FedPolicy: An RL-Guided Redistribution Policy for Synergizing Local-Global Optimization in Federated Learning

Anonymous authors
Paper under double-blind review

Abstract

Statistical heterogeneity remains a central challenge in federated learning. Existing methods primarily address this problem through improved local objectives, aggregation strategies, or personalization mechanisms, while the post-aggregation redistribution step is typically applied uniformly and receives little explicit treatment. This design becomes problematic under heterogeneous client distributions, where repeatedly overwriting local models with the same aggregated parameters can disrupt client-specific adaptation and induce negative transfer. We propose FedPolicy, an RL-guided post-aggregation redistribution framework that treats the return path of the aggregated model as a client-specific decision problem. Rather than broadcasting the same update to every client, FedPolicy learns which part of the aggregated model should be transferred back to each client by selecting among full-model, backbone-only, and head-only parameter blocks. This formulation identifies post-aggregation redistribution as a previously underexplored control axis in federated optimization, improving the balance between global transfer and local specialization. Extensive experiments under heterogeneous federated settings show that FedPolicy consistently outperforms strong baselines across FMNIST, CIFAR-10, and CIFAR-100, with the clearest gains appearing in the more challenging heterogeneous regimes. Across all datasets and heterogeneity settings, FedPolicy achieves an average relative gain of 3.93% over the strongest baseline, with the largest improvement reaching 8.40% on CIFAR-100 under severe heterogeneity, while converging faster and delivering a more favorable cost-to-accuracy trade-off with negligible overhead. These results highlight client-specific post-aggregation redistribution as an underexplored yet impactful design dimension in heterogeneous federated learning. Code for reproducing the results is available at `https://anonymous.4open.science/r/A-FedPolicy-816B`.

1 Introduction

Federated learning (FL) systems commonly assume that once a global model is obtained through aggregation, it can be applied *uniformly* to all participating clients without adverse effects. However, in practice, clients often differ substantially in data distribution, representation maturity, and class-level learning difficulty. Under such heterogeneity, directly overwriting local models with aggregated parameters uniformly can induce negative transfer, destabilize training dynamics, and degrade the effectiveness of subsequent aggregation rounds. Despite substantial progress in federated learning, the question of *how aggregated parameters should be applied to heterogeneous clients* remains underexplored.

Intuitively, clients that have already learned stable and specialized representations may suffer representation erosion when aggressively aligned with the global model, whereas clients struggling with specific classes may propagate noisy or biased updates if forced to fully absorb global parameters. The effect is not limited to a single round. Post-aggregation initialization determines which local trajectory each client follows before returning updates to the server. If initialization is mismatched, the returned updates may become noisier or less aligned with local data geometry, which then degrades the next aggregate and propagates instability across rounds. By contrast, client-specific redistribution can preserve useful local structure while still exploiting transferable global progress. Addressing this mismatch becomes crucial, especially under severe non-independent and identically distributed (non-IID) data distributions, as client updates directly influence

the quality and stability of future aggregation, yet existing FL pipelines treat this step as a fixed and uniform operation.

In the standard federated learning framework, clients perform local optimization, the server aggregates their updates, and the resulting global model is redistributed to all clients for the next round McMahan et al. (2017). Most prior federated learning methods address heterogeneity through two main design axes: *local objective design* and *server-side aggregation design*. Regularization-based approaches such as FedProx Li et al. (2020) and control-variate methods such as SCAFFOLD Karimireddy et al. (2020) constrain local optimization to mitigate client drift. Adaptive aggregation strategies Reddi et al. (2021); Wang et al. (2020b); Chen et al. (2023) seek to combine heterogeneous updates more robustly at the server, while client-adaptive methods Arivazhagan et al. (2019); Wang et al. (2024b); Grinwald et al. (2024) and prototype- or representation-learning approaches Tan et al. (2022); Collins et al. (2021); Yang et al. (2024) further tailor learned representations to client-specific distributions. More recently, our prior work FedCA Chowdhury & Halder (2026) extended this direction by jointly addressing heterogeneity through confusion-aware local optimization and class-prioritized aggregation under extreme non-IID settings. Readers may refer to Appendix A for more details.

However, a common assumption still remains unchanged across these lines of work: once the global model is formed, it is redistributed back to all participating clients in the same form. Although prior work has substantially improved both *how clients are trained locally* and *how their updates are aggregated globally*, it still leaves underexplored *how the aggregated global model should be transferred back to heterogeneous clients*. This gap is especially consequential under strong heterogeneity, where the same global update need not be equally beneficial to all clients. A client may benefit from global representation transfer while needing to preserve its local classifier, whereas another may require boundary correction without disturbing already adapted features. Treating redistribution as uniformly beneficial therefore introduces a mismatch between the aggregated model and the client's local learning state, which can degrade both local adaptation and the quality of subsequent aggregation rounds.

In this work, we argue that effective federated learning under heterogeneity requires not only better local objectives or better aggregation operators, but also explicit control over *how the aggregated global model is transferred back to individual clients*. This post-aggregation step is inherently sequential and non-stationary: each initialization decision changes a client's local training trajectory, and the resulting local updates affect the aggregate model used in the subsequent round. Moreover, because the participating clients, class composition, and model states vary across rounds, the learning environment is itself non-stationary. As a result, fixed heuristic rules may not remain effective throughout training. These characteristics naturally motivate a reinforcement learning (RL) formulation.

To this end, we propose FedPolicy, an RL policy-guided federated learning framework that treats post-aggregation redistribution as an adaptive, client-specific decision problem, while leaving local optimization and server-side aggregation unchanged. After the server forms the global model at a communication round, it does not simply broadcast the same model to every selected client. Instead, it observes a client-specific state and decides whether that client should receive the full global model, the backbone only, or the classifier head only. A key motivation for this design is the functional asymmetry of modern classifiers: backbone parameters primarily encode transferable representations, whereas head parameters are more sensitive to client-specific class imbalance and coverage Li et al. (2023). The resulting selective parameter sharing mechanism balances global knowledge transfer with preservation of local specialization.

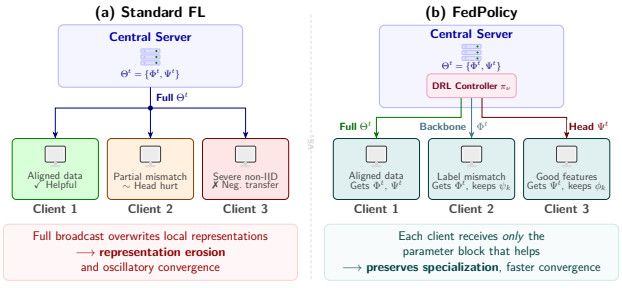

Figure 1: **Selective parameter broadcast under heterogeneity.** **(a)** Uniform full-model broadcast can induce representation drift and negative transfer when client data are non-IID. **(b)** FedPolicy uses a DRL policy to choose a client-specific broadcast action among full model (Θ^t), backbone-only (Φ^t), and head-only (Ψ^t). This reflects the backbone/head functional separation and selectively preserves beneficial local structure while leveraging global progress.

To support this decision process, we introduce a **semantic error topology** representation, formed as an exponential moving average of each client's *soft confusion matrix* to capture persistent class-level confusability patterns. We further augment this representation with optimization-status signals, including validation performance gaps and parameter divergence, yielding a compact state descriptor for the server-side controller. Based on this state, FedPolicy trains a **Deep Reinforcement Learning (DRL)** agent that learns a client-specific selective parameter sharing policy by maximizing a reward that balances client-level improvement and global training stability. This selective-initialization perspective is illustrated in Fig. 1.

To summarize, the contributions of this paper are as follows:

- We propose FedPolicy, a **Policy-Guided Selective Global Redistribution** framework that formulates client-specific post-aggregation redistribution as a sequential decision-making problem via a DRL policy, enabling adaptive control over which model components are transmitted to each client.
- We design a composite client state representation that captures **semantic error topology** via soft confusion matrices and **optimization status** via parameter divergence and validation performance gaps, providing a rich and compact signal for heterogeneity-aware server-side control.
- We develop a DRL agent that adaptively selects client-specific parameter sharing actions (Full, Backbone, Head), improving convergence stability and the quality of subsequent aggregation rounds.
- We conduct extensive experiments on FMNIST, CIFAR-10, and CIFAR-100 under diverse levels of data heterogeneity, showing that FedPolicy consistently improves accuracy, convergence, and communication efficiency over strong federated baselines, with gains of up to 8.40% on CIFAR-100 under severe heterogeneity.

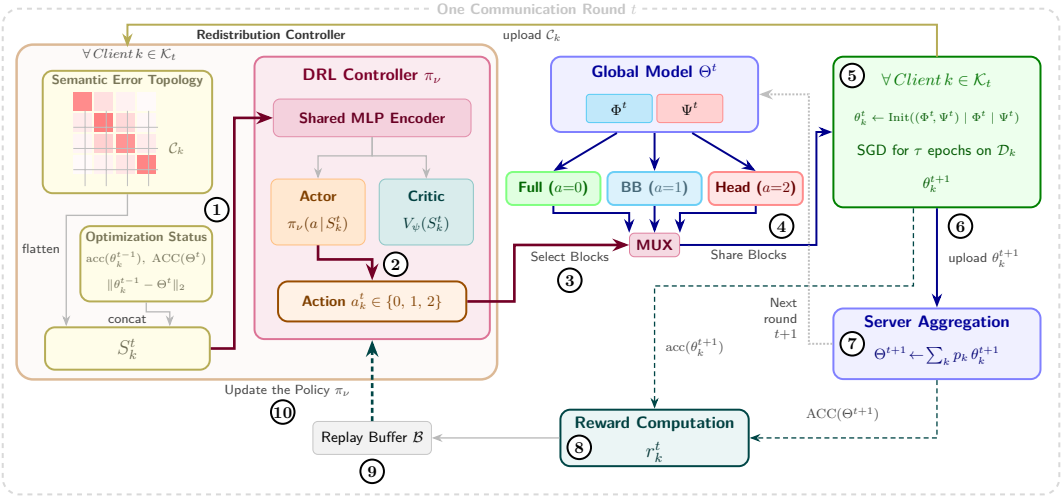

Figure 2: Overview of the FedPolicy training workflow.

2 Methodology

This section presents FedPolicy, a policy-guided federated learning framework for client-specific post-aggregation parameter sharing under statistical heterogeneity. Figure 2 illustrates the workflow of a training round. The redistribution controller is the key component of FedPolicy, which allows the server to construct a client-specific state representation for each selected client and obtain a sharing action, that is, whether to share the full model, the backbone only, or the head only (Steps 1–2). Based on this action, the server determines and transmits the corresponding parameter block (Steps 3–4). Following this, the client performs local training by incorporating the received block into its local model, and returns the updated model together with the confusion statistics required for policy evaluation (Steps 5–6). Notably, FedPolicy does not introduce adaptivity at the aggregation stage. The server forms the global model using the standard mean

aggregation rule, and measures global progress on a small server-held-out validation split, which is then combined with client-side validation improvement to assess the effect of the selected redistribution actions and generate the reward signal for training the redistribution controller (Steps 7–10). In this way, FedPolicy concentrates adaptivity in the post-aggregation parameter redistribution stage while retaining standard local optimization and server aggregation. Table 1 summarizes the key mathematical symbols and notation used throughout the paper.

2.1 Problem Formulation

Table 1: Key notations and symbols.

Symbol	Description
$\mathcal{A}$	Action space of the controller
a_k^t	Action selected for client k at round t
β	EMA momentum for soft confusion tracking
C_k	Soft confusion matrix of client k
$\mathcal{B}$	Replay buffer for DRL training
$\mathcal{D}_k, \mathcal{D}_k^{\text{val}}$	Local training and validation datasets of client k
$\text{acc}(\cdot)$	Client-side validation accuracy
$\text{ACC}(\cdot)$	Server-side held-out validation accuracy
ΔAcc_k^t	Local validation improvement of client k
$\text{vec}(\cdot)$	Vectorization operator
η	Local learning rate
γ	DRL discount factor
$\mathcal{J}(\Theta)$	Global objective
$\mathcal{K}_t$	Set of participating clients at round t
λ_{glob}	Weight of the global reward term
$\mathcal{L}_k$	Empirical risk on client k
Φ	Backbone (representation encoder) parameters
π_ν	DRL policy with parameters ν
p_k	Aggregation weight for client k
r_k^t	Reward of client k at round t
$\mathcal{S}$	State space
S_k^t	State of client k at round t
τ	Number of local training epochs
$\Theta^t = \{\Phi^t, \Psi^t\}$	Global model at round t
$\theta_k^t = \{\phi_k^t, \psi_k^t\}$	Local model of client k at round t
Ψ	Classifier head parameters

We decompose the global model parameters Θ into two functional subspaces: the representation encoder (backbone) Φ and the classifier head Ψ, i.e., $\Theta = \{\Phi, \Psi\}$. This decomposition reflects the observation that statistical heterogeneity affects representation learning and decision boundaries asymmetrically. The backbone primarily captures transferable features, whereas the classifier head is more sensitive to local label composition.

The federated optimization process proceeds in communication rounds $t = 1, \ldots, T$. At round t, the server maintains the aggregated model $\Theta^t = \{\Phi^t, \Psi^t\}$ and selects a subset of clients $\mathcal{K}_t$ for participation. For each selected client $k \in \mathcal{K}_t$, the server chooses an action a_k^t that determines which parameter block of Θ^t should be shared with that client. We denote the transmitted block by

$$\zeta_k^t = \text{Share}(\Theta^t, a_k^t), \tag{1}$$

where ζ_k^t may correspond to the *full model*, the *backbone* only, or the *head* only. The client then combines the received block with its previous local model θ_k^{t-1} to form the effective model θ_k^t used for local training.

Given this selectively initialized model, client k performs local optimization on its private dataset $\mathcal{D}_k$ for τ epochs to obtain an updated model θ_k^{t+1}:

$$\theta_k^{t+1} \leftarrow \theta_k^t - \eta \nabla \mathcal{L}_k(\theta_k^t), \tag{2}$$

where η is the local learning rate and $\mathcal{L}_k$ denotes the empirical risk on client k.

After receiving the updated local models from the participating clients, the server forms the next global model through standard aggregation:

$$\Theta^{t+1} \leftarrow \sum_{k \in \mathcal{K}_t} p_k \theta_k^{t+1}, \tag{3}$$

where client's contribution weight $p_k = 1/|\mathcal{K}_t|$ for all $k \in \mathcal{K}_t$, for mean (FedAvg-style) aggregation.

Our objective is to learn a policy π_ν that selects actions a_k^t so as to improve both client-level adaptation and the quality of future global aggregation over the communication horizon. Formally, we seek

$$\min_{\pi_\nu} \mathbb{E}\big[\mathcal{J}(\Theta^T)\big] = \min_{\pi_\nu} \mathbb{E}\left[\frac{1}{N}\sum_{k=1}^{N} \mathcal{L}_k(\Theta^T)\right], \tag{4}$$

where N denotes the total number of clients in the federation. This formulation elevates post-aggregation parameter sharing from a fixed implementation step to a sequential control problem in federated learning.

2.2 Fixed Local Training and Aggregation Pipeline

FedPolicy preserves the standard client–server optimization loop: participating clients perform local empirical risk minimization, and the server aggregates the resulting updates into the next global model. The intervention of the proposed method occurs only after this aggregation step, when the aggregated model is redistributed to clients for the next round of local training. For a selected client k, the local objective is,

$$\mathcal{L}_k(\theta_k^t) = \frac{1}{|\mathcal{D}_k|} \sum_{(x,y)\in\mathcal{D}_k} \ell\Big(f_{\theta_k^t}(x), y\Big), \tag{5}$$

where $\ell(\cdot,\cdot)$ denotes the task loss and $f_{\theta_k^t}$ is the local model obtained after selective parameter sharing. After τ local epochs, each client returns the updated model θ_k^{t+1}, the corresponding soft confusion statistics C_k, and the validation accuracy signal $\mathrm{acc}(\theta_k^{t+1})$ used for policy learning.

The global model is then updated using the standard mean aggregation rule in Eq. (3). This design choice is deliberate as prior heterogeneity-aware studies including FedCA, improved robustness by adapting local objectives and aggregation-side coordination. In contrast, FedPolicy targets a different unresolved stage of the FL pipeline: once an aggregated model has already been formed, how should it be redistributed back to heterogeneous clients? By keeping aggregation fixed, the method isolates redistribution as the control point and attributes gains to client-specific parameter sharing rather than aggregation complexity.

2.3 RL-Guided Post-Aggregation Redistribution Controller

We cast post-aggregation redistribution as a sequential decision problem executed at the server. At communication round t, once the aggregated model Θ^t has been obtained, the server must decide, for each selected client $k \in \mathcal{K}_t$, which parameter block of Θ^t should be shared with that client, denoted as $a_k^t \in \mathcal{A}$. The action a_k^t determines the shared block ζ_k^t and, through client-side initialization, the effective local model θ_k^t used for the next local optimization phase. However, a central challenge is that the quality of a_k^t cannot be reliably judged from immediate observations alone. The selected sharing action influences both the client's short-term local training trajectory and the quality of the returned update θ_k^{t+1}, which in turn shapes the subsequent aggregate Θ^{t+1}. The effect of each decision therefore propagates through coupled client-level and federation-level dynamics, making post-aggregation redistribution a sequential control problem rather than a static assignment rule. Accordingly, we model the controller as a DRL policy

$$\pi_\nu(a_k^t \mid S_k^t), \tag{6}$$

where S_k^t is a client-specific state representation and ν denotes the controller parameters. For each selected client, the server observes S_k^t, selects an action a_k^t, transmits the corresponding block ζ_k^t, and receives feedback only after local optimization and subsequent aggregation. The resulting interaction induces a finite-horizon Markov decision process with transitions of the form $(S_k^t, a_k^t, r_k^t, S_k^{t+1})$, from which the controller is trained.

The objective of the controller is to maximize the expected long-term utility of client-specific redistribution over the communication horizon:

$$\max_{\nu}\ \mathbb{E}_{\pi_\nu}\left[\sum_{t=0}^{T-1} \gamma^t r_k^t\right], \tag{7}$$

where $\gamma \in [0,1]$ is the discount factor.

2.4 State Representation: Semantic and Optimization Signals

The controller requires a client-specific state S_k^t that captures both the nature of the client's learning difficulty and its relationship to the current global model. In FedPolicy, the state is designed to encode two complementary aspects of heterogeneity: semantic error structure and optimization status.

2.4.1 Semantic Error Structure

A scalar metric such as total loss or top-1 accuracy cannot reveal which classes a client consistently confuses. Yet such class-level structure is important for determining whether the client should preserve its local classifier, adopt the global classifier, or absorb the full global model. To represent this information compactly, we use a soft confusion matrix $C_k \in \mathbb{R}^{C\times C}$, where C is the number of classes and each entry reflects the tendency of the current local model to predict class j for a sample whose true label is class i. The flattened representation $\mathrm{vec}(C_k)$ therefore serves as a compact summary of the client's class-level confusion structure.

2.4.2 Optimization Status

In addition to semantic confusion, the controller must assess how the client's local model relates to the current global model. For this purpose, we include two optimization-status signals.

First, we use the client-side validation accuracy $\mathrm{acc}(\theta_k^{t-1})$ from the previous round together with the current server-side held-out validation accuracy $\mathrm{ACC}(\Theta^t)$. These quantities indicate whether the incoming global model is likely to be beneficial relative to the client's current local state.

Second, we measure the parameter-space mismatch between the local and global models as $d_k^t = \left\|\theta_k^{t-1} - \Theta^t\right\|_2$. A large mismatch suggests that the client has evolved along a trajectory substantially different from the global consensus, indicating that selective redistribution may be preferable to a full-model overwrite.

Combining both semantic and optimization signals, the state vector for client k at round t is defined as

$$S_k^t = \Big[\mathrm{vec}(C_k) \oplus \mathrm{acc}(\theta_k^{t-1}) \oplus \mathrm{ACC}(\Theta^t) \oplus d_k^t\Big], \tag{8}$$

where $\oplus$ denotes vector concatenation.

2.5 Parameter Block Selection

Given the client state S_k^t, the controller selects an action $a_k^t \in \mathcal{A} = \{0, 1, 2\}$, which specifies the parameter-sharing rule and the transmitted global block ζ_k^t, thereby determining the effective local model θ_k^t.

Concretely, (i) for $a_k^t = 0$, the server broadcasts the full aggregated model $\zeta_k^t = \{\Phi^t, \Psi^t\}$, yielding $\theta_k^t = \{\Phi^t, \Psi^t\}$ and recovering standard federated learning; this is suitable when the client is well aligned with the global model. (ii) For $a_k^t = 1$, only the global backbone is transmitted, $\zeta_k^t = \Phi^t$, while the client retains its local head, resulting in $\theta_k^t = \{\Phi^t, \psi_k^{t-1}\}$; this preserves client-specific decision boundaries while sharing feature representations. (iii) For $a_k^t = 2$, only the global head is transmitted, $\zeta_k^t = \Psi^t$, yielding $\theta_k^t = \{\phi_k^{t-1}, \Psi^t\}$; this preserves locally adapted features while aligning the classifier with the global decision structure.

Algorithm 1: Federated Learning with Policy-Guided Selective Parameter Sharing (FedPolicy)

Require: Global model Θ^t, previous client models $\{\theta_k^{t-1}\}$, controller policy π_ν, replay buffer $\mathcal{B}$
Ensure: Next global model Θ^{t+1}, updated controller parameters ν
// Phase 1: State construction and policy-guided sharing
for each selected client $k \in \mathcal{K}_t$ **in parallel do**
  Construct state S_k^t using θ_k^{t-1} and Θ^t (Eq. 8)
  Select action $a_k^t \sim \pi_\nu(\cdot \mid S_k^t)$
  Determine shared block $\zeta_k^t \leftarrow \mathrm{Share}(\Theta^t, a_k^t)$
  Initialize θ_k^t from θ_k^{t-1} using ζ_k^t
end for
// Phase 2: Local training
for each selected client $k \in \mathcal{K}_t$ **in parallel do**
  Update θ_k^{t+1} via local training on $\mathcal{D}_k$
  Observe reward r_k^t and next state S_k^{t+1}
  Store transition $(S_k^t, a_k^t, r_k^t, S_k^{t+1})$ in $\mathcal{B}$
end for
// Phase 3: Server aggregation
$\Theta^{t+1} \leftarrow \sum_{k\in\mathcal{K}_t} p_k \theta_k^{t+1}$, with $p_k = 1/|\mathcal{K}_t|$
// Phase 4: Controller update
if update interval reached **then**
  Sample a minibatch from $\mathcal{B}$
  Update controller parameters ν using the chosen DRL objective
end if

2.6 Policy Learning and Reward Design

We define the controller problem as a finite-horizon Markov decision process $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space defined by Eq. (8), $\mathcal{A}$ is the action space of parameter-block broadcast, $\mathcal{P}$ captures the stochastic transition dynamics induced by local training and server-side aggregation, $\mathcal{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor.

To guide learning, we define a reward that combines client-level improvement with federation-level progress:

$$r_k^t = \Delta \mathrm{Acc}_k^t + \lambda_{\mathrm{glob}} \Big(\mathrm{ACC}(\Theta^{t+1}) - \mathrm{ACC}(\Theta^t) \Big), \tag{9}$$

where $\Delta \mathrm{Acc}_k^t = \mathrm{acc}(\theta_k^{t+1}) - \mathrm{acc}(\theta_k^{t-1})$, and λ_{glob} controls the contribution of the federation-level progress term measured on the server-held-out validation split. The first term rewards local improvement induced by the selected broadcast action, while the second term provides a shared federation-level signal that reflects whether the resulting round contributes positively to overall validation progress. The test set is reserved strictly for final reporting and is never used for controller training, reward computation, state construction, or policy updates.

The controller is trained to maximize the expected discounted return in Eq. (7). Since the DRL module operates entirely at the server, clients do not perform any policy-learning computation locally; they only receive the selected parameter block and then follow the standard federated training routine.

Overall, FedPolicy provides an end-to-end, policy-guided mechanism for client-specific post-aggregation parameter transfer, enabling the federation to balance global knowledge sharing with preservation of locally specialized structure under heterogeneous data distributions. Algorithm 1 summarizes the complete training workflow. For each selected client, the server constructs a client state, selects a broadcast action, determines the corresponding parameter block, and forms the effective local model. The selected clients then perform local training and return their updated models together with the statistics required for policy evaluation. The server computes reward and next-state information, stores the resulting transition in the replay buffer, aggregates client models using the fixed mean/FedAvg-style rule, and updates the controller from replayed transitions. This cycle is repeated at each communication round until convergence or until the communication budget is exhausted.

3 Theoretical Analysis

We provide a structural optimization analysis of client-specific selective redistribution versus uniform overwrite under standard smoothness and heterogeneity assumptions. The analysis does not constitute a full convergence theory of the learned RL controller; instead, it abstracts away policy-learning dynamics and characterizes how selective initialization can improve optimization behavior once a redistribution action is chosen. Full derivations are deferred to Appendix B.

Assumption 1 (Smoothness)**.** *For each client k, the local objective $\mathcal{L}_k$ is L-smooth.*

Assumption 2 (Stochastic Gradient Variance)**.** *Client stochastic gradients are unbiased and have bounded variance: $\mathbb{E}\|g_k(w) - \nabla \mathcal{L}_k(w)\|^2 \le \sigma^2$.*

Assumption 3 (Gradient Dissimilarity)**.** *Client gradients are heterogeneity-bounded: $\sum_k p_k \|\nabla \mathcal{L}_k(w) - \nabla f(w)\|^2 \le B^2$, where $f(w) = \sum_k p_k \mathcal{L}_k(w)$.*

Assumption 4 (Block-wise Strong Convexity)**.** *In a neighborhood of candidate updates, $\mathcal{L}_k([\phi; \psi])$ is μ-strongly convex in each block when the other block is fixed.*

Assumption 5 (Drift-Bounded Block Gradient)**.** *For the selected client state, the relevant block gradient norm is bounded by $\epsilon_{\mathrm{drift}}$.*

Lemma 1 (Initialization Risk Bound)**.** *Let $\tilde{w}_k^t$ denote the selectively initialized client model and w^t the aggregated model at round t. Then $\mathcal{L}_k(\tilde{w}_k^t) - \mathcal{L}_k(w^t) \le \langle \nabla \mathcal{L}_k(w^t), \delta_k^t \rangle + \frac{L}{2} \|\delta_k^t\|^2$, where $\delta_k^t := \tilde{w}_k^t - w^t$.*

Corollary 1 (Strict Advantage Condition)**.** *If $\langle \nabla \mathcal{L}_k(w^t), \delta_k^t \rangle + \frac{L}{2} \|\delta_k^t\|^2 < 0$, then $\mathcal{L}_k(\tilde{w}_k^t) < \mathcal{L}_k(w^t)$.*

Theorem 1 (Policy Advantage over Uniform Overwrite)**.** *Let $\mathbf{w}_{\mathit{Best}}^t$ denote the candidate among $\mathbf{w}_{\mathit{Full}}^t$, $\mathbf{w}_{\mathit{Back}}^t$, and $\mathbf{w}_{\mathit{Head}}^t$ with the smallest client loss. Under Assumptions 4 and 5, the best candidate satisfies a lower bound on the loss reduction relative to uniform Full overwrite, with the bound scaling with block mismatch magnitude and decreasing with the drift penalty.*

Theorem 2 (Global Convergence under Selective Parameter Sharing)**.** *Under Assumptions 1, 2, and 3, with learning rate $\eta \le 1/(LE)$ and initialization drift Γ^2 induced by selective parameter redistribution, the*

averaged stationarity metric satisfies

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(w^t)\|^2 \leq \frac{2(\mathbb{E}[f(w^0)]-f^\star)}{\eta ET} + 2L\eta\sigma^2 + 2L\eta EB^2 + \frac{L}{E}\Gamma^2,$$

Theorem 3 (Convergence Improvement via Drift Reduction)**.** *If selective redistribution yields* $\Gamma^2_{\mathit{Pol}} \leq \rho\,\Gamma^2_{\mathit{Full}}$ *for some* $\rho < 1$*, then the convergence upper bound in Theorem 2 is strictly tighter than that of uniform Full overwrite.*

4 Experimental Evaluation

We evaluate FedPolicy under heterogeneous client distributions generated by both probabilistic label-skew and deterministic class-absence partitions. The study is designed to answer four questions: (i) whether client-specific post-aggregation redistribution improves accuracy under heterogeneity, (ii) whether it accelerates convergence, (iii) whether the gain is attributable to a learned redistribution policy rather than to a fixed heuristic or only to the local loss, and (iv) whether the resulting improvement is obtained with practical computational and communication cost. During evaluation, baseline methods are kept in their original algorithmic form, including their defining optimization or coordination mechanisms and their tuned settings, unless explicitly stated otherwise. By contrast, FedPolicy uses simple mean (FedAvg-style) aggregation and introduces adaptivity only through client-specific post-aggregation redistribution. This design isolates redistribution as the sole adaptive component of the proposed method, ensuring that the observed gains are not attributable to any additional aggregation-side advantage, but to the client-specific adaptive model redistribution mechanism itself. Comprehensive experimental details, including implementation settings, training configurations, and data partitioning procedures, are provided in Appendix C.

Table 2: Top-1 accuracy (%) on FMNIST, CIFAR-10, and CIFAR-100 under Dirichlet label-skew with concentration β. Lower β indicates stronger heterogeneity. Best per column in **bold**.

Method	FMNIST			CIFAR-10			CIFAR-100		
	β=0.3	β=0.1	β=0.05	β=0.3	β=0.1	β=0.05	β=0.3	β=0.1	β=0.05
FedAvg	82.51±0.38	78.75±0.52	73.11±0.91	64.76±0.61	31.25±1.14	16.11±1.52	49.83±0.67	43.92±0.95	37.22±1.38
FedProx	82.65±0.35	78.89±0.49	73.72±0.87	64.82±0.58	33.79±1.09	16.31±1.47	49.87±0.64	43.97±0.91	37.23±1.35
FedNTD	83.95±0.32	80.04±0.51	74.23±0.89	69.20±0.62	44.08±1.15	21.52±1.54	50.19±0.58	44.93±1.17	38.34±1.23
Ditto	83.13±0.41	78.15±0.59	73.23±0.96	65.00±0.67	31.25±1.21	17.18±1.61	50.00±0.68	37.64±1.08	28.96±1.52
CCVR	83.63±0.39	78.97±0.54	71.78±0.99	67.31±0.59	38.57±1.12	15.78±1.64	48.95±0.73	42.68±0.98	36.41±1.41
FedSAM	82.06±0.45	78.87±0.61	70.83±1.03	64.13±0.79	27.25±1.28	14.66±1.71	46.98±0.78	38.67±1.15	32.15±1.58
FedFA	83.31±0.43	77.48±0.63	74.59±0.92	64.51±0.72	28.31±1.25	17.35±1.55	49.27±0.75	37.78±1.11	31.74±1.53
FedCA	84.64±0.56	81.24±0.47	78.89±0.84	73.46±0.52	59.01±1.48	51.62±1.76	50.65±0.78	45.62±0.98	39.41±1.29
FedPolicy	**85.84±0.61**	**82.68±0.57**	**79.94±0.89**	**74.28±0.52**	**62.84±1.13**	**55.25±1.38**	**51.83±0.74**	**48.12±1.13**	**42.72±1.24**

4.1 Performance under Heterogeneous Data

We first evaluate FedPolicy on two primary axes: best top-1 accuracy under heterogeneous partitions and convergence speed toward useful operating points. We then analyze whether the observed gains are consistent with the proposed redistribution mechanism.

4.1.1 Accuracy under Label Heterogeneity

Table 2 compares FedPolicy against eight baselines on FMNIST, CIFAR-10, and CIFAR-100 under Dirichlet heterogeneity with $\beta \in \{0.3, 0.1, 0.05\}$. These benchmarks span different class counts (10 vs. 100), image modalities (grayscale vs. RGB), and

Table 3: CIFAR-10 local-loss attribution. CE versus C3E accuracy (%) across baselines and FedPolicy under Dirichlet heterogeneity. Best result in each column is shown in **bold**.

	β=0.3		β=0.1		β=0.05	
Method	CE	C3E	CE	C3E	CE	C3E
FedAvg	64.76	71.30	31.25	49.27	16.11	39.19
FedProx	64.82	72.10	33.79	44.20	16.31	40.12
Ditto	65.00	72.40	31.25	43.10	17.18	40.50
CCVR	67.31	73.78	38.57	51.80	15.78	48.30
FedSAM	54.13	61.80	27.25	38.70	14.66	46.10
FedNTD	69.20	72.98	44.08	53.77	21.52	32.54
FedFA	63.51	71.80	28.31	41.60	17.35	46.83
FedCA	69.28	73.46	40.68	59.01	23.83	51.62
FedPolicy	**71.41**	**74.28**	**48.67**	**62.84**	**28.43**	**55.25**

task difficulties, thereby providing a broad assessment of robustness under heterogeneous client distributions. Across the three datasets, FedPolicy achieves the best result in every setting. The advantage is modest when heterogeneity is mild and becomes more pronounced on the more challenging benchmarks and under stronger heterogeneity settings. On CIFAR-10, for example, the improvement over the strongest baseline increases from +1.12% at β=0.3 to +7.03% at β=0.05. A similar trend appears on CIFAR-100, where the margin increases from +2.33% to +8.40% as β decreases. On FMNIST, the gains are smaller in relative terms but remain consistent across all settings, ranging from +1.42% to +1.77%. This pattern indicates that, under stronger skew, uniform redistribution of the global model becomes increasingly mismatched to local learning states, and the benefit of client-specific redistribution becomes larger.

Furthermore, Table 3 indicates that the gain of FedPolicy is not solely attributable to stronger local optimization. When relative improvement is computed against the best baseline at each heterogeneity level and then averaged across $\beta \in \{0.3, 0.1, 0.05\}$, FedPolicy achieves 10.93% under CE and 4.73% under C3E. Concretely, the CE improvements are +3.07%, +10.42%, and +19.30%, while the C3E improvements are +0.68%, +6.49%, and +7.03%. These results show that although stronger local optimization contributes to performance, it does not account for the full gain; the learned selective parameter-sharing policy provides an additional and consistent improvement beyond that effect.

Overall, Table 2 and Table 3 support the central empirical claim of the paper: even without a specialized aggregation rule, adaptive client-specific post-aggregation parameter sharing yields consistent gains over strong heterogeneous FL baselines, with larger benefits in regimes where uniform rebroadcast is least reliable. The improvements also cannot be explained by local-loss optimization alone, indicating a distinct contribution from the learned selective sharing policy.

4.1.2 Convergence Behavior

Table 4 reports the first communication round at which each method reaches predefined accuracy thresholds on CIFAR-10 (β=0.3), while Figure 3 shows full learning curves under the harder β=0.1 setting. Together, these results evaluate convergence speed rather than only final accuracy.

FedPolicy reaches useful operating points earlier than all baselines. In Table 4, it reaches 40% at round 3 (vs. round 5 for FedCA and round 6 for CCVR) and 60% at round 11 (vs. round 16 for FedCA and round 29 for FedNTD). At the highest threshold, both FedPolicy and FedCA exceed 70%, but FedPolicy does so substantially earlier (round 31 vs. round 46). These margins show that the gain is not limited to endpoint performance; it appears throughout training, especially in the early and mid stages.

Table 4: Convergence on CIFAR-10 (β=0.3): first round reaching each accuracy threshold (lower is better). "×" indicates the threshold was not reached within 105 rounds.

Method	≥ 20%	≥ 40%	≥ 60%	≥ 70%
FedAvg	1	16	63	×
FedProx	1	14	57	×
Ditto	1	17	49	×
CCVR	2	6	33	×
FedSAM	1	27	61	×
FedFA	1	8	72	×
FedNTD	2	16	29	×
FedCA	1	5	16	46
FedPolicy	**1**	**3**	**11**	**31**

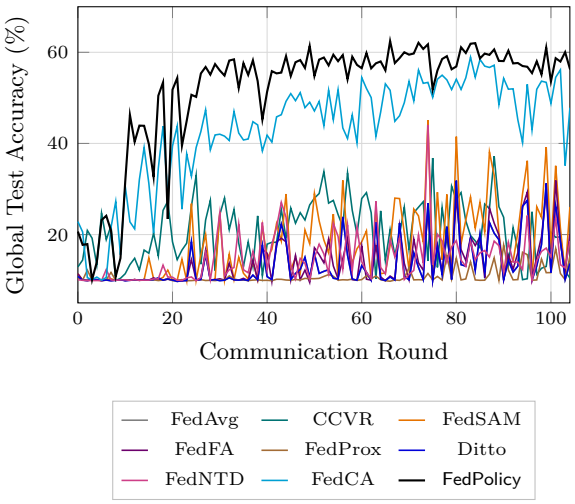

Figure 3: Global test accuracy over communication rounds on CIFAR-10 under moderate heterogeneity (β=0.1). FedPolicy consistently achieves higher accuracy and faster convergence compared to all baselines.

Figure 3 confirms the same pattern under stronger heterogeneity (β=0.1): FedPolicy separates from FedCA within roughly 15–20 rounds and maintains a consistent gap thereafter. Thus, the improvement is trajectory-level, not a late-round fluctuation. Overall, these convergence results strengthen the main claim that client-specific post-aggregation sharing improves both final accuracy and training efficiency under heterogeneous data.

4.2 DRL Policy Behavior under Data Heterogeneity

Figures 4 and 5 show that the learned redistribution policy varies substantially across DRL controllers, with the differences becoming clearer under stronger heterogeneity. Under severe heterogeneity ($\beta = 0.05$), SAC and PPO are both strongly Backbone-dominant, allocating 83.0% and 93.0% of actions to Backbone sharing, respectively, whereas DQN remains close to a uniform allocation (34% Full, 35% Backbone, 32% Head). At $\beta = 0.1$, the separation is more pronounced: SAC follows a structured mixed policy (51% Full, 40% Backbone, 9% Head), PPO shifts to a Head-heavy pattern (65.0% Head), and DQN again remains near-uniform (35%/34%/31%). Under milder heterogeneity ($\beta = 0.3$), SAC still places the largest share on Backbone updates (52%) ues to show little specialization.

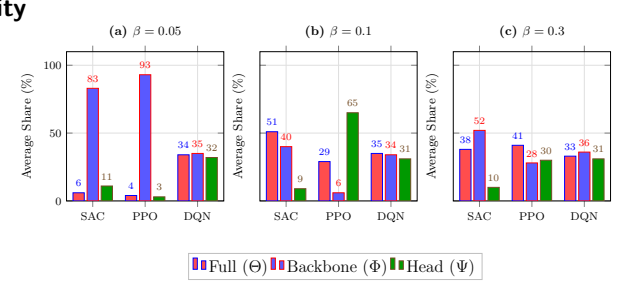

Figure 4: Average action proportions for three DRL controllers across three Dirichlet heterogeneity levels.

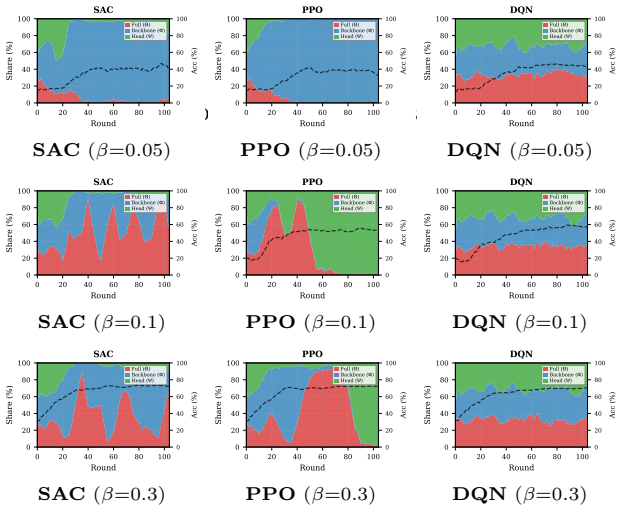

Figure 5: Policy evolution for SAC, PPO, DQN across three Dirichlet levels.

The temporal trajectories in Figure 5 are consistent with the same pattern: SAC converges to a clearer redistribution preference over training, PPO varies more across regimes, and DQN remains comparatively diffuse. A broader controller comparison and additional policy diagnostics are provided in Appendix C.2. Taken together, the evidence suggests that the benefit of FedPolicy depends not merely on using a DRL controller, but on whether that controller can learn a heterogeneity-aware and stable redistribution policy.

4.3 Ablation Studies

We next examine which components of FedPolicy are responsible for the observed gains. The ablations address four questions: which state signals are most informative for the controller, how the reward should be defined, whether the benefit comes from a learned policy rather than a fixed sharing rule, and whether the method remains effective under different local loss functions.

4.3.1 Learned versus Static Sharing Strategies

Table 5 compares five DRL controllers with four static parameter-sharing heuristics on CIFAR-10 under deterministic fixed-class splits and probabilistic Dirichlet label skew. The goal is to test whether gains come from learning the policy itself, rather than from any fixed sharing rule.

Under fixed-class partitions, learned controllers are consistently competitive and become clearly superior as class support narrows. At `fc`=6 and `fc`=4, TRPO reaches 78.78% and 75.41%, respectively, exceeding the best static heuristics (77.12% and 73.78%). At the hardest setting (`fc`=2), DQN attains 55.35%, outperforming the best static baseline (Random, 52.81%) by +2.54 points. Head-Only remains weak (30.48%), indicating that classifier-only transfer is insufficient under severe class scarcity.

The same trend appears more strongly under Dirichlet heterogeneity. At β=0.3, the best DRL result is slightly below the best static heuristic (74.28% vs. 74.75%, −0.47). As heterogeneity increases, the benefit of learned policies becomes large: at β=0.1, TRPO reaches 64.32% versus 58.99% for the best static rule (+5.33), and at β=0.05, SAC reaches 54.38% versus 42.42% (+11.96). No single static heuristic is consistently best across settings, whereas learned policies adapt with regime difficulty. Notably, all five DRL methods outperform the best static rule at β=0.05, indicating that the gain is due to adaptive client-specific sharing rather than a controller-specific artifact.

4.4 State Representation and Reward Design

Table 6 isolates the contributions of state features and reward design. For state representation, parameter divergence is the most consistently useful signal: configurations without it show the largest drops (e.g., 74.27% → 71.56% at β=0.3, and 62.59% → 60.44% at β=0.1). This supports the view that local–global mismatch is a primary driver of redistribution decisions.

No single state variant dominates all regimes. At β=0.3, NCF is best (74.27%); at β=0.1, the full state is best (62.59%); and at β=0.05, NCFNLA is best (52.54%). This pattern suggests that informative signals shift with heterogeneity severity, while divergence remains broadly valuable.

For reward design, the proposed combined reward is best at β=0.3 and β=0.1 (73.03%, 61.84%), and close to the best at β=0.05 (vs. 51.94% for Global-only). Overall, balancing client-level and global-progress signals yields the most reliable behavior across settings.

4.5 Practical Efficiency

A final question is whether the gain of FedPolicy is obtained at a meaningful systems cost. The answer is favorable.

4.5.1 Cost to Accuracy Trade-off

Table 7 summarizes the cumulative communication and computation required to first reach a sequence of target accuracies on CIFAR-10 under β=0.1. This analysis measures practical efficiency, not only best accuracy. For baselines, communication per round is fixed because each selected client exchanges the full model in both directions. For FedPolicy, communication is asymmetric: clients still upload full models, while only the downlink is selective. Consequently, cumulative communication gains arise from the combination of reduced downlink payload and faster convergence in rounds.

The results show that FedPolicy is markedly more efficient than the compared methods across all meaningful milestones and is also faster than FedCA at the shared thresholds. It reaches 30% accuracy in 11 rounds using 7.5 GB and 3.7 minutes, whereas FedCA requires 13 rounds, 10.8 GB, and 4.6 minutes. The gap becomes larger at higher thresholds: FedPolicy reaches 40% accuracy in 11 rounds, compared with 21 rounds

Table 5: Controller ablation on CIFAR-10 under two client-partition settings. **Left:** deterministic fixed-class-per-client splits (controlled by classes/client). **Right:** probabilistic Dirichlet label-skew splits (controlled by the concentration parameter). Best result in each column is shown in **bold**.

	Fixed-Class (Deterministic)			*Dirichlet (Probabilistic)*		
Controller	`fc=6`	`fc=4`	`fc=2`	β=0.3	β=0.1	β=0.05
Static Heuristics						
Full-Only	77.12	73.64	49.87	72.67	54.94	42.42
Backbone-Only	76.32	73.78	48.22	**74.75**	56.40	41.17
Head-Only	69.64	59.85	30.48	59.16	25.22	22.57
Random Policy	73.40	72.88	52.81	72.37	58.99	42.07
DRL Algorithms						
SAC	77.59	74.22	54.92	74.28	62.84	**54.38**
TRPO	**78.78**	75.41	50.39	73.88	**64.32**	54.34
A2C	78.60	72.96	54.97	73.80	63.26	52.85
PPO	76.33	72.16	51.00	73.81	58.52	52.56
DQN	77.21	73.21	**55.35**	72.58	62.19	50.66

Table 6: Ablation study on state representation and reward strategy for FedPolicy on CIFAR-10 across heterogeneity levels (β). The full state (All) comprises the soft confusion matrix ($\vec{C}_k$), local accuracy (acc_k), server validation accuracy (ACC), and parameter divergence ($\|\Delta\theta\|$). Best results per column are in **bold**.

Configuration	β=0.3	β=0.1	β=0.05
State Representation			
All: $[\vec{C}_k \oplus \text{acc}_k \oplus \text{ACC} \oplus \|\Delta\theta\|]$	73.83	**62.59**	51.80
NCF: $[\text{acc}_k \oplus \text{ACC} \oplus \|\Delta\theta\|]$	**74.27**	62.20	50.95
NCFNLA: $[\text{ACC} \oplus \|\Delta\theta\|]$	73.08	62.45	**52.54**
OCMLA: $[\vec{C}_k \oplus \text{acc}_k \oplus \text{ACC}]$	71.56	60.44	49.92
OPD: $[\vec{C}_k \oplus \text{acc}_k \oplus \|\Delta\theta\|]$	73.27	61.95	52.20
Reward Strategy			
Local-only (Δacc_k)	72.67	61.21	51.17
Global-only (ΔACC)	72.43	61.72	**51.94**
Proposed ($\Delta\text{acc}_k + \lambda_{\text{glob}}\,\Delta\text{ACC}$)	**73.03**	**61.84**	51.69

Table 7: Cost-to-accuracy analysis on CIFAR-10 (β=0.1). For each method, we report the first round reaching 20%, 30%, 40%, and 50% global test accuracy, together with cumulative communication (GB) and computation time (minutes) at that point. "×" indicates a threshold not reached within 105 rounds. Lower is better. Best values are in **bold**.

Method	Round (↓)				Comm. (GB) (↓)				Comp. (min) (↓)				Best (%)
	20%	30%	40%	50%	20%	30%	40%	50%	20%	30%	40%	50%	
FedAvg	43	80	×	×	36.7	67.5	×	×	15.3	28.2	×	×	31.9
FedProx	40	74	×	×	33.3	61.7	×	×	14.2	26.1	×	×	31.9
FedSAM	24	56	74	×	20.8	47.5	62.5	×	8.1	18.5	24.3	×	45.1
FedFA	68	101	×	×	57.5	85.0	×	×	31.6	46.7	×	×	31.9
CCVR	2	25	×	×	2.5	21.7	×	×	1.6	13.5	×	×	37.2
FedNTD	30	74	74	×	25.8	62.5	62.5	×	10.8	26.1	26.1	×	44.1
Ditto	43	80	×	×	36.7	67.5	×	×	15.3	28.2	×	×	31.9
FedCA	1	13	21	87	0.8	10.8	17.5	72.5	0.3	4.6	7.4	30.5	59.0
FedPolicy (Ours)	**1**	**11**	**11**	**23**	**0.7**	**7.5**	**7.5**	**12.1**	**0.3**	**3.7**	**3.7**	**5.6**	**63.5**

for FedCA, and reaches 50% by round 23 with 12.1 GB of communication and 5.6 minutes of computation. FedCA also reaches 50% within the budget, but only at round 87, while the remaining baselines do not reach this level. FedPolicy ultimately reaches 63.5% accuracy.

These results show that the benefit of FedPolicy is not limited to improved best accuracy. The method reaches useful operating points much earlier and at substantially lower cumulative cost. In particular, Head-only broadcasts reduce the downlink payload from 42.66 MB to 0.02 MB for a given client, which explains the large savings in the early training phase. Taken together, the computational and communication analyses indicate that adaptive post-aggregation redistribution is practically lightweight while yielding a substantially better cost-to-accuracy trade-off under heterogeneous data.

A detailed discussion, including key observations, privacy considerations, and limitations, is provided in Appendix D.

5 Conclusion and Future Directions

This paper examined a common assumption in federated learning: the aggregated global model is shared back to all clients in the same form at every round. Under statistical heterogeneity, such uniform reuse can induce negative transfer by disrupting locally adapted representations. To address this limitation, we proposed FedPolicy, a policy-guided selective redistribution mechanism that learns whether each client should receive the full model, the backbone, or the classifier head after aggregation. Across FMNIST, CIFAR-10, and CIFAR-100, FedPolicy consistently outperformed strong baselines, with the advantage becoming more pronounced as heterogeneity increased. On CIFAR-10, for example, the relative improvement over the strongest baseline grew from +1.12% at β=0.3 to +7.03% at β=0.05, while on CIFAR-100 it increased from +2.33% to +8.40%. Beyond improving final accuracy, the method also accelerated convergence, reaching 60% accuracy on CIFAR-10 at β=0.3 in 11 rounds compared with 29 rounds for FedNTD, and attained the same operating point under β=0.1 with substantially lower communication cost than FedAvg. These gains were obtained with negligible additional overhead, with the controller contributing only 0.021 s per round. Taken together, these results support the central claim of the paper: client-specific post-aggregation redistribution is an important and underexplored design dimension in federated learning under heterogeneity.

In future, we plan to extend the current discrete action space to finer-grained sharing strategies, such as layer-wise selection or continuous mixing. It would also be valuable to examine the method under broader deployment conditions, including asynchronous participation, system heterogeneity, and time-varying client distributions.

References

Manojkumar Varma Arivazhagan, Vinay Aggarwal, Aneesh Kumar Singh, and Sunav Choudhary. Federated learning with personalized layers. *arXiv preprint arXiv:1912.00818*, 2019.

Dengsheng Chen, Jie Hu, Vince Junkai Tan, Xiaoming Wei, and Enhua Wu. Elastic aggregation for federated optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12187–12197, June 2023.

Sujit Chowdhury and Raju Halder. Confusion-calibrated cross-entropy and class-specialized aggregation for robust federated learning under extreme data heterogeneity. *Knowledge-Based Systems*, 338:115497, 2026. ISSN 0950-7051. doi: 10.1016/j.knosys.2026.115497. URL `https://www.sciencedirect.com/science/article/pii/S095070512600239X`.

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

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9268–9277, 2019.

Dennis Grinwald, Philipp Wiesner, and Shinichi Nakajima. Federated learning over connected modes. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=JL2eMCfDW8`.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870. PMLR, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. *International Conference on Machine Learning*, pp. 5132–5143, 2020.

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

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Gihun Lee, Minchan Shin, Sangmin Yun, Sungjoo Son, and Sungho Yun. Preservation of the global knowledge by not-true distillation in federated learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 38461–38474, 2022.

Bo Li, Mikkel N Schmidt, Tommy S Alstrøm, and Sebastian U Stich. On the effectiveness of partial variance reduction in federated learning with heterogeneous data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3964–3973, 2023.

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

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

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

Mi Luo et al. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34:5972–5984, 2021.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2017.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Yuexiang Tan, Weitao Xu, Yu Wang, and Jiashi Feng. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8482–8490, 2022.

Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 1698–1707. IEEE, 2020a.

Haolin Wang, Xuefeng Liu, Jianwei Niu, Wenkai Guo, and Shaojie Tang. Why go full? elevating federated learning through partial network updates. In *Advances in Neural Information Processing Systems*, 2024a.

Jian Wang et al. Flute: Federated learning with unbiased task-specific embedding. *arXiv preprint arXiv:2401.00000*, 2024b.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33: 7611–7623, 2020b.

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

Xin Yang, Hao Zhang, Wei Qi, and Ke Zhang. Fedfa: Federated feature augmentation. In *International Conference on Learning Representations*, 2023.

Xiyuan Yang, Wenke Huang, and Mang Ye. Fedas: Bridging inconsistency in personalized federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11986–11995, June 2024.

Jie Zhang, Zhiqi Li, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, and Chao Wu. Federated learning with label distribution skew via logits calibration. In *International Conference on Machine Learning*, pp. 26311–26329. PMLR, 2022a. arXiv preprint arXiv:2108.13329 (2021).

Jie Zhang et al. Federated learning with label distribution skew via logits calibration. In *International Conference on Machine Learning*, pp. 26311–26329. PMLR, 2022b.

Zhe Zhang et al. Federated learning with sam: Sharpness-aware minimization. In *International Conference on Machine Learning*, 2023.

Appendix

A Related Work

Statistical heterogeneity remains a central challenge in federated learning (FL), motivating extensive research on improving convergence stability and generalization under non-IID data distributions. Early work such as FedAvg McMahan et al. (2017) establishes the standard paradigm of local training followed by server-side aggregation, but suffers from client drift when local objectives differ significantly. To mitigate this, methods such as FedProx Li et al. (2020) introduce proximal regularization to constrain local updates, while SCAFFOLD Karimireddy et al. (2020) employs control variates to correct drift induced by heterogeneous data. Related server-side optimization strategies further stabilize training through normalized or adaptive aggregation rules. Despite their effectiveness, these approaches implicitly assume that the aggregated global model can be uniformly and fully reapplied to all clients, leaving the post-aggregation client update step fixed and unmodeled.

Personalized federated learning (PFL) relaxes the single-global-model assumption by allowing client-specific parameters through multi-task learning, clustering, mixture models, or partial parameter sharing Arivazhagan et al. (2019); Collins et al. (2021). A common design separates models into shared and private components, often retaining a global backbone while personalizing classifier heads. These approaches demonstrate improved client-level performance by reducing negative transfer across heterogeneous distributions. However, personalization structures are typically predefined and static, and optimization is driven by client-local objectives. As a result, PFL methods do not explicitly model how client update decisions influence future aggregation quality or global convergence behavior.

Another line of work improves robustness under heterogeneity by adapting server-side coordination mechanisms, including dynamic client weighting, similarity-aware aggregation, and fairness-oriented reweighting Li et al. (2021). More recently, reinforcement learning (RL) has been employed to automate coordination decisions, such as client selection or aggregation weight tuning, using validation feedback as reward Wang et al. (2020a). While these RL-based methods introduce adaptivity at the aggregation stage, they still apply the resulting global model back to clients using a uniform overwrite or interpolation rule, thereby leaving client-side initialization dynamics unaddressed.

Selective and partial parameter update strategies further demonstrate that *which* parameters are synchronized plays a crucial role in federated optimization. Recent work such as FedPart Wang et al. (2024a) shows that full-model synchronization can induce layer mismatch under heterogeneity, and proposes restricting updates to subsets of layers using predefined schedules. Related approaches freeze or alternate updates between backbone and head to improve stability and efficiency. While these methods highlight the importance of parameter subspace selection, update schedules are typically heuristic and shared across clients, rather than learned as a function of evolving client learning states.

Class imbalance and label skew further exacerbate heterogeneity in FL, motivating imbalance-aware objectives and reweighting strategies Zhang et al. (2022a). However, global class reweighting is often misaligned with client-specific label distributions, and privacy constraints limit direct access to global class statistics. Moreover, existing imbalance-aware methods primarily focus on improving local optimization rather than leveraging class-level error structure as a coordination signal for adaptive system-level control.

For positioning, we emphasize a clear novelty boundary. Prior heterogeneity-aware studies have addressed confusion-aware local objectives and class-prioritized aggregation. The present paper does not reuse aggregation-side prioritization; instead, it addresses a different stage that remained fixed in those formulations: post-aggregation redistribution. By keeping aggregation standard (mean/FedAvg-style) and introducing adaptivity only in client-specific redistribution, the proposed formulation isolates the contribution and avoids conflating redistribution effects with aggregation complexity.

In contrast to prior work that emphasizes local optimization, aggregation weighting, personalization structure, or heuristic partial updates, this work focuses on an underexplored but consequential dimension of federated learning: *how heterogeneous clients should initialize from aggregated parameters after standard aggregation*. We retain standard aggregation operators (e.g., FedAvg) and instead learn a policy that con-

trols post-aggregation client initialization (full model, backbone-only, or head-only updates). By leveraging confusion-aware local training signals that compactly characterize class-level struggle patterns, our approach enables principled policy learning for client-specific update control and explicitly models the long-term impact of initialization decisions on future aggregation quality.

B Detailed Theoretical Proofs

This appendix provides detailed proofs for the theoretical analysis presented in Section 3. The analysis isolates the effect of selective post-aggregation parameter sharing while keeping server aggregation unchanged and abstracting away the policy-learning dynamics of the RL controller.

B.1 Proof Notation

Table 8 summarizes the symbols used in the proofs below.

Table 8: Notation used in the theoretical proofs.

Symbol	Description
$\mathbf{w}^t$	Aggregated global model at communication round t
$\tilde{\mathbf{w}}_k^t$	Client-specific initialized model for client k at round t
$\boldsymbol{\delta}_k^t$	Initialization difference, $\tilde{\mathbf{w}}_k^t - \mathbf{w}^t$
$\mathcal{L}_k(\cdot)$	Local objective of client k
$\nabla\mathcal{L}_k(\cdot)$	Gradient of the client-k objective
L	Smoothness constant of each $\mathcal{L}_k$
μ	Block-wise strong convexity constant
ϵ_{drift}	Upper bound on the relevant block gradient magnitude
$\mathbf{w}_{\mathsf{Full}}^t$	Full global overwrite candidate
$\mathbf{w}_{\mathsf{Back}}^t$	Backbone-only overwrite candidate
$\mathbf{w}_{\mathsf{Head}}^t$	Head-only overwrite candidate
$\mathbf{w}_{\mathsf{Best}}^t$	Best candidate selected among the available sharing actions
$\boldsymbol{\phi}^t$	Global backbone parameters at round t
$\boldsymbol{\psi}^t$	Global head parameters at round t
$\boldsymbol{\phi}_k^{t-1}$	Previous local backbone parameters of client k
$\boldsymbol{\psi}_k^{t-1}$	Previous local head parameters of client k
$\mathcal{U}$	Neighborhood on which block-wise strong convexity holds
$\langle\cdot,\cdot\rangle$	Standard Euclidean inner product
$\|\cdot\|$	Standard Euclidean norm

B.2 Assumptions Used in the Proofs

For completeness, we state below the assumptions used in the results of this appendix.

Assumption 1 (Smoothness)**.** *For each client k, the local objective $\mathcal{L}_k$ is L-smooth. Equivalently, for all $u, v \in \mathbb{R}^d$,*

$$\mathcal{L}_k(u) \le \mathcal{L}_k(v) + \langle \nabla\mathcal{L}_k(v),\, u - v\rangle + \frac{L}{2}\|u - v\|^2.$$

Assumption 2 (Stochastic gradient variance)**.** *Client stochastic gradients are unbiased and have bounded variance:*

$$\mathbb{E}\|g_k(w) - \nabla\mathcal{L}_k(w)\|^2 \le \sigma^2.$$

Assumption 3 (Gradient dissimilarity)**.** *Client gradients are heterogeneity-bounded:*

$$\sum_k p_k \|\nabla\mathcal{L}_k(w) - \nabla f(w)\|^2 \le B^2, \qquad f(w) = \sum_k p_k \mathcal{L}_k(w).$$

Assumption 4 (Block-wise strong convexity)**.** *There exists a neighborhood $\mathcal{U}$ containing the candidate initialization points such that, for each client k:*

- *with the backbone fixed, the function*
$$h(\boldsymbol{\psi}) := \mathcal{L}_k([\boldsymbol{\phi}^t; \boldsymbol{\psi}])$$
is μ-strongly convex in $\boldsymbol{\psi}$ on $\mathcal{U}$;
- *with the head fixed, the function*
$$g(\boldsymbol{\phi}) := \mathcal{L}_k([\boldsymbol{\phi}; \boldsymbol{\psi}^t])$$
is μ-strongly convex in $\boldsymbol{\phi}$ on $\mathcal{U}$.

Equivalently, for any differentiable μ-strongly convex block-restricted function q and any $u, v \in \mathcal{U}$,

$$q(u) - q(v) \geq \langle \nabla q(v),\, u - v \rangle + \frac{\mu}{2}\|u - v\|^2.$$

Assumption 5 (Block-gradient drift bound)**.** *The candidate partially updated models satisfy the following block-gradient bounds:*

$$\|\nabla_{\boldsymbol{\psi}} \mathcal{L}_k(\mathbf{w}^t_{\mathit{Back}})\| \leq \epsilon_{\text{drift}}, \qquad \|\nabla_{\boldsymbol{\phi}} \mathcal{L}_k(\mathbf{w}^t_{\mathit{Head}})\| \leq \epsilon_{\text{drift}}.$$

The local comparison proofs below use Assumptions 1, 4, and 5. The global convergence proofs use Assumptions 1, 2, and 3.

B.3 Initialization Risk Bound

Lemma 2 (Initialization risk bound)**.** *Under Assumption 1, for any client k and round t,*

$$\mathcal{L}_k(\tilde{\mathbf{w}}^t_k) - \mathcal{L}_k(\mathbf{w}^t) \leq \langle \nabla \mathcal{L}_k(\mathbf{w}^t),\, \boldsymbol{\delta}^t_k \rangle + \frac{L}{2}\|\boldsymbol{\delta}^t_k\|^2,$$

where

$$\boldsymbol{\delta}^t_k := \tilde{\mathbf{w}}^t_k - \mathbf{w}^t.$$

Proof. By Assumption 1, the function $\mathcal{L}_k$ is L-smooth. Therefore, for any $u, v \in \mathbb{R}^d$,

$$\mathcal{L}_k(u) \leq \mathcal{L}_k(v) + \langle \nabla \mathcal{L}_k(v),\, u - v \rangle + \frac{L}{2}\|u - v\|^2. \tag{10}$$

We now choose

$$u = \tilde{\mathbf{w}}^t_k, \qquad v = \mathbf{w}^t.$$

Substituting these into Eq. (10), we obtain

$$\mathcal{L}_k(\tilde{\mathbf{w}}^t_k) \leq \mathcal{L}_k(\mathbf{w}^t) + \langle \nabla \mathcal{L}_k(\mathbf{w}^t),\, \tilde{\mathbf{w}}^t_k - \mathbf{w}^t \rangle + \frac{L}{2}\|\tilde{\mathbf{w}}^t_k - \mathbf{w}^t\|^2.$$

By definition,

$$\boldsymbol{\delta}^t_k := \tilde{\mathbf{w}}^t_k - \mathbf{w}^t.$$

Hence the previous inequality becomes

$$\mathcal{L}_k(\tilde{\mathbf{w}}^t_k) \leq \mathcal{L}_k(\mathbf{w}^t) + \langle \nabla \mathcal{L}_k(\mathbf{w}^t),\, \boldsymbol{\delta}^t_k \rangle + \frac{L}{2}\|\boldsymbol{\delta}^t_k\|^2.$$

Subtracting $\mathcal{L}_k(\mathbf{w}^t)$ from both sides yields

$$\mathcal{L}_k(\tilde{\mathbf{w}}^t_k) - \mathcal{L}_k(\mathbf{w}^t) \leq \langle \nabla \mathcal{L}_k(\mathbf{w}^t),\, \boldsymbol{\delta}^t_k \rangle + \frac{L}{2}\|\boldsymbol{\delta}^t_k\|^2,$$

which proves the result.

B.4 Sufficient Condition for Strict Improvement

Corollary 2 (Sufficient condition for strict improvement)**.** *If*

$$\langle \nabla \mathcal{L}_k(\mathbf{w}^t),\, \boldsymbol{\delta}_k^t \rangle + \frac{L}{2}\|\boldsymbol{\delta}_k^t\|^2 < 0,$$

then

$$\mathcal{L}_k(\tilde{\mathbf{w}}_k^t) < \mathcal{L}_k(\mathbf{w}^t).$$

Proof. From Lemma 2,

$$\mathcal{L}_k(\tilde{\mathbf{w}}_k^t) - \mathcal{L}_k(\mathbf{w}^t) \le \langle \nabla \mathcal{L}_k(\mathbf{w}^t),\, \boldsymbol{\delta}_k^t \rangle + \frac{L}{2}\|\boldsymbol{\delta}_k^t\|^2.$$

By the stated condition, the right-hand side is strictly negative. Therefore,

$$\mathcal{L}_k(\tilde{\mathbf{w}}_k^t) - \mathcal{L}_k(\mathbf{w}^t) < 0.$$

Adding $\mathcal{L}_k(\mathbf{w}^t)$ to both sides gives

$$\mathcal{L}_k(\tilde{\mathbf{w}}_k^t) < \mathcal{L}_k(\mathbf{w}^t),$$

which proves the claim.

B.5 Policy Advantage over Uniform Overwrite

Theorem 4 (Policy advantage over uniform overwrite)**.** *Under Assumptions 4 and 5,*

$$\mathcal{L}_k(\mathbf{w}_{\mathit{Full}}^t) - \mathcal{L}_k(\mathbf{w}_{\mathit{Best}}^t) \ge \frac{\mu}{4}\min\{\|\boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1}\|^2,\ \|\boldsymbol{\phi}^t - \boldsymbol{\phi}_k^{t-1}\|^2\} - \frac{\epsilon_{\text{drift}}^2}{\mu}.$$

Proof. We define the three candidate initialized models:

$$\mathbf{w}_{\mathsf{Full}}^t = [\boldsymbol{\phi}^t; \boldsymbol{\psi}^t], \qquad \mathbf{w}_{\mathsf{Back}}^t = [\boldsymbol{\phi}^t; \boldsymbol{\psi}_k^{t-1}], \qquad \mathbf{w}_{\mathsf{Head}}^t = [\boldsymbol{\phi}_k^{t-1}; \boldsymbol{\psi}^t].$$

The proof first compares Full with Back, then Full with Head, and finally uses the fact that the best candidate among the available options attains the smallest loss.

Step 1: Compare Full with Back. Fix the backbone at $\boldsymbol{\phi}^t$ and define the block-restricted function

$$h(\boldsymbol{\psi}) := \mathcal{L}_k([\boldsymbol{\phi}^t; \boldsymbol{\psi}]).$$

By Assumption 4, the function h is μ-strongly convex in $\boldsymbol{\psi}$ on $\mathcal{U}$. Therefore, for any $u, v \in \mathcal{U}$,

$$h(u) - h(v) \ge \langle \nabla h(v),\, u - v \rangle + \frac{\mu}{2}\|u - v\|^2. \tag{11}$$

Choose

$$u = \boldsymbol{\psi}^t, \qquad v = \boldsymbol{\psi}_k^{t-1}.$$

Then Eq. (11) gives

$$h(\boldsymbol{\psi}^t) - h(\boldsymbol{\psi}_k^{t-1}) \ge \langle \nabla h(\boldsymbol{\psi}_k^{t-1}),\, \boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1} \rangle + \frac{\mu}{2}\|\boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1}\|^2.$$

Using the definition of h,

$$h(\boldsymbol{\psi}^t) = \mathcal{L}_k(\mathbf{w}_{\mathsf{Full}}^t), \qquad h(\boldsymbol{\psi}_k^{t-1}) = \mathcal{L}_k(\mathbf{w}_{\mathsf{Back}}^t),$$

and

$$\nabla h(\boldsymbol{\psi}_k^{t-1}) = \nabla_{\boldsymbol{\psi}} \mathcal{L}_k(\mathbf{w}_{\mathsf{Back}}^t).$$

Hence

$$\mathcal{L}_k(\mathbf{w}^t_{\mathsf{Full}}) - \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Back}}) \geq \left\langle \nabla_{\boldsymbol{\psi}} \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Back}}),\, \boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1} \right\rangle + \frac{\mu}{2} \|\boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1}\|^2. \tag{12}$$

By Assumption 5,

$$\left\| \nabla_{\boldsymbol{\psi}} \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Back}}) \right\| \leq \epsilon_{\text{drift}}.$$

Using the Cauchy–Schwarz inequality,

$$\left\langle \nabla_{\boldsymbol{\psi}} \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Back}}),\, \boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1} \right\rangle \geq - \left\| \nabla_{\boldsymbol{\psi}} \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Back}}) \right\| \cdot \left\| \boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1} \right\| \geq -\epsilon_{\text{drift}} \left\| \boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1} \right\|.$$

Next we apply Young's inequality in the form

$$ab \leq \frac{\mu}{4} b^2 + \frac{1}{\mu} a^2.$$

Setting

$$a = \epsilon_{\text{drift}}, \qquad b = \|\boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1}\|,$$

we obtain

$$-\epsilon_{\text{drift}} \|\boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1}\| \geq -\frac{\mu}{4} \|\boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1}\|^2 - \frac{\epsilon^2_{\text{drift}}}{\mu}.$$

Substituting this into Eq. (12), we get

$$\mathcal{L}_k(\mathbf{w}^t_{\mathsf{Full}}) - \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Back}}) \geq \left(\frac{\mu}{2} - \frac{\mu}{4} \right) \|\boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1}\|^2 - \frac{\epsilon^2_{\text{drift}}}{\mu}.$$

Therefore,

$$\mathcal{L}_k(\mathbf{w}^t_{\mathsf{Full}}) - \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Back}}) \geq \frac{\mu}{4} \|\boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1}\|^2 - \frac{\epsilon^2_{\text{drift}}}{\mu}. \tag{13}$$

Step 2: Compare Full with Head. Fix the head at $\boldsymbol{\psi}^t$ and define

$$g(\boldsymbol{\phi}) := \mathcal{L}_k([\boldsymbol{\phi}; \boldsymbol{\psi}^t]).$$

By Assumption 4, the function g is μ-strongly convex in $\boldsymbol{\phi}$ on $\mathcal{U}$. Thus, for any $u, v \in \mathcal{U}$,

$$g(u) - g(v) \geq \langle \nabla g(v),\, u - v \rangle + \frac{\mu}{2} \|u - v\|^2.$$

Choose

$$u = \boldsymbol{\phi}^t, \qquad v = \boldsymbol{\phi}_k^{t-1}.$$

Then

$$g(\boldsymbol{\phi}^t) - g(\boldsymbol{\phi}_k^{t-1}) \geq \left\langle \nabla g(\boldsymbol{\phi}_k^{t-1}),\, \boldsymbol{\phi}^t - \boldsymbol{\phi}_k^{t-1} \right\rangle + \frac{\mu}{2} \|\boldsymbol{\phi}^t - \boldsymbol{\phi}_k^{t-1}\|^2.$$

Using the definition of g,

$$g(\boldsymbol{\phi}^t) = \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Full}}), \qquad g(\boldsymbol{\phi}_k^{t-1}) = \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Head}}),$$

and

$$\nabla g(\boldsymbol{\phi}_k^{t-1}) = \nabla_{\boldsymbol{\phi}} \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Head}}).$$

Hence

$$\mathcal{L}_k(\mathbf{w}^t_{\mathsf{Full}}) - \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Head}}) \geq \left\langle \nabla_{\boldsymbol{\phi}} \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Head}}),\, \boldsymbol{\phi}^t - \boldsymbol{\phi}_k^{t-1} \right\rangle + \frac{\mu}{2} \|\boldsymbol{\phi}^t - \boldsymbol{\phi}_k^{t-1}\|^2. \tag{14}$$

Again, by Assumption 5,

$$\left\| \nabla_{\boldsymbol{\phi}} \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Head}}) \right\| \leq \epsilon_{\text{drift}}.$$

Applying Cauchy–Schwarz and Young's inequality as above gives

$$\left\langle \nabla_{\boldsymbol{\phi}} \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Head}}),\, \boldsymbol{\phi}^t - \boldsymbol{\phi}^{t-1}_k \right\rangle \geq -\frac{\mu}{4}\|\boldsymbol{\phi}^t - \boldsymbol{\phi}^{t-1}_k\|^2 - \frac{\epsilon^2_{\text{drift}}}{\mu}.$$

Substituting this into Eq. (14), we obtain

$$\mathcal{L}_k(\mathbf{w}^t_{\mathsf{Full}}) - \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Head}}) \geq \frac{\mu}{4}\|\boldsymbol{\phi}^t - \boldsymbol{\phi}^{t-1}_k\|^2 - \frac{\epsilon^2_{\text{drift}}}{\mu}. \tag{15}$$

Step 3: Use the best-candidate selection rule. By definition,

$$\mathcal{L}_k(\mathbf{w}^t_{\mathsf{Best}}) = \min\left\{\mathcal{L}_k(\mathbf{w}^t_{\mathsf{Full}}),\, \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Back}}),\, \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Head}})\right\}.$$

Therefore,

$$\mathcal{L}_k(\mathbf{w}^t_{\mathsf{Full}}) - \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Best}}) \geq \max\left\{\mathcal{L}_k(\mathbf{w}^t_{\mathsf{Full}}) - \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Back}}),\, \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Full}}) - \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Head}})\right\}.$$

Using Eqs. (13) and (15), we get

$$\mathcal{L}_k(\mathbf{w}^t_{\mathsf{Full}}) - \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Best}}) \geq \frac{\mu}{4}\max\left\{\|\boldsymbol{\psi}^t - \boldsymbol{\psi}^{t-1}_k\|^2,\, \|\boldsymbol{\phi}^t - \boldsymbol{\phi}^{t-1}_k\|^2\right\} - \frac{\epsilon^2_{\text{drift}}}{\mu}.$$

Finally, for any nonnegative scalars x and y,

$$\max\{x, y\} \geq \min\{x, y\}.$$

Applying this gives the stated conservative lower bound:

$$\mathcal{L}_k(\mathbf{w}^t_{\mathsf{Full}}) - \mathcal{L}_k(\mathbf{w}^t_{\mathsf{Best}}) \geq \frac{\mu}{4}\min\left\{\|\boldsymbol{\psi}^t - \boldsymbol{\psi}^{t-1}_k\|^2,\, \|\boldsymbol{\phi}^t - \boldsymbol{\phi}^{t-1}_k\|^2\right\} - \frac{\epsilon^2_{\text{drift}}}{\mu}.$$

This completes the proof.

B.6 Global Convergence Proofs

This subsection provides the deferred proof details for Theorem 2 and Theorem 3. In contrast to the local comparison results above, these bounds rely on Assumptions 1, 2, and 3.

Proof of Theorem 2. Let $f(w) := \sum_k p_k \mathcal{L}_k(w)$ and let w^t be the server model at round t. With E local steps and step size $\eta \leq 1/(LE)$, standard smoothness descent for one communication round gives

$$\mathbb{E}[f(w^{t+1})] \leq \mathbb{E}[f(w^t)] - \frac{\eta E}{2}\,\mathbb{E}\|\nabla f(w^t)\|^2 + L\eta^2 E\sigma^2 + L\eta^2 E^2 B^2 + \frac{L\eta}{2}\,\Gamma_t^2, \tag{16}$$

where:

- the $L\eta^2 E\sigma^2$ term comes from Assumption 2,
- the $L\eta^2 E^2 B^2$ term comes from Assumption 3,
- and Γ_t^2 is the per-round redistribution drift (mismatch induced by selective initialization).

Assume $\Gamma_t^2 \leq \Gamma^2$ for all t.

Rearranging Eq. (16):

$$\frac{\eta E}{2}\,\mathbb{E}\|\nabla f(w^t)\|^2 \leq \mathbb{E}[f(w^t)] - \mathbb{E}[f(w^{t+1})] + L\eta^2 E\sigma^2 + L\eta^2 E^2 B^2 + \frac{L\eta}{2}\,\Gamma^2.$$

Summing from $t = 0$ to $T - 1$ yields

$$\frac{\eta E}{2} \sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(w^t)\|^2 \leq \mathbb{E}[f(w^0)] - \mathbb{E}[f(w^T)] + TL\eta^2 E\sigma^2 + TL\eta^2 E^2 B^2 + T\frac{L\eta}{2}\Gamma^2.$$

Since $f(w^T) \geq f^\star$, we have

$$\frac{\eta E}{2} \sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(w^t)\|^2 \leq \mathbb{E}[f(w^0)] - f^\star + TL\eta^2 E\sigma^2 + TL\eta^2 E^2 B^2 + T\frac{L\eta}{2}\Gamma^2.$$

Divide by $\eta ET/2$ to obtain

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(w^t)\|^2 \leq \frac{2(\mathbb{E}[f(w^0)] - f^\star)}{\eta ET} + 2L\eta\sigma^2 + 2L\eta EB^2 + \frac{L}{E}\Gamma^2,$$

which is exactly the claimed bound.

Proof of Theorem 3. Apply Theorem 2 to two schemes with identical $L, \eta, E, \sigma^2, B^2$ and different drift levels:

$$\mathcal{U}_{\mathsf{Pol}} = \frac{2(\mathbb{E}[f(w^0)] - f^\star)}{\eta ET} + 2L\eta\sigma^2 + 2L\eta EB^2 + \frac{L}{E}\Gamma^2_{\mathsf{Pol}},$$

$$\mathcal{U}_{\mathsf{Full}} = \frac{2(\mathbb{E}[f(w^0)] - f^\star)}{\eta ET} + 2L\eta\sigma^2 + 2L\eta EB^2 + \frac{L}{E}\Gamma^2_{\mathsf{Full}}.$$

If $\Gamma^2_{\mathsf{Pol}} \leq \rho\,\Gamma^2_{\mathsf{Full}}$ with $\rho < 1$, then

$$\mathcal{U}_{\mathsf{Pol}} - \mathcal{U}_{\mathsf{Full}} = \frac{L}{E}\left(\Gamma^2_{\mathsf{Pol}} - \Gamma^2_{\mathsf{Full}}\right) \leq -\frac{L}{E}(1-\rho)\Gamma^2_{\mathsf{Full}} < 0$$

whenever $\Gamma^2_{\mathsf{Full}} > 0$. Hence the selective redistribution bound is strictly tighter than uniform Full overwrite, proving the theorem.

C Additional Experimental Evaluation

This appendix collects supporting experiments that complement the main paper: implementation details, controller behavior diagnostics, loss and reward ablations, and the overhead analysis.

C.1 Experimental Setup

C.1.1 Implementation Details

All methods are implemented in `PyTorch` using Python 3.9. We evaluate on three standard image classification benchmarks of increasing difficulty: Fashion-MNIST (FMNIST) Xiao et al. (2017), which contains 10 classes of 28×28 grayscale images with 70K total samples; CIFAR-10 Krizhevsky (2009), which comprises 10 classes of 32×32 RGB images with 60K samples; and CIFAR-100 Krizhevsky (2009), which consists of 100 classes of 32×32 RGB images with 60K samples. For FMNIST, we use LeNet-5 LeCun et al. (1998) as the base model. For CIFAR-10 and CIFAR-100, we use ResNet-18 He et al. (2016) as the feature extractor, followed by a linear classifier head Ψ with $C \times 512 + C$ parameters, where C denotes the number of classes. The ResNet-18 backbone contains $|\Phi| = 11{,}176{,}512$ parameters. For CIFAR-10, the classifier head contains $|\Psi| = 5{,}130$ parameters, accounting for only $\approx$0.046% of the full model. In float32 precision, the complete ResNet-18-based model occupies 42.66 MB, whereas the classifier head requires only 0.02 MB. This pronounced backbone-head asymmetry is a key architectural property underlying our design. Because the classifier head constitutes only a negligible fraction of the total parameter budget, it enables selective global parameter sharing with fine-grained control over the balance between transferable global representations and client-specific classifier adaptation.

C.1.2 Data Partitioning

We consider two heterogeneous data-generation settings:

- **Dirichlet-based heterogeneity.** We simulate label-distribution skew across clients using a Dirichlet-based partitioning scheme Hsu et al. (2019) with concentration parameter $\beta \in \{0.3, 0.1, 0.05\}$. Smaller β values induce more severe heterogeneity by producing increasingly imbalanced class proportions across clients, whereas larger β values yield more balanced local distributions. To ensure fair comparison, all methods use the same client partitions generated with a fixed random seed.
- **Fixed-class-per-client heterogeneity.** To evaluate robustness under explicit class absence, we construct partitions in which each client is assigned data from at most $\texttt{fc} \in \{6, 4, 2\}$ classes. Smaller `fc` values therefore correspond to more extreme heterogeneity, as clients become increasingly specialized to a limited subset of classes.

These two settings provide complementary views of statistical heterogeneity: the Dirichlet-based scheme models stochastic variation in label proportions, whereas the fixed-class setting imposes a more structured and severe form of class-distribution skew.

C.1.3 Training Configuration

Unless otherwise stated, all experiments use the same default configuration. We simulate a federation of N=100 clients, with C_r=10 clients sampled uniformly at random in each communication round. Each selected client performs E=5 local epochs of SGD with learning rate η=10^{-2}, momentum 0.9, weight decay 5×10^{-4}, and batch size 64. The local learning rate is fixed at 0.01 after hyperparameter tuning over the range $[10^{-4}, 10^{-1}]$. All methods are trained for T=105 communication rounds. For the baseline methods, we retain the local loss specified in their original formulations. For FedPolicy, we evaluate several local objectives, including standard cross-entropy (CE), Focal Loss (FL) Lin et al. (2017), logits-calibrated loss (FedLC) Zhang et al. (2022b), Class-Balanced loss (CB) Cui et al. (2019), Label Smoothing (LS) Szegedy et al. (2016), and Confusion-Calibrated Cross-Entropy (C3E) Chowdhury & Halder (2026). Unless explicitly noted otherwise, reported results correspond to the best Top-1 global test accuracy (%) achieved over the training trajectory and averaged over three independent runs with different random seeds. A server-held-out validation split is used only to compute ACC($\cdot$) for controller state/reward; test labels are never used for controller training, model selection, or policy updates.

For server-side aggregation, each baseline follows the aggregation mechanism defined in its original paper, while FedPolicy uses a fixed mean/FedAvg-style rule. The default FedPolicy controller in the main paper is SAC Haarnoja et al. (2018), parameterized by an MLP with 64 hidden units, trained with learning rate 0.05 (fixed by tuning over $[10^{-4}, 10^{-1}]$). Alternative RL controllers (PPO, A2C, TRPO, and DQN) are evaluated only in the policy-ablation study. All methods share identical random seeds, data partitions, and model initializations to ensure strict comparability.

C.1.4 Baselines

We benchmark FedPolicy against eight popular FL baselines spanning diverse algorithmic families: FedAvg McMahan et al. (2017) (standard aggregation), FedProx Li et al. (2020) (drift correction), FedNTD Lee et al. (2022) (knowledge distillation), CCVR Luo et al. (2021) and FedFA Yang et al. (2023) (representation alignment), FedSAM Zhang et al. (2023) (sharpness-aware optimization), Ditto Li et al. (2021) (personalization), and FedCA Chowdhury & Halder (2026) (confusion-aware local optimization, originally proposed with class-prioritized aggregation). These baselines provide strong points of comparison because they address heterogeneity through different design principles—local regularization, representation alignment, personalization, modified optimization, or confusion-aware coordination—whereas FedPolicy targets the post-aggregation global knowledge transfer stage. All methods are evaluated under the same experimental protocol, with matched communication rounds and identical client partitions.

C.2 DRL-Policy Behavior over Data Heterogeneity

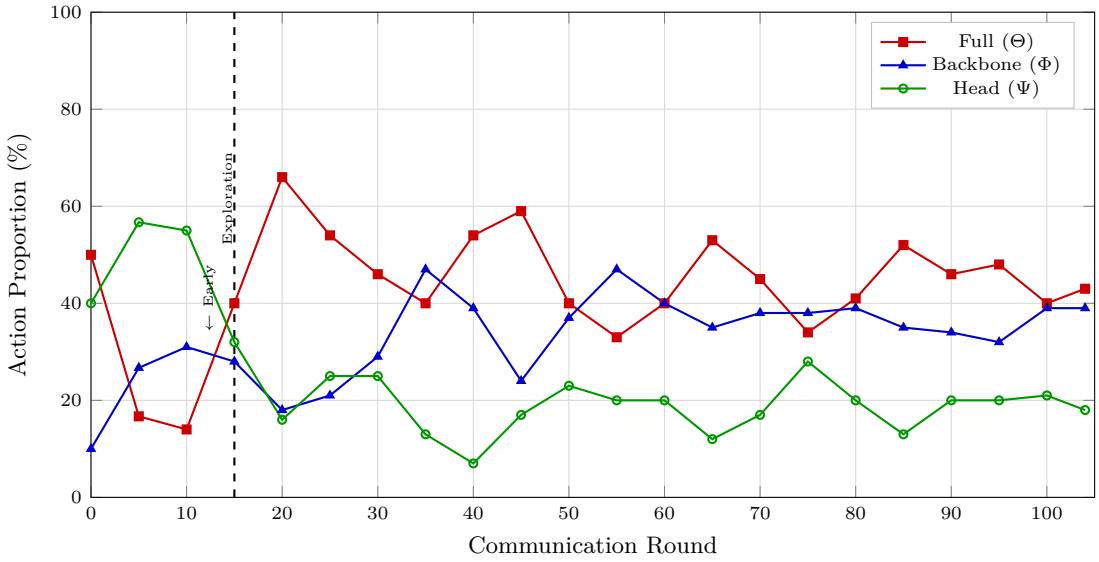

Figure 6: DRL controller action distribution for FedPolicy on CIFAR-10 (β=0.1).

Figure 6 shows how the SAC controller changes its action preference over training. In the initial rounds (0–15), Head updates are most frequent (about 55%), indicating that the policy first favors local classifier adaptation while limiting global interference. As training proceeds, the share of Full updates rises to roughly 40–60%, whereas Backbone updates remain consistently active at around 30–40%. This suggests that the controller first allows client-specific specialization and later increases global coordination as training stabilizes.

Figures 7 and 8 show that the learned policy depends strongly on both the RL algorithm and the heterogeneity level. A clear distinction appears between the actor–critic methods (SAC, TRPO, PPO, and A2C) and DQN. While the actor–critic methods learn non-uniform and clearly structured policies, DQN remains close to a uniform allocation across Full, Backbone, and Head updates for all β values. For example, its action split is 33.7%/34.6%/31.7% at β=0.05 and 32.9%/35.9%/31.2% at β=0.3, with entropy remaining near 1.0 throughout training. This indicates that DQN fails to specialize, even though such a uniform strategy remains reasonably competitive under mild heterogeneity.

Among the actor–critic methods, backbone-dominant policies emerge consistently as heterogeneity increases. Under severe heterogeneity (β=0.05), all four methods allocate most actions to Backbone updates, with shares ranging from 77.9% to 93.3%. TRPO is the most stable in this respect, maintaining a backbone share of 77.9%–80.4% across all three β values and achieving the best or near-best accuracy throughout. SAC shows a similar tendency, favoring Backbone updates at β=0.05 (83.0%) while shifting toward more Full updates as heterogeneity weakens, and attains the highest accuracy at both β=0.05 (54.4%) and β=0.3 (74.3%). A2C also remains largely backbone-oriented and performs strongly at β=0.1 and β=0.3. PPO is the least stable: although it is strongly backbone-dominant at β=0.05 (93.3%), it shifts to a head-heavy policy at β=0.1 (65.0% Head), where its accuracy drops to 58.5%.

The effect of policy choice is most visible at β=0.1. Here, TRPO's backbone-heavy policy (73.4% Backbone) reaches 64.3% accuracy, and A2C's mainly Full/Backbone allocation (49.0%/44.8%) reaches 63.3%, whereas PPO's head-dominant policy performs substantially worse. By contrast, at β=0.3, the accuracy gap across algorithms narrows to only 1.7 percentage points, indicating that policy specialization matters less when heterogeneity is mild. Overall, the results point to a consistent conclusion: as data heterogeneity becomes stronger, effective policies place increasing emphasis on Backbone sharing, suggesting that feature-level transfer is more reliable than uniform full-model synchronization in highly non-IID settings.

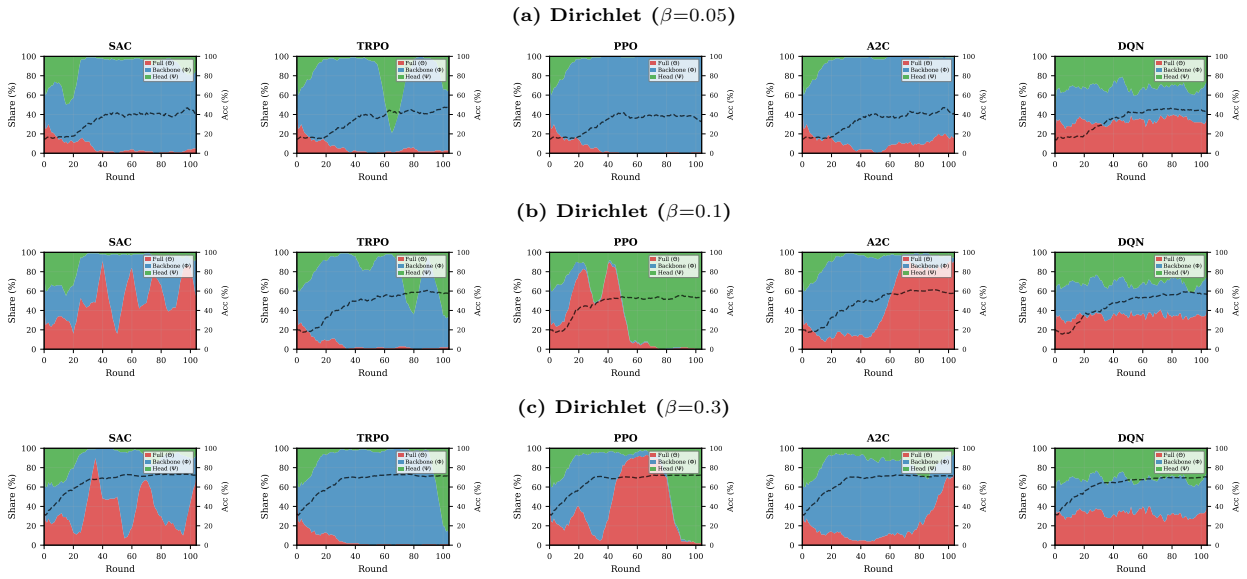

Figure 7: Per-algorithm action evolution (10-round sliding window) for five DRL algorithms across three Dirichlet heterogeneity levels. Each panel shows stacked action proportions for Full (Θ), Backbone (Φ), and Head (Ψ), together with global test accuracy (dashed line, right axis).

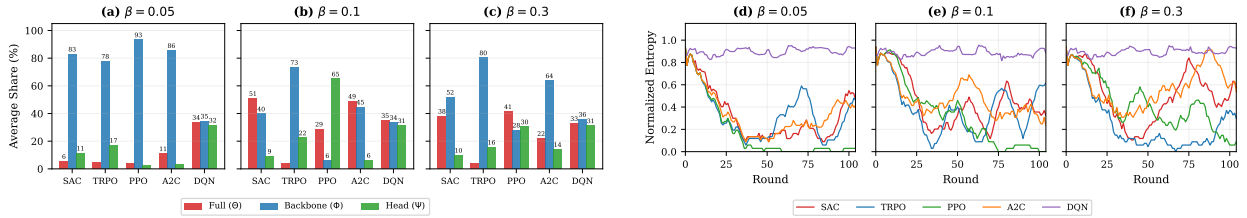

Figure 8: Multi-algorithm policy summary. **Left (a–c):** grouped bar charts of average action proportions per algorithm at each β. **Right (d–f):** normalized Shannon entropy $H/\log 3$ of action distributions over communication rounds.

C.3 Ablation Study

C.3.1 Local-Loss Analysis

Figure 9 compares six local loss functions on CIFAR-10 within FedPolicy under three Dirichlet heterogeneity levels. The goal of this study is to identify which local objective is most compatible with the proposed adaptive parameter-sharing framework under heterogeneous client distributions.

The results show a clear separation between C3E and the alternative losses, and this difference becomes more pronounced as heterogeneity increases. Under severe skew (β=0.05), C3E is the only loss that sustains learning in the 55% accuracy range. By contrast, CE, CB, FL, and LS remain close to chance level for most of training, while FedLC improves only briefly before deteriorating. At β=0.1, C3E again maintains the strongest and most stable

Table 9: Local-loss comparison on CIFAR-10. CE versus C3E accuracy (%) across baselines and FedPolicy at $\beta \in \{0.3, 0.1, 0.05\}$. Best result of each heterogeneity level is shown in **bold**.

	β=0.3		β=0.1		β=0.05	
Method	CE	C3E	CE	C3E	CE	C3E
FedAvg	64.76	71.30	31.25	49.27	16.11	39.19
FedProx	64.82	72.10	33.79	44.20	16.31	40.12
Ditto	65.00	72.40	31.25	43.10	17.18	40.50
CCVR	67.31	73.78	38.57	51.80	15.78	48.30
FedSAM	54.13	61.80	27.25	38.70	14.66	46.10
FedFA	63.51	71.80	28.31	41.60	17.35	46.83
FedNTD	69.20	72.98	44.08	53.77	21.52	32.54
FedCA	69.28	73.46	40.68	59.01	23.83	51.62
FedPolicy	71.41	**74.20**	48.67	**63.57**	28.43	**55.25**

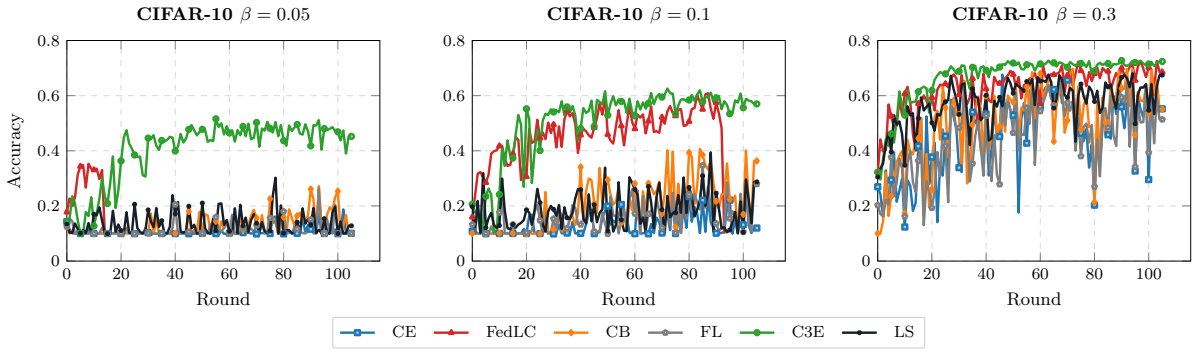

Figure 9: Convergence trajectories of FedPolicy on CIFAR-10 with various local loss functions under $\beta \in \{0.05, 0.1, 0.3\}$.

convergence, whereas the remaining losses either saturate early or exhibit unstable trajectories. Even at the mildest setting (β=0.3), where the gap is smaller, C3E still achieves the best peak accuracy with the smoothest training behavior.

Table 9 shows that replacing CE with C3E consistently strengthens all compared methods across all heterogeneity levels. However, the advantage of FedPolicy cannot be attributed to the local loss alone. Even when the baselines are equipped with the same C3E objective, FedPolicy remains the top-performing method at every $\beta \in \{0.3, 0.1, 0.05\}$. In this appendix table, CCVR is the strongest C3E baseline at β=0.3, while FedCA is strongest at β=0.1 and β=0.05, reaching 73.78%, 59.01%, and 51.62%, respectively. FedPolicy reaches 74.20%, 63.57%, and 55.25%, corresponding to gains of +0.42, +4.56, and +3.63 percentage points. Under the standard CE setting, FedCA is likewise the strongest baseline at all three heterogeneity levels; FedPolicy improves over it by +2.13, +7.99, and +4.60 percentage points. These results indicate that the benefit of FedPolicy is additive to that of stronger local optimization, and arises from the proposed adaptive post-aggregation parameter-sharing policy.

C.3.2 Reward Weight Sensitivity

Table 10 examines the effect of the global reward weight λ_{glob} within the proposed reward formulation. No single value is uniformly optimal across all settings, but $\lambda_{glob} = 0.25$ and $\lambda_{glob} = 1.0$ perform consistently well. In particular, $\lambda_{glob} = 0.25$ achieves the best accuracy at β=0.3 and β=0.05, while $\lambda_{glob} = 1.0$ is best at β=0.1. The variation across settings is modest, suggesting that the method is not unduly sensitive to precise reward weighting as long as both local and global signals are retained.

Table 10: Sensitivity analysis of the hyperparameter λ_{glob} of our proposed reward function.

λ_{glob}	β=0.3	β=0.1	β=0.05
0.10	72.69	60.54	51.56
0.25	**73.03**	61.74	**51.69**
0.50	72.85	60.20	51.45
0.75	72.37	61.02	51.11
1.00	72.91	**61.84**	51.63

C.4 Cost Analysis

For practical deployment, the adaptive controller should improve performance without introducing a meaningful systems burden. We therefore evaluate FedPolicy along three complementary dimensions: per-round computational overhead, communication cost, and the joint cost–accuracy trade-off.

Table 11: Per-round wall-clock time decomposition (seconds) on CIFAR-10 (β=0.1). The — indicates that baseline costs do not include controller overhead.

Method	Local Train (s)	Agg. (s)	DRL (s)	Total (s)
FedAvg	20.90	0.012	—	20.91
FedProx	20.90	0.012	—	20.91
FedNTD	20.90	0.012	—	20.91
Ditto	20.90	0.012	—	20.91
FedSAM	19.44	0.026	—	19.47
FedFA	27.44	0.030	—	27.47
CCVR	31.06	0.023	—	31.08
FedCA	20.92	0.087	—	21.79
FedPolicy	20.92	0.011	0.021	20.952

C.4.1 Computational Overhead

Table 11 reports the per-round wall-clock time on CIFAR-10 under β=0.1, decomposed into local training, server aggregation, and controller cost. The additional cost of the DRL controller is only 0.021 s per round, which amounts to approximately 0.10% of the total runtime and about 2.2 s over the full 105-round training budget. As a result, the overall per-round time of FedPolicy is 20.95 s, essentially matching FedAvg-style training (20.91 s) and remaining well below more expensive baselines such as FedCA (21.79 s), FedFA (27.47 s), and CCVR (31.08 s). This indicates that the performance gain of FedPolicy is not achieved at the expense of noticeable computational overhead.

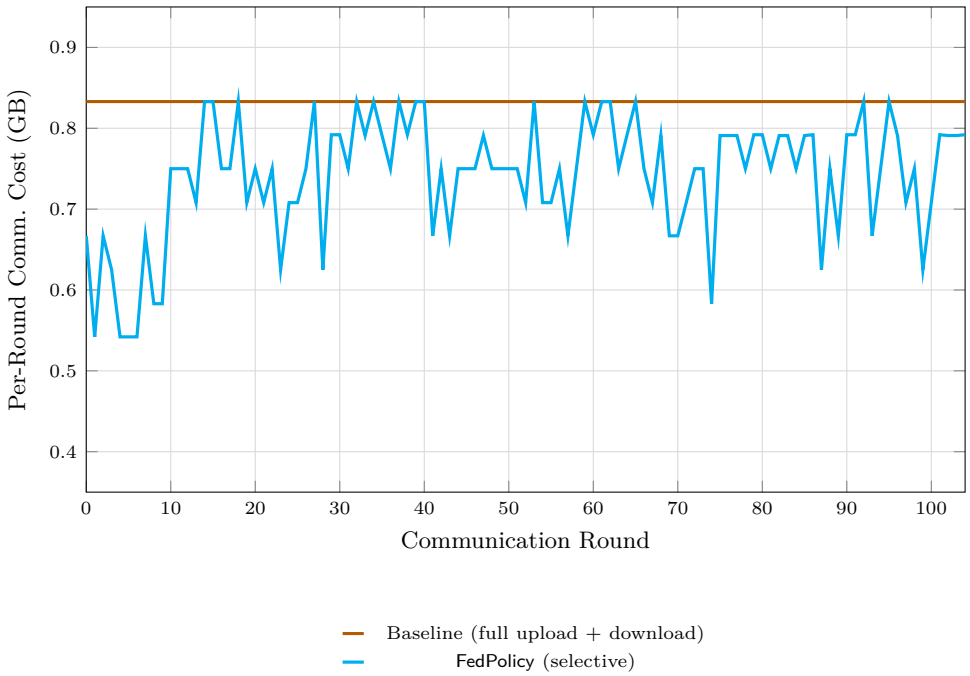

Figure 10: CIFAR-10 (β=0.1): per-round communication cost. Baselines incur a constant 0.833 GB/round because each selected client uploads and downloads the full model in every round; FedPolicy remains lower in early rounds and approaches baseline as Full updates become more frequent.

C.4.2 Communication Efficiency

We next examine communication accounting under selective broadcast. In baseline methods, each of the C_r=10 selected clients uploads and downloads the full model in every round, yielding a total bidirectional cost of

$$C_r \times 2 \times |\Theta| = 10 \times 2 \times 42.66 \text{ MB} \approx 0.833 \text{ GB/round}.$$

Over T=105 rounds, this corresponds to approximately 87.5 GB of total communication. In FedPolicy, uploads remain full-model, while downloads are selective (Full/Backbone/Head). Therefore, per-round savings come from reduced downlink payload, and cumulative savings additionally reflect fewer rounds needed to reach a target accuracy.

Figure 10 shows that FedPolicy remains below the baseline cost in the early stage of training, where Head and Backbone updates are selected more frequently, and then gradually approaches the full-model baseline as Full updates become more common. In practice, the per-round cost of FedPolicy starts at roughly 0.54–0.67 GB and increases later in training as the policy shifts toward broader synchronization. This behavior is consistent with the learned action dynamics: communication is reduced precisely in the regime where selective sharing is used most aggressively.

D Discussion

D.1 Key Observations

- The main contribution of FedPolicy lies in improving the match between global redistribution and client-specific learning conditions. Its advantage is modest when heterogeneity is mild, but becomes increasingly pronounced as client distributions grow more skewed, where uniform rebroadcast becomes less appropriate.
- The learned policy behavior is consistent with this trend. As heterogeneity increases, the controller places progressively greater emphasis on Backbone sharing, suggesting that feature-level transfer remains broadly useful across clients, whereas classifier-level parameters are more tightly coupled to local label structure. The ablations support the same interpretation: parameter divergence is the most informative state signal, and combining local and global reward terms yields the most stable behavior.
- The advantage of FedPolicy cannot be explained solely by the use of C3E. Although C3E improves all compared methods, FedPolicy remains superior even when the same loss is applied to the baselines, and the same pattern is already visible under standard CE. This shows that the gain arises from the proposed adaptive post-aggregation sharing mechanism in addition to stronger local optimization.
- The comparison with static sharing rules further indicates that the improvement comes from learning the redistribution policy itself, rather than from any fixed preference for Full, Backbone, or Head updates. No single static heuristic performs best across all heterogeneity regimes, whereas learned policies adapt their behavior according to client state and data conditions.
- The performance gain is achieved with negligible practical overhead. The controller adds only a very small fraction to the total per-round runtime, while sharing selective parameter blocks to clients reduces communication cost, leading to a more favorable cost-to-accuracy trade-off.

D.2 Privacy Considerations

FedPolicy does not require raw data sharing, but it should not be viewed as providing any additional privacy guarantee beyond the standard federated setting. The controller operates on model updates together with auxiliary client-side statistics such as soft confusion information and local accuracy. Although these signals are compact, they are still derived from private local data and may reveal information about class composition or client-specific difficulty. The present results therefore establish utility, not formal privacy protection.

D.3 Limitations

The present evaluation is limited to image classification, with LeNet-5 on FMNIST and a backbone—head decomposition based primarily on ResNet-18 for CIFAR experiments. While the general idea is not tied to a specific architecture, its usefulness depends on whether a meaningful separation exists between transferable representation parameters and more client-specific prediction layers. The current action space is also deliberately coarse, consisting only of Full, Backbone, and Head updates. This makes the controller simple and interpretable, but it also limits the granularity of adaptation. Finally, the experiments focus on statistical heterogeneity under synchronous training, and do not address other deployment factors such as client dropouts, asynchronous participation, or time-varying data distributions.