Rethinking Fair Federated Learning from Parameter and Client View

Kaiqi Guan[1†], **Wenke Huang**[1†], **Xianda Guo**[1],
Yueyang Yuan[1], **Bin Yang**[1], **Mang Ye**[1*]
[1] National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence,
Hubei Key Laboratory of Multimedia and Network Communication Engineering,
School of Computer Science, Wuhan University, Wuhan, China.
{guankaiqi, wenkehuang, yemang}@whu.edu.cn

Abstract

Federated Learning is a promising technique that enables collaborative machine learning while preserving participant privacy. With respect to multi-party collaboration, achieving performance fairness acts as a critical challenge in federated systems. Existing explorations mainly focus on considering all parameter-wise fairness and consistently protecting weak clients to achieve performance fairness in federation. However, these approaches neglect two critical issues. 1) Parameter Redundancy: Redundant parameters that are unnecessary for fairness training may conflict with critical parameters update, thereby leading to performance degradation. 2) Persistent Protection: Current fairness mechanisms persistently enhance weak clients throughout the entire training cycle, hindering global optimization and causing lower performance alongside unfairness. To address these, we propose a strategy with two key components: First, parameter adjustment with mask and rescale which discarding redundant parameter and highlight critical ones, preserving key parameter updates and decrease conflict. Second, we observe that the federated training process exhibits distinct characteristics across different phases. We propose a dynamic aggregation strategy that adaptively weights clients based on local update directions and performance variations. Empirical results on single-domain and cross-domain scenarios demonstrate the effectiveness of the proposed solution and the efficiency of crucial modules. The code is available at https://github.com/guankaiqi/FedPW.

1 Introduction

Federated Learning (FL) is a collaborative machine learning framework [26, 57, 30, 27, 58, 19] that enables multiple clients to jointly train a global model [38, 31, 18] without sharing raw data. Clients process data locally and periodically send model updates to the server, which aggregates these updates into a global model. This training paradigm effectively addresses data island and privacy issues. However, due to data heterogeneity [58, 23], intermittent client participation, and system heterogeneity, the model is prone to unfairness, which diminishes FL's generalization capability.

Improving **performance fairness** [47, 19] is a central research focus in federated learning. Existing approaches can be categorized into three types: client selection [16, 42], weight allocation [39, 35, 48], and personalized local models [33]. For instance, to enhance the performance of underperforming clients, the federated server may assign them larger aggregation weights, amplifying their influence

[†] Equal Contribution.
[*] Corresponding Author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

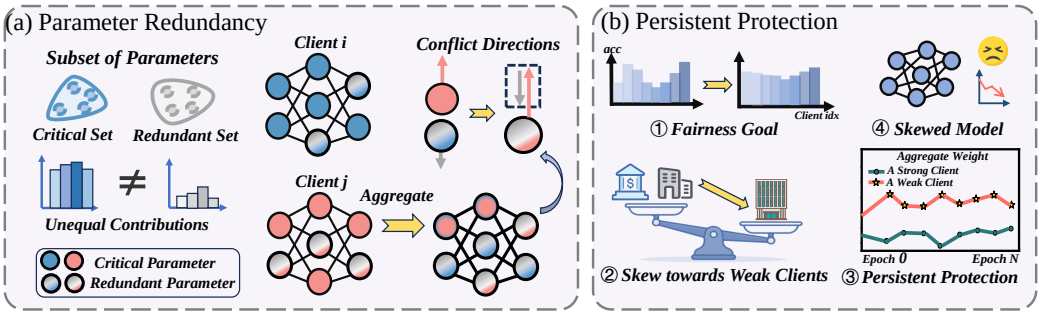


Figure 1: **Problem illustration** of existing fairness methods: Traditional approaches consider all model parameters for fairness, but not all parameters are equally important. Redundant parameters can conflict with key ones, disrupting their crucial contributions. Additionally, a long-term weighting strategy that favors weak clients is a classic approach. However, since weak clients often deviate from the global trend, this strategy can lead to a skewed initial training direction, resulting in an undertrained model and performance degradation.

on model updates. However, these methods typically compromise the performance of the global model. But training an effective global model is the primary goal of FL. This provokes our thinking:

> *Can we design an algorithm for FL that promotes fairness while improving the performance of the global model?*

We identify two primary causes of global model degradation: **I Parameter Redundancy**: *Due to the high redundancy in model parameters, a significant portion of parameters are inherently non-essential, making it unnecessary for all parameters to participate in model aggregation.* Owing to the over-parameterized characteristics of deep neural networks [10, 51, 59], not all parameters contribute equally to fitting the domain distribution. Specifically, only a subset of critical core parameters plays a decisive role in model training, while other non-core, less important parameters introduce noise that disrupts the parameter space. Moreover, these marginal parameters are prone to conflicting with other critical parameters, leading to performance degradation and unfairness. **II Persistent Protection**: *Fairness approaches typically protect weak clients throughout the entire training cycle, leading to suboptimal model optimization*. From a conventional fairness perspective, the existence of weak clients is paramount to ensuring equitable performance across heterogeneous participants. However, we argue that such clients may inherently act as outliers within the federated ecosystem. If the system rigidly applies a static prioritization strategy favoring underperforming clients across all training phases, the global model may become excessively influenced by their gradient directions during early stages, leading to a deviation in the model's optimization trajectory and resulting in performance degradation and unfairness.

To address the aforementioned problems, we propose FedPW(Fair **Fed**erated Learning via **P**arameter Adjustment and Adaptive **W**eighting). For problem **I**, we leverage Parameter Adjustment to address this challenge. Specifically, we observe that discarding small updates to parameters can reduce conflicts with clients from other domains without negatively affecting the model's performance on its own domain. Interestingly, we also find that different domains exhibit varying levels of tolerance for parameter discarding, which is inversely correlated with the domain's complexity. Therefore, we apply a domain-specific drop rate to discard the tail parameters of each client, ensuring fairness in the discarding process while mitigating conflicts between domains. However, the effect of the discarding operation is limited. To further reduce confusion during aggregation, we identify a set of consensus parameters for amplification, making the global model's update direction more stable and consistent, while scaling the discarded parameter updates back to their original magnitude, which is shown to benefit the model's performance [60, 15].

For Problem **II**, we propose an adaptive weighting mechanism that dynamically adjusts model aggregation weights based on the evolving training dynamics. We observe that the training process exhibits distinct phases. Early stages may benefit from reinforcing consensus to stabilize joint training, while later stages require emphasis on domain diversity to enhance fairness. To implement this insight, we dynamically allocate aggregation weights using the dot product between client parameter updates and loss variations. Our method achieves a transition from reinforcing consensus to emphasizing fairness, resulting in excellent performance and fairness. The details are presented in Sec. 3.3.

In this paper, FedPW consists of two main components. First, parameter adjustment alleviates parameter update conflicts by discarding certain parameters, and strengthens consensus to emphasize important parameters across multiple parties. Second, through the reweighting strategy, FedPW ensures balanced performance across clients from different domains, guaranteeing fairness in the diverse update directions during multi-party collaboration. The method is straightforward to implement and focuses on improving the aggregation step, making it easily compatible with other federated learning methods. The main contributions are summarized as follows:

❶ ***Re-examining Why might Fairness Methods Harm Model Performance***. Full parameter participation introduce unnecessary conflicts via redundant parameters, undermining model training.

❷ ***A Novel Partial-parameter Dynamic Aggregation Framework***. We discard minor updates while amplifying critical ones to adjust the model parameters. Furthermore, we propose an adaptive weighting strategy based on the dynamic characteristics of the training process, which mitigates the negative impact of lagging clients hindering the global model during the early stages of training.

❸ ***Extensive Experimental Validation***. We conduct experiments on single-domain and cross-domain scenarios. With ablations, we validate the efficacy of FedPW and the indispensability of modules.

2 Related Work

2.1 Heterogeneous Federated Learning

Statistical heterogeneity across parties, commonly referred to as the non-IID problem, poses significant challenges in Federated Learning (FL). The pioneering work FedAvg [38] demonstrated notable performance degradation under heterogeneous data settings. To address this, many approaches employ regularization terms to constrain local training. For instance, FedProx [34] introduced a proximal term to mitigate divergence between local and global models, while FedDyn [1], FedCurv, and pFedMe [49] adopted similar regularization strategies. Methods like MOON [31], FCCL [17], FedUFO [62], FedProto [50], FPL [18], and FedProc [40] incorporate alignment-based penalty terms to harmonize feature representations across clients, addressing data heterogeneity. SCAFFOLD [24] proposed a control variate mechanism to correct client drift by reducing gradient divergence. Other approaches tackle heterogeneity via prototype-based communication. FedProto [50] aligns global and local class prototypes to handle label distribution skew, though it primarily targets single-domain label skew. For cross-domain challenges, FPL [18] leverages clustered prototypes to generate unbiased global representations, while FedGA [63] and FedDG [36] focus on domain generalization for unseen target domains. However, these methods often involve full parameter updates during training, which introduces redundancy. Our approach emphasizes parameter adjustment to prioritize critical parameters, effectively resolving conflicts arising from redundant parameters across diverse domains.

2.2 Fair Federated Learning

Fairness has been a key focus in Federated Learning, with various concepts proposed, such as Performance Fairness [39, 22], Collaboration Fairness [37, 64, 55], and Group Fairness [9, 61, 6]. Performance Fairness, which aims to ensure similar accuracy across clients, is one of the most widely studied areas [32, 39]. Some methods address this by modifying client selection strategies. For instance, UCB-CS [5] uses a communication-efficient selection strategy based on multi-armed bandit theory, choosing clients with higher local loss to promote fairness and consistency. Other approaches focus on adjusting aggregation weights to unify training outcomes. A notable example is AFL [39], which minimizes maximum loss to improve the performance of the worst-performing devices. In contrast, q-FFL [35] introduces exponentially scaled weights to penalize clients with higher loss, leading to a more balanced accuracy distribution. FedHEAL [3] leverages the distance between local models and the global model to constrain unfair disparities. FedFV [53] uses cosine similarity to detect and resolve gradient conflicts iteratively, converging to a Pareto-stable solution. Ditto proposes a personalized federated learning framework that employs a penalty term to control the degree of model personalization, thereby achieving fairness and robustness. FedCE [22] leverages client contribution estimation as global model aggregation weights, demonstrating improved Performance Fairness and Collaboration Fairness. However, these methods often exhibit a persistent tendency to protect underperforming clients throughout the entire training cycle, which may hinder model training. Our method leverages the training dynamics, applying different weighting strategies at different stages, thereby achieving dual excellence in both generalization and fairness.

3 Methodology

3.1 Preliminary

Federated Learning and Performance Fairness. In typical federated learning, a system consists of K clients, each with private data $D_k = \{x_i, y_i\}_{i=1}^{N_k}$. At the start of each communication round t, the server shares the global model w^t with all clients. Each client initializes its local model w_k^t with w^t, performs local optimization using its data, and sends the updated parameters back to the server. The server then aggregates these updates using weighted averaging:

$$w_k^t \leftarrow w_k^t - \eta \nabla \sum_{i \in B_k} l(w_k^t, \xi_i), \quad w^{t+1} = \sum_k \lambda_k w_k^t. \tag{1}$$

Here, B_k is a mini-batch sampled from the local dataset D_k, ξ represents a query instance, and η is the local learning rate. The optimization objective is to minimize global loss:

$$\min_w F(w) = \sum_{k=1}^{K} \lambda_k f_k(w), \tag{2}$$

where λ_k is the weight of client and $f_k(w)$ is the loss of local model with parameters w.

Definition 3.1. (Performance Fairness) *Given two trained models, w and $\tilde{w}$, model w is considered to provide a fairer solution to the federated learning objective (2) if its performance across the m devices is more uniform compared to model $\tilde{w}$, i.e.* $\text{var}\left\{F_k(w)\right\}_{k\in[K]} < \text{var}\left\{F_k(\tilde{w})\right\}_{k\in[K]}$.

3.2 Parameter Adjustment

Motivation. Previous work has shown that model parameters often contain significant redundancy [60, 15, 2, 7, 8, 52], with only a small set of key parameters driving performance, while most parameters are redundant and ineffective. To investigate parameter redundancy, we conducted an observational experiment on the Digits dataset. We trained on 5 clients across four domains for 20 epochs, gradually increasing the mask rate while monitoring performance. As shown in Fig. 2, all domains showed almost no performance change when a small number of parameters were discarded, with performance degradation only occurring when a large proportion of parameters was discarded, indicating the redundancy of the parameters. Our findings revealed that each domain exhibited significant parameter redundancy, and domains with lower performance had less redundancy(Fig. 2). If redundant parameters are included in aggregation, they can overwhelm important updates, leading to confusion in the global model. Therefore, minimizing this redundancy is essential to ensure effective model updates. Our approach focuses on discarding unimportant parameters and enhancing the updates of key ones. The specific process is as follows:

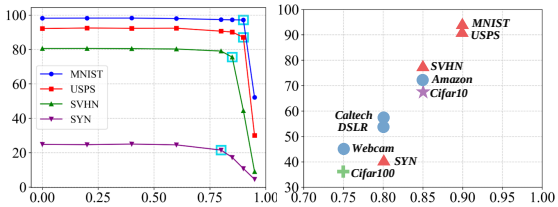


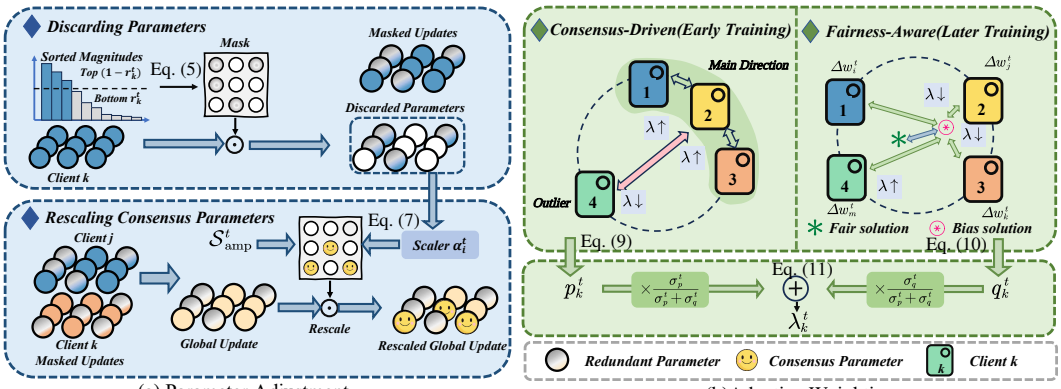
Figure 2: **Study on Parameter Redundancy**. Left: Model accuracy declines with increasing mask ratio. Right: Redundancy-performance relationship across datasets.

Selection of Unimportant Parameters. As previously established, the parameter updates Δw_k^t exhibit significant variation: while most parameters undergo negligible changes ($|\Delta w_{k,i}^t| \to 0$), a critical minority demonstrate substantial updates. We prune insignificant parameters by first representing the G-dimensional update vector:

$$\Delta w_k^t = [\Delta w_{k,1}^t, \ldots, \Delta w_{k,G}^t]. \tag{3}$$

Unimportant parameters are defined as those below threshold $\tau_k = \text{sorted}(|\Delta w_k^t|)[(1 - r_k^t)G]$, where $r_k^t \in (0, 1]$ is the client-specific mask rate. As demonstrated in Fig. 2, parameter redundancy inversely correlates with client performance (quantified by training loss). In our experiments, various methods to increase the parameter redundancy for clients with lower loss were proven effective, with the inverse of loss being the simplest and most effective approach. Thus, we compute r_k^t using the inverse of smoothed loss q_k^t from Eq. (9), where c is a hyper-parameter representing the average mask ratio.

$$r_k^t = c \cdot \frac{1/q_k^t}{\sum_{k=1}^{K} 1/q_k^t}. \tag{4}$$



(a) Parameter Adjustment (b)Adaptive Weighting

Figure 3: **Architecture illustration** of FedPW. FedPW consists of two core components: ❶ The left box refers to **Parameter Adjustment (PA)**, which adaptively discards redundant parameters and amplifies consensus ones (Sec. 3.2), reducing interference among parameters. ❷ The right box represents **Adaptive Weighted Aggregation (AWA)**, where we dynamically assign weights to each client according to the evolving training process (Sec. 3.3). In this way, conflicts among clients can be mitigated while reinforcing consensus.

Discarding Redundant Parameters. We use a mask $\mathcal{M}_{k,i}^t \in \mathbb{R}^G$ based on magnitudes to focus on important parameters which is used to discard the smallest subset of updates. The mask is defined as:

$$\mathcal{M}_{k,i}^t = \begin{cases} 1, & \text{if } |\Delta w_{k,i}^t| \geq \tau_k, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Finally, the masked updates are $\Delta \mathbf{w}_k^t = \Delta w_k^t \odot \mathcal{M}_k$, where $\odot$ denotes the Hadamard product. These masked updates are subsequently aggregated through $\Delta \mathcal{W}^t = \sum_{k=1}^K \lambda_k^t \Delta \mathbf{w}_k^t$, where λ_k^t denotes the weighting coefficients derived from Eq. (11).

Consensus Parameter Rescaling. To enhance directional consensus in global updates, we amplify parameters exhibiting cross-client agreement through a rescale process. First, we construct the client update matrix $\Delta \mathcal{W}^t = [\Delta w_1^t, \ldots, \Delta w_K^t]^\top \in \mathbb{R}^{K \times G}$ and perform normalization:

$$\widehat{\Delta w}_k^t = \frac{\Delta \mathbf{w}_k^t}{\|\Delta \mathbf{w}_k^t\|}, \quad \widetilde{\Delta w}_{k,i}^t = \frac{\widehat{\Delta w}_{k,i}^t}{\sum_{j=1}^G \widehat{\Delta w}_{k,j}^t}. \tag{6}$$

This applies row-wise and column-wise normalization to $\Delta \mathcal{W}^t$, enabling cross-client comparability of update directions while preserving their relative importance.

The consensus degree of parameter i is quantified by its standard deviation $\sigma_i^t = \text{std}(\{\widetilde{\Delta w}_{k,i}^t\}_{k=1}^K)$ across clients. We then amplify the most consistent parameters (those with the lowest σ_i) corresponding to the bottom ρ-quantile, where $\rho^t = \frac{1}{K}\sum_{k=1}^K r_k^t$ is the average mask rate. The purpose of using ρ^t to determine the amplification set is to rescale the aggregated gradients back to their original magnitude, which is shown to benefit the model's performance [60, 15].

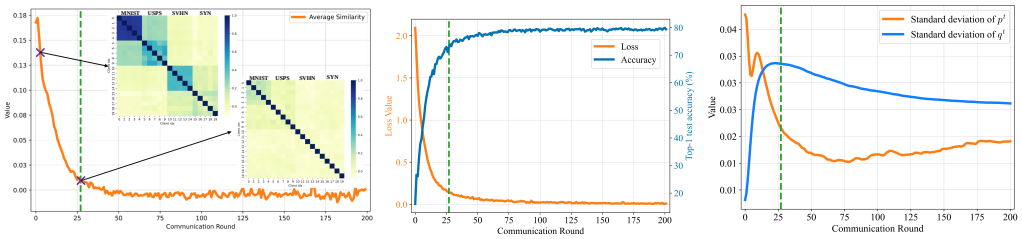
$$\alpha_i^t = \begin{cases} 1 + \frac{m_d^t}{m_a^t}, & i \in \mathcal{S}_{\text{amp}}^t = \{i \mid \sigma_i^t \leq \text{sorted}(\sigma^t)[(1-\rho^t)G]\}, \\ 1, & \text{otherwise,} \end{cases} \tag{7}$$

where m_d^t and m_a^t denote the average magnitudes of discarded and amplified parameters respectively. This rescaling ensures model outputs remain stable relative to the pre-discarding state. The final aggregated update becomes $\mathcal{W}^{t+1} = \mathcal{W}^t + \alpha^t \odot \Delta \mathcal{W}^t$.

3.3 Adaptive Weighted Aggregation

Motivation. Our analysis of federated learning reveals staged characteristics in collaborative training, as shown in 4. The two sides of the green line exhibit different behaviors, representing two phases:

Phase I: Early Training. The initial phase exhibits strong client alignment, where test accuracy improves rapidly through collective gradient coherence. However, while most clients converge to beneficial update directions, the clients in the SYN domain show lower cosine similarity with other clients. These divergent directions may skew the dominant direction, compromising the generalization



(a) Cosine similarity between clients (b) Loss and accuracy curve (c) Variation of σ_p^t and σ_q^t

Figure 4: **Illustration of Training Dynamics**. The federated training has two phases: early phase (left of green line) and later phase (right of green line). The two phases exhibit distinctly different characteristics. Please refer to Sec. 3.3 for a detailed discussion.

ability of the global model. Our strategy therefore enforces directional consensus during this phase, selectively amplifying clients contributing to coherent, generalizable updates.

Phase II: Later Training. As training progresses, client updates evolve toward orthogonality (near-zero cosine similarity), signaling exhausted consensus-driven gains. Here, the very underperformers that posed risks in Phase I become valuable sources of exploratory gradient diversity. Their divergent directions now prevent premature convergence and enable fairer parameter distributions across heterogeneous clients. We correspondingly shift weighting priorities to elevate these previously marginalized contributors, transforming gradient conflicts into fairness-enhancing signals.

This phased paradigm fundamentally motivates our dual-term adaptive weighting framework. By dynamically rebalancing generalization and fairness priorities in response to emergent training signatures, we transcend the limitations of static aggregation schemes.

3.3.1 Consensus-Driven Generalization

The design of our generalization term stems from a fundamental intricate and interdependent relationship between gradient alignment patterns and collective learning progress dynamics. By analyzing the first-order Taylor expansion of the global loss reduction:

$$\begin{aligned}\Delta\mathcal{L}_{total}^t &= \sum_{i=1}^{K}\Delta\mathcal{L}_i^t = \sum_{i=1}^{K}\left(\mathcal{L}_i^t\left(\mathbf{w}^t-\eta\mathbf{d}^t\right)-\mathcal{L}_i^t\left(\mathbf{w}^t\right)\right)\\ &\approx \sum_{i=1}^{K}-\eta\cdot\langle\mathbf{g}_i^t,\mathbf{d}^t\rangle = \sum_{i=1}^{K}-\eta\cdot\langle\mathbf{g}_i^t,\sum_{j=1}^{K}\lambda_j\mathbf{g}_j^t\rangle\\ &= -\eta\cdot\sum_{i=1}^{K}\sum_{j=1}^{K}\lambda_i\langle\mathbf{g}_i^t,\mathbf{g}_j^t\rangle.\end{aligned} \tag{8}$$

We establish that clients exhibiting higher gradient coherence (larger $\sum_j\langle\mathbf{g}_i^t,\mathbf{g}_j^t\rangle$) contribute more significantly to overall loss minimization. In practical training, $\mathbf{g}_j^t$ is equivalent to $\Delta\mathbf{w}_i^t$, and we assign more weight to those with a greater sum of dot products with other clients. This pairwise dot product formulation serves two purposes: it robustly captures directional consensus across skewed distributions through gradient similarity metrics, while simultaneously suppressing clients with divergent updates that could destabilize the global model. Additionally, we employ momentum updates [25], with the momentum coefficient decaying using a parameter γ_p, which helps constrain the oscillations that typically occur as the system approaches convergence.

$$\begin{aligned}\Delta p_m^t &= (1-\beta\gamma_p)\Delta p_k^{t-1}+\beta\gamma_p\frac{\sum_{i=1}^{K}\langle\Delta\mathbf{w}_i^t,\Delta\mathbf{w}_j^t\rangle}{\sum_{i,j}\langle\Delta\mathbf{w}_i^t,\Delta\mathbf{w}_j^t\rangle},\\ \gamma_p &= \frac{\overline{\langle\Delta\mathbf{w}_i^t,\Delta\mathbf{w}_j^t\rangle}}{\overline{\langle\Delta\mathbf{w}_i^0,\Delta\mathbf{w}_j^0\rangle}},\ p_k^t=p_k^{t-1}+\Delta p_k^t,\ p_k^t=\frac{p_k^t}{\sum_{j=1}^{K}p_j^t},\end{aligned} \tag{9}$$

where $\overline{\langle\Delta\mathbf{w}_i^t,\Delta\mathbf{w}_j^t\rangle}$ represents the average value of the dot products between clients during epoch t.

3.3.2 Diversity-Enhanced Fairness

A natural approach to achieving fairness, as defined in (1), is to reweight the aggregation process by assigning higher weights to devices with poor performance. We use the loss as a proxy for performance and assign higher weights to clients with higher loss values, applying the same momentum

update to the fairness score as described previously.

$$\Delta q_k^t = (1-\beta\gamma_q)\Delta q_k^{t-1} + \beta\gamma_q \frac{\mathcal{L}_i^t}{\sum_j^M \mathcal{L}_i^t}, \gamma_q = \frac{\overline{\mathcal{L}_i^t}}{\overline{\mathcal{L}_0^t}},$$
$$q_k^t = q_k^{t-1} + \Delta q_k^t, \quad q_k^t = \frac{q_k^t}{\sum_{j=1}^K q_j^t}. \tag{10}$$

To reduce the number of hyper-parameters, we adopt the same β as in Eq. (9), and γ_q serves the same role as γ_p. This mechanism aims to shift the weights in favor of disadvantaged clients, thereby achieving more uniform performance while also accounting for domain diversity.

3.3.3 Phase-Adaptive Weight Synthesis

Our phased analysis reveals a critical duality in federated optimization objectives: the alignment of updates and the disparity in loss exhibit opposing dominance patterns across different training phases. To empirically validate this phase-dependent dichotomy, we quantify the client-wise standard deviations of generalization (σ_p^t) and fairness (σ_q^t) weights throughout training. As shown in Fig. 4c, σ_p^t dominates in the early rounds, while σ_q^t surpasses it in later stages.

- **Early Phase**: $\sigma_p^t > \sigma_q^t$, high gradient alignment variability among clientsleads to prioritizing consensus-driven updates to maximize collective progress.
- **Later Phase**: $\sigma_q^t > \sigma_p^t$, the later-phase exhibits diminishing gradient coherence but increasing loss disparity, necessitating interventions that emphasize fairness.

This systematic inversion motivates our adaptive weighting mechanism, which dynamically rebalances the two objectives based on their relative variability:

$$\lambda_i = \frac{\sigma_p^t}{\sigma_p^t + \sigma_q^t} p_i^t + \frac{\sigma_q^t}{\sigma_p^t + \sigma_q^t} q_i^t. \tag{11}$$

This variance-regulated synthesis automatically shifts emphasis between consensus-seeking and disparity-reduction modes throughout the training dynamics.

3.4 Discussion and Limitation

Comparison with Analogous Methods. AFL [39], q-FFL [35], and FedCE [22] prioritize weak clients using single metrics (*e.g.* loss/accuracy), but simply increasing their weights is not appropriate. In contrast, our method first suppresses then amplifies updates for straggling clients, aligning with natural learning dynamics. This strategy maintains update consistency without sacrificing domain diversity. While prior works [44, 43] like FedLF address update conflicts via gradient projection, trivial parameters still disrupt global optimization. Our parameter adjustment selectively discards non-critical updates to preserve essential ones, enhancing both generalization and fairness.

Discussion on Parameter Adjustment. Our method addresses update conflicts in large-scale FL through selective parameter pruning. This strategy enables performance gains for conflicting clients with minimal self-degradation, achieving collective enhancement. Concurrent gradient rescaling preserves original gradient norm magnitudes while amplifying critical parameters, effectively reducing parameter space conflicts and stabilizing multi-client collaboration.

Limitations. Our method employs parameter adjustment and adaptive weighting to adjust model aggregation. However, setting the hyper-parameter c to excessively high values may cause instability, exhibits sensitivity to selection. Our approach requires all participating clients to maintain identical network architecture specifications, which may limit the broader applicability of this method.

4 Experiments

We perform experiments on image classification tasks in various single-domain and cross-domain scenarios to validate the superiority of our framework FedPW.

4.1 Experiment Setup

Datasets. Following [18, 21, 43], we evaluate our method on single-domain datasets Fashion-Mnist [54], Cifar10 [28], Cifar100, and cross-domain datasets Digits [29] and Office-Caltech [11].

Table 1: Comparison of Average Accuracy (and Standard Deviation) with baselines in single-domain scenarios.

Methods	FMNIST			CIFAR-10			CIFAR-100		
	Dir(0.1)	Pat-1	Pat-2	Dir(0.1)	Pat-1	Pat-2	Dir(0.1)	Pat-1	Pat-2
FedAvg	87.0(11.5)	82.3(14.2)	82.9(11.4)	68.1(15.0)	56.2(20.4)	68.7(19.3)	37.1(7.5)	20.5(15.7)	21.9(15.3)
q-FFL	86.1(9.9)	84.0(13.5)	79.3(9.6)	67.8(14.1)	58.6(17.5)	68.8(18.1)	37.9(6.8)	17.6(15.1)	23.1(13.9)
AFL	87.0(9.5)	82.2(17.2)	83.7(11.8)	64.3(13.8)	56.5(14.4)	65.6(14.0)	39.1(7.0)	17.5(16.6)	26.7(14.6)
Ditto	79.9(10.6)	75.5(21.4)	77.8(10.2)	57.9(13.1)	45.7(11.3)	52.6(14.2)	27.9(7.4)	18.1(20.1)	11.2(11.4)
FedProx	81.9(10.0)	84.1(11.9)	84.3(8.8)	53.4(13.4)	57.3(11.9)	56.8(12.2)	18.7(5.9)	20.3(14.7)	19.1(12.0)
FedFV	82.8(9.4)	88.5(11.7)	81.6(13.4)	68.1(13.3)	55.1(18.3)	69.4(18.9)	37.3(6.9)	18.9(14.5)	21.9(13.7)
FedCKA	89.0(10.1)	79.3(18.8)	87.4(10.9)	67.3(14.3)	56.4(20.4)	69.3(14.4)	37.1(7.0)	19.6(15.7)	21.3(14.4)
FedSAC	84.1(13.1)	74.5(23.6)	83.4(18.1)	58.7(14.3)	44.8(15.3)	55.8(15.3)	33.4(6.1)	25.4(14.3)	7.5(3.1)
FedGCR	85.7(10.0)	83.2(13.5)	84.0(11.3)	66.6(13.8)	61.5(19.1)	65.5(16.9)	38.6(6.7)	20.6(14.5)	19.9(15.6)
FedHeal	85.7(11.3)	85.8(13.9)	84.0(11.6)	71.0(14.4)	58.7(22.4)	66.8(14.1)	38.7(6.8)	19.6(14.9)	22.8(14.3)
FedAA	86.8(7.7)	85.7(8.8)	87.9(7.0)	73.8(12.7)	73.2(10.4)	71.4(11.3)	38.1(7.0)	27.4(13.2)	34.2(12.7)
FedPW	**88.2(7.1)**	**89.4(8.0)**	**90.2(6.6)**	**75.3(11.1)**	**75.1(10.3)**	**76.3(9.9)**	**41.2(6.6)**	**35.2(14.2)**	**38.4(12.6)**

Table 2: Cross-domain comparison of Average Accuracy (**AVG**) and Standard Deviation (**STD**) with baseline.

Methods	Digits						Office-Caltech					
	MNIST	USPS	SVHN	SYN	AVG ↑	STD ↓	Amazon	DSLR	Caltech	Webcam	AVG ↑	STD ↓
FedAvg	93.23	91.01	79.13	40.02	75.85	24.67	72.36	56.93	59.19	45.92	58.60	10.85
+AFL	93.73	93.42	75.42	44.25	76.71 ↑0.86	23.27 ↓1.40	64.34	65.65	57.21	47.52	58.68 ↑0.08	8.31 ↓2.54
+q-FFL	93.89	90.63	77.93	44.68	76.78 ↑0.93	22.48 ↓2.19	59.41	64.75	52.60	51.33	57.02 ↓1.58	6.26 ↓4.59
+FedHEAL	93.12	94.12	79.13	46.36	78.18 ↑2.33	22.29 ↓2.38	67.49	66.81	59.82	54.83	62.24 ↑3.64	6.03 ↓4.82
+**FedPW**	94.28	93.62	80.76	49.43	**79.52** ↑3.67	**21.00** ↓3.67	68.64	65.95	59.76	58.62	**63.24** ↑4.64	**4.83** ↓6.02
FedProx	93.64	91.14	80.53	41.93	76.81	23.94	70.31	57.83	59.86	44.79	58.20	10.48
+AFL	93.72	95.23	75.44	43.15	76.89 ↑0.08	24.22 ↑0.28	67.18	63.52	59.65	53.08	60.86 ↑2.66	6.03 ↓4.45
+q-FFL	94.05	93.49	75.73	44.36	76.91 ↑0.10	23.31 ↓0.63	62.27	73.62	54.39	55.46	61.44 ↑3.24	8.84 ↑1.64
+FedHEAL	92.15	93.58	78.89	44.61	77.31 ↑0.50	22.78 ↓1.16	66.17	72.65	58.09	56.95	63.46 ↑5.26	7.37 ↓3.11
+**FedPW**	94.33	92.46	80.19	48.62	**78.90** ↑2.09	**21.14** ↓2.80	68.40	70.67	59.86	58.94	**64.47** ↑6.27	**5.94** ↓4.54
MOON	92.65	92.81	80.51	39.63	76.40	25.18	73.01	60.29	59.66	47.54	60.13	10.40
+AFL	93.14	95.12	74.68	44.48	76.86 ↑0.46	23.46 ↓1.72	66.70	68.20	61.54	54.50	62.74 ↑2.61	6.18 ↓4.22
+q-FFL	92.31	94.51	75.98	43.67	76.62 ↑0.22	23.47 ↓1.71	64.90	65.85	53.88	58.93	60.89 ↑0.76	5.59 ↓4.81
+FedHEAL	93.23	94.31	80.81	45.12	78.37 ↑1.97	23.00 ↓2.18	68.16	64.95	59.17	59.51	62.95 ↑2.82	4.37 ↓6.03
+**FedPW**	93.71	94.61	81.53	48.27	**79.53** ↑3.13	**21.68** ↓3.50	66.73	65.84	59.47	60.34	**63.10** ↑2.97	**3.72** ↓6.68
FedDyn	94.15	94.82	80.29	40.48	77.44	25.53	70.11	61.56	59.78	48.15	59.90	9.04
+AFL	94.37	96.14	70.95	41.28	75.69 ↓1.75	25.65 ↑0.12	70.84	57.86	60.57	50.99	60.06 ↑0.16	8.24 ↓0.80
+q-FFL	94.71	94.26	75.33	42.79	76.77 ↓0.67	24.39 ↓1.14	62.99	66.44	55.76	55.36	60.14 ↑0.24	5.47 ↓3.57
+FedHEAL	94.61	95.72	79.94	43.72	78.50 ↑1.06	24.27 ↓1.26	67.55	60.53	58.86	53.38	60.08 ↑0.18	5.84 ↓3.20
+**FedPW**	94.61	95.72	80.06	46.84	**79.31** ↑1.87	**22.79** ↓2.74	66.93	63.27	58.83	54.67	**60.93** ↑1.03	**5.33** ↓3.71
Independent Methods												
Ditto	93.62	91.58	79.55	40.65	76.42	24.65	56.94	69.67	56.73	56.82	60.04	6.42
FedFV	94.92	94.70	76.77	40.83	76.81	25.45	61.83	71.72	54.97	58.92	61.96	7.15
FedCKA	91.53	94.72	78.82	46.48	78.25	22.48	67.24	64.03	59.76	49.10	60.03	7.91
FedGCR	93.14	95.12	78.26	44.48	77.75	23.42	66.70	68.20	62.54	54.50	62.99	6.14
FedSAC	92.16	93.75	78.61	41.67	76.55	24.22	52.39	51.68	55.37	44.16	50.90	4.77
FedAA	92.91	92.28	78.57	47.16	77.73	21.43	62.89	68.71	56.83	56.02	61.11	5.92
FedPW	94.28	93.62	80.76	49.43	**79.52**	**21.00**	68.64	65.95	59.76	58.62	**63.24**	**4.83**

Data Heterogeneity.. To simulate heterogeneous clients in FL, we consider three scenarios: (1) Dir(α): We simulate m clients in Dirichlet heterogeneous partition. The smaller α is, the more imbalanced the local distribution is. (2) Pat-1: It constructs a difficult data-island scenario where each client only has data from one class. (3) Pat-2: We follow FedAvg to build pathological non-IID data where each client has data from two classes.

Model. For the single-domain scenario, we conduct experiments with CNN(two convolutional layers) [53]. For cross-domain scenarios, we use ResNet-10 [14] whose feature vector dimension is 512. Note that all methods use the same network architecture for fair comparison across different tasks.

Counterparts. We compare our method with FedAvg [38] and fairness-focused FL approaches: AFL [39] , q-FFL [35], and FedHEAL[3] (both integrable). For non-integrable frameworks like Ditto [33], FedFV [53], FedGCR[4], FedSAC[56] and FedAA[13], we perform full end-to-end benchmarking. This ensures comprehensive evaluation across all baseline categories.

Implement Details. Following [3, 43], in the single-domain setting, we employ 100 clients for 3,000 communication epochs, where all federated learning methods exhibit minimal or no accuracy

improvement beyond this point. Each epoch involves 10% client participation. We use the SGD optimizer with a learning rate of 0.1 and a batch size of 50. For the cross-domain setting, we allocate 20 clients per task and equal clients per domain, with clients randomly assigned to domains. The training runs for E = 200 communication epochs with T = 10 local updates per round. Each epoch involves all clients. The SGD uses a learning rate of 0.001, and momentum is 0.9. The batch sizes are 64 for Digits and 16 for Office-Caltech. We fix the random seed to ensure reproduction and conduct experiments on the NVIDIA 3090Ti. The hyperparameter settings are detailed in Sec. 4.3.

Evaluation Metric. Following [34, 38], Top-1 accuracy is adopted for model performance evaluation. For the single-domain setting, we use the standard deviation of accuracy across clients, while for the cross-domain setting, we use the standard deviation across domains for fair evaluation. We conduct experiments three times and utilize the accuracy of the last five epochs as the final performance.

4.2 Comparison to State-of-the-Arts

We benchmark FedPW against contemporary approaches addressing Performance Fairness in FL, with comprehensive results presented in Tab. 1 and Sec. 4.1. Our method establishes new state-of-the-art performance, achieving superior mean accuracy while maintaining the lowest standard deviation across several scenarios. The convergence analysis in Fig. 5 further demonstrates FedPW's accelerated training dynamics compared to existing methods and several key observations are summarized:

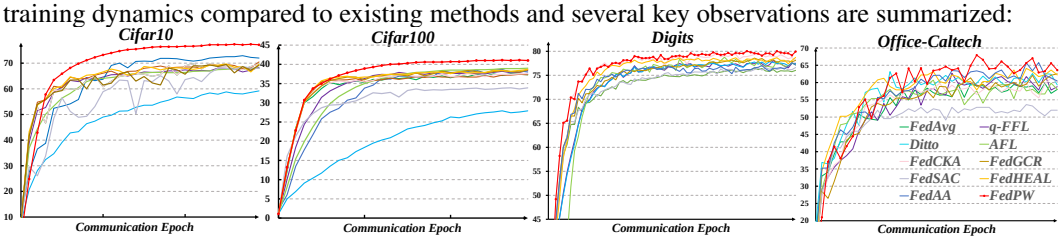


Figure 5: **Comparison of convergence of average accuracy** with counterparts. Please see details in Sec. 4.2.

❶ ***FedPW achieves an optimal trade-off between overall performance and fairness.*** Through the synergistic collaboration of the PA and AWA modules, FedPW simultaneously promotes fairness while improving the performance of the global model.

❷ ***Existing fairness-oriented methods risk degrading the accuracy of global model or advantaged clients.*** For instance, q-FFL and AFL exhibit performance degradation compared to FedAvg on Cifar-10, and similar phenomena occur with SVHN in the Digits benchmark.

❸ ***FedPW safeguards performance for disadvantaged parties.*** Underperforming domains like SYN in Digits and Webcam in Office-Caltech achieve improvements under FedPW's framework.

4.3 Diagnostic Experiments

Compatibility Study. To validate the compatibility of FedPW, we compared the results of several widely-adopted FL methods, FedAvg [38], FedProx [34], FedDyn [1], without and with FedPW. The results are shown in Sec. 4.1.

Table 3: **Ablation study** on multiple datasets. Please refer to Sec. 4.3 for detailed discussion.

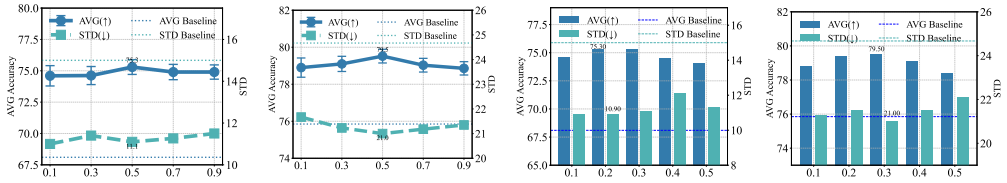
Setting		FMNIST		CIFAR-10		CIFAR-100		Digits		Office-Caltech	
PA	AWA	AVG ↑	STD ↓	AVG ↑	STD ↓	AVG ↑	STD ↓	AVG ↑	STD ↓	AVG ↑	STD ↓
✗	✗	87.03	11.21	68.13	15.02	37.12	7.48	75.85	24.67	58.60	10.85
✓	✗	88.05	10.34	72.88	13.94	38.64	7.81	77.85	23.62	60.65	9.92
✗	✓	87.34	8.62	72.12	12.84	40.01	7.27	78.58	22.08	62.39	5.05
✓	✓	**88.22**	**7.13**	**75.28**	**11.10**	**41.22**	**6.57**	**79.52**	**21.00**	**63.24**	**4.83**

Ablation Study. We conducted an ablation study to analyze the contributions of Parameter Adjustment (PA) and Adaptive Weighted Aggregation (AWA) components, as summarized in Tab. 3. Our findings indicate that each module positively contributes to performance, with optimal results achieved through their combination.

Table 4: AVG(STD) under different number of clients on Digits.

Methods	Client scales		
	20	60	100
FedAvg	75.9(24.7)	86.2(16.7)	87.1(16.0)
FedPW	79.5(21.0)	88.9(9.8)	91.3(7.8)

Hyper-parameter Analysis. We systematically investigate the impact of two critical hyper-parameters: the client selection rate c (Eq. (4)) and the momentum coefficient β (Eq. (9)). Here, β is solely used for momentum updates to mitigate drastic fluctuations during model training. Fig. 6 shows that β has a limited impact on model performance, though the error bars indicate that increasing β leads to more stable accuracy. The optimal values are set as defaults for subsequent experiments.



(a) Cifar10 for β (b) Digits for β (c) Cifar10 for c (d) Digits for c

Figure 6: **Hyper-parameter study** with variant β (Eq. (9)) and variant c (Eq. (4)). See details in Sec. 4.3.

5 Conclusion

In this paper, we explore the fairness challenges arising from domain skew in heterogeneous federated learning. We propose a simple yet effective federated learning algorithm, FedPW, to address two critical issues: Parameter Redundancy and Persistent Favoritism. Specifically, we utilize gradient information from model training to selectively discard and reinforce parameters. Furthermore, by leveraging training dynamics across epochs, our method achieves adaptive weighted aggregation. The effectiveness of FedPW has been extensively validated against several popular methods across various classification tasks. We hope that this work will serve as a foundation for future research.

Acknowledgement

This work is supported by National Natural Science Foundation of China under Grant (62361166629, 623B2080, 62501428), the Major Project of Science and Technology Innovation of Hubei Province (2024BCA003, 2025BEA002), and the Innovative Research Group Project of Hubei Province under Grants 2024AFA017. The supercomputing system at the Supercomputing Center of Wuhan University supported the numerical calculations in this paper.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *ICLR*, 2021.

[2] Tian Bowen, Lai Songning, Wu Jiemin, Shuai Zhihao, Ge Shiming, and Yue Yutao. Beyond task vectors: Selective task arithmetic based on importance metrics. *arXiv preprint arXiv:2411.16139*, 2024.

[3] Yuhang Chen, Wenke Huang Huang, and Mang Ye. Fair federated learning under domain skew with local consistency and domain diversity. In *CVPR*, 2024.

[4] Shu-Ling Cheng, Chin-Yuan Yeh, Ting-An Chen, Eliana Pastor, and Ming-Syan Chen. Fedgcr: Achieving performance and fairness for federated learning with distinct client types via group customization and reweighting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11498–11506, 2024.

[5] Yae Jee Cho, Samarth Gupta, Gauri Joshi, and Osman Yağan. Bandit-based communication-efficient client selection strategies for federated learning. In *2020 54th Asilomar Conference on Signals, Systems, and Computers*, pages 1066–1069. IEEE, 2020.

[6] Sen Cui, Weishen Pan, Jian Liang, Changshui Zhang, and Fei Wang. Addressing algorithmic disparity and performance inconsistency in federated learning. In *NeurIPS*, pages 26091–26102, 2021.

[7] Pala Tej Deep, Rishabh Bhardwaj, and Soujanya Poria. Della-merging: Reducing interference in model merging through magnitude-based sampling. *arXiv preprint arXiv:2406.11617*, 2024.

[8] Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim Kuan Goh, Ho-Kin Tang, Daojing He, et al. Parameter competition balancing for model merging. *arXiv preprint arXiv:2410.02396*, 2024.

[9] Yahya H Ezzeldin, Shen Yan, Chaoyang He, Emilio Ferrara, and A Salman Avestimehr. Fairfed: Enabling group fairness in federated learning. In *AAAI*, pages 7494–7502, 2023.

[10] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[11] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073, 2012.

[12] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[13] Jialuo He, Wei Chen, and Xiaojin Zhang. Fedaa: A reinforcement learning perspective on adaptive aggregation for fair and robust federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 17085–17093, 2025.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[15] Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. Emr-merging: Tuning-free high-performance model merging. *arXiv preprint arXiv:2405.17461*, 2024.

[16] Tiansheng Huang, Weiwei Lin, Wentai Wu, Ligang He, Keqin Li, and Albert Y Zomaya. An efficiency-boosting client selection scheme for federated learning with fairness guarantee. *IEEE Transactions on Parallel and Distributed Systems*, 32(7):1552–1564, 2020.

[17] Wenke Huang, Mang Ye, and Bo Du. Learn from others and be yourself in heterogeneous federated learning. In *CVPR*, 2022.

[18] Wenke Huang, Mang Ye, Zekun Shi, He Li, and Bo Du. Rethinking federated learning with domain shift: A prototype view. In *CVPR*, pages 16312–16322, 2023.

[19] Wenke Huang, Mang Ye, Zekun Shi, Guancheng Wan, He Li, Bo Du, and Qiang Yang. A federated learning for generalization, robustness, fairness: A survey and benchmark. *arXiv*, 2023.

[20] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE PAMI*, pages 550–554, 1994.

[21] Yongzhe Jia, Xuyun Zhang, Hongsheng Hu, Kim-Kwang Raymond Choo, Lianyong Qi, Xiaolong Xu, Amin Beheshti, and Wanchun Dou. Dapperfl: Domain adaptive federated learning with model fusion pruning for edge devices. *arXiv preprint arXiv:2412.05823*, 2024.

[22] Meirui Jiang, Holger R Roth, Wenqi Li, Dong Yang, Can Zhao, Vishwesh Nath, Daguang Xu, Qi Dou, and Ziyue Xu. Fair federated medical image segmentation via client contribution estimation. In *CVPR*, 2023.

[23] Xuefeng Jiang, Sheng Sun, Yuwei Wang, and Min Liu. Towards federated learning against noisy labels via local self-regularization. In *CIKM*, pages 862–873, 2022.

[24] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. In *ICML*, 2020.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[27] Zhiqiang Kou, Jing Wang, Jiawei Tang, Yuheng Jia, Boyu Shi, and Xin Geng. Exploiting multi-label correlation in label distribution learning. In Kate Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 4326–4334. International Joint Conferences on Artificial Intelligence Organization, 8 2024. Main Track.

[28] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[29] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, pages 2278–2324, 1998.

[30] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *ICDE*, pages 965–978, 2022.

[31] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *CVPR*, pages 10713–10722, 2021.

[32] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Federated multi-task learning for competing constraints. *arXiv preprint arXiv:2012.04221*, 2020.

[33] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *ICML*, pages 6357–6368, 2021.

[34] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *MLSys*, 2020.

[35] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. In *ICLR*, 2020.

[36] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *CVPR*, pages 1013–1023, 2021.

[37] Lingjuan Lyu, Xinyi Xu, Qian Wang, and Han Yu. Collaborative fairness in federated learning. *Federated Learning: Privacy and Incentive*, pages 189–204, 2020.

[38] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pages 1273–1282, 2017.

[39] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *ICML*, pages 4615–4625. PMLR, 2019.

[40] Xutong Mu, Yulong Shen, Ke Cheng, Xueli Geng, Jiaxuan Fu, Tao Zhang, and Zhiwei Zhang. Fedproc: Prototypical contrastive federated learning on non-iid data. *arXiv preprint arXiv:2109.12273*, 2021.

[41] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop*, 2011.

[42] Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE international conference on communications (ICC)*, pages 1–7. IEEE, 2019.

[43] Zibin Pan, Chi Li, Fangchen Yu, Shuyi Wang, Haijin Wang, Xiaoying Tang, and Junhua Zhao. Fedlf: Layer-wise fair federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14527–14535, 2024.

[44] Zibin Pan, Shuyi Wang, Chi Li, Haijin Wang, Xiaoying Tang, and Junhua Zhao. Fedmdfg: Federated learning with multi-gradient descent and fair guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9364–9371, 2023.

[45] Prasun Roy, Subhankar Ghosh, Saumik Bhattacharya, and Umapada Pal. Effects of degradations on deep neural network architectures. *arXiv preprint arXiv:1807.10108*, 2018.

[46] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, 2010.

[47] Yuxin Shi, Han Yu, and Cyril Leung. Towards fairness-aware federated learning. *IEEE TNNLS*, 2023.

[48] Yaqi Sun, Shijing Si, Jianzong Wang, Yuhan Dong, Zhitao Zhu, and Jing Xiao. A fair federated learning framework with reinforcement learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.

[49] Canh T. Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. In *NeurIPS*, pages 21394–21405, 2020.

[50] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *AAAI*, 2022.

[51] Yuandong Tian, Tina Jiang, Qucheng Gong, and Ari Morcos. Luck matters: Understanding training dynamics of deep relu networks. *arXiv preprint arXiv:1905.13405*, 2019.

[52] Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. *arXiv preprint arXiv:2405.07813*, 2024.

[53] Zheng Wang, Xiaoliang Fan, Jianzhong Qi, Chenglu Wen, Cheng Wang, and Rongshan Yu. Federated learning with fair averaging. In *IJCAI*, 2021.

[54] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[55] Xinyi Xu, Lingjuan Lyu, Xingjun Ma, Chenglin Miao, Chuan Sheng Foo, and Bryan Kian Hsiang Low. Gradient driven rewards to guarantee fairness in collaborative machine learning. In *NeurIPS*, volume 34, pages 16104–16117, 2021.

[56] Kunda Yan, Sen Cui, Abudukelimu Wuerkaixi, Jingfeng Zhang, Bo Han, Gang Niu, Masashi Sugiyama, and Changshui Zhang. Balancing similarity and complementarity for federated learning. 2024.

[57] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM TIST*, pages 1–19, 2019.

[58] Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *CSUR*, 2023.

[59] Yun Ye, Ganmei You, Jong-Kae Fwu, Xia Zhu, Qing Yang, and Yuan Zhu. Channel pruning via optimal thresholding. In *Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 18–22, 2020, Proceedings, Part V 27*, pages 508–516. Springer, 2020.

[60] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

[61] Yuchen Zeng, Hongxu Chen, and Kangwook Lee. Improving fairness via federated learning. *arXiv preprint arXiv:2110.15545*, 2021.

[62] Lin Zhang, Yong Luo, Yan Bai, Bo Du, and Ling-Yu Duan. Federated learning for non-iid data via unified feature learning and optimization objective alignment. In *ICCV*, pages 4420–4428, 2021.

[63] Ruipeng Zhang, Qinwei Xu, Jiangchao Yao, Ya Zhang, Qi Tian, and Yanfeng Wang. Federated domain generalization with generalization adjustment. In *CVPR*, pages 3954–3963, June 2023.

[64] Zirui Zhou, Lingyang Chu, Changxin Liu, Lanjun Wang, Jian Pei, and Yong Zhang. Towards fair federated learning. In *ACM SIGKDD*, pages 4100–4101, 2021.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main claims of our work, including an overview of FedPW. In addition, the main contributions and existing limitations are logically outlined in Sec. 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Justification: In the dedicated section (Sec. 3.4), we analyze the limitations of FedPW, including its sensitivity to hyper-parameter. Moreover, all clients should maintain identical network architecture specifications, which may limit the broader applicability.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not include formal theoretical results or theorems that require specific assumptions and complete proofs. The methodology presented in Sec. 3 is empirical and algorithmic, accompanied by the corresponding descriptions and formulas.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: This paper provides sufficient experimental details to reproduce the results, including model backbone types, datasets with partitioning methods, the number of tasks, related training hyper-parameters, specific evaluation metrics and exact baseline settings.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is accessible in this paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all necessary training and evaluation details, including model, datasets, dataset splits per task, batch size, and methods to control the non-IID level in Sec. 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Please refer to Sec. 4.2.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to Sec. 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work addresses the issue of catastrophic performance degradation caused by traditional fairness methods and has been validated across diverse data scenarios. The study does not involve human subjects or sensitive personal data, presents no adverse real-world impacts, and fully complies with the NeurIPS ethical guidelines.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer:[NA]

Justification: There is no societal impact of the work performed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work does not release any new pretrained models, image generators, or datasets that pose a risk of misuse. We conducted the evaluation using publicly available datasets and backbones. The additional settings required in the experiment are all commonly used in current methods, and there was no misuse threat.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All the external assets used in this paper, including the ResNet10 backbone and all baselines, as well as the Fasion-MNIST, CIFAR-10, CIFAR-100, Digits and Office-Caltech datasets, have been correctly cited and accompanied by the corresponding references. Their licenses are respected and no unauthorized or crawled content has been used.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: None of the core methods, including any important, original, or non-standard components, rely on LLMs. We also do not use LLMs to generate data, experimental results, or similar

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Method Algorithm

Algorithm 1: FedPW

Input: Communication rounds T, local epochs $\mathcal{E}$, number of participants K, k^{th} participant private data D_k, private model w_k.
Output: The final global model w^T

Server: initialize the global model w^0
for $t = 0, 1, 2, ..., T-1$ **do**
  Client:
  for $k = 1, 2, ..., K$ **in parallel do**
    $w_k^t \leftarrow \mathcal{W}^t$
    for $e = 1, 2, ..., \mathcal{E}$ **do**
      $w_k^t \leftarrow w_k^t - \eta \nabla \mathbf{CE}(w_k^t, D_k)$
  $\Delta w_k^t \leftarrow w_k^t - \mathcal{W}^t$
  Server:
  $\Delta \mathcal{W}_k^t \leftarrow \mathbf{FedPW}(\Delta w^t)$
  for $i = 1, 2, \ldots, G$ **do**
    $\mathcal{W}_i^{t+1} = \mathcal{W}_i^t + \Delta \mathcal{W}_i^t$

FedPW(Δw^t) : **for** $k = 1, 2, \ldots, K$ **do**
  `/* Adaptive Weighting */`
  $p_k^t, \Delta p_k^t, \leftarrow (p_k^{t-1}, \Delta p_k^{t-1}, \beta)$ in Eq. (9)
  $q_k^t, \Delta q_k^t \leftarrow (q_k^{t-1}, \Delta q_k^{t-1}, \beta)$ in Eq. (10)
  $\lambda_k^t \leftarrow (p_k^t, p_k^t)$ in Eq. (11)
  `/* Parameter Adjustment */`
  for $i = 1, 2, \ldots, G$ **do**
    $\mathcal{M}_{k,i}^t \leftarrow$ Eq. (5)
    $\Delta \mathbf{w}_{k,i}^t = \Delta w_{k,i}^t \cdot \mathcal{M}_{k,i}^t$
    $\alpha_i^t \leftarrow (\Delta w_{k,i}^t)$ in Eq. (7)
    $\Delta \mathcal{W}_i^t = \alpha_i^t \sum_{k=1}^K \lambda_k \Delta \mathbf{w}_{k,i}^t$

return $\Delta \mathcal{W}_i^t$

Parameter Adjustment. This module refines client updates through a two-stage process. In the first stage, redundant parameters are pruned by computing client-specific mask rates r_k^t from inverse training losses and applying binary masks $\mathcal{M}_{k,i}^t$ in Eq. (5). Parameters below the adaptive threshold τ_k are removed, eliminating about $(1 - r_k^t) \times 100\%$ of the least significant parameters and reducing aggregation noise. In the second stage, consensus-based rescaling is applied using Eq. (7): masked updates are normalized, consistent parameters are identified via cross-client standard deviation, and emphasized by scaling factors α_i^t to preserve gradient magnitude while reinforcing aligned updates that support stable convergence.

Adaptive Weighting. This module balances generalization and fairness across training. In each round, generalization weights p_k^t are computed via gradient alignment (Eq. (9)), while fairness weights q_k^t are derived from training losses (Eq. (10)). The two components are then adaptively combined using the variance-based rule in Eq. (11), shifting from alignment-driven weighting in early rounds to fairness-oriented weighting later, guided by the relative variability of the two distributions.

B Details of Experiments

Datasets. Following [18, 21, 3, 43], we evaluate the efficacy of our method on single-domain datasets Fashion-Mnist, Cifar10, Cifar100, and cross-domain datasets Digits and Office-caltech.

- **Fasion-MNIST** [54] has 60k train and 10k test examples from 10 classes.
- **Cifar10** [28] contains $50k$, $10k$ images for training, validation. Each image is in 32×32 size from 10 different classes, *e.g.*, airplanes, cars, and birds.
- **Cifar100** [28] contains $50k$ and $10k$ images with 32×32 for 100 classes.
- **Digits** [29, 20, 41, 45]] includes four domains: MNIST(M), USPS (U), SVHN (SV) and SYN (SY) with 10 cat- 427 egories (digit number from 0 to 9).

- **Office-Caltech** [11] consists four domains: Caltech (C), Amazon (A), Webcam (W) and DSLR (D), which is formed of ten overlapping classes between Office31 [46] and Caltech-256 [12].

Hyper-parameter Study Our method involves only two hyperparameters: the mask ratio c and the momentum coefficient β. The mask ratio c governs the average masking level in the PA component, where smaller values make FedPW resemble FedAvg and reduce potential gains, while excessively large values may cause instability due to insufficient trainable parameters. The coefficient β controls the smoothness of adaptive updates in the AWA component; extremely small values degrade it to a non-momentum variant, while overly large values may delay the system's responsiveness to training dynamics. The following tables present comprehensive hyperparameter evaluation results.

Table 5: **AVG(STD) under varying** β across different datasets.

β	0.1	0.3	0.5	0.7	0.9
FMNIST	85.1(8.3)	87.7(7.2)	**88.2(7.1)**	87.2(7.8)	86.7(6.8)
CIFAR-10	74.5(10.9)	74.6(11.4)	**75.3(11.1)**	75.0(11.3)	74.9(11.6)
CIFAR-100	39.6(6.8)	41.2(7.0)	**41.2(6.6)**	39.8(6.3)	39.3(6.6)
Digits	78.8(21.7)	79.1(21.4)	**79.5(21.0)**	78.5(21.1)	77.9(21.4)
Office-Caltech	62.7(6.3)	61.9(4.8)	**63.2(4.6)**	62.7(5.9)	62.3(5.8)

Table 6: **AVG(STD) under varying** c across different datasets.

c	0.1	0.2	0.3	0.4	0.5
FMNIST	86.1(7.1)	87.6(7.4)	**88.2(7.1)**	86.3(6.9)	86.1(6.7)
CIFAR-10	74.6(10.9)	**75.3(10.9)**	75.3(11.1)	74.5(12.1)	74.1(11.3)
CIFAR-100	40.2(6.8)	41.0(6.6)	**41.2(6.6)**	39.3(6.9)	38.6(7.4)
Digits	78.8(21.3)	79.4(21.5)	**79.5(21.0)**	79.1(21.5)	78.4(22.1)
Office-Caltech	62.1(5.1)	62.3(5.1)	**63.2(4.6)**	60.1(3.8)	58.9(4.2)