FedPAF: Federated Personalized Alignment Framework

Rongbo Zhou
School of Computer Science and Technology
Xinjiang University
Urumqi, China
107552503757@stu.xju.edu.cn

Jiao Tian*
School of Computer Science and Technology
Xinjiang University
Urumqi, China
jiaotian@xju.edu.cn

Liejun Wang*
School of Computer Science and Technology
Xinjiang University
Urumqi, China
wljxju@xju.edu.cn

Yihao Yin
School of Computer Science and Technology
Xinjiang University
Urumqi, China
107552503684@stu.xju.edu.cn

***Abstract*—Federated learning under client heterogeneity faces non-IID data distributions and client-specific feature statistics that degrade performance. While prior personalized federated learning methods partially mitigate statistical mismatch through local adaptation or normalization decoupling, they struggle to jointly model client-dependent discriminative cues and maintain cross-client semantic consistency. We propose FedPAF, a personalized federated framework that improves performance by jointly addressing normalization heterogeneity, feature heterogeneity, and semantic drift. FedPAF adopts a personalization strategy with normalization decoupling, leveraging Per-FedAvg-style meta-learning for adaptation while maintaining FedBN-style local batch normalization, enabling transferable representation learning while preserving client-specific statistics. It further highlights client-dependent discriminative cues through a lightweight multi-spectral channel reweighting module with negligible overhead, and aligns class semantics via global prototype aggregation to reduce semantic drift and enhance class-wise separability. Extensive experiments on Digits-Five, PACS, and DomainNet demonstrate consistent improvements over strong baselines in both accuracy and Macro-F1 under heterogeneous client settings.**

***Index Terms*—federated learning, client heterogeneity, personalized federated learning, frequency-aware feature modeling, prototype alignment**

I. Introduction

Federated learning (FL) enables multiple clients to collaboratively train models without sharing raw data, making it appealing for privacy-sensitive and decentralized applications. In practice, however, client data are rarely independent and identically distributed (IID). *Client heterogeneity* is particularly common: clients may differ in devices, environments, or local data distributions, resulting in client-specific feature statistics and heterogeneous data characteristics. Under such heterogeneity, standard FL often exhibits unstable optimization and degraded performance for individual clients.

A central challenge lies in jointly achieving *normalization decoupling*, *client-adaptive personalization*, and *cross-client semantic consistency* under heterogeneous data distributions. FedAvg [1] aggregates model parameters assuming compatible feature distributions, which is frequently violated under heterogeneous clients. FedProx [2] stabilizes local optimization with a proximal term, but does not explicitly address client-specific feature statistics that affect normalization. FedBN [3] alleviates statistical mismatch by keeping batch normalization (BN) local, yet it can still struggle when clients exhibit different discriminative patterns and when semantic inconsistency leads to class-wise misalignment. From complementary perspectives, Per-FedAvg [4] improves local adaptation via meta-learning personalization, FedProto [5] promotes semantic consistency through prototype-based information exchange, and FedUV [6] regularizes representation uniformity and variance for more stable training. Despite their effectiveness, these methods typically emphasize one aspect at a time, making it difficult to jointly preserve client-specific statistics, capture client-dependent cues, and maintain consistent class semantics under heterogeneous client settings.

To bridge these gaps, we target three coupled challenges in personalized federated learning under client heterogeneity: (i) incompatible normalization statistics across clients that hinder effective aggregation, (ii) client-dependent discriminative patterns that require adaptive feature modeling, and (iii) cross-client semantic inconsistency that causes class-wise misalignment. We propose FedPAF (*Federated Personalized Alignment Framework*), which addresses these challenges through three complementary mechanisms: personalization with normalization decoupling to preserve client-specific statistics while enabling shared learning, lightweight frequency-aware channel reweighting to capture client-dependent patterns, and global prototype alignment to mitigate semantic drift and enhance class-wise separability.

Contributions. The main contributions are:

*Corresponding authors.

- We propose a personalization strategy with normalization decoupling, where meta-learning adaptation provides personalized updates and batch normalization is kept local, enabling shared representation learning while preserving client-specific statistics under client heterogeneity.
- We develop a lightweight frequency-aware channel reweighting mechanism to capture client-dependent discriminative patterns with negligible computational overhead.
- We devise a global prototype alignment regularization that aggregates class prototypes across clients to mitigate semantic drift and improve class-wise separability in heterogeneous client settings.

As this research is ongoing, the source code will be released in future publications.

II. Related Work

A. Federated Learning under Statistical Heterogeneity

Statistical heterogeneity (non-IID data) is a fundamental challenge in federated learning, where clients exhibit diverse data distributions. Existing approaches can be categorized into optimization-based, normalization-based, and regularization-based strategies.

Optimization-based strategies focus on stabilizing convergence under heterogeneous client updates. FedAvg [1] establishes the baseline by simple model averaging, but suffers from client drift under non-IID data. FedDyn [7] introduces dynamic regularization to balance local and global objectives, while MOON [8] leverages contrastive learning to align representations across rounds. These methods improve optimization stability but do not explicitly handle client-specific feature statistics or normalization behaviors.

Normalization-based strategies address the mismatch in feature statistics across clients due to variations in data collection protocols, device calibrations, or local preprocessing pipelines. FedBN [3] keeps batch normalization (BN) parameters local while aggregating other parameters, preventing the mixing of incompatible normalization statistics across heterogeneous clients. This approach proves effective when clients exhibit statistical shifts in feature distributions, but does not jointly capture client-dependent discriminative patterns or maintain semantic consistency across clients.

B. Personalized and Prototype-based Federated Learning

Personalized federated learning (PFL) aims to produce client-adaptive models rather than a single global model. Per-FedAvg [4] formulates PFL from a meta-learning perspective, learning an initialization that can be quickly adapted to each client with a few local steps. Other representative approaches achieve personalization via regularized local objectives (e.g., pFedMe [9]), explicit global–local trade-offs (e.g., Ditto [10]), or partial parameter sharing (e.g., FedPer [11]). Although personalization improves local adaptation, many PFL methods focus primarily on model-level customization and often do not jointly address statistical heterogeneity in feature distributions and semantic inconsistency across clients.

To reduce semantic mismatch across clients, prototype-based FL communicates class-level information. FedProto [5] shares class prototypes as compact semantic summaries for inter-client alignment, and FedNH [12] further addresses class imbalance by distributing prototypes uniformly in latent space while infusing class semantics. Most prototype-based approaches emphasize *prototype-based communication*; in contrast, prototypes can also serve as *auxiliary semantic anchors* integrated into parameter-aggregation FL through regularization, which is more flexible when combined with personalization and normalization.

C. Feature Enhancement and Representation Learning

FedUV [6] improves robustness by regularizing the *uniformity* and *variance* of representations under heterogeneity. Meanwhile, feature enhancement modules such as SE and CBAM [13], [14] reweight informative channels/spatial cues to strengthen discriminative representations. In addition, frequency-domain modeling has shown effectiveness in capturing discriminative patterns, and FcaNet [15] demonstrates that frequency channel attention provides useful interpretations (e.g., global average pooling relates to the DC/low-frequency component). However, existing frequency-aware designs often rely on explicit transforms (e.g., DCT-based multi-frequency extraction/selection), which introduce non-trivial computation and implementation overhead, making them less attractive for resource-constrained edge clients in FL. This motivates efficient frequency-aware feature modeling that preserves the benefits of spectral cues while remaining practical in federated deployments.

III. Methodology

This section presents FedPAF, a personalized federated learning framework tackling client heterogeneity. As illustrated in Fig. 1, the overall architecture is designed to address three coupled challenges: *normalization-level statistical heterogeneity*, *feature-level discriminative heterogeneity*, and *semantic-level inconsistency*.

To this end, FedPAF unifies three complementary components: (A) *normalization decoupling* (via local BN) to preserve client-specific statistics while enabling shared representation learning; (B) *lightweight frequency-aware feature enhancement* to efficiently capture client-dependent discriminative patterns; and (C) *global prototype alignment* to mitigate semantic drift using class-level anchors.

Together, these modules systematically address heterogeneity across the normalization, feature, and semantic levels, enabling stable shared learning and improved client-adaptive performance. The operating principles of each component are detailed below.

A. Personalization with Normalization Decoupling

FedPAF maintains a shared model at the server while allowing each client to preserve client-specific normalization behavior and retain a personalized component for fast local adaptation. At the beginning of communication round t, client

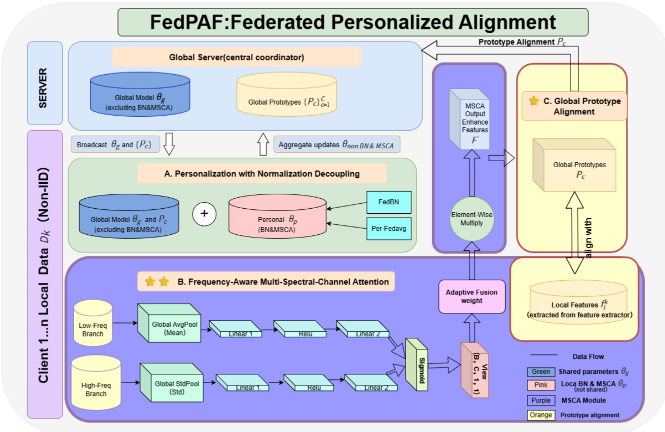


Fig. 1: Overall architecture of FedPAF. The server broadcasts global model θ_g (excluding BN/MSCA) and prototypes $\{P_c\}$. Each client trains through three components: (A) Personalization with normalization decoupling mixes θ_g with local θ_p while keeping BN local; (B) Frequency-aware attention enhances features using multi-spectral statistics; (C) Prototype alignment aligns features with global prototypes for semantic consistency. The server aggregates shared parameters only.

Algorithm 1 FedPAF Training

Input: K clients, rounds T, local steps τ, mixing α, prototype weight λ_{proto}
Initialize: $\theta_g^{(0)}, \{P_c^{(0)}\}_{c=1}^{C}, \{\theta_p^{(k)}\}_{k=1}^{K}$

```
for round t = 0, ..., T−1 do
    Server broadcasts θ_g^(t) and {P_c^(t)} to all clients
    for each client k in parallel do
        θ_k ← α · θ_g^(t) + (1−α) · θ_p^(k)        // personalized mixing
        for j = 1, ..., τ do
            Sample mini-batch B from D_k
            Extract features and apply MSCA: F' ← MSCA(g(B; θ_k))
            Compute L = L_CE + λ_proto L_proto and update θ_k
        end for
        Compute local prototypes P_c^(k) ← s_c^(k) / max(1, N_c^(k)) for all c
        Meta-update: θ_p^(k) ← θ_k − β∇_θ L(θ_k; B^val)
        Upload θ_{k,non-BN} and {P_c^(k), N_c^(k)} to server
    end for
    Server aggregation: θ_g^(t+1) ← (1/K) Σ_{k=1}^K θ_{k,non-BN}
    Prototype aggregation: P_c^(t+1) ← (Σ_k N_c^(k) P_c^(k)) / (Σ_k N_c^(k)) for all c
end for
```

k initializes its trainable parameters by mixing the current shared parameters and its personalized component (Section A in Fig. 1):

$$\theta_k \leftarrow \alpha \cdot \theta_g^{(t)} + (1-\alpha) \cdot \theta_p^{(k)}, \tag{1}$$

where $\alpha \in [0,1]$ controls the global–personalized trade-off. Batch normalization (BN) parameters and running statistics remain strictly local (pink blocks in Fig. 1) to avoid aggregating incompatible client-specific statistics.

Given local objective $\mathcal{L}_k(\cdot)$, client k performs τ inner updates with learning rate η:

$$\theta_k^{(j+1)} = \theta_k^{(j)} - \eta \nabla_\theta \mathcal{L}_k(\theta_k^{(j)}), \quad j = 0, \ldots, \tau-1. \tag{2}$$

A meta-style update is then applied to the personalized component using learning rate β:

$$\theta_p^{(k)} \leftarrow \theta_p^{(k)} - \beta \nabla_\theta \mathcal{L}_k(\theta_k^{(\tau)}; \mathcal{D}_k^{\text{val}}), \tag{3}$$

where $\mathcal{D}_k^{\text{val}}$ can be implemented by a held-out batch (or a support/query split). For efficiency, a first-order approximation is adopted. Importantly, BN-related parameters remain local and are updated only through standard local optimization, preserving client-specific normalization patterns.

B. Frequency-Aware Multi-Spectral Channel Attention

To strengthen client-dependent discriminative patterns under client heterogeneity, a lightweight frequency-aware channel reweighting module (MSCA) is inserted into the backbone to modulate feature representations (Section B in Fig. 1). Unless otherwise specified, MSCA is placed *after the last residual block of each stage* of the backbone.

Concretely, for a ResNet-style network with four stages, MSCA is attached to the stage outputs $\{\mathbf{F}^{(s)}\}_{s=1}^{4}$, i.e., after `layer1–layer4`:

$$\mathbf{F}_{\text{out}}^{(s)} = \text{MSCA}\left(\mathbf{F}_{\text{in}}^{(s)}\right), \quad s \in \{1,2,3,4\}, \tag{4}$$

and the subsequent blocks consume $\mathbf{F}_{\text{out}}^{(s)}$ as input. For lighter backbones used on Digits-Five, MSCA is inserted once at the last convolutional feature map before global pooling. For AlexNet-style networks, MSCA is placed after the last convolutional group (e.g., `conv5`) before the pooling/classifier.

Given a feature map $\mathbf{F} \in \mathbb{R}^{C\times H\times W}$, MSCA extracts low-frequency and high-frequency statistics using efficient spatial summaries:

$$\mathbf{F}_{\text{low}} = \text{GAP}(\mathbf{F}) = \frac{1}{HW}\sum_{h,w} \mathbf{F}_{:,h,w} \in \mathbb{R}^{C}, \tag{5}$$

$$\mathbf{F}_{\text{high}} = \text{StdDev}(\mathbf{F}) = \sqrt{\frac{1}{HW}\sum_{h,w} \left(\mathbf{F}_{:,h,w} - \mathbf{F}_{\text{low}}\right)^2} \in \mathbb{R}^{C}. \tag{6}$$

These statistics serve as efficient proxies for discriminative frequency cues without explicit frequency transforms (e.g., DCT/FFT), which keeps the module computation linear in the feature-map size.

The two statistics are passed through lightweight FC branches and fused to produce channel attention:

$$\mathbf{w}_{\text{attn}} = \lambda_{\text{low}} \cdot \text{FC}_{\text{low}}(\mathbf{F}_{\text{low}}) + \lambda_{\text{high}} \cdot \text{FC}_{\text{high}}(\mathbf{F}_{\text{high}}), \tag{7}$$

where $\lambda_{\text{low}}, \lambda_{\text{high}}$ are learnable fusion weights normalized by softmax. For an input feature map $\mathbf{F} \in \mathbb{R}^{C\times H\times W}$, MSCA incurs computational complexity $\mathcal{O}(CHW + 2C^2/r)$, where r is the reduction ratio in FC layers. Compared with a standard 3×3 convolutional layer with complexity $\mathcal{O}(9C^2HW)$, the overhead ratio is $\mathcal{O}(1/(9C) + 2/(9rHW))$, which is negligible for typical feature map sizes (e.g., $H, W \geq 14$). The enhanced feature map is computed by residual reweighting:

$$\mathbf{F}' = \mathbf{F} \odot \left(1 + \sigma(\mathbf{w}_{\text{attn}})\right), \tag{8}$$

where $\sigma(\cdot)$ denotes Sigmoid and $\odot$ is channel-wise multiplication. To preserve client-specific frequency preferences, MSCA parameters are maintained locally and are not aggregated across clients.

C. Global Prototype Alignment

To mitigate semantic inconsistency across clients, FedPAF introduces global class prototypes as semantic anchors (Section C in Fig. 1). During local training, each client accumulates class prototypes from pooled features $f_i^{(k)} \in \mathbb{R}^d$ (after MSCA and pooling):

$$P_c^{(k)} = \frac{1}{N_c^{(k)}} \sum_{(x,y)\in\mathcal{D}_k,\, y=c} f^{(k)}(x), \tag{9}$$

where $N_c^{(k)}$ is the number of observed samples of class c. After receiving $\{P_c^{(k)}\}$ from clients, the server aggregates global prototypes using class-frequency weighting:

$$P_c^{(t+1)} = \frac{\sum_{k=1}^{K} N_c^{(k)} \cdot P_c^{(k)}}{\sum_{k=1}^{K} N_c^{(k)}}. \tag{10}$$

Clients align their feature representations to the broadcasted global prototypes via an auxiliary objective:

$$\mathcal{L}_{\text{proto}} = \frac{1}{|\mathcal{B}|} \sum_{(x,y)\in\mathcal{B}} d\Big(f^{(k)}(x),\, P_y^{(t)}\Big), \tag{11}$$

where $d(\cdot,\cdot)$ is instantiated as MSE on Digits-Five and cosine distance on PACS/DomainNet. The total local loss is:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_{\text{proto}}(t)\cdot\mathcal{L}_{\text{proto}}, \qquad \lambda_{\text{proto}}(t) = \lambda_{\text{proto}}\cdot\mathbb{I}[t \geq t_{\text{warmup}}], \tag{12}$$

where warm-up prevents early-round instability before prototypes become informative.

Communication cost and privacy considerations. Beyond the standard parameter upload (model weights), each client additionally transmits C class prototypes $\{P_c^{(k)}\}_{c=1}^{C}$ with $\mathcal{O}(Cd)$ overhead, where d is the feature dimension. For typical settings (C=10, d=512), this adds ~5KB per round, negligible compared to model parameters (e.g., ~10MB for ResNet-18). However, prototypes are aggregated feature representations that may leak information about local data distributions. To mitigate privacy risks, differential privacy mechanisms [16] or secure aggregation [17] can be applied to prototype transmission, though we leave this integration to future work and focus on demonstrating the semantic alignment benefits in the current study.

D. Discussion and Analysis

The above components address client heterogeneity in a complementary manner. Personalization with normalization decoupling stabilizes training under client-specific feature statistics and supports client-adaptive updates without aggregating incompatible BN behaviors. Frequency-aware channel reweighting strengthens client-dependent discriminative patterns, improving robustness to feature-level heterogeneity across clients. Global prototype alignment further reduces cross-client semantic inconsistency by introducing class-level anchors, which improves class-wise separability and mitigates semantic drift. By operating jointly at the normalization-statistics, feature-representation, and semantic levels, FedPAF consistently improves both overall accuracy and Macro-F1 under heterogeneous client settings.

TABLE I: Ablation study on component effectiveness. We report the best average Accuracy (Acc) and Macro-F1 (%).

Dataset	Variant	Modules Activation			Acc.	Macro-F1
		Inn. 1	Inn. 2	Inn. 3	(%)	(%)
Digits-Five	No structure (FedAvg)	–	–	–	82.95	82.50
	FedBN (baseline)	–	–	–	85.73	84.85
	Ours (Inn. 1)	✓	–	–	87.18	85.63
	Ours (Inn. 1+2)	✓	✓	–	87.62	87.08
	Ours (Inn. 1+2+3)	✓	✓	✓	**88.54**	**88.00**
PACS	No structure (FedAvg)	–	–	–	61.86	60.40
	FedBN (baseline)	–	–	–	63.53	62.17
	Ours (Inn. 1)	✓	–	–	67.22	66.02
	Ours (Inn. 1+2)	✓	✓	–	67.81	66.64
	Ours (Inn. 1+2+3)	✓	✓	✓	**68.46**	**67.31**
DomainNet	No structure (FedAvg)	–	–	–	76.86	71.76
	FedBN (baseline)	–	–	–	78.43	72.06
	Ours (Inn. 1)	✓	–	–	79.98	74.55
	Ours (Inn. 1+2)	✓	✓	–	80.71	73.78
	Ours (Inn. 1+2+3)	✓	✓	✓	**81.71**	**76.84**

- **Inn. 1** denotes personalization with normalization decoupling, which enables client-wise personalized optimization while keeping batch normalization local to preserve client-specific statistics.
- **Inn. 2** denotes the lightweight frequency-aware channel reweighting module (MSCA) to enhance client-dependent discriminative patterns with negligible overhead.
- **Inn. 3** denotes global prototype alignment regularization, aggregating class prototypes across clients to mitigate semantic drift and improve class-wise separability.

IV. Experiments and Results

A. Experimental Setup

Experiments are conducted with K heterogeneous clients: Digits-Five uses K=5 clients (MNIST, MNIST-M, SVHN, SynthDigits, USPS), PACS uses K=4 clients (Photo, Art, Cartoon, Sketch), and DomainNet uses K=4 clients (Clipart, Painting, Real, Sketch) with ten classes. All methods share identical backbones and hyperparameters. Training runs for R=150 rounds on Digits-Five, R=100 on PACS, and R=150 on DomainNet. Results report average test Accuracy and Macro-F1 across clients, with best performance over rounds recorded. All results are averaged over three random seeds.

B. Ablation Study

We conduct ablation studies to quantify the contribution of each FedPAF component. We compare five variants: (i) FedAvg (no structure), (ii) FedBN (baseline), (iii) Ours (Inn. 1) enabling only the first innovation, (iv) Ours (Inn. 1+2) adding the second innovation, and (v) Ours (Inn. 1+2+3), the full model. All variants share the same backbone, data splits, and training hyperparameters; only the target modules are enabled/disabled.

Analysis of Results.Table I summarizes the best average *accuracy* and *macro F1* on Digits-Five, PACS, and DomainNet. Compared with FedAvg, FedBN consistently yields better performance, confirming the importance of decoupling client-specific normalization statistics. Building upon this baseline, Inn. 1 brings substantial gains across all datasets, indicating that personalization with normalization decoupling effectively improves client-adaptive learning. Further activating Inn. 2 improves both metrics on Digits-Five and PACS, and increases accuracy on DomainNet, suggesting that lightweight frequency-aware enhancement better captures client-dependent discriminative patterns. The full model achieves the best results across all benchmarks.

Notably, on DomainNet, Inn. 2 slightly reduces Macro-F1. This is likely because frequency-aware enhancement without semantic constraints may amplify class-wise imbalance across heterogeneous clients, to which Macro-F1 is highly sensitive. Adding Inn. 3 recovers this drop and delivers a clear net improvement, implying that prototype alignment effectively mitigates semantic drift and stabilizes class-wise separability under severe client heterogeneity. Fig. 2 further visualizes these progressive improvements.

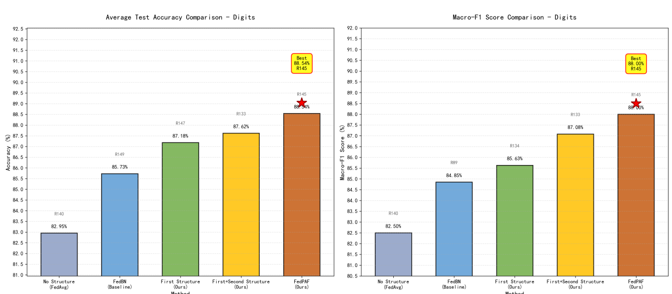


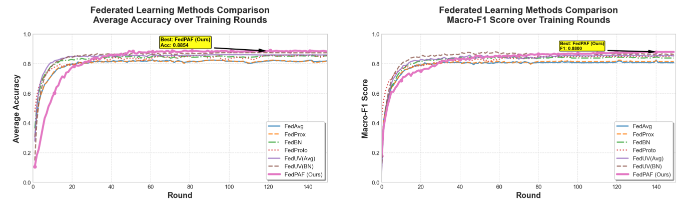
Fig. 2: Ablation comparison of average Accuracy and Macro-F1 on Digits-Five. Progressive activation of the three modules yields consistent gains.

C. Comparison with State-of-the-Art Methods

We compare FedPAF with representative federated learning baselines: FedAvg [1], FedProx [2], FedBN [3], FedUV [6], and FedProto [5]. Table II reports the per-client and averaged Accuracy and Macro-F1 (%) on Digits-Five, PACS, and DomainNet.

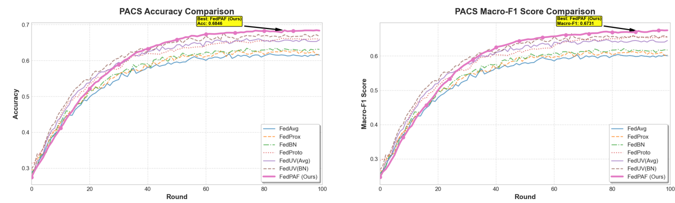
Since FedUV originally relies on FedAvg-style aggregation, we additionally implement FedUV (BN)—combining FedUV's regularization with FedBN's local normalization—to ensure a fair comparison under the same normalization-aware protocol.

As shown in Table II, FedPAF achieves the best average performance across all benchmarks. Compared to its foundational baseline, FedBN, FedPAF significantly boosts average accuracy by +2.81%, +4.93%, and +3.28% on Digits-Five, PACS, and DomainNet, respectively, with corresponding Macro-F1 improvements of +3.15%, +5.14%, and +4.78%. This confirms that localizing BN alone is insufficient, and



(a) Digits-Five: Accuracy (b) Digits-Five: Macro-F1

Fig. 3: Learning curves on Digits-Five for Accuracy and Macro-F1. FedPAF maintains the top trajectory and the best final plateau.



(a) PACS: Accuracy (b) PACS: Macro-F1

Fig. 4: Learning curves on PACS for Accuracy and Macro-F1. FedPAF keeps a clear and stable margin over strong baselines.

jointly addressing discriminative patterns and semantic consistency is highly beneficial. Furthermore, against recent strong methods like FedUV (BN), FedPAF still delivers consistent accuracy (+0.86%, +1.16%, +1.18%) and Macro-F1 (+1.00%, +1.21%, +0.65%) gains. Overall, FedPAF consistently outperforms optimization-based (FedAvg/FedProx), prototype-based (FedProto), and normalization-based baselines, validating its effectiveness under client heterogeneity.

The per-client breakdown in Table II shows the most pronounced gains on clients with highly distinct data characteristics (e.g., SVHN in Digits-Five, Sketch in PACS/DomainNet), directly validating our frequency-aware enhancement and prototype alignment. Despite marginal trade-offs on specific clients (e.g., Clipart on DomainNet), FedPAF achieves a superior overall balance and higher Macro-F1.

Finally, the learning curves in Figs. 3, 4, and 5 further demonstrate FedPAF's consistent advantages and stability in accuracy and Macro-F1 throughout the communication rounds.

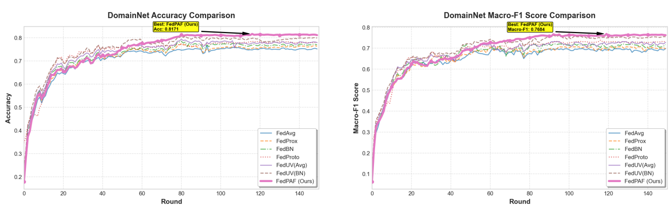
V. Conclusion

In this paper, we studied personalized federated learning under client heterogeneity, where clients exhibit both client-specific feature statistics and semantic inconsistency. We proposed FedPAF to address these challenges through three complementary mechanisms: personalization with normalization decoupling, frequency-aware feature enhancement, and class-level semantic alignment. By jointly optimizing local adaptation, client-dependent feature modeling, and cross-client consistency, FedPAF balances client-specific optimization with global semantic coherence.

Experiments on Digits-Five, PACS, and DomainNet show that FedPAF consistently improves both Accuracy and Macro-F1 over strong baselines, with stable optimization under se-

TABLE II: Comprehensive per-client and average performance comparison. For each method, we report Accuracy (Acc) and Macro-F1 (F1) as separate columns. Average results are reported as $\text{mean}_{\pm\text{std}}$ over 3 random seeds. Best results per metric are in **bold**.

Dataset	Client	FedAvg		FedProx		FedBN		FedProto		FedUV (Avg)		FedUV (BN)		Ours	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Digits-Five	MNIST	96.44	96.41	96.50	96.48	96.93	96.83	96.84	96.83	97.46	97.26	**98.05**	97.55	97.44	**97.43**
	SVHN	66.33	62.96	65.65	63.01	73.34	69.56	73.19	70.45	71.92	69.39	76.51	74.37	**78.71**	**77.00**
	USPS	96.02	95.64	96.29	96.00	97.10	96.29	97.26	96.34	97.21	96.15	**97.76**	**97.20**	97.51	96.67
	SynthDigits	81.76	82.21	82.29	81.85	83.44	83.11	85.25	85.23	86.34	85.15	86.48	85.98	**86.88**	**86.83**
	MNIST-M	77.00	77.83	78.18	78.28	80.11	79.90	83.31	83.17	85.85	81.21	85.86	81.82	**86.12**	**86.06**
	Average	$82.88_{\pm 0.32}$	$82.50_{\pm 0.28}$	$82.95_{\pm 0.35}$	$82.43_{\pm 0.31}$	$85.73_{\pm 0.29}$	$84.85_{\pm 0.26}$	$86.58_{\pm 0.38}$	$85.82_{\pm 0.34}$	$86.30_{\pm 0.41}$	$85.61_{\pm 0.37}$	$87.68_{\pm 0.28}$	$87.00_{\pm 0.25}$	$\mathbf{88.54}_{\pm 0.21}$	$\mathbf{88.00}_{\pm 0.19}$
PACS	Art	60.96	58.22	61.85	60.30	61.97	61.43	66.28	62.15	65.93	65.65	67.15	66.92	**68.49**	**68.35**
	Cartoon	64.51	65.75	65.82	66.55	67.81	67.63	70.54	71.13	69.91	69.81	71.35	71.42	**72.44**	**72.46**
	Photo	77.46	75.02	77.93	76.03	78.81	78.17	81.43	80.96	80.82	80.30	82.10	81.62	**83.50**	**83.13**
	Sketch	45.26	42.87	45.72	43.86	45.98	44.82	47.92	45.88	47.01	44.84	48.60	46.50	**49.50**	**47.47**
	Average	$61.86_{\pm 0.45}$	$60.40_{\pm 0.42}$	$62.68_{\pm 0.48}$	$61.55_{\pm 0.44}$	$63.53_{\pm 0.39}$	$62.17_{\pm 0.36}$	$66.48_{\pm 0.51}$	$65.00_{\pm 0.47}$	$65.89_{\pm 0.53}$	$64.62_{\pm 0.49}$	$67.30_{\pm 0.35}$	$66.10_{\pm 0.32}$	$\mathbf{68.46}_{\pm 0.27}$	$\mathbf{67.31}_{\pm 0.24}$
DomainNet	Clipart	80.58	74.74	82.04	75.13	83.01	77.79	83.18	78.31	80.10	78.51	**83.43**	78.97	82.52	**78.78**
	Painting	67.71	57.15	68.75	58.77	69.58	59.23	71.70	59.38	70.23	62.12	70.01	62.21	**72.92**	**63.79**
	Real	87.85	79.92	88.46	81.59	88.91	82.58	89.08	86.86	89.38	82.41	90.00	86.03	**92.00**	**87.73**
	Sketch	75.84	75.85	75.98	77.34	76.56	77.80	81.74	81.21	79.00	78.81	81.18	82.75	**84.27**	**84.25**
	Average	$76.86_{\pm 0.38}$	$69.82_{\pm 0.52}$	$77.45_{\pm 0.41}$	$70.77_{\pm 0.48}$	$78.43_{\pm 0.35}$	$72.06_{\pm 0.43}$	$78.53_{\pm 0.46}$	$73.86_{\pm 0.50}$	$79.42_{\pm 0.49}$	$73.20_{\pm 0.54}$	$80.53_{\pm 0.33}$	$76.19_{\pm 0.38}$	$\mathbf{81.71}_{\pm 0.26}$	$\mathbf{76.84}_{\pm 0.29}$



(a) DomainNet: Accuracy (b) DomainNet: Macro-F1

Fig. 5: Learning curves on DomainNet for Accuracy and Macro-F1. FedPAF converges more stably and reaches higher final performance.

vere client heterogeneity. These results suggest that jointly addressing statistical and semantic heterogeneity is beneficial for robust federated learning. The design principles introduced here—local adaptation, frequency-aware modeling, and semantic consistency—provide valuable insights for building robust personalized federated systems. Future work may explore partial participation and adaptive aggregation under more realistic settings.

Acknowledgment

This work was supported in part by the Joint Resource Optimization Method for MEC Systems Based on Deep Reinforcement Learning Project under Grant 62472368, and in part by the Basic Scientific Research Business Expenses Project for Universities in Xinjiang Uyghur Autonomous Region under Grant XJEDU2024J032 (Construction Type Project).

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat. (AISTATS)*, 2017, pp. 1273–1282.

[2] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Machine Learning Syst. (MLSys)*, 2020.

[3] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "FedBN: Federated learning on non-IID features via local batch normalization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[4] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 3557–3568.

[5] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang, "FedProto: Federated prototype learning across heterogeneous clients," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2022, pp. 8432–8440.

[6] H. M. Son, M.-H. Kim, T.-M. Chung, C. Huang, and X. Liu, "FedUV: Uniformity and variance for heterogeneous federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 5863–5872.

[7] A. Acar, Y. Zhao, R. Matas, M. Mattina, P. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[8] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 10713–10722.

[9] C. T. Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with Moreau envelopes," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 21394–21405.

[10] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 6357–6368.

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

[12] Y. Dai, Z. Chen, J. Li, S. Heinecke, L. Sun, and R. Xu, "Tackling data heterogeneity in federated learning with class prototypes," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2023, pp. 7314–7322.

[13] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 7132–7141.

[14] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[15] Z. Qin, P. Zhang, F. Wu, and X. Li, "FcaNet: Frequency channel attention networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 783–792.

[16] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, 2016, pp. 308–318.

[17] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, 2017, pp. 1175–1191.