# 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, where uniform post-aggregation broadcast can overwrite client-specific structure and cause negative transfer. Existing approaches primarily address heterogeneity through local objectives, server aggregation, or personalization, while clients typically receive the same parameter block after aggregation. 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 CIFAR-10, CIFAR-100, and ISIC2019, with the clearest gains appearing in the more challenging heterogeneous regimes. Across the settings in the main comparison, FedPolicy achieves an average relative gain of approximately 7.30% over the strongest baseline, with the largest improvement reaching 16.47% on ISIC2019 under severe heterogeneity; the supplementary FMNIST and UCI HAR study shows the same direction of improvement. It also converges faster and improves the cost-to-accuracy trade-off, while adding only 0.021 seconds of controller computation per round. These results identify client-specific post-aggregation redistribution as an effective and underexplored 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) commonly applies an aggregated global model *uniformly* to participating clients. Clients differ substantially in data distribution, representation quality, and class-level learning difficulty. Under heterogeneity, uniformly overwriting local models with aggregated parameters can induce negative transfer, destabilize training, and degrade subsequent aggregation. Despite substantial progress, *how aggregated parameters should be applied to heterogeneous clients* remains underexplored.

Intuitively, clients with stable, 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. These effects compound across rounds: post-aggregation initialization determines each client's local trajectory before it returns an update. A mismatch may produce noisier updates misaligned with local data geometry, degrading the next aggregate and propagating instability. By contrast, client-specific redistribution can preserve useful local structure while still exploiting transferable global progress. This trade-off is especially important under severe non-IID distributions.

In standard FL, 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



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

methods address heterogeneity through *local objectives* and *server aggregation*. 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 robustly combine heterogeneous client updates at the server, while client-adaptive methods Arivazhagan et al. (2019); Liu et al. (2024); Grinwald et al. (2024) and prototype- or representation-learning approaches Tan et al. (2022); Collins et al. (2021); Yang et al. (2024) tailor learned representations to client-specific distributions. FedCA Chowdhury & Halder (2026) further combines confusion-aware local optimization with class-prioritized aggregation under extreme non-IID settings; Appendix C provides additional discussion.

Yet these methods retain a common assumption: the global model is redistributed identically to all participating clients. Although prior work has substantially improved both *how clients are trained locally* and *how their updates are aggregated globally*, it 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.

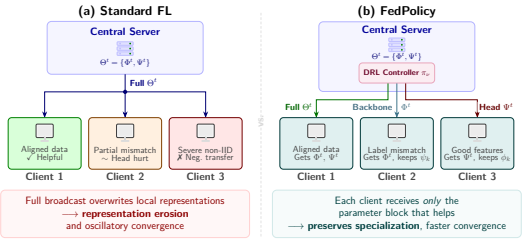



Figure 1: **Selective parameter broadcast under heterogeneity. (a)** Uniform full-model broadcast can induce representation drift and negative transfer when client data are non-IID. **(b)** FedPolicy uses a DRL policy to choose a client-specific broadcast action among full model ($\Theta^t$), backbone-only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

In this work, we argue that effective federated learning under heterogeneity requires not only better local objectives or aggregation operators, but also explicit control over *how the aggregated model is transferred back to individual clients*. This step is sequential and non-stationary: each initialization decision changes a client's local trajectory and thus the updates shaping the next aggregate, while participating clients, class composition, and model states vary across rounds. Fixed heuristics may therefore not remain effective throughout training, naturally motivating 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. After the server forms the global model each round, FedPolicy selects whether each participating client receives the full model, only the backbone, or only the classifier head, while leaving local optimization and server-side aggregation unchanged. This selective-initialization perspective is illustrated in Fig. 1. The overall contributions are summarized as follows:

- We introduce FedPolicy, a **Policy-Guided Selective Global Redistribution** framework that formulates client-specific post-aggregation redistribution as a sequential decision-making problem and adaptively selects Full, Backbone, or Head redistribution for each selected client.
- We design a composite client state that captures **class-level confusion structure** through soft confusion matrices and **optimization status** through client–global parameter divergence and local/global validation levels, providing a compact signal for heterogeneity-aware redistribution control.
- We evaluate FedPolicy on CIFAR-10, CIFAR-100, and ISIC2019 under heterogeneous federated settings against standard, personalized, confusion-aware, and partial-sharing baselines, with controller, state, reward, dropout, and aggregation-compatibility ablations.

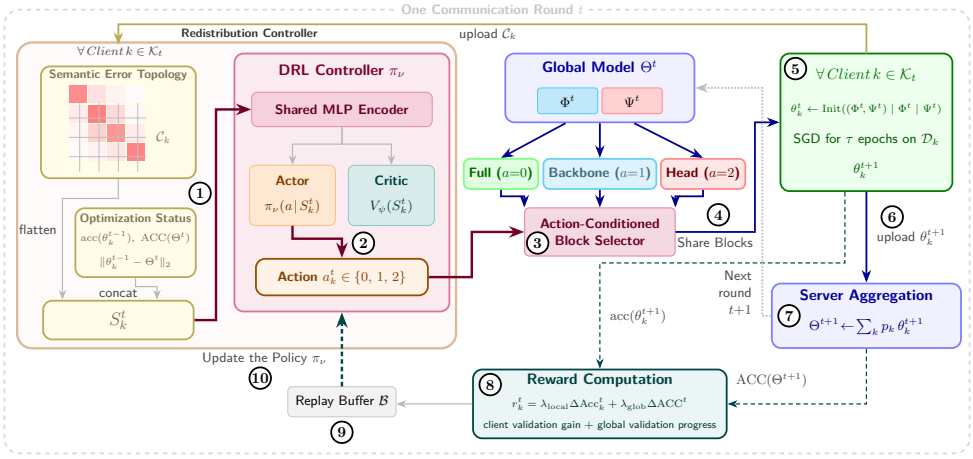



Figure 2: Overview of the FedPolicy training workflow.

## 2 METHODOLOGY

This section presents FedPolicy, a policy-guided framework for client-specific post-aggregation parameter sharing under statistical heterogeneity. Figure 2 illustrates one training round. The redistribution controller converts the normally fixed broadcast into a client-specific decision. For each selected client, the server constructs a state and selects full-model, backbone-only, or head-only sharing (Steps 1–2), then transmits the corresponding parameter block (Steps 3–4). The client incorporates this block, trains locally, and returns its updated model and confusion statistics (Steps 5–6). After mean aggregation, the server combines progress on a small held-out split with client-side validation improvement, stores the resulting transition, and updates the controller (Steps 7–10). The held-out split supplies controller feedback only; it does not alter aggregation or local optimization.

Thus, FedPolicy confines adaptivity to post-aggregation redistribution while retaining standard local optimization and server aggregation. Table 1 summarizes the notation.

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 $t$ |
| $\rho$ | EMA coefficient for soft confusion tracking |
| $C_k^t$ | Soft confusion matrix of client $k$ after round $t$ |
| $\mathcal{B}$ | 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 |
| $\Delta\text{Acc}_k^t$ | Local validation improvement of client $k$ |
| $\Delta\text{ACC}^t$ | Server-side held-out validation improvement after round $t$ |
| $\text{vec}(\cdot)$ | Vectorization operator |
| $\eta$ | Local learning rate |
| $\gamma$ | DRL discount factor |
| $\mathcal{J}(\Theta)$ | Global objective |
| $\mathcal{K}_t$ | Set of participating clients at round $t$ |
| $\lambda_{\text{local}}$ | Weight of the local reward term |
| $\lambda_{\text{glob}}$ | Weight of the global reward term |
| $\mathcal{L}_k$ | Empirical risk on client $k$ |
| $\Phi$ | Backbone (representation encoder) parameters |
| $\pi_\nu$ | DRL policy with parameters $\nu$ |
| $p_k$ | Population weight of client $k$ in the global objective |
| $w_k^t$ | Aggregation weight of selected client $k$ at round $t$ |
| $r_k^t$ | Reward of client $k$ at round $t$ |
| $\mathcal{S}$ | State space |
| $S_k^t$ | State of client $k$ at round $t$ |
| $\tau$ | 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 $t$ |
| $\Psi$ | Classifier head parameters |

### 2.1 PROBLEM FORMULATION

Given $N$ clients with local empirical risks $\mathcal{L}_k$ and population weights $p_k \geq 0$ satisfying $\sum_k p_k = 1$, conventional FL learns a single global model by minimizing

$$\min_{\Theta} \mathcal{J}(\Theta) = \min_{\Theta} \sum_{k=1}^{N} p_k \mathcal{L}_k(\Theta), \tag{1}$$

and uniformly broadcasts the aggregate at the next round. In our experiments, $p_k = 1/N$ defines the equal-client population objective. Because clients are sampled uniformly, the server estimates this objective in each round using the selected-client mean, with $w_k^t = 1/|\mathcal{K}_t|$ in Eq. (2). Thus, $p_k$ specifies the target objective, whereas $w_k^t$ specifies the realized aggregation among participating clients. FedPolicy leaves both choices unchanged and modifies only the post-aggregation redistribution step.

We decompose $\Theta = \{\Phi, \Psi\}$ into a representation encoder (backbone) $\Phi$, which captures transferable features, and a classifier head $\Psi$, which is more sensitive to local label composition.

At round $t$, the server holds $\Theta^t = \{\Phi^t, \Psi^t\}$, selects clients $\mathcal{K}_t$, and chooses an action $a_k^t$ for each $k \in \mathcal{K}_t$, yielding the transmitted block $\zeta_k^t = \mathrm{Share}(\Theta^t, a_k^t)$: the full model, backbone, or head. The client combines this block with its most recently stored model $\theta_k^{t-1}$ to initialize $\theta_{k,0}^t$ and performs $\tau$ local epochs, $\theta_{k,e+1}^t = \theta_{k,e}^t - \eta g_k(\theta_{k,e}^t)$ for $e = 0, \ldots, \tau - 1$, where $g_k$ is the stochastic local gradient; it returns $\theta_k^{t+1} := \theta_{k,\tau}^t$.

The server then applies mean (FedAvg-style) aggregation,

$$\Theta^{t+1} \leftarrow \sum_{k \in \mathcal{K}_t} w_k^t \theta_k^{t+1}, \tag{2}$$

with $w_k^t = 1/|\mathcal{K}_t|$ for $k \in \mathcal{K}_t$. We distinguish these round-specific mean-aggregation weights from the population weights $p_k$ in Eq. (1).

We learn a policy $\pi_\nu$ that selects $a_k^t$ to improve client adaptation and future global aggregation over the communication horizon by solving

$$\min_{\pi_\nu} \mathbb{E}\big[\mathcal{J}(\Theta^T)\big] = \min_{\pi_\nu} \mathbb{E}\left[\sum_{k=1}^{N} p_k \mathcal{L}_k(\Theta^T)\right]. \tag{3}$$

This turns post-aggregation sharing from a fixed operation into a sequential control problem.

### 2.2 Fixed Local Training and Aggregation Pipeline

FedPolicy preserves the standard client–server optimization loop: participating clients minimize local empirical risk, and the server aggregates their updates. It intervenes only when the aggregate is redistributed for the next round. For a selected client $k$, the local objective is

$$\mathcal{L}_k(\theta_k^t) = \frac{1}{|\mathcal{D}_k|} \sum_{(x,y) \in \mathcal{D}_k} \ell\Big(f_{\theta_k^t}(x), y\Big), \tag{4}$$

where $\ell(\cdot, \cdot)$ is the task loss and $f_{\theta_k^t}$ denotes the model at its selectively initialized starting point (equivalently, $\theta_{k,0}^t$ in the local-iterate notation above). Selective sharing changes the starting point of local optimization, not which parameters are trained. After $\tau$ epochs, the client returns $\theta_k^{t+1}$, its updated soft confusion statistics $C_k^t$ for the next state, and validation accuracy $\mathrm{acc}(\theta_k^{t+1})$ for policy learning.

The server updates the global model using Eq. (2). This fixed aggregation rule is deliberate: whereas heterogeneity-aware methods such as FedCA adapt local objectives and aggregation-side coordination, FedPolicy targets how an already aggregated model is returned to heterogeneous clients. It therefore isolates redistribution as the control point and separates its gains from aggregation complexity.

### 2.3 RL-Guided Post-Aggregation Redistribution Controller

We cast post-aggregation redistribution as a sequential decision problem at the server. Unlike RL-based FL methods that control client selection or aggregation weights Wang et al. (2020a), FedPolicy selects which block of an already aggregated model each client receives. Its novelty therefore lies in the controlled stage, not in the use of RL alone. At round $t$, the server observes the collection of selected-client states and samples the joint action $A^t = (a_k^t)_{k \in \mathcal{K}_t}$ using a shared, client-conditioned policy. Each action determines $\zeta_k^t$ and the client's effective initialization, while all returned updates jointly determine $\Theta^{t+1}$. Thus, the environment transition is federation-level rather than an independent transition for each client; the factorized per-client policy is a tractable centralized-control approximation. Accordingly, we model the controller as a DRL policy

$$\pi_\nu(a_k^t \mid S_k^t), \tag{5}$$

where $S_k^t$ is the observation for client $k$ and $\nu$ denotes shared controller parameters. Writing $S^t = (S_k^t)_{k\in\mathcal{K}_t}$, the full environment state also includes the global model and stored states of nonparticipating clients, but the implemented policy conditions on the selected-client observations and factorizes as $\pi_\nu(A^t \mid S^t) = \prod_{k\in\mathcal{K}_t} \pi_\nu(a_k^t \mid S_k^t)$. We therefore use per-client tuples $(S_k^t, a_k^t, r_k^t, S_k^{t+1})$ for learning while retaining the coupling induced by joint aggregation in the reward and next observations.

The controller maximizes the expected discounted mean reward over the communication horizon:

$$\max_{\nu} \mathbb{E}_{\pi_\nu}\left[\sum_{t=0}^{T-1} \gamma^t \bar{r}^t\right], \tag{6}$$

where $\bar{r}^t = |\mathcal{K}_t|^{-1} \sum_{k\in\mathcal{K}_t} r_k^t$ and $\gamma \in [0, 1)$ 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. We therefore use a row-normalized soft confusion matrix $C_k^t \in \mathbb{R}^{C\times C}$. For labeled examples processed by client $k$ in round $t$, $\widehat{C}_{k,ij}^t$ is the mean predicted probability assigned to class $j$ among examples with label $i$; absent rows retain their previous estimate. We smooth this statistic as $C_k^t = \rho C_k^{t-1} + (1-\rho)\widehat{C}_k^t$, where $\rho \in [0, 1)$, and use $\mathrm{vec}(C_k^{t-1})$ as the round-$t$ state signal. This supplies class-level confusion structure without transmitting examples or hard prediction lists.

#### 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 of the client's most recently stored model, $\mathrm{acc}(\theta_k^{t-1})$, 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 indicates that the client has evolved along a trajectory substantially different from the global consensus. We use this quantity as a state signal for the controller, not as evidence that selective redistribution is always preferable to a full-model overwrite.

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

$$S_k^t = \left[\mathrm{vec}(C_k^{t-1}) \oplus \mathrm{acc}(\theta_k^{t-1}) \oplus \mathrm{ACC}(\Theta^t) \oplus d_k^t\right], \tag{7}$$

where $\oplus$ denotes vector concatenation.

### 2.5 Parameter Block Selection

Given client state $S_k^t$, the controller selects $a_k^t \in \mathcal{A} = \{0, 1, 2\}$, which determines the transmitted global block $\zeta_k^t$ and effective local model $\theta_k^t$. Specifically, (i) **Full** ($a_k^t = 0$) sends $\zeta_k^t = \{\Phi^t, \Psi^t\}$ and sets $\theta_k^t = \{\Phi^t, \Psi^t\}$, recovering standard FL; (ii) **Backbone** ($a_k^t = 1$) sends $\zeta_k^t = \Phi^t$ and sets $\theta_k^t = \{\Phi^t, \psi_k^{t-1}\}$, sharing representations while preserving the local decision boundary; and (iii) **Head** ($a_k^t = 2$) sends $\zeta_k^t = \Psi^t$ and sets $\theta_k^t = \{\phi_k^{t-1}, \Psi^t\}$, preserving local features while aligning the classifier globally. This compact action space spans full-model transfer, representation transfer, and classifier alignment.

The action affects only post-aggregation redistribution; local training and aggregation remain unchanged. FedPart Wang et al. (2024) instead follows a predefined layer-wise schedule for training and aggregation, whereas FedPolicy learns which block to return to each client in each round. Unlike personalized FL with persistent private components or predefined shared/private partitions, FedPolicy adds neither a permanent personalized component nor a separate personalization objective; a client's action may change as its state evolves. Thus, the contribution is learned control of post-aggregation transfer, not partial parameter exchange alone.

These actions expose an optimization trade-off. Under block-wise regularity and bounded gradient drift, retaining a local head or backbone can lower immediate loss relative to Full overwrite when the corresponding client–global discrepancy exceeds an explicit threshold. Yet retained parameters introduce mismatch into subsequent aggregation: a one-step stationarity bound yields a vanishing average gradient when the time-averaged mismatch vanishes and otherwise characterizes an error neighborhood. This motivates client- and round-specific selection rather than uniform Full or partial redistribution. These results characterize prescribed actions, not optimality of the learned policy. Appendix E provides the assumptions, theorem statements, and proofs.

### 2.6 Policy Learning and Reward Design

We implement the centralized sequential-control problem with an MDP-style, factorized policy. The client observation space is defined by Eq. (7), $\mathcal{A}$ contains the three redistribution actions, the environment transition is induced jointly by local training and aggregation, and $\gamma \in [0, 1)$ is the discount factor. This formulation does not assume that an individual $S_k^t$ is Markov in isolation.

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

$$r_k^t = \lambda_{\text{local}} \Delta \text{Acc}_k^t + \lambda_{\text{glob}} \Big( \text{ACC}(\Theta^{t+1}) - \text{ACC}(\Theta^t) \Big), \tag{8}$$

Here, $\Delta \text{Acc}_k^t = \text{acc}(\theta_k^{t+1}) - \text{acc}(\theta_k^{t-1})$ and $\Delta \text{ACC}^t = \text{ACC}(\Theta^{t+1}) - \text{ACC}(\Theta^t)$, with $\lambda_{\text{local}}$ and $\lambda_{\text{glob}}$ weighting their contributions. The first difference uses client-held validation data, whereas the second uses the server-held validation split. The models $\theta_k^{t-1}$ and $\theta_k^{t+1}$ denote the client's pre-broadcast model and its model after selective initialization and local training, respectively; thus, $\Delta \text{Acc}_k^t$ measures improvement over the round-$t$ transition. The global increment is shared by all clients selected in round $t$: it supplies federation-level credit but is not a marginal or causal estimate of client $k$'s action, because all selected-client actions jointly determine $\Theta^{t+1}$. The mixed reward therefore balances client adaptation and aggregate progress without claiming exact per-action credit assignment or guaranteed avoidance of destabilizing actions. Appendix F.5.3 reports sensitivity to both weights. Test labels are never used for state construction, reward computation, policy updates, or hyperparameter tuning; peak test accuracy is reported only as a retrospective summary of each fixed-budget trajectory.

The controller maximizes the expected discounted return in Eq. (6). Since the DRL module operates entirely at the server, clients do not perform policy-learning computation locally; they only receive the selected parameter block and follow the standard federated training routine.

After fixed mean aggregation, the server computes the reward and next state, stores the transition, and updates the controller from replay. Keeping local optimization and aggregation fixed isolates learning to redistribution decisions. This cycle repeats until convergence or the communication budget is exhausted. The reader may refer to Algorithm 1 in Appendix D for the complete learning procedure.

## 3 Experimental Evaluation

We evaluate FedPolicy under probabilistic label-skew and deterministic class-absence partitions. We ask whether client-specific redistribution improves accuracy and convergence, whether learning the policy adds value beyond static heuristics and the local loss, and whether the gains justify their computational and communication costs. Baselines retain their defining mechanisms and tuned settings unless stated otherwise. FedPolicy uses mean (FedAvg-style) aggregation, placing all adap-

tivity in redistribution so that aggregation changes do not confound the comparison. Appendix F provides the implementation, training, and partition details.

Table 2: Top-1 accuracy (%) on CIFAR-10, CIFAR-100, and ISIC2019 under size-varying Dirichlet label skew. Lower $\beta$ indicates stronger heterogeneity. Each experiment was conducted using three independent random seeds (0, 1, and 2), and the reported values denote the mean $\pm$ standard deviation of the peak global test accuracy across these runs. Each method uses its configured main-benchmark local objective; this table is therefore distinct from the controlled CE/C3E reruns in Table 3. The best result in each column is highlighted in **bold**.

| Method | CIFAR-10 | | | CIFAR-100 | | | ISIC2019 | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\beta$=0.3 | $\beta$=0.1 | $\beta$=0.05 | $\beta$=0.3 | $\beta$=0.1 | $\beta$=0.05 | $\beta$=0.3 | $\beta$=0.1 | $\beta$=0.05 |
| FedAvg | 64.76±0.61 | 31.25±1.14 | 16.11±1.52 | 49.83±0.67 | 43.92±0.95 | 37.22±1.38 | 55.82±0.86 | 49.24±1.09 | 48.68±1.21 |
| FedProx | 64.82±0.58 | 33.79±1.09 | 16.31±1.47 | 49.87±0.64 | 43.97±0.91 | 37.23±1.35 | 57.67±0.64 | 49.68±1.16 | 49.16±1.43 |
| FedNTD | 69.20±0.62 | 44.08±1.15 | 21.52±1.54 | 50.19±0.58 | 44.93±1.17 | 38.34±1.23 | 63.73±0.79 | 50.13±0.96 | 48.85±1.29 |
| Ditto | 65.00±0.67 | 31.25±1.21 | 17.18±1.61 | 50.00±0.68 | 37.64±1.08 | 28.96±1.52 | 61.83±0.82 | 52.36±1.09 | 50.31±1.27 |
| CCVR | 67.31±0.59 | 38.57±1.12 | 15.78±1.64 | 48.95±0.73 | 42.68±0.98 | 36.41±1.41 | 62.48±0.75 | 53.23±0.99 | 51.42±1.28 |
| FedSAM | 64.13±0.79 | 27.25±1.28 | 14.66±1.71 | 46.98±0.78 | 38.67±1.15 | 32.15±1.58 | 59.74±0.91 | 50.23±1.02 | 48.37±1.36 |
| FedFA | 64.51±0.72 | 28.31±1.25 | 17.35±1.55 | 49.27±0.75 | 37.78±1.11 | 31.74±1.53 | 65.33±0.59 | 56.90±1.13 | 52.73±1.17 |
| FedPart | 66.59±0.67 | 27.89±1.47 | 13.93±1.61 | 45.97±0.68 | 40.46±1.08 | 29.40±1.52 | 53.70±0.66 | 48.71±1.19 | 47.52±1.36 |
| FedCA | 73.46±0.52 | 59.01±1.48 | 51.62±1.76 | 50.65±0.78 | 45.62±0.98 | 39.41±1.29 | 66.92±0.31 | 58.18±0.84 | 54.51±0.91 |
| **FedPolicy** | **74.28±0.52** | **62.84±1.13** | **55.25±1.38** | **51.83±0.74** | **48.12±1.13** | **42.72±1.34** | **68.23±0.64** | **67.71±0.47** | **63.49±0.98** |

### 3.1 Performance under Heterogeneous Data

We first measure top-1 accuracy and convergence under heterogeneous partitions, then test whether the gains are consistent with the proposed redistribution mechanism.

#### 3.1.1 Accuracy under Label Heterogeneity

Table 2 compares FedPolicy with nine baselines on CIFAR-10, CIFAR-100, and ISIC2019 under size-varying Dirichlet label skew $\beta \in \{0.3, 0.1, 0.05\}$. These benchmarks span natural and medical images with 10–100 classes; Table 10 in Appendix F jointly reports FMNIST and the non-image UCI HAR scope study. FedPolicy performs best in all nine main-table settings, with margins that generally widen as heterogeneity increases. From $\beta$=0.3 to 0.05, its relative gain over the strongest baseline grows from $+1.12\%$ to $+7.03\%$ on CIFAR-10 and from $+2.33\%$ to $+8.40\%$ on CIFAR-100.

Table 3: Controlled local-loss study on CIFAR-10 under size-varying Dirichlet label skew (seed 0). Entries report peak Top-1 accuracy (%) when every listed method is rerun with CE or C3E. These single-seed values are not the three-seed estimates in Table 2. Best result in each column is shown in **bold**.

| | $\beta$=0.3 | | $\beta$=0.1 | | $\beta$=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 |
| FedPart | 66.59 | 72.94 | 23.08 | 53.21 | 13.93 | 22.86 |
| 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** |

On ISIC2019, FedPolicy obtains $68.23\%$, $67.71\%$, and $63.49\%$ at $\beta$=0.3, 0.1, and 0.05, versus $66.92\%$, $58.18\%$, and $54.51\%$ for the strongest baseline, FedCA. The corresponding gains are $+1.31$, $+9.53$, and $+8.98$ points. FedFA remains competitive at $\beta$=0.3 and 0.05, whereas FedPart trails the strongest methods. These Top-1 results show the same larger advantage under stronger heterogeneity, but do not by themselves establish class-balanced or clinical performance on this imbalanced dataset.

Table 3 controls for the client objective. FedPolicy remains strongest under both CE and C3E at every heterogeneity level: relative to the best baseline, its gains are $+3.07\%$, $+10.42\%$, and $+19.30\%$ under CE and $+0.68\%$, $+6.49\%$, and $+7.03\%$ under C3E for $\beta \in \{0.3, 0.1, 0.05\}$. Thus, the gains persist without specialized aggregation and are not explained by the stronger local loss alone; adaptive redistribution contributes most under severe skew.

Appendix F.3 further evaluates architectural portability on UCI HAR using a transformer encoder over sensor features. In the combined table, the HAR FedPolicy$^{\dagger}$ row denotes selective redistribu-

tion with FedACG aggregation and C3E; it leads at $\beta$=0.1 and 0.05. We interpret this as a non-image scope check, not evidence of universal superiority.

#### 3.1.2 Convergence Behavior

Table 4 and Figure 3 evaluate convergence rather than endpoint accuracy alone. At $\beta$=0.3, FedPolicy reaches 40%, 60%, and 70% at rounds 3, 11, and 31, respectively, versus 5, 16, and 46 for FedCA; FedPart requires rounds 19 and 71 for the first two thresholds and never reaches 70%. Under the harder $\beta$=0.1 setting, FedPolicy separates from FedCA within roughly 15–20 rounds and retains its advantage, while FedPart improves later but remains below it. The gain therefore persists across the training trajectory.

Table 4: Convergence on CIFAR-10 ($\beta$=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 | × |
| FedPart | 1 | 19 | 71 | × |
| FedCA | 1 | 5 | 16 | 46 |
| **FedPolicy** | **1** | **3** | **11** | **31** |

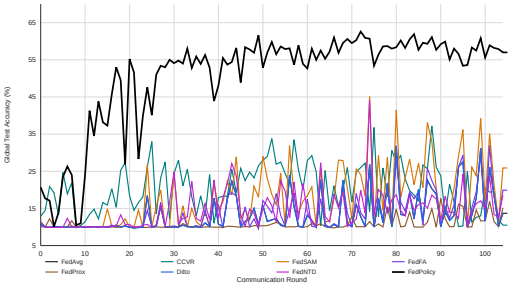



Figure 3: Global test accuracy over communication rounds on CIFAR-10 under size-varying Dirichlet label skew ($\beta$=0.1). The curve is shown for retrospective evaluation only and is not used by the controller.

### 3.2 DRL Policy Behavior under Data Heterogeneity

The extended controller analysis in Appendix F.4 shows that policy specialization becomes more pronounced as heterogeneity increases: effective actor–critic controllers favor Backbone transfer under severe skew, whereas DQN remains comparatively diffuse. This behavior is consistent with the learned-versus-static comparison below and indicates that performance depends on learning a structured redistribution policy rather than merely introducing a controller.

### 3.3 Ablation Studies

We next test whether the gains require a learned policy and examine the state, reward, and local-loss components.

#### 3.3.1 Learned versus Static Sharing Strategies

Table 5 compares five DRL controllers, four static heuristics, and a contextual bandit under matched architecture, partitions, budget, and C3E loss. Full-Only is therefore the matched uniform control, not the CE-based FedAvg entry in Table 2; the bandit uses the same state and actions but optimizes only immediate reward.

Learned controllers become most useful as heterogeneity strengthens. Under fixed-class partitions, TRPO leads at fc=6 and 4, while DQN reaches 55.35% at fc=2, 2.54 points above the best static rule. Under Dirichlet skew, Backbone-Only is slightly better at $\beta$=0.3 (74.75% vs. 74.28%), but learned policies lead by 5.33 points at $\beta$=0.1 and 11.96 at $\beta$=0.05; all five DRL methods beat the best static rule in the latter regime. The contextual bandit trails both static and sequential alterna-

Table 5: Single-seed controller ablation on CIFAR-10 with C3E loss. **Left:** deterministic fixed-class-support partitions by classes/client. **Right:** size-varying Dirichlet label-skew partitions by concentration $\beta$. Full-Only denotes uniform full-model redistribution. Column-best results are in **bold**.

| | *Fixed-Class (Deterministic)* | | | *Dirichlet (Probabilistic)* | | |
|---|---|---|---|---|---|---|
| Controller | fc=6 | fc=4 | fc=2 | $\beta$=0.3 | $\beta$=0.1 | $\beta$=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 |
| *Contextual Bandit Baseline* | | | | | | |
| Contextual Bandit | 56.17 | 53.51 | 42.44 | 50.84 | 44.06 | 39.32 |
| *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 |

tives, consistent with delayed effects through subsequent updates and aggregation. Controller complexity is therefore most justified under moderate-to-severe heterogeneity; these results support state-dependent redistribution but do not claim policy optimality.

### 3.4 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., $74.27\% \rightarrow 71.56\%$ at $\beta$=0.3, and $62.59\% \rightarrow 60.44\%$ at $\beta$=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 $\beta$=0.3, NCF is best (74.27%); at $\beta$=0.1, the full state is best (62.59%); and at $\beta$=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 $\beta$=0.3 and $\beta$=0.1 (73.03%, 61.84%), and close to the best at $\beta$=0.05 (vs. 51.94% for Global-only). Overall, balancing client-level and global-progress signals yields the most reliable behavior across settings.

Table 6: State and reward ablation for FedPolicy on CIFAR-10 across heterogeneity levels ($\beta$). The full state (All) includes $\vec{C}_k$, $\mathrm{acc}_k$, ACC, and $\|\Delta\theta\|$. Best results per column are in **bold**.

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

### 3.5 Practical Efficiency

We finally assess whether FedPolicy's gains incur meaningful systems cost.

#### 3.5.1 Cost-to-Accuracy Trade-off

Table 7 jointly measures convergence and communication. FedPolicy reaches 20%, 30%, 40%, and 50% in 1, 11, 11, and 23 rounds and peaks at 63.5%. At 50%, it uses 12.1 GB versus FedCA's 72.5 GB; FedPart uses 11.2 GB but requires 88 rounds and peaks at 53.2%. Thus, FedPolicy provides the strongest joint accuracy–convergence trade-off, not the lowest communication at every operating point. Appendix F.6.1 gives complete computation accounting.

Table 7: Cost to accuracy on CIFAR-10 ($\beta$=0.1): first round and cumulative communication (GB) to each threshold. "×": not reached within 105 rounds. Rounds/communication: lower is better; peak accuracy: higher is better.

| Method | Round | | | | Communication (GB) | | | | Best (%) |
|---|---|---|---|---|---|---|---|---|---|
| | 20% | 30% | 40% | 50% | 20% | 30% | 40% | 50% | |
| FedAvg | 43 | 80 | × | × | 36.7 | 67.5 | × | × | 31.9 |
| FedProx | 40 | 74 | × | × | 33.3 | 61.7 | × | × | 31.9 |
| FedSAM | 24 | 56 | 74 | × | 20.8 | 47.5 | 62.5 | × | 45.1 |
| FedFA | 68 | 101 | × | × | 57.5 | 85.0 | × | × | 31.9 |
| CCVR | 2 | 25 | × | × | 2.5 | 21.7 | × | × | 37.2 |
| FedNTD | 30 | 74 | 74 | × | 25.8 | 62.5 | 62.5 | × | 44.1 |
| Ditto | 43 | 80 | × | × | 36.7 | 67.5 | × | × | 31.9 |
| FedCA | 1 | 13 | 21 | 87 | 0.8 | 10.8 | 17.5 | 72.5 | 59.0 |
| FedPart | 18 | 52 | 65 | 88 | 4.3 | 10.2 | 10.3 | **11.2** | 53.2 |
| **FedPolicy** | **1** | **11** | **11** | **23** | **0.7** | **7.5** | **7.5** | 12.1 | **63.5** |

## 4 Conclusion

We introduced FedPolicy, which treats post-aggregation redistribution as a client- and round-specific decision among Full, Backbone, and Head transfer, while keeping local optimization and server aggregation unchanged. Across CIFAR-10, CIFAR-100, and ISIC2019, FedPolicy achieves the highest mean peak global accuracy in all nine main settings, with an average relative gain of 7.30% over the strongest baseline and a maximum gain of 16.47% under severe ISIC2019 heterogeneity. It also reaches target accuracies earlier and provides a favorable cost-to-accuracy trade-off, while adding only 0.021 s of controller computation per round. Controlled local-loss and learned-versus-static comparisons show that these gains cannot be attributed solely to a stronger client objective or a fixed partial-sharing rule. State and reward ablations further highlight local–global parameter divergence as an important control signal, while policy analysis shows that effective controllers increasingly favor Backbone transfer as heterogeneity becomes stronger. Robustness to client dropout and gains across multiple aggregation operators further support the stability of selective redistribution. Supplementary FMNIST and UCI HAR results provide additional evidence of architectural portability beyond the main benchmarks. Overall, FedPolicy identifies adaptive post-aggregation redistribution as a distinct and practical design dimension for heterogeneous federated learning; limitations and future directions are discussed in Appendix G.

## 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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186. Association for Computational Linguistics, 2019.

Jean Ogier du Terrail, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold, Tanguy Marchand, Othmane Marfoq, Erum Mushtaq, Boris Muzellec, Constantin Philippenko, Santiago Silva, Maria Teleńczuk, Shadi Albarqouni, Salman Avestimehr, Aurélien Bellet, Aymeric Dieuleveut, Martin Jaggi, Sai Praneeth Karimireddy, Marco Lorenzi, Giovanni Neglia, Marc Tommasi, and Mathieu Andreux. FLamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5315–5334, 2022.

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.

Geeho Kim, Jinkyu Kim, and Bohyung Han. Communication-efficient federated learning with accelerated client gradient. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12385–12394, 2024. doi: 10.1109/CVPR52733.2024.01177.

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.

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.

Renpu Liu, Cong Shen, and Jing Yang. Federated representation learning in the under-parameterized regime. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 31808–31843. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/liu24ba.html.

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*, 2024.

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.

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.

Tianfei Zhou and Ender Konukoglu. FedFA: Federated feature augmentation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=U9yFP90jU0.

# APPENDIX

This appendix provides supplementary material, including the reproducibility statement (Appendix A), disclosure on the use of large language models (Appendix B), related work (Appendix C), the complete FedPolicy training procedure (Appendix D), the theoretical analysis and proofs (Appendix E), additional experimental evaluation (Appendix F), and further discussion, privacy considerations, limitations, and future directions (Appendix G).

## A REPRODUCIBILITY STATEMENT

The anonymous repository mentioned in the abstract contains the implementation and configurations used in the experiments. Appendix D gives the complete training procedure, and Appendix F specifies the datasets, partition conventions, optimization settings, controller settings, random seeds, and evaluation budgets.

## B USE OF LARGE LANGUAGE MODELS

The technical content presented in this paper is entirely original and was conceived by the authors. Large language models were used only as supplementary tools for grammar correction, language polishing, and internal consistency checks during manuscript preparation. They were not used to formulate the research problem, design the method, conduct experiments, generate reported data, or determine scientific conclusions. The authors verified the technical content, citations, and numerical claims and take full responsibility for the manuscript.

## C 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. FedACG Kim et al. (2024) follows this direction by accelerating client gradients: the server broadcasts a lookahead global model and local training is regularized toward this overshot model, improving communication efficiency and stability under low participation. 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. (2024) 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. FedCA is a recent confusion-aware FL method in this family. It introduces C3E as a confusion-calibrated local objective that uses each client's recurring class-confusion patterns to emphasize difficult class pairs during local training. FedCA further couples this local objective with a class-prioritized server aggregation rule, so that aggregation gives additional attention to classes that are more difficult under the current heterogeneous training dynamics.

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 controls 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.

## D Overall FedPolicy Training Procedure

Algorithm 1 summarizes one communication round of FedPolicy with the default off-policy discrete SAC controller. The server first constructs a state for each selected client, samples a redistribution action from the policy, and transmits the corresponding full-model, backbone, or head parameters. Each client uses the received block to initialize its local model, performs standard local training, and returns the updated model and feedback statistics. After aggregating the returned models with the fixed mean rule, the server computes each client's reward and next state, stores the transition $(S_k^t, a_k^t, r_k^t, S_k^{t+1})$ in the replay buffer, and periodically updates the controller from sampled transitions. Thus, action selection precedes local training, whereas its reward is observed only after

all selected-client updates have jointly influenced the next aggregate. The PPO, A2C, and TRPO ablations replace the replay-buffer update with their native on-policy rollout updates.

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

**Require:** Global model $\Theta^t$, previous client models $\{\theta_k^{t-1}\}$, controller policy $\pi_\nu$, replay buffer $\mathcal{B}$
**Ensure:** Next global model $\Theta^{t+1}$, updated controller parameters $\nu$
**// 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 $\theta_k^{t-1}$ and $\Theta^t$ (Eq. 7)
  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 $\theta_k^t$ from $\theta_k^{t-1}$ using $\zeta_k^t$
**end for**
**// Phase 2: Local training**
**for** each selected client $k \in \mathcal{K}_t$ **in parallel do**
  Update $\theta_k^{t+1}$ via local training on $\mathcal{D}_k$
  Return $\theta_k^{t+1}$ and client-side feedback to the server
**end for**
**// Phase 3: Server aggregation**
Aggregate returned models using Eq. (2)
**for** each selected client $k \in \mathcal{K}_t$ **do**
  Compute reward $r_k^t$ using Eq. (8) 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 4: Controller update**
**if** update interval reached **then**
  Sample a minibatch from $\mathcal{B}$
  Update controller parameters $\nu$ using the chosen DRL objective
**end if**

# E Theoretical Analysis of Selective Initialization

This appendix provides a self-contained theoretical characterization of selective post-aggregation initialization. The analysis isolates the effect of a specified redistribution action while keeping server aggregation unchanged and abstracting away the policy-learning dynamics of the RL controller. It therefore does not establish controller convergence, policy optimality, systematic mismatch reduction, or an ordering between FedPolicy and Full redistribution.

## E.1 Proof Notation

Table 8 summarizes the symbols used in the proofs below.

## E.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** (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 $\mu$-strongly convex in $\boldsymbol{\psi}$ on $\mathcal{U}$;*

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, $\tilde{\mathbf{w}}_k^t - \mathbf{w}^t$ |
| $\bar{\boldsymbol{\delta}}^t$ | Weighted mean initialization difference, $\sum_k p_k \boldsymbol{\delta}_k^t$ |
| $D_t^2$ | Weighted initialization mismatch, $\sum_k p_k \|\boldsymbol{\delta}_k^t\|^2$ |
| $\mathcal{L}_k(\cdot)$ | Local objective of client $k$ |
| $\nabla \mathcal{L}_k(\cdot)$ | Gradient of the client-$k$ objective |
| $L$ | Smoothness constant of each $\mathcal{L}_k$ |
| $\mu$ | Block-wise strong convexity constant |
| $\epsilon_{\text{drift}}$ | 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 |
| $\boldsymbol{\phi}^t$ | 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 |

- *with the head fixed, the function*

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

*is $\mu$-strongly convex in $\boldsymbol{\phi}$ on $\mathcal{U}$.*

*Equivalently, for any differentiable $\mu$-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 3** (Block-gradient drift bound)**.** *The candidate partially updated models satisfy the following block-gradient bounds:*

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

The local comparison uses Assumptions 2 and 3. The stationarity characterization uses smoothness together with the explicitly stated one-step, full-participation update; it does not imply convergence of the learned policy.

### E.3 Block-wise Redistribution Comparison Bound

For client $k$ at round $t$, define the candidate initializations $\mathbf{w}_{\mathsf{Full}}^t = [\boldsymbol{\phi}^t; \boldsymbol{\psi}^t]$, $\mathbf{w}_{\mathsf{Back}}^t = [\boldsymbol{\phi}^t; \boldsymbol{\psi}_k^{t-1}]$, and $\mathbf{w}_{\mathsf{Head}}^t = [\boldsymbol{\phi}_k^{t-1}; \boldsymbol{\psi}^t]$.

**Theorem 1** (Block-wise redistribution comparison bound)**.** *Under Assumptions 2 and 3,*

$$\mathcal{L}_k(\mathbf{w}_{\mathit{Full}}^t) - \mathcal{L}_k(\mathbf{w}_{\mathit{Back}}^t) \geq \frac{\mu}{4}\|\boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1}\|^2 - \frac{\epsilon_{\text{drift}}^2}{\mu}, \tag{9}$$

$$\mathcal{L}_k(\mathbf{w}_{\mathit{Full}}^t) - \mathcal{L}_k(\mathbf{w}_{\mathit{Head}}^t) \geq \frac{\mu}{4}\|\boldsymbol{\phi}^t - \boldsymbol{\phi}_k^{t-1}\|^2 - \frac{\epsilon_{\text{drift}}^2}{\mu}. \tag{10}$$

*Consequently, Backbone redistribution has strictly lower client loss than Full overwrite whenever $\|\boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1}\| > 2\epsilon_{\text{drift}}/\mu$, and Head redistribution has strictly lower client loss whenever $\|\boldsymbol{\phi}^t - \boldsymbol{\phi}_k^{t-1}\| > 2\epsilon_{\text{drift}}/\mu$.*

*Proof.* We define the three candidate initialized models:

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

The proof compares Full separately with Back and Head, yielding action-specific comparison bounds.

**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 2, the function $h$ is $\mu$-strongly convex in $\boldsymbol{\psi}$ on $\mathcal{U}$. Therefore, for any $u, v \in \mathcal{U}$,

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

Choose

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

Then Eq. (11) gives

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

Using the definition of $h$,

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

and

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

Hence

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

By Assumption 3,

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

Using the Cauchy–Schwarz inequality,

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

Next we apply Young's inequality in the form

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

Setting

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

we obtain

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

Substituting this into Eq. (12), we get

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

Therefore,

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

**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 2, the function $g$ is $\mu$-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 $g$,

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

and

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

Hence

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

Again, by Assumption 3,

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

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

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

Substituting this into Eq. (14), we obtain

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

Equations (13) and (15) establish the desired block-wise comparison bounds. If $\|\boldsymbol{\psi}^t - \boldsymbol{\psi}_k^{t-1}\| > 2\epsilon_{\text{drift}}/\mu$, then the right-hand side of the lower bound for $\mathbf{w}_{\mathsf{Back}}^t$ is strictly positive, implying $\mathcal{L}_k(\mathbf{w}_{\mathsf{Back}}^t) < \mathcal{L}_k(\mathbf{w}_{\mathsf{Full}}^t)$. Similarly, if $\|\boldsymbol{\phi}^t - \boldsymbol{\phi}_k^{t-1}\| > 2\epsilon_{\text{drift}}/\mu$, then the right-hand side of the lower bound for $\mathbf{w}_{\mathsf{Head}}^t$ is strictly positive, implying $\mathcal{L}_k(\mathbf{w}_{\mathsf{Head}}^t) < \mathcal{L}_k(\mathbf{w}_{\mathsf{Full}}^t)$. This completes the proof.

### E.4 Stationarity Characterization under Selective-Initialization Error

**Theorem 2** (Stationarity characterization under selective-initialization error)**.** *Under Assumption 1, let $f(\mathbf{w}) = \sum_{k=1}^N p_k \mathcal{L}_k(\mathbf{w})$, where $p_k \geq 0$ and $\sum_k p_k = 1$. Consider the full-participation, one-exact-local-step abstraction*

$$\tilde{\mathbf{w}}_k^t = \mathbf{w}^t + \boldsymbol{\delta}_k^t, \qquad \mathbf{w}^{t+1} = \sum_{k=1}^N p_k \left(\tilde{\mathbf{w}}_k^t - \eta \nabla \mathcal{L}_k(\tilde{\mathbf{w}}_k^t)\right), \tag{16}$$

*with $0 < \eta \leq 1/L$. Define $D_t^2 := \sum_k p_k \|\boldsymbol{\delta}_k^t\|^2$ and $\bar{\boldsymbol{\delta}}^t := \sum_k p_k \boldsymbol{\delta}_k^t$. If $f$ is bounded below by $f^\star$, then*

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla f(\mathbf{w}^t)\|^2 \leq \frac{2(f(\mathbf{w}^0) - f^\star)}{\eta T} + \frac{2L^2}{T}\sum_{t=0}^{T-1} D_t^2 + \frac{2}{\eta^2 T}\sum_{t=0}^{T-1} \|\bar{\boldsymbol{\delta}}^t\|^2. \tag{17}$$

*Thus the average stationarity measure converges to zero when the two time-averaged mismatch terms on the right vanish; otherwise the result characterizes the corresponding error neighborhood. For Full overwrite, $\boldsymbol{\delta}_k^t = 0$ and both mismatch terms disappear.*

We next prove Theorem 2. The redistribution error is defined from the actual aggregated update, and the one-round descent inequality is derived explicitly.

*Proof of Theorem 2.* Because each $\mathcal{L}_k$ is $L$-smooth and the weights form a convex combination, $f = \sum_k p_k \mathcal{L}_k$ is also $L$-smooth. Define the gradient perturbation

$$\mathbf{e}_t := \sum_{k=1}^{N} p_k \left( \nabla \mathcal{L}_k(\tilde{\mathbf{w}}_k^t) - \nabla \mathcal{L}_k(\mathbf{w}^t) \right) - \frac{\bar{\boldsymbol{\delta}}^t}{\eta}. \tag{18}$$

Using $\tilde{\mathbf{w}}_k^t = \mathbf{w}^t + \boldsymbol{\delta}_k^t$ and $\sum_k p_k = 1$, Eq. (16) becomes exactly

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \bar{\boldsymbol{\delta}}^t - \eta \sum_k p_k \nabla \mathcal{L}_k(\tilde{\mathbf{w}}_k^t) \tag{19}$$

$$= \mathbf{w}^t - \eta \left( \nabla f(\mathbf{w}^t) + \mathbf{e}_t \right). \tag{20}$$

Thus $\mathbf{e}_t$ is not an assumed drift variable; it is the exact perturbation induced by evaluating client gradients at selectively initialized models and by averaging the retained parameter offsets.

Let $\mathbf{g}_t := \nabla f(\mathbf{w}^t)$. Smoothness of $f$, Eq. (20), and $L\eta \le 1$ give

$$f(\mathbf{w}^{t+1}) \le f(\mathbf{w}^t) - \eta \langle \mathbf{g}_t, \mathbf{g}_t + \mathbf{e}_t \rangle + \frac{L\eta^2}{2} \|\mathbf{g}_t + \mathbf{e}_t\|^2 \tag{21}$$

$$\le f(\mathbf{w}^t) - \eta \langle \mathbf{g}_t, \mathbf{g}_t + \mathbf{e}_t \rangle + \frac{\eta}{2} \|\mathbf{g}_t + \mathbf{e}_t\|^2 \tag{22}$$

$$= f(\mathbf{w}^t) - \frac{\eta}{2} \|\mathbf{g}_t\|^2 + \frac{\eta}{2} \|\mathbf{e}_t\|^2. \tag{23}$$

It remains to bound the perturbation. By $\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2$, Jensen's inequality, and gradient Lipschitzness,

$$\|\mathbf{e}_t\|^2 \le 2 \left\| \sum_k p_k \left( \nabla \mathcal{L}_k(\tilde{\mathbf{w}}_k^t) - \nabla \mathcal{L}_k(\mathbf{w}^t) \right) \right\|^2 + \frac{2}{\eta^2} \|\bar{\boldsymbol{\delta}}^t\|^2 \tag{24}$$

$$\le 2 \sum_k p_k \left\| \nabla \mathcal{L}_k(\tilde{\mathbf{w}}_k^t) - \nabla \mathcal{L}_k(\mathbf{w}^t) \right\|^2 + \frac{2}{\eta^2} \|\bar{\boldsymbol{\delta}}^t\|^2 \tag{25}$$

$$\le 2L^2 D_t^2 + \frac{2}{\eta^2} \|\bar{\boldsymbol{\delta}}^t\|^2. \tag{26}$$

Rearranging Eq. (23), summing from $t = 0$ to $T-1$, using $f(\mathbf{w}^T) \ge f^\star$, and dividing by $\eta T/2$ yield

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla f(\mathbf{w}^t)\|^2 \le \frac{2(f(\mathbf{w}^0) - f^\star)}{\eta T} + \frac{1}{T} \sum_{t=0}^{T-1} \|\mathbf{e}_t\|^2. \tag{27}$$

Substitution of Eq. (26) gives Eq. (17) and completes the proof.

### E.5 Discounted Reward and Global Validation Progress

The preceding analysis characterizes the optimization effect of a specified redistribution sequence. We next give a complementary, assumption-light identity for the implemented reward. This result does not assert convergence of the neural controller; it only makes precise how the global component of the reward evaluates progress across communication rounds.

Let $A_t = \mathrm{ACC}(\Theta^t)$ and let $\ell_t$ denote the participating-client average of the local reward component in Eq. (8), including its weight $\lambda_{\mathrm{local}}$. For a policy $\pi$, define

$$J_T(\pi) = \mathbb{E}_\pi \left[ \sum_{t=0}^{T-1} \gamma^t \big( \ell_t + \lambda_{\mathrm{glob}} (A_{t+1} - A_t) \big) \right], \qquad U_T(\pi) = \mathbb{E}_\pi \left[ \sum_{t=0}^{T-1} \gamma^t \ell_t \right].$$

For any upper accuracy target $A^\star$, set

$$\omega_t = (1-\gamma)\gamma^{t-1} \quad (1 \le t < T), \qquad \omega_T = \gamma^{T-1},$$

and define the weighted global-accuracy gap

$$\mathcal{E}_T(\pi) = \mathbb{E}_\pi\left[\sum_{t=1}^{T} \omega_t (A^\star - A_t)\right].$$

The weights are nonnegative and sum to one for $0 < \gamma \leq 1$.

**Lemma 1** (Exact discounted-reward decomposition)**.** *For every policy, including a history-dependent policy,*

$$J_T(\pi) = U_T(\pi) + \lambda_{\text{glob}}\left(A^\star - \mathbb{E}A_0 - \mathcal{E}_T(\pi)\right). \tag{28}$$

*Proof.* Reindexing the discounted global-improvement terms gives the pathwise identity

$$\sum_{t=0}^{T-1} \gamma^t (A_{t+1} - A_t) = -A_0 + (1-\gamma)\sum_{t=1}^{T-1} \gamma^{t-1} A_t + \gamma^{T-1} A_T.$$

Substituting the definitions of $\omega_t$ and $\mathcal{E}_T$, and then taking expectations, proves Eq. (28).

When $\gamma = 1$, the global term is exactly $\lambda_{\text{glob}}(A_T - A_0)$ and therefore evaluates terminal global validation accuracy. For $\gamma < 1$, it evaluates a weighted accuracy trajectory rather than only the final model.

**Theorem 3** (Return-to-global-progress transfer)**.** *Let $\pi_0$ be a reference redistribution policy and suppose*

$$J_T(\widehat{\pi}) \geq J_T(\pi_0) - \varepsilon, \qquad U_T(\widehat{\pi}) - U_T(\pi_0) \leq B_T,$$

*where $\lambda_{\text{glob}} > 0$. Then*

$$\mathcal{E}_T(\widehat{\pi}) \leq \mathcal{E}_T(\pi_0) + \frac{\varepsilon + B_T}{\lambda_{\text{glob}}}. \tag{29}$$

*For $\gamma = 1$, this equivalently implies*

$$\mathbb{E}A_T^{\widehat{\pi}} \geq \mathbb{E}A_T^{\pi_0} - \frac{\varepsilon + B_T}{\lambda_{\text{glob}}}.$$

*Proof.* Subtracting Eq. (28) for the two policies cancels the common initial-state term and yields

$$\lambda_{\text{glob}}\big(\mathcal{E}_T(\widehat{\pi}) - \mathcal{E}_T(\pi_0)\big) = U_T(\widehat{\pi}) - U_T(\pi_0) + J_T(\pi_0) - J_T(\widehat{\pi}).$$

Applying the two assumed bounds proves Eq. (29). The terminal-accuracy statement follows because $\mathcal{E}_T(\pi) = A^\star - \mathbb{E}A_T$ when $\gamma = 1$.

The residual $B_T$ is essential: local training can recover accuracy lost at redistribution without improving the global model. Consequently, the theorem does not claim that maximizing local reward alone guarantees global progress, nor does it establish convergence of SAC. Instead, it isolates the condition under which near-reference long-horizon return transfers to a comparable global validation trajectory.

# F Additional Experimental Evaluation

This appendix collects supporting experiments that complement the main paper: implementation details, a non-image architectural-scope evaluation, controller behavior analysis, local-training-loss comparison, unified controller-sensitivity analysis, client-dropout robustness, aggregation-compatibility evaluation, and the overhead analysis.

## F.1 Experimental Setup

### F.1.1 Implementation Details

All methods are implemented in `PyTorch` using Python 3.9. We evaluate four image classification benchmarks of increasing difficulty: Fashion-MNIST (FMNIST) Xiao et al. (2017), which contains 10 classes of $28 \times 28$ grayscale images with 70K total samples; CIFAR-10 Krizhevsky (2009), which comprises 10 classes of $32 \times 32$ RGB images with 60K samples; CIFAR-100 Krizhevsky (2009), which consists of 100 classes of $32 \times 32$ RGB images with 60K samples; and ISIC2019, an eight-class dermoscopic skin-lesion classification benchmark included through FLamby (du Terrail et al., 2022). We additionally study non-image sequence classification on UCI HAR, using a two-layer BERT encoder to classify human activities from sensor features. The dataset–model assignments are fixed across methods: LeNet-5 LeCun et al. (1998) is used for FMNIST; ResNet-18 He et al. (2016) with a linear classifier head is used for CIFAR-10, CIFAR-100, and ISIC2019; and the non-image UCI HAR study uses a two-layer BERT model Devlin et al. (2019). For the ResNet-18 models, the classifier head $\Psi$ contains $C \times 512 + C$ parameters, where $C$ denotes the number of classes, while the backbone contains $|\Phi| = 11{,}176{,}512$ parameters. For CIFAR-10, the head has $|\Psi| = 5{,}130$ parameters, approximately $0.046\%$ of the full model. In float32 precision, the complete model occupies 42.66 MB, whereas the head requires only 0.02 MB. Section F.3 details the BERT architecture. 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.

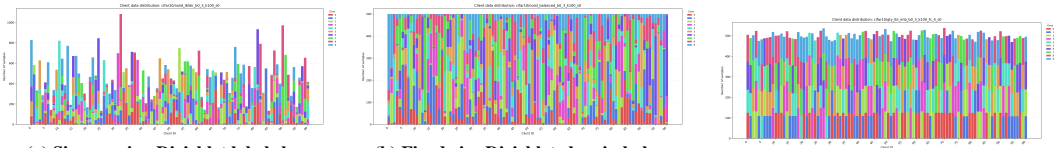
### F.1.2 Data Partitioning

We use two primary heterogeneous data-generation protocols in the experiments: size-varying Dirichlet label skew and fixed-class-support heterogeneity. The size-varying Dirichlet protocol is used for most main and appendix experiments, while the fixed-class-support protocol provides an additional stress test with explicit class absence.

- **Size-varying Dirichlet label skew.** 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 $\beta$ values induce more severe heterogeneity by producing increasingly imbalanced class proportions across clients, whereas larger $\beta$ 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-support 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.
- **Fixed-size Dirichlet class imbalance.** For contextual comparison with the external CIFAR-10 result referenced in the reviewer comment, we also visualize a fixed-size Dirichlet partition convention. This partition fixes the number of training samples per client and varies only class composition; it is not used as the main evaluation protocol in our experiments.

The two primary study protocols provide complementary views of statistical heterogeneity: the size-varying Dirichlet scheme models stochastic variation in both label proportions and realized client sample counts, whereas the fixed-class-support setting imposes a more structured and severe form of class-distribution skew.

Figure 4 highlights an important reproducibility detail. Our reported Dirichlet setting does not enforce equal numbers of samples per client; as a result, $\beta$ controls label skew while the realized partition also contains quantity skew. This setting is therefore a stronger heterogeneity stress test than a fixed-size Dirichlet partition that uses the same $\beta$ value while keeping every client at the same sample count. The fixed-class-support example illustrates the second primary study protocol, where heterogeneity is induced by restricting the class support available to each client. We therefore compare baselines only under matched partitions generated by our data pipeline.



**(a) Size-varying Dirichlet label skew** ($\beta$=0.3)

**(b) Fixed-size Dirichlet class imbalance** ($\beta$=0.3)

**(c) Fixed-class-support** ($\beta$=0.3, fc=4)

Figure 4: CIFAR-10 client data distributions under three partition conventions. In the size-varying Dirichlet label-skew partition used in most of our experiments, the allocation changes both class composition and the number of training samples assigned to each client. In the fixed-size Dirichlet class-imbalance convention, each client has exactly 500 training samples and only the class composition varies. The fixed-class-support partition with $\beta$=0.3 and fc=4 assigns each client data from at most four classes, creating explicit class absence. Thus, identical Dirichlet concentration values can correspond to materially different federated data distributions when the client-size constraint differs, and fixed-class-support splits represent a separate heterogeneity protocol.

Table 9: Effect of partition convention on CIFAR-10. Entries report peak global Top-1 accuracy (%) under three client-partition conventions. All runs use the same evaluation budget and sampling setup (100 clients, 10 clients per round, 105 rounds, five local epochs, batch size 64, and seed 0), but this comparison uses its own training profile rather than the profile used in the main comparison table: ResNet-18, SGD with learning rate 0.01, momentum 0.9, learning-rate decay 0.998, and weight decay $5\times10^{-4}$. Method-specific objectives, regularizers, and controllers are kept according to each method's configured setting. Best result in each column is shown in **bold**.

| Method | Size-varying Dirichlet $\beta$=0.3 | Fixed-size Dirichlet $\beta$=0.3 | Fixed-class support fc=4 |
|---|---|---|---|
| FedAvg | 65.07 | 72.71 | 72.16 |
| FedProx | 56.93 | 74.17 | 72.08 |
| FedNTD | 70.38 | 74.72 | 75.11 |
| Ditto | 55.47 | 74.18 | 72.27 |
| CCVR | 64.10 | 73.72 | 72.87 |
| FedSAM | 69.58 | 74.68 | 75.22 |
| FedFA | 65.16 | 71.74 | 71.72 |
| FedPart | 66.81 | 71.19 | 58.67 |
| FedACG | 72.92 | 75.52 | 71.00 |
| FedCA | 74.39 | 75.10 | 75.23 |
| FedPolicy | **75.66** | **79.37** | **79.16** |

### F.1.3 Training Configuration

Unless otherwise stated, the main benchmark experiments use the following common training protocol. We simulate a federation of $N$=100 clients and use size-varying Dirichlet label skew with $\beta \in \{0.3, 0.1, 0.05\}$; lower $\beta$ indicates stronger label heterogeneity, and the size-varying protocol also induces client quantity skew. In each communication round, $C_r$=10 clients are sampled uniformly at random. Each selected client performs $E$=5 local epochs using SGD with learning rate $\eta$=$10^{-2}$, momentum 0.9, weight decay $5 \times 10^{-4}$, and batch size 64. The local learning rate is fixed at 0.01 after tuning over $[10^{-4}, 10^{-1}]$. Unless a table or subsection explicitly states a different evaluation budget, all methods are trained for $T$=105 communication rounds using the dataset-specific model described in Section F.1.1. For the main benchmark, each baseline retains the local objective in its configured method implementation; the controlled loss study instead reruns every listed method with the explicitly labeled CE or C3E objective. The two tables therefore represent different experimental protocols, and equal-looking method/dataset entries should not be assumed to share a loss or seed summary unless the caption says so. For FedPolicy, we additionally evaluate Focal Loss (FL) Lin et al. (2017), logits-calibrated loss (FedLC) Zhang et al. (2022b), Class-Balanced loss (CB) Cui et al. (2019), and Label Smoothing (LS) Szegedy et al. (2016). C3E Chowdhury & Halder (2026) gives extra weight to class pairs that the local model repeatedly confuses, whereas CE treats wrong classes uniformly. The main comparison in Table 2 is evaluated over three independent random seeds $\{0, 1, 2\}$ and reported as mean $\pm$ standard deviation of the peak Top-1 global

test accuracy. The supporting ablation, sensitivity, robustness, partition-convention, and compatibility studies are controlled single-seed runs using seed 0 unless their caption states otherwise. Each client-held validation subset is excluded from its local SGD batches. A disjoint server-held validation subset is used only for $\mathrm{ACC}(\cdot)$ in the controller state and reward; the reference configurations use 1,500 examples for CIFAR-10 and 2,000 for CIFAR-100 and ISIC2019, with sensitivity reported in Table 13. Test labels are not used for controller training or hyperparameter selection. Test accuracy is logged for evaluation, and the reported peak is a retrospective fixed-budget trajectory statistic rather than a validation-selected deployment checkpoint.

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 aggregation-compatibility evaluation is the exception: there, the aggregation operator is varied deliberately and stated in Table 15. The default FedPolicy controller uses SAC Haarnoja et al. (2018) with the mixed local–global reward and the full state representation. The controller is parameterized by an MLP with hidden width 128, trained with learning rate $10^{-3}$, updated every three communication rounds, and optimized for five controller epochs per update. Alternative RL controllers (PPO, A2C, TRPO, and DQN) are evaluated only in the policy-ablation study. For ablation and supporting studies, these default optimizer, local-training, partitioning, and evaluation settings remain unchanged unless the corresponding table caption or discussion explicitly states otherwise. All methods share identical random seeds, data partitions, and model initializations to ensure strict comparability.

#### F.1.4 Baselines

We benchmark FedPolicy against nine 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 Zhou & Konukoglu (2023) (representation alignment), FedSAM Zhang et al. (2023) (sharpness-aware optimization), Ditto Li et al. (2021) (personalization), FedPart Wang et al. (2024) (scheduled partial training and aggregation), and FedCA Chowdhury & Halder (2026) (confusion-aware local optimization with class-prioritized aggregation). These baselines address heterogeneity through different design principles, whereas FedPolicy targets post-aggregation global knowledge transfer. We report the global-model Top-1 accuracy for every method, including Ditto; the table does not report Ditto's personalized-client accuracy. Within each comparison, methods use matched communication budgets and identical client partitions.

### F.2 Effect of Partition Convention on Empirical Difficulty

Table 9 shows that the same nominal $\beta$=0.3 value can lead to substantially different empirical difficulty depending on the partition convention. This comparison uses the hyperparameter profile stated in the caption and is intended to clarify the effect of the partition rule rather than replace the main benchmark protocol. For most methods, the size-varying Dirichlet setting is harder than the fixed-size Dirichlet setting because it combines label skew with quantity skew; for example, FedAvg changes from $72.71\%$ under fixed-size Dirichlet to $65.07\%$ under size-varying Dirichlet, and FedProx changes from $74.17\%$ to $56.93\%$. The fixed-class-support protocol captures a different form of heterogeneity by imposing explicit class absence, which affects methods differently; for instance, FedPart drops to $58.67\%$ under $\mathtt{fc}{=}4$, whereas several other methods remain near the low-to-mid $70\%$ range. The strongest non-FedPolicy baseline also depends on the partition convention: FedCA is strongest under size-varying Dirichlet and fixed-class-support heterogeneity ($74.39\%$ and $75.23\%$), while FedACG is strongest under fixed-size Dirichlet class imbalance ($75.52\%$). FedPolicy is strongest under all three partition conventions, reaching $75.66\%$, $79.37\%$, and $79.16\%$, respectively. Thus, Table 9 complements Figure 4: $\beta$ alone is not sufficient to identify a matched FL benchmark unless the client-size constraint and partition rule are also specified.

### F.3 Non-Image Architectural Scope: HAR Sensor Classification

We include UCI Human Activity Recognition (HAR) as a non-image scope check. This experiment asks whether selective redistribution can be instantiated for a transformer encoder over sensor features; it is not used to claim universal superiority beyond image classification. HAR contains 561 engineered sensor features and six activity classes. The BERT model maps each scalar fea-

ture to width 64, adds learned positional embeddings and a classification token, and applies two transformer-encoder layers with four attention heads, feedforward width 128, GELU activations, and dropout 0.1. A normalized classification-token representation feeds a six-class linear head. The model is trained from scratch and uses the same Full, Backbone, and Head redistribution actions as the image models.

The study uses synthetic size-varying Dirichlet partitions, 100 configured clients, 10 eligible clients sampled per round, 105 communication rounds, and six seeds. Clients with fewer than 10 training examples are excluded; the resulting active-client counts range from 99–100 at $\beta$=0.3, 78–89 at $\beta$=0.1, and 53–63 at $\beta$=0.05. Local training uses SGD with initial learning rate 0.01, momentum 0.9, weight decay $10^{-5}$, batch size 64, and five local epochs. These partitions are synthetic and should not be interpreted as natural subject-level federations.

Table 10: Top-1 accuracy (%) on FMNIST and UCI HAR under size-varying Dirichlet label skew. Lower $\beta$ indicates stronger heterogeneity. FMNIST uses LeNet-5 and three seeds; UCI HAR uses BERT and six seeds. For HAR, FedPolicy$^\dagger$ denotes selective redistribution with FedACG aggregation and the C3E local objective (the Part-FedACG-CS configuration). Best result in each column is bold; "–" denotes an unreported configuration.

| | FMNIST | | | UCI HAR | | |
|---|---|---|---|---|---|---|
| Method | $\beta$=0.3 | $\beta$=0.1 | $\beta$=0.05 | $\beta$=0.3 | $\beta$=0.1 | $\beta$=0.05 |
| FedAvg | 82.51±0.38 | 78.75±0.52 | 73.11±0.91 | 41.76±0.61 | 40.41±2.28 | 29.80±8.71 |
| FedProx | 82.65±0.35 | 78.89±0.49 | 73.72±0.87 | 41.88±0.75 | 40.73±1.63 | 31.66±6.67 |
| FedNTD | 83.95±0.32 | 80.04±0.51 | 74.23±0.89 | 42.06±1.39 | 40.71±1.05 | 40.22±2.48 |
| Ditto | 83.13±0.41 | 78.15±0.59 | 73.23±0.96 | 41.60±0.98 | 40.58±1.16 | 29.52±9.25 |
| CCVR | 83.63±0.39 | 78.97±0.54 | 71.78±0.99 | 41.85±0.78 | 41.29±1.43 | 34.49±5.83 |
| FedSAM | 82.06±0.45 | 78.87±0.61 | 70.83±1.03 | 41.44±0.74 | 40.23±0.67 | 31.55±6.77 |
| FedFA | 83.31±0.43 | 77.48±0.63 | 74.59±0.92 | 41.98±0.65 | 40.73±0.50 | 32.15±6.89 |
| FedPart | 77.79±0.47 | 71.73±0.52 | 41.82±1.03 | 39.92±1.78 | 34.84±2.18 | 25.84±6.59 |
| FedCA | 84.64±0.56 | 81.24±0.47 | 78.89±0.84 | **43.36±0.63** | 42.39±1.21 | 42.42±1.05 |
| **FedPolicy$^\dagger$** | **85.84±0.61** | **82.68±0.57** | **79.94±0.89** | 42.71±1.19 | **43.06±0.50** | **42.68±0.75** |

Table 10 shows that the redistribution mechanism extends to a transformer-based sensor model without relying on a convolutional backbone. The strongest method depends on heterogeneity: FedCA leads at $\beta$=0.3, whereas FedPolicy$^\dagger$—the FedACG+C3E configuration defined in the table caption—leads at $\beta$=0.1 and 0.05. We interpret this as evidence of architectural portability and competitive behavior, not as evidence that the learned controller or a particular aggregation rule dominates on all non-image tasks.

### F.4 DRL-Policy Behavior over Data Heterogeneity

Figure 5 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 6 and 7 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 $\beta$ values. For example, its action split is 33.7%/34.6%/31.7% at $\beta$=0.05 and 32.9%/35.9%/31.2% at $\beta$=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 ($\beta$=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,

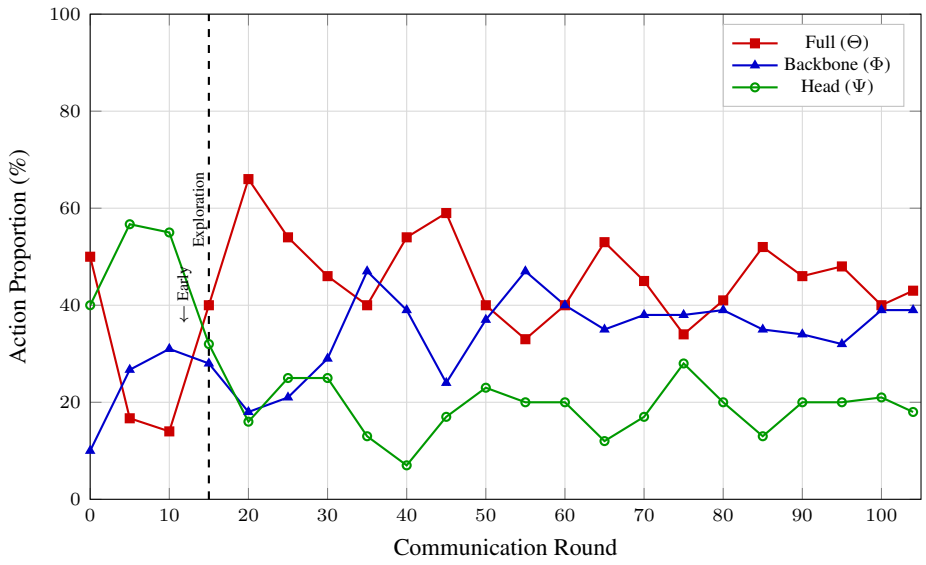



Figure 5: DRL controller action distribution for FedPolicy on CIFAR-10 ($\beta$=0.1).

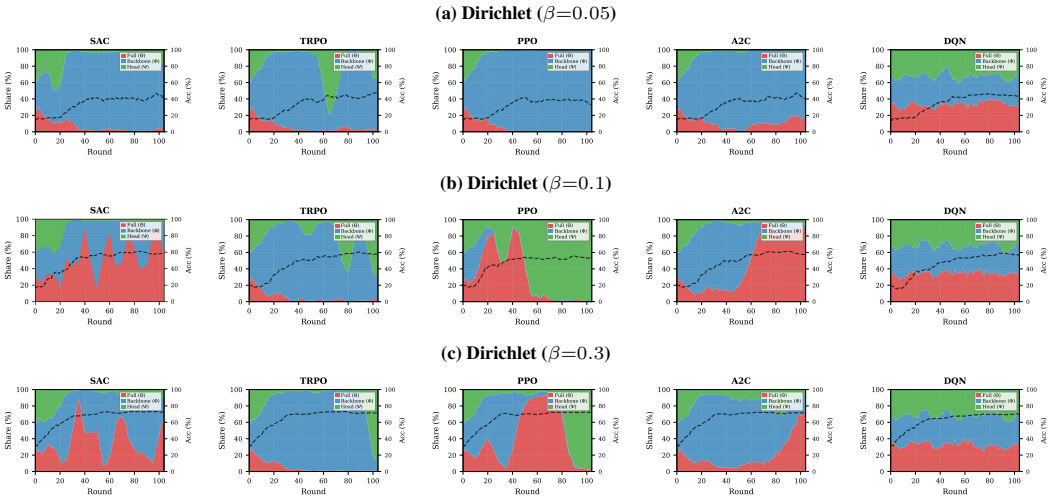



Figure 6: Per-algorithm action evolution (10-round sliding window) for five DRL algorithms across three size-varying Dirichlet label-skew levels. Each panel shows stacked action proportions for Full (Θ), Backbone (Φ), and Head (Ψ), together with global test accuracy (dashed line, right axis).

maintaining a backbone share of 77.9%–80.4% across all three $\beta$ values and achieving the best or near-best accuracy throughout. SAC shows a similar tendency, favoring Backbone updates at $\beta$=0.05 (83.0%) while shifting toward more Full updates as heterogeneity weakens, and attains the highest accuracy at both $\beta$=0.05 (54.4%) and $\beta$=0.3 (74.3%). A2C also remains largely backbone-oriented and performs strongly at $\beta$=0.1 and $\beta$=0.3. PPO is the least stable: although it is strongly backbone-dominant at $\beta$=0.05 (93.3%), it shifts to a head-heavy policy at $\beta$=0.1 (65.0% Head), where its accuracy drops to 58.5%.

The effect of policy choice is most visible at $\beta$=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 $\beta$=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

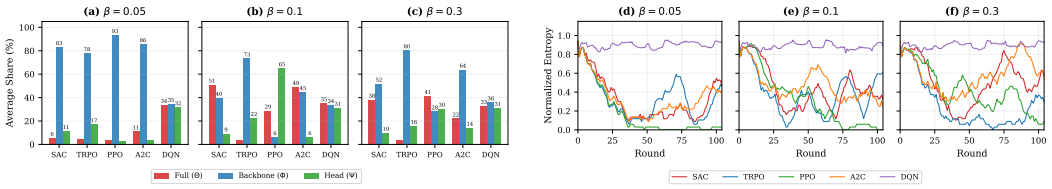



Figure 7: Multi-algorithm policy summary. **Left (a–c):** grouped bar charts of average action proportions per algorithm at each $\beta$. **Right (d–f):** normalized Shannon entropy $H/\log 3$ of action distributions over communication rounds.

Table 11: State and reward ablation for FedPolicy on CIFAR-10 across heterogeneity levels ($\beta$). The full state (All) contains $\vec{C}_k$, $\mathrm{acc}_k$, ACC, and $\|\Delta\theta\|$. Best results per column are in **bold**.

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

emphasis on Backbone sharing, suggesting that feature-level transfer is more reliable than uniform full-model synchronization in highly non-IID settings.

### F.5 Ablation Study

#### F.5.1 State Representation and Reward Design

No reduced state dominates every regime, but removing parameter divergence causes the largest consistent degradation, supporting local–global mismatch as a useful control signal. The combined local–global reward is best at $\beta$=0.3 and 0.1 and remains within 0.25 points of the best result at $\beta$=0.05, making it the most reliable choice across settings.

#### F.5.2 Local-Loss Analysis

Figure 8 compares six local loss functions on CIFAR-10 within FedPolicy under three size-varying Dirichlet label-skew 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 ($\beta$=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 $\beta$=0.1, C3E again maintains the strongest and most stable convergence, whereas the remaining losses either saturate early or exhibit unstable trajectories. Even at the mildest setting ($\beta$=0.3), where the gap is smaller, C3E still achieves the best peak accuracy with the smoothest training behavior.

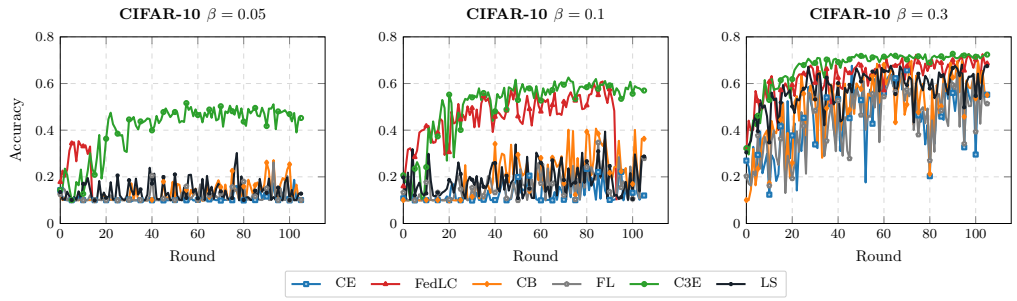



Figure 8: Convergence trajectories of FedPolicy on CIFAR-10 with various local loss functions under $\beta \in \{0.05, 0.1, 0.3\}$.

Table 12: Controlled local-loss study on CIFAR-10 under size-varying Dirichlet label skew (seed 0). Entries report peak Top-1 accuracy (%) when each method is rerun with CE or C3E. These values are not the three-seed main-benchmark summaries in Table 2. Best result in each column is shown in **bold**.

| | $\beta$=0.3 | | $\beta$=0.1 | | $\beta$=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 |
| FedPart | 66.59 | 72.94 | 23.08 | 53.21 | 13.93 | 22.86 |
| 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** |

Table 12 shows that replacing CE with C3E consistently strengthens all compared methods across all heterogeneity levels. However, the advantage of FedPolicy cannot be explained by the local loss alone. Even when the baselines are equipped with the same C3E objective, FedPolicy remains the top-performing method at every $\beta \in \{0.3, 0.1, 0.05\}$. CCVR is the strongest C3E baseline at $\beta$=0.3, while FedCA is strongest at $\beta$=0.1 and $\beta$=0.05, reaching 73.78%, 59.01%, and 51.62%, respectively. FedPolicy reaches 74.28%, 62.84%, and 55.25%, corresponding to gains of +0.50, +3.83, 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. FedPart is included under the same CE and C3E local-loss settings for this comparison. These results indicate that the benefit of FedPolicy is additive to that of stronger local optimization and becomes especially important under severe skew.

### F.5.3 Controller Hyperparameter Sensitivity

We examine whether the learned redistribution policy depends on a narrowly selected controller configuration. Table 13 reports a one-factor-at-a-time sensitivity analysis over the controller learning rate $\eta_c$, local reward weight $\lambda_{\text{local}}$, global reward weight $\lambda_{\text{glob}}$, hidden width $d_h$, controller-training interval $I_c$, controller update epochs $E_c$, and server validation-subset size. Within each parameter block, only the indicated hyperparameter is varied, while the state representation and all remaining controller settings are fixed at the corresponding dataset-specific reference configuration. All experiments use size-varying Dirichlet label skew ($\beta$=0.1), 100 clients, 5 participating clients per round, and 55 global communication rounds.

Table 13: Hyperparameter sensitivity analysis of the FedPolicy controller under size-varying Dirichlet label skew ($\beta$=0.1). All experiments are evaluated after 55 global communication rounds. Within each parameter block, only the indicated hyperparameter is varied, while the state representation and all remaining controller settings are fixed at the corresponding dataset-specific reference configuration. Entries report the peak global Top-1 accuracy (%). The best-performing configuration within each parameter block is highlighted in bold.

| Controller parameter | Value | CIFAR-10 | CIFAR-100 | ISIC2019 |
|---|---|---|---|---|
| Controller learning rate $\eta_c$ | $1 \times 10^{-4}$ | 49.12 | 31.08 | 56.94 |
| | $5 \times 10^{-4}$ | 50.87 | 32.74 | **58.50** |
| | $1 \times 10^{-3}$ | **55.41** | 33.42 | 56.73 |
| | $5 \times 10^{-3}$ | 52.34 | **35.33** | 57.21 |
| Local reward weight $\lambda_{\text{local}}$ | 0.25 | 50.86 | 31.42 | 56.91 |
| | 0.50 | 53.74 | 33.56 | 57.68 |
| | 0.75 | 51.92 | 32.21 | 56.74 |
| | 1.00 | **55.41** | **35.33** | **58.50** |
| | 1.25 | 49.63 | 30.88 | 56.12 |
| Global reward weight $\lambda_{\text{glob}}$ | 0.25 | 52.18 | 31.27 | 56.48 |
| | 0.50 | 49.94 | 33.41 | **58.50** |
| | 0.75 | 53.07 | **35.33** | 57.14 |
| | 1.00 | **55.41** | 31.86 | 56.66 |
| Hidden width $d_h$ | 64 | 52.36 | 31.18 | 57.02 |
| | 128 | **55.41** | 32.46 | 56.41 |
| | 256 | 49.74 | **35.33** | **58.50** |
| Training interval $I_c$ | 1 | 50.71 | 31.04 | **58.50** |
| | 3 | **55.41** | 32.16 | 57.43 |
| | 5 | 52.84 | **35.33** | 56.51 |
| Update epochs $E_c$ | 3 | 51.24 | 31.72 | **58.50** |
| | 5 | **55.41** | 33.04 | 57.11 |
| | 8 | 49.88 | **35.33** | 56.46 |
| Server validation-subset size $V_s$ | 500 | 49.82 | 31.06 | 56.72 |
| | 1000 | 52.93 | 33.14 | 57.61 |
| | 1500 | **55.41** | 32.28 | 57.08 |
| | 2000 | 54.49 | **35.33** | **58.50** |
| | 2500 | 52.66 | 31.42 | 56.34 |

Table 13 shows that FedPolicy remains effective across a range of controller settings while still benefiting from dataset-specific tuning. For CIFAR-10, the strongest settings are close to the reference configuration, including $\eta_c$=$10^{-3}$, $\lambda_{\text{local}}$=1.0, $\lambda_{\text{glob}}$=1.0, $d_h$=128, $I_c$=3, and $E_c$=5. CIFAR-100 and ISIC2019 favor somewhat different controller schedules and model widths, indicating that the best controller configuration can depend on dataset complexity and the validation signal available to the policy. The reward-weight blocks show that the local reward term is important across all three datasets, with $\lambda_{\text{local}}$=1.0 giving the best result in each case, while the preferred global reward weight is dataset-dependent. The validation-subset block shows that a moderate held-out subset is sufficient: CIFAR-10 peaks at $V_s$=1500, whereas CIFAR-100 and ISIC2019 peak at $V_s$=2000, and increasing the subset to 2500 does not improve performance.

### F.5.4 Client-Dropout Robustness

We evaluate robustness to synchronous client dropout on CIFAR-10 with size-varying Dirichlet label skew ($\beta$=0.1), 100 clients, 10 candidate clients sampled per round, ResNet-18, five local epochs, and 105 communication rounds. After the candidate set is sampled, each selected client is independently removed with probability $p_{\text{drop}} \in \{0.1, 0.3, 0.5\}$ before local training and aggregation. Thus, $p_{\text{drop}}$ denotes the per-selected-client failure probability, not the fraction of all 100 clients removed

Table 14: Synchronous client-dropout robustness on CIFAR-10 under size-varying Dirichlet label skew ($\beta$=0.1). In each round, 10 candidate clients are first sampled, and each candidate is independently removed before local training with probability $p_{\text{drop}}$. Entries report peak global Top-1 accuracy (%) over 105 communication rounds. Best result in each column is shown in bold.

| Method | $p_{\text{drop}}$=0.1 | $p_{\text{drop}}$=0.3 | $p_{\text{drop}}$=0.5 |
|---|---|---|---|
| FedAvg | 38.50 | 38.16 | 32.18 |
| FedProx | 38.91 | 38.01 | 30.74 |
| Ditto | 39.84 | 38.47 | 35.62 |
| FedNTD | 36.63 | 38.91 | 34.98 |
| FedSAM | 36.38 | 34.18 | 33.14 |
| CCVR | 43.78 | 42.16 | 39.71 |
| FedFA | 44.29 | 42.83 | 40.36 |
| FedPart | 42.51 | 40.72 | 38.43 |
| FedCA | 46.84 | 44.57 | 44.50 |
| **FedPolicy** | **53.80** | **51.63** | **51.92** |

permanently. With 10 candidates per round, the corresponding expected numbers of participating clients are 9, 7, and 5, respectively. If all candidates are removed, one is retained to preserve a valid synchronous round. The same data partition, candidate-client sampling, seed, and training budget are used for FedAvg, FedProx, FedNTD, FedSAM, FedCA, and FedPolicy. Each method retains its configured local objective and aggregation mechanism. The FedPolicy entry uses the same SAC redistribution controller profile as the corresponding robustness experiment.

Table 14 shows that FedPolicy remains effective across the evaluated dropout probabilities and outperforms all compared methods in each column. Its peak accuracy stays within a 2.17-point range, from 51.63% to 53.80%, and exceeds the strongest non-FedPolicy baseline, FedCA, by 6.96, 7.06, and 7.42 percentage points at $p_{\text{drop}}$=0.1, 0.3, and 0.5, respectively. The completed baseline set, including Ditto, CCVR, FedFA, and FedPart, follows the same dropout protocol and remains below FedCA and FedPolicy in this evaluation. The non-monotonic peak values should not be interpreted as evidence that a particular dropout rate improves learning. Under heterogeneous data, changing the number and composition of surviving clients changes the update trajectory, and the maximum observed over 105 rounds is sensitive to transient fluctuations. Because each configuration uses one seed, the study supports resistance to performance collapse under the evaluated random-dropout protocol, but not a monotonic relationship between dropout and accuracy.

### F.5.5 Aggregation-Compatibility and Selective-Initialization Stability

Selective redistribution mixes a newly aggregated block with a retained local block, so the retained component may differ from the current global model. We do not interpret this retained component as a stale asynchronous update. Rather, the retained block is an intentional initialization choice: for example, when a client receives only the global head, its local backbone is preserved so that additional local optimization can continue adapting representation parameters to that client's local distribution, while the classifier head receives global correction. The relevant question is therefore whether the resulting selective initialization remains compatible with server aggregation.

To examine this point, Table 15 reports a matched aggregation-compatibility evaluation on CIFAR-10 with size-varying Dirichlet label skew ($\beta$=0.1), 100 clients, 10 participating clients per round, ResNet-18, C3E as the local training loss, and 105 communication rounds. We compare uniform Full redistribution and FedPolicy selective redistribution under the same aggregation operators, including FedCA.

The results show no evidence of collapse from combining retained and newly aggregated blocks in this evaluation. Selective redistribution improves peak global Top-1 accuracy under all six matched aggregation settings, with gains ranging from +2.86 to +10.52 percentage points. We therefore treat block mismatch as a quantity that should be monitored and controlled, rather than as evidence that retained local blocks are inherently destabilizing.

Table 15: Aggregation-compatibility evaluation on CIFAR-10 under size-varying Dirichlet label skew ($\beta$=0.1) using C3E as the local training loss. For each aggregation method, the baseline column uses uniform Full redistribution, while the FedPolicy column uses selective post-aggregation redistribution with the same aggregation method. Entries report peak global Top-1 accuracy (%) over 105 communication rounds, and Gain is reported in percentage points (p.p.).

| Aggregation Method | Full redistribution (Baseline) | Selective redistribution (FedPolicy) | Gain (p.p.) |
|---|---|---|---|
| FedCA | 59.01 | **61.87** | +2.86 |
| Mean | 56.84 | **62.84** | +6.00 |
| FedAvg | 49.27 | **56.68** | +7.41 |
| FedAvgM | 48.41 | **55.89** | +7.48 |
| FedAdam | 58.46 | **63.25** | +4.79 |
| Elastic | 41.99 | **52.51** | +10.52 |

Table 16: Complete cost-to-accuracy results on CIFAR-10 ($\beta$=0.1). Entries report the first round reaching each accuracy threshold and the corresponding cumulative communication (GB) and computation time (minutes). "×" denotes a threshold not reached within 105 rounds. Lower is better for rounds and cost; higher is better for peak accuracy.

| Method | Round | | | | Communication (GB) | | | | Computation (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 |
| FedPart | 18 | 52 | 65 | 88 | 4.3 | 10.2 | 10.3 | **11.2** | 6.3 | 18.3 | 22.9 | 31.0 | 53.2 |
| **FedPolicy** | **1** | **11** | **11** | **23** | **0.7** | **7.5** | **7.5** | 12.1 | **0.3** | **3.7** | **3.7** | **8.0** | **63.5** |

### F.6 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.

#### F.6.1 Complete Cost-to-Accuracy Trade-off

Table 16 provides the complete accounting behind the compact main-text comparison. For each target accuracy, it reports the first round at which the target is reached, cumulative model-parameter communication through that round, and cumulative computation time. Communication includes full-model client uploads and the method-specific downlinks; the compact controller-feedback payload is excluded from these tensor-only totals and quantified in Section F.6.3. Computation uses the measured wall-clock cost of each method, while "×" indicates that the target was not reached within the 105-round budget.

FedPolicy ties FedCA in reaching 20% at round 1 and reaches the 30%, 40%, and 50% targets earliest, at rounds 11, 11, and 23. At 50%, it requires 12.1 GB and 8.0 minutes, compared with 72.5 GB and 30.5 minutes for FedCA. FedPart uses 11.2 GB at the same target, 0.9 GB less than FedPolicy, but requires 88 rounds and 31.0 minutes and attains a lower peak accuracy (53.2% versus 63.5%). Hence, selective redistribution does not minimize every individual cost entry; its advantage is reaching stronger operating points substantially earlier with a favorable combined communication–computation profile. The following analyses separate the per-round computational overhead from the communication savings that produce this result.

Table 17: Per-round wall-clock time decomposition (seconds) on CIFAR-10 ($\beta$=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 |

### F.6.2 Computational Overhead

Table 17 reports the per-round wall-clock time on CIFAR-10 under $\beta$=0.1, decomposed into local training, server aggregation, and controller cost. The additional cost of the DRL controller is 0.021 s per round, approximately 0.10% of the measured total and about 2.2 s over 105 rounds. The overall per-round time of FedPolicy is 20.95 s, close to FedAvg-style training (20.91 s) and below FedCA (21.79 s), FedFA (27.47 s), and CCVR (31.08 s) in the same measurement environment. These wall-clock values support a relative overhead comparison within that environment; they should not be interpreted as hardware-independent deployment latency.

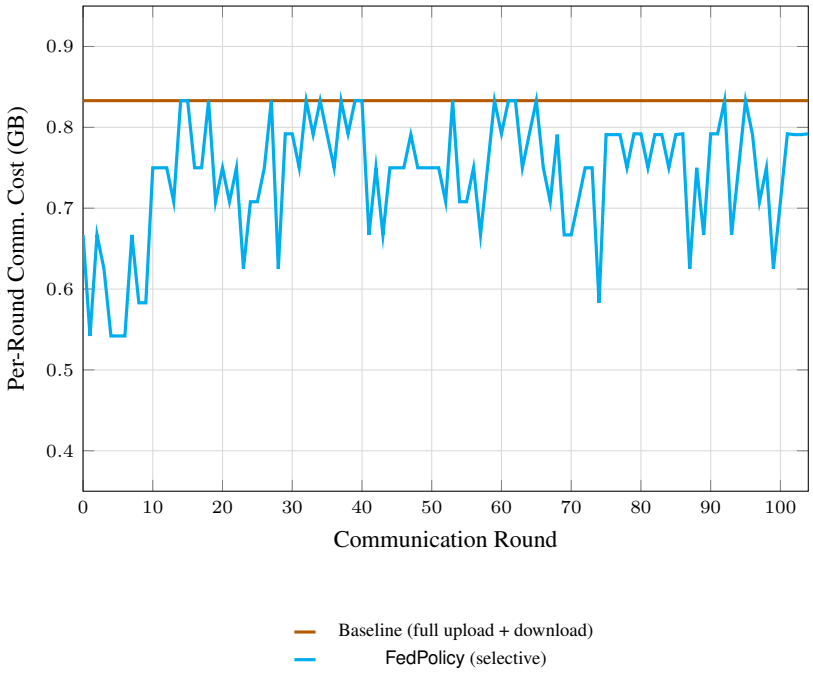



Figure 9: CIFAR-10 ($\beta$=0.1): per-round communication cost. Baselines incur a constant 0.833 GB/round because each selected client uploads and downloads the full model in every round; FedPolicy remains lower in early rounds and approaches baseline as Full updates become more frequent.

### F.6.3 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 model-parameter 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. The accounting excludes the auxiliary controller feedback: per participating client, a $C \times C$ soft-confusion matrix and a small number of scalar statistics are uploaded. This payload is at most about 40 KB in our image benchmarks ($C$=100, float32), compared with a 42.66 MB model, but it is not zero and is disclosed separately from the tensor-only totals.

Figure 9 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.

## G DISCUSSION

### G.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 controller-ablation and local-training-loss comparison isolate post-aggregation redistribution as the active design choice in FedPolicy. The proposed method keeps server aggregation fixed to the mean rule and learns only the post-aggregation sharing decision. This comparison includes both standard CE and C3E settings, showing that the gain arises from the proposed adaptive post-aggregation sharing mechanism in addition to stronger local optimization.
- The comparison with static and Random sharing rules provides empirical evidence that the improvement in stronger heterogeneous regimes is associated with state-dependent learned redistribution rather than a single fixed preference for Full, Backbone, or Head updates. No single static heuristic performs best across all regimes. This evidence does not imply policy optimality or guarantee that each selected action minimizes redistribution mismatch.
- 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.

### G.2 PRIVACY CONSIDERATIONS

FedPolicy does not require raw data sharing, but the method does not by itself provide a formal 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. When formal privacy guarantees are required, FedPolicy can be paired with standard privacy-preserving techniques such as differential privacy, clipping with calibrated noise, secure aggregation, and explicit privacy accounting.

### G.3 Limitations

The primary evaluation covers image classification with LeNet-5 on FMNIST and ResNet-18-style backbone–head decompositions on CIFAR and ISIC2019. The supplementary UCI HAR experiment provides a limited non-image, transformer-based scope check, but uses engineered sensor features and synthetic client partitions; it does not establish broad cross-modal generality. The method also depends on a meaningful decomposition between transferable representation parameters and prediction layers. Its deliberately coarse Full/Backbone/Head action space improves interpretability but limits adaptation granularity. Further study is needed for natural user-level federations, encoder–decoder or adapter-based models, and architectures without an explicit backbone/head split. The present state and reward are classification-specific because they use confusion statistics and validation accuracy; other tasks require task-specific states, actions, and rewards. Finally, ISIC2019 is evaluated only with Top-1 accuracy, so class-balanced and clinically relevant performance require additional metrics such as macro-F1, balanced accuracy, or per-class sensitivity.

### G.4 Future Directions

Promising directions include extending the discrete action space to finer-grained strategies such as layer-wise selection or continuous parameter mixing, and evaluating the method under asynchronous participation, system heterogeneity, and time-varying client distributions. Alternative adaptive controllers—including supervised action predictors and threshold-based decision rules—could also be studied within the same redistribution framework.