototype Refinement

Simple averaging can produce poor global prototypes under severe non-IID conditions: when only a few clients contribute prototypes for a given class, the average may be noisy, and there is no guarantee that prototypes of different classes are well-separated. The server initializes a refined prototype $\tilde{p}_c$ from each aggregate $\bar{p}_c$ and optimizes:

$$\mathcal{L}_{\text{server}} = \mathcal{L}_{\text{agree}} + \lambda_S \mathcal{L}_{\text{sep}} \tag{4}$$

Agreement loss. Keeps refined prototypes close to client submissions:

$$\mathcal{L}_{\text{agree}} = \sum_c \sum_{k\in\mathcal{K}_c} (1 - p_{k,c} \cdot \tilde{p}_c) \tag{5}$$

Each term $(1 - p_{k,c} \cdot \tilde{p}_c)$ measures the angular distance between a client's prototype and the current refined prototype. When the two are perfectly aligned, the loss is zero.

Separation loss. Enforces a minimum angular margin m between prototypes of different classes:

$$\mathcal{L}_{\text{sep}} = \sum_{c\neq c'} \max(0,\ \tilde{p}_c \cdot \tilde{p}_{c'} - m) \tag{6}$$

This penalizes any pair of class prototypes whose cosine similarity exceeds m, preventing semantically distinct classes from collapsing to nearby regions of the shared embedding space. Pairs below the margin incur no penalty. This is to enforce discriminability in the prototype space.

Optimization. The server initializes a $C \times d$ prototype matrix $\tilde{P} = [\tilde{p}_1; \ldots; \tilde{p}_C]$ from the aggregated prototypes and runs T steps of SGD (with momentum 0.9) on the unconstrained parameters,

normalizing in the forward pass for loss computation. After optimization, the prototypes are projected back to the unit sphere. The optimization involves only a $C \times d$ matrix, with no data and no backpropagation through client models. The refined prototypes $\tilde{p}_c$ are then broadcast to clients.

3.6 Client-Side Training

At the start of each round, clients receive the refined prototypes and perform three steps:

1. Classifier anchoring The classifier weight matrix is overwritten with the refined prototypes: $W_c \leftarrow \tilde{p}_c / \|\tilde{p}_c\|_2$ for all c. Biases are set to zero. This synchronizes all clients' decision boundaries at the start of each round. Without anchoring, each client's classifier evolves independently and may drift away from the prototype space. Since the classifier operates on L_2-normalized consensus features, the anchored classifier computes cosine similarity between the input feature and each class prototype.

2. Composite loss The client optimizes:

$$\mathcal{L}_{\text{client}} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{global}} + \lambda_E \mathcal{L}_{\text{ent}} \tag{7}$$

- $\mathcal{L}_{\text{CE}}$: Standard cross-entropy on the local classifier logits.
- $\mathcal{L}_{\text{global}}$: Cross-entropy on prototype-based logits $\ell_c = (\hat{z} \cdot \tilde{p}_c)/\beta$, where β is a temperature parameter. This aligns the client's features with the refined prototypes. Smaller β produces sharper distributions that encourage features to cluster tightly around their class prototype.
- $\mathcal{L}_{\text{ent}} = -\frac{1}{|B|}\sum_{x \in B}\frac{1}{C}\sum_c \log \text{softmax}(\ell_c)$: Minimizes $\text{KL}(U\|p)$ where U is the uniform distribution and p is the softmax over prototype logits, encouraging high-entropy predictions. This prevents clients with skewed class distributions from collapsing their features onto a small number of prototypes.

3. Prototype upload After training, the client computes updated prototypes and sends them to the server.

3.7 Inference

At test time, classification uses cosine similarity to prototypes:

$$\hat{y} = \arg\max_c \ \hat{z} \cdot \tilde{p}_c \tag{8}$$

Algorithm 1 FedPAGR Training

Require: M clients with models $\{f_k, g_k\}$, rounds R, participation ratio ρ
Initialize $\{\tilde{p}_c\}_{c=1}^{C}$ randomly on unit sphere
for round $r = 1$ to R **do**
  Sample $\mathcal{S}_r \subset \{1, \ldots, M\}$ with $|\mathcal{S}_r| = \lfloor \rho M \rfloor$
  Broadcast $\{\tilde{p}_c\}$ to clients in $\mathcal{S}_r$
  for each client $k \in \mathcal{S}_r$ **in parallel do**
    $W_c \leftarrow \tilde{p}_c / \|\tilde{p}_c\|_2, \quad b_c \leftarrow 0 \quad \forall c$ {anchor classifier}
    for each local epoch **do**
      for each batch (x, y) **do**
        $\hat{z} \leftarrow \text{Normalize}(g_k(f_k(x)))$
        $\mathcal{L}_{\text{CE}} \leftarrow \text{CrossEntropy}(W\hat{z},\ y)$
        $\ell_c \leftarrow (\hat{z} \cdot \tilde{p}_c)/\beta \quad \forall c$
        $\mathcal{L}_{\text{global}} \leftarrow \text{CrossEntropy}(\boldsymbol{\ell},\ y)$
        $\mathcal{L}_{\text{ent}} \leftarrow -\frac{1}{|B|}\sum_{x \in B}\frac{1}{C}\sum_c \log \text{softmax}(\ell_c)$
        Update f_k, g_k via SGD on $\mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{global}} + \lambda_E \mathcal{L}_{\text{ent}}$
      end for
    end for
    $p_{k,c} \leftarrow \text{Normalize}\Big(\frac{1}{|D_{k,c}|}\sum_{i:y_i=c} \hat{z}_i\Big) \quad \forall c \in \mathcal{Y}_k$
    Send $\{p_{k,c}\}$ to server
  end for
  Server aggregation:
  $\bar{p}_c \leftarrow \text{Normalize}\Big(\frac{1}{|\mathcal{K}_c|}\sum_{k \in \mathcal{K}_c} p_{k,c}\Big) \quad \forall c$
  Server refinement: initialize $\tilde{P} \leftarrow [\bar{p}_1; \ldots; \bar{p}_C]$
  for $t = 1$ to T **do**
    $\hat{P} \leftarrow \text{row-Normalize}(\tilde{P})$
    Compute $\mathcal{L}_{\text{agree}}(\hat{P}) + \lambda_S \mathcal{L}_{\text{sep}}(\hat{P})$
    Update $\tilde{P}$ via SGD (momentum 0.9)
  end for
  $\tilde{p}_c \leftarrow \tilde{P}_c / \|\tilde{P}_c\|_2 \quad \forall c$ {project to unit sphere}
end for

4 Experiments

4.1 Setup

Datasets. We evaluate on four standard image classification benchmarks and one clinical NLP task:

- CIFAR-10 / CIFAR-100 (Krizhevsky, 2009): 32×32 color images with 10 and 100 classes, respectively. 50K training / 10K evaluation images each.
- Tiny ImageNet (Le and Yang, 2015): a 200-class subset of ImageNet with 64×64 images. 100K training / 10K evaluation images.
- Fashion-MNIST (Xiao et al., 2017): 28×28 grayscale images of clothing items across 10 classes. 60K training / 10K evaluation images, resized to 32×32 and expanded to 3 channels.
- MIMIC-IV (Johnson et al., 2023, 2024): we use the MIMIC-IV-Ext-BHC labeled clinical

notes dataset (Aali et al., 2025, 2024) for hospital discharge disposition prediction. 270K clinical notes from 128K patients, classified into 6 categories: Home, Home Health, SNF, Rehab, Expired, and Other. Data is split by patient: 70% for federated training (Dirichlet-partitioned across clients, each further split 80/20 into client train/test), and 30% reserved for validation and testing.

Label heterogeneity across clients is simulated via Dirichlet allocation with $\alpha = 0.1$, which produces severe class imbalance where most clients observe only a small subset of the total classes.

Architectures. For image tasks, each federation of M=20 clients uses five architectures assigned round-robin: CNN, MLP, ResNet-18, GoogLeNet, and MobileNetV2. All share a 512-dimensional consensus space. For MIMIC-IV, we use ClinicalBERT variants with different numbers of trainable layers (frozen, 2-layer, 4-layer fine-tuning) and a 128-dimensional consensus space.

Baselines. We compare against FedProto (Tan et al., 2022a), FedTGP (Zhang et al., 2024), and FedProc (Mu et al., 2023). All baselines use prototype exchange for communication in our heterogeneous setting.

Hyperparameters. Image tasks: 400 rounds, 1 local epoch, SGD with lr=0.01, momentum=0.9, batch size 32, 50% client participation. FedPAGR: β=0.1, λ_E=0.1, λ_S=0.5, m=0.3, 5 refinement steps at lr=0.01. For CIFAR-100 and Tiny ImageNet: m=0.1, 3 refinement steps. MIMIC-IV: 100 rounds, lr=2×10^{-5}, batch size 8, 20% participation, β=0.5.

Metrics. We report *local test accuracy* (weighted average of each client's accuracy on its held-out test set) and *ensemble accuracy* (argmax of the mean predicted class probabilities across all client models on a shared global test set).

4.2 Results Under Label Heterogeneity

All experiments use Dirichlet allocation with α=0.1 to simulate severe label heterogeneity, where each client observes only a partial, overlapping subset of the total classes. Each federation consists of 20 clients with 5 distinct architectures. Table 1 reports steady-state accuracy averaged over the last 5 evaluation points (rounds 320 to 400 for image tasks, rounds 80 to 100 for MIMIC-IV).

4.2.1 Ensemble (Server-Side) Performance

Ensemble accuracy measures cross-architecture generalization by averaging softmax outputs across all client models on a shared global test set. This metric reflects how well the server-refined prototypes align heterogeneous representations into a coherent global predictor.

FedPAGR achieves the highest ensemble accuracy on all four image datasets (Table 1): 40.76% on CIFAR-10, 23.10% on CIFAR-100, 10.48% on Tiny ImageNet, and 80.20% on Fashion-MNIST. Ensemble evaluation was not performed on MIMIC-IV due to computational constraints. FedTGP is the strongest baseline, reaching 38.40%, 22.98%, 9.62%, and 79.30% on the same datasets. FedProto's ensemble accuracy is substantially lower (25.62%, 4.04%, 1.68%, 62.10%), consistent with its lack of server-side prototype optimization. FedProc achieves 24.08%, 5.45%, 2.30%, and 59.57%.

The gap between FedPAGR and FedTGP is largest on CIFAR-10 (2.36 points) and smallest on CIFAR-100 (0.12 points). On high-class datasets, the separation loss sums over all $C(C-1)$ ordered class pairs, producing nearly 40,000 terms for Tiny ImageNet's 200 classes. We reduce the margin from m=0.3 to m=0.1 and the refinement steps from 5 to 3 for CIFAR-100 and Tiny ImageNet to maintain stability, which limits the refinement's expressiveness on these datasets.

4.2.2 Client-Side (Local) Performance

Local test accuracy is the weighted average of each client's accuracy on its own held-out test set, measuring personalized performance under the client's specific label distribution. Each client trains a different architecture (CNN, MLP, ResNet-18, GoogLeNet, or MobileNetV2 for image tasks; ClinicalBERT variants for MIMIC-IV) and observes a different subset of classes due to the Dirichlet partition.

FedPAGR achieves the highest local accuracy on CIFAR-10 (83.05%), Fashion-MNIST (95.96%), and MIMIC-IV (76.38%). On CIFAR-10, FedPAGR outperforms FedProto by 5.94 points and FedTGP by 8.07 points (Table 1). On MIMIC-IV, FedPAGR outperforms FedTGP by 4.99 points (76.38% vs. 71.39%).

On CIFAR-100 and Tiny ImageNet, FedTGP achieves higher local accuracy than FedPAGR (44.15% vs. 41.86% and 24.18% vs. 23.07%, respectively). FedProto and FedProc perform poorly

	CIFAR-10		CIFAR-100		Tiny ImageNet		Fashion-MNIST		MIMIC-IV
	Local	Ens.	Local	Ens.	Local	Ens.	Local	Ens.	Local
FedProto	77.11	25.62	30.50	4.04	11.93	1.68	95.53	62.10	11.98
FedTGP	74.98	38.40	**44.15**	22.98	**24.18**	9.62	94.73	79.30	71.39
FedProc	64.69	24.08	15.14	5.45	6.78	2.30	86.66	59.57	7.89
FedPAGR	**83.05**	**40.76**	41.86	**23.10**	23.07	**10.48**	**95.96**	**80.20**	**76.38**

Table 1: Benchmark results under label heterogeneity (α=0.1, 20 clients, 5 architectures). Local: weighted average client test accuracy (%). Ens.: ensemble accuracy (%). Best in bold.

on these high-class datasets, with FedProc scoring below 15% local accuracy on both.

4.2.3 Cross-Domain Generalization to Clinical NLP

The MIMIC-IV experiment demonstrates that prototype alignment is effective for heterogeneous transformer-based language models in realistic federated settings. Clients run ClinicalBERT variants with different numbers of trainable transformer layers (frozen, 2-layer, and 4-layer fine-tuning), producing representations that vary substantially in quality and structure. Despite these differences, FedPAGR achieves 76.38% local accuracy, outperforming FedTGP by 4.99 points (Table 1). FedProto (11.98%) and FedProc (7.89%) fail on this task. Prototype alignment is effective for text representations because class prototypes in the shared embedding space capture semantic centroids that are meaningful regardless of the underlying encoder's fine-tuning depth. The angular separation enforced by the refinement objective prevents class prototypes from collapsing in regions of the embedding space where frozen and fine-tuned encoders produce similar representations. This setting reflects real-world clinical NLP deployments, where different pretrained models and fine-tuning strategies are used while operating under strict data-sharing constraints.

4.3 Ablation Studies

We ablate FedPAGR on CIFAR-10 (α=0.1) by varying one hyperparameter at a time from the default configuration (Table 2).

The alignment temperature (β) is the most critical component. Removing the prototype alignment terms (β=0.0) causes accuracy to collapse to random chance (9.97%), confirming that prototype-based alignment is essential (Table 2). A larger temperature (β=0.5) reduces ensemble accuracy from 40.76% to 34.40%, showing that sharper alignment is beneficial.

Setting	Local	Ens.
Default	**83.05**	**40.76**
Separation weight (λ_S)		
$\lambda_S = 0.0$ (no separation)	82.74	38.03
$\lambda_S = 1.0$	82.63	39.27
Separation margin (m)		
$m = 0.1$	82.51	38.55
$m = 0.5$	83.00	37.30
Alignment temperature (β)		
$\beta = 0.0$ (remove alignment terms)	9.97	10.00
$\beta = 0.5$	82.59	34.40
Entropy weight (λ_E)		
$\lambda_E = 0.0$ (no entropy)	82.66	37.44
$\lambda_E = 0.05$	82.96	39.30
Refinement steps (T)		
$T = 1$	82.87	38.44
$T = 3$	82.94	39.63
$T = 10$	82.82	40.19

Table 2: Ablation of FedPAGR components on CIFAR-10 (α=0.1). Default: λ_S=0.5, m=0.3, β=0.1, λ_E=0.1, T=5.

Separation (λ_S) and entropy (λ_E) primarily affect ensemble accuracy. Removing separation (λ_S=0.0) drops ensemble accuracy by 2.73 points, while removing entropy (λ_E=0.0) drops it by 3.32 points. Local accuracy is less sensitive to these components.

Refinement steps (T) show a clear trend in ensemble accuracy: 38.44% at T=1, 39.63% at T=3, 40.19% at T=10, and 40.76% at the default T=5. Local accuracy remains stable across all values (82.8% to 83.1%). This confirms that refinement primarily improves cross-architecture alignment rather than individual client performance.

4.4 Communication Cost

FedPAGR communicates only class prototypes: $C \times d$ floats per client per round in each direction. For CIFAR-10 with d=512, this is $10 \times 512 \times 4 =$ 20KB uploaded and 20KB downloaded per client per round, compared to full model transmission which ranges from hundreds of KB to tens of MB.

All prototype-based methods (FedProto, FedTGP, FedProc, FedPAGR) have identical communication cost since they exchange the same prototype vectors.

5 Conclusion

We presented FedPAGR, a federated learning framework that enables collaboration across heterogeneous model architectures through prototype exchange in a shared consensus space. The server refines aggregated prototypes via a geometric objective that enforces agreement with client submissions and angular separation between classes. Clients anchor their classifiers to the refined prototypes and train with a composite loss combining classification, prototype alignment, and entropy regularization.

Experiments across four image benchmarks and a clinical NLP task show that FedPAGR achieves the highest ensemble accuracy on all four image datasets and the highest local accuracy on CIFAR-10, Fashion-MNIST, and MIMIC-IV. On MIMIC-IV, where clients run ClinicalBERT variants with different fine-tuning depths, FedPAGR outperforms the strongest baseline by 4.99 points, demonstrating that prototype alignment generalizes from vision to language understanding. Ablation studies confirm that prototype-based alignment is essential, with the anchoring temperature β being the most critical component. Future work includes scaling the separation objective to larger class spaces, evaluating with heterogeneous large language models, and testing under additional forms of distribution shift such as non-overlapping class partitions and data-type heterogeneity.

Limitations

FedPAGR's separation loss sums over $C(C-1)$ ordered class pairs, scaling quadratically with the number of classes. For CIFAR-100 and Tiny ImageNet, we reduce the margin and refinement steps to maintain stability, which limits the refinement's expressiveness on high-class datasets.

The current evaluation uses Dirichlet-based synthetic heterogeneity. Real-world systems may exhibit additional forms of distribution shift not captured by label skew alone.

References

Asad Aali, Dave Van Veen, Yamin Arefeen, Jason Hom, Christian Bluethgen, Eduardo Pontes Reis, Sergios Gatidis, Namuun Clifford, Joseph Daws, Arash Tehrani, Jangwon Kim, and Akshay Chaudhari. 2025. MIMIC-IV-Ext-BHC: Labeled Clinical Notes Dataset for Hospital Course Summarization. *PhysioNet*. Version 1.2.0.

Asad Aali, Dave Van Veen, Yamin Ishraq Arefeen, Jason Hom, Christian Bluethgen, Eduardo Pontes Reis, Sergios Gatidis, Namuun Clifford, Joseph Daws, Arash S Tehrani, Jangwon Kim, and Akshay S Chaudhari. 2024. A dataset and benchmark for hospital course summarization with adapted large language models. *Journal of the American Medical Informatics Association*, 32(3):470–479.

Samiul Alam, Luyang Liu, Ming Yan, and Mi Zhang. 2022. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. In *Advances in Neural Information Processing Systems*.

Yutong Dai, Zeyuan Chen, Junnan Li, Shelby Heinecke, Lichao Sun, and Ran Xu. 2023. Tackling data heterogeneity in federated learning with class prototypes. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Enmao Diao, Jie Ding, and Vahid Tarokh. 2021. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. In *International Conference on Learning Representations*.

Boyu Fan, Siyang Jiang, Xiang Su, Sasu Tarkoma, and Pan Hui. 2024. A survey on model-heterogeneous federated learning: Problems, methods, and prospects. *IEEE International Conference on Big Data*.

Samuele Fonio, Mirko Polato, and Roberto Esposito. 2024. Fedhp: Federated learning with hyperspherical prototypical regularization. In *European Symposium on Artificial Neural Networks*.

A. E. W. Johnson, L. Bulgarelli, L. Shen, and et al. 2023. Mimic-iv, a freely accessible electronic health record dataset. *Scientific Data*, 10.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Brian Gow, Benjamin Moody, Steven Horng, Leo Anthony Celi, and Roger Mark. 2024. MIMIC-IV. *PhysioNet*. Version 3.1.

Peter Kairouz and H. Brendan McMahan. 2021. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1-2):1–210.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR.

Alex Krizhevsky. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Ya Le and Xuan S. Yang. 2015. Tiny imagenet visual recognition challenge.

Daliang Li and Junpu Wang. 2019. Fedmd: Heterogenous federated learning via model distillation. In *NeurIPS Workshop on Federated Learning for Data Privacy and Confidentiality*.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450.

Xiang Li, Yue Zhang, Junpu Wang, Jian Chen, and 1 others. 2024. Federated learning via knowledge distillation: A survey. *arXiv preprint*, arXiv:2404.08564.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. 2020. Ensemble distillation for robust model fusion in federated learning. In *Advances in Neural Information Processing Systems*.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR.

Xutong Mu, Yulong Shen, Ke Cheng, Xueli Geng, Jiaxuan Fu, Tao Zhang, and Zhiwei Zhang. 2023. Fedproc: Prototypical contrastive federated learning on non-iid data. *Future Generation Computer Systems*, 143:93–104.

Felix Sattler, Tim Korjakow, Roman Rischke, and Wojciech Samek. 2021. Fedaux: Leveraging unlabeled auxiliary data in federated learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*.

Changlin Song, Divya Saxena, Jiannong Cao, and Yuqing Zhao. 2024. Feddistill: Global model distillation for local model de-biasing in non-iid federated learning. *arXiv preprint arXiv:2404.09210*.

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

Yue Tan, Guodong Long, Jie Ma, Lu Liu, Tianyi Zhou, and Jing Jiang. 2022b. Federated learning from pre-trained models: A contrastive learning approach. *arXiv preprint arXiv:2209.10083*.

Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *Preprint*, arXiv:1708.07747.

Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. 2023. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44.

Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. 2024. Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. *Preprint*, arXiv:2401.03230.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data.