FedDO: Dynamic Client Optimization for Adaptive Federated Learning

Peixuan Tang[1] and Xuehe Wang[1*]

School of Artificial Intelligence, Sun Yat-sen University, Zhuhai, China
tangpx3@mail2.sysu.edu.cn, wangxuehe@mail.sysu.edu.cn

Abstract. Federated Learning (FL) allows multiple devices to collaboratively train machine learning models while keeping their raw data private. However, in real-world scenarios, FL often struggles with unstable optimization and slow convergence. This is largely caused by heterogeneous client data, where both the data distributions and dataset sizes vary significantly across devices. To address these challenges, we propose FedDO, an adaptive federated optimization framework guided by reinforcement learning. Instead of using predefined rules or binary selection, FedDO formulates client coordination as a continuous control problem. Specifically, we employ a sample-efficient Distributional Soft Actor-Critic with Three Refinements (DSAC-T) agent to dynamically allocate a continuous data-usage ratio for each client per round. This fine-grained adjustment effectively mitigates gradient drift. Furthermore, we introduce a low-rank parameterization technique to compress the agent's action space, ensuring scalability to thousands of clients. Extensive experiments on standard image classification benchmarks show that FedDO consistently achieves higher accuracy and much faster convergence than state-of-the-art FL baselines, especially under highly unbalanced and heterogeneous data settings. Code is available at `https://github.com/leafuan/FedDO_code`.

Keywords: Federated Learning · Reinforcement Learning · Adaptive Control

1 Introduction

Federated Learning (FL) enables collaborative model training across distributed clients without sharing raw data, effectively addressing privacy and regulatory concerns. This decentralized approach enhances data privacy by ensuring that sensitive information remains localized while still enabling global model updates. However, real-world FL systems face significant challenges due to both *statistical heterogeneity* and *scale heterogeneity*. Statistical heterogeneity refers to the non-independent and identically distributed (non-IID) nature of client data, meaning that data across clients can have varying distributions, which can hinder model

* Corresponding author

convergence. Scale heterogeneity, on the other hand, arises from uneven dataset sizes across clients, leading to imbalances in the contribution of each client to the global model. Consequently, clients with massive amounts of data can easily dominate the training direction, overshadowing the critical but distinct knowledge held by clients with scarce data. These factors significantly complicate the optimization process, making it harder to obtain efficient and stable convergence. Non-IID data cause gradient drift and client divergence [9], where the global model struggles to represent the diversity of local client data, leading to suboptimal performance. In addition, scale imbalance results in inconsistent aggregation weights during model updates, creating oscillations and slower convergence rates.

In dynamic environments where client availability and data characteristics evolve over time, fixed participation rates or sampling rules often fail to maintain model stability. This volatility exacerbates the challenge of achieving robust convergence [28]. Foundational algorithms like FedAvg [24] are highly sensitive to these heterogeneous conditions, frequently resulting in slow and unreliable convergence. Therefore, there is a strong need for an adaptive, round-level control mechanism that can dynamically assign client data-usage ratios to mitigate the imbalances and stabilize the optimization process.

Existing approaches have explored the use of reinforcement learning (RL) to enable adaptive control in FL systems. RL offers a promising framework for optimizing client selection and data utilization, as it is capable of learning dynamic strategies through interaction with the environment. For example, RL has been applied to guide client selection in FL [31, 38]. These approaches demonstrate that RL agents can capture temporal patterns and client-specific dynamics, making them well-suited for optimizing decision-making processes in distributed systems. However, while these works show the potential of RL in FL, most RL–FL frameworks treat client selection as a binary decision—selecting or rejecting clients for each round—without considering the continuous nature of the *data usage ratio*. This limitation prevents such methods from effectively addressing issues related to data-scale imbalance and the diversity of local datasets [37]. Additionally, many existing RL–FL frameworks rely on on-policy RL algorithms, which are inherently less sample-efficient and require frequent online interactions to train the RL agent. This results in high communication overhead and further contributes to instability, as on-policy methods often require large amounts of data to converge effectively [26]. As a result, these methods struggle to provide stable and efficient adaptation, particularly under non-IID, asynchronous, and noisy conditions. Thus, while the potential for RL to enable fine-grained adaptive control in FL is clear, the problem remains largely unsolved, especially when dealing with real-world, complex FL settings.

We propose **FedDO**, a dynamic federated learning framework that leverages reinforcement learning to adaptively control the optimization process in heterogeneous environments. In each communication round, an RL agent jointly controls both *client selection* and the *data usage ratio* of each selected client, enabling fine-grained adjustment of contribution strength according to data scale,

distribution, and recent performance. This joint and continuous control mechanism allows FedDO to balance participation among heterogeneous clients, mitigate gradient bias caused by unbalanced data, and stabilize global convergence across rounds. In addition, the agent adaptively adjusts the utilization of local data within each training round, achieving an implicit trade-off between computation efficiency and statistical representativeness.

To achieve this adaptive control, FedDO adopts the **Distributional Soft Actor–Critic with Three Refinements (DSAC-T)** [6] as its policy-learning backbone. DSAC-T is an off-policy deep reinforcement learning (DRL) algorithm that efficiently reuses historical transitions for policy updates while supporting continuous action spaces. Its distributional critic models the full return distribution rather than only expected Q-values, providing uncertainty-aware value estimation, while the twin-critic architecture reduces over-optimistic bias and enhances training stability. Compared with conventional SAC [11] or PPO [29] methods, DSAC-T achieves superior stability, sample efficiency, and robustness under stochastic or noisy reward settings, making it particularly well-suited for the communication-limited and non-stationary nature of FL. Overall, FedDO integrates these strengths to realize a practical and scalable adaptive-control mechanism that dynamically coordinates client participation and data usage under highly heterogeneous and evolving environments.

Our main contributions are summarized as follows:

- We propose a novel adaptive control framework for federated optimization that treats the **data usage ratio** as a continuous control variable, jointly optimized with client selection to alleviate data-scale heterogeneity.
- We integrate the **DSAC-T** algorithm as a sample-efficient and stable DRL agent, enabling reliable decision-making under communication and statistical constraints.
- Extensive experiments on FashionMNIST, CIFAR-10, CIFAR-100, and Tiny-ImageNet demonstrate that **FedDO** substantially accelerates convergence and achieves higher final accuracy than strong baseline methods.

2 Related Works

Heterogeneity. Federated learning inherently suffers from heterogeneity, where clients possess non-IID data, varying dataset sizes, and class imbalances. Such disparities degrade convergence speed and global accuracy. Optimization solutions mitigate client drift or objective inconsistency, such as FedProx [21], SCAFFOLD [16], and FedDyn [1], which stabilize local updates through regularization or dynamic adjustments. Clustered FL frameworks [10] further group clients by data similarity to reduce gradient conflicts and stabilize global updates, especially when distributions exhibit strong multimodal structures. Personalized FL extends this line by tailoring models to individual client distributions, typically via meta-learning [8], shared representations [3], or ensemble-based aggregation [2], thereby capturing client-specific patterns without sacrificing collaboration benefits. Representation alignment [20] and prototype sharing [4] strengthen

generalization under class imbalance by harmonizing global and local feature spaces and reducing feature drift across heterogeneous clients. In addition, fairness in FL aims to equalize performance among clients by mitigating disparities induced by skewed data or device resources. Agnostic FL [25] optimizes the worst-case client loss, while FairFed [7] explicitly adjusts aggregation weights to balance group accuracies and ensure equitable contributions. More recent works pursue multi-dimensional fairness across data quality and network conditions, such as FedLF [27], FedISM [33], and FedTrade [12], integrating fairness objectives with robustness and personalization to better handle real-world deployment constraints.

Adaptive Control. Recent advances employ reinforcement learning to enable adaptive control and automation in FL. Instead of relying on fixed heuristics, RL formulates client selection, scheduling, and aggregation as sequential decision-making processes, allowing agents to learn optimal strategies over time. Multi-agent RL approaches [32,38] dynamically choose clients according to contribution, latency, and energy constraints, significantly improving convergence under system heterogeneity. RL-driven schedulers further optimize communication efficiency by adjusting aggregation timing and update frequency based on model staleness [22]. Server-side RL controllers, such as in FedAA [13], adapt aggregation weights to maintain robustness and fairness in non-IID environments. Compared with heuristic selection policies like Oort [18], these learning-based frameworks achieve fine-grained, self-tuning coordination between clients and the server, balancing accuracy, fairness, and communication overhead. Overall, RL provides a flexible mechanism to orchestrate federated optimization dynamically, bridging system and data heterogeneity in large-scale deployments.

3 Method

3.1 Problem Statement

We consider a FL system with K clients, where each client k holds a private local dataset $\mathcal{D}_k$. In each communication round t, the server broadcasts the global model w_t and assigns a data usage ratio $d_t^k \in [0,1]$ to every client. Based on this ratio, client k randomly and unbiasedly selects a subset $\mathcal{S}_t^k \subseteq \mathcal{D}_k$ for local training, and clients with $d_t^k > 0$ constitute the participating set $\mathcal{K}_t$. After performing local updates on $\mathcal{S}_t^k$, participating clients return their updated models w_t^k to the server. Let $\mathcal{S}_t = \cup_{k\in\mathcal{K}_t}\mathcal{S}_t^k$ denote the union of all selected samples, and let $\ell(\cdot)$ denote the per-sample loss function. The server aims to minimize the global loss over participating clients. Formally, the FL objective in round t is formulated as

$$\begin{aligned} &\min_{w_t}\left\{F(w_t) := \frac{1}{|\mathcal{S}_t|}\sum_{\xi\in\mathcal{S}_t}\ell(w_t;\,\xi)\right\}, \\ &\text{s.t.}\quad w_t = \sum_{k\in\mathcal{K}_t}\frac{|\mathcal{S}_t^k|}{|\mathcal{S}_t|}\,w_t^k. \end{aligned} \tag{1}$$

3.2 The Agent based on DSAC-T

To adaptively regulate the training process and improve federated optimization efficiency, we treat the FL server as a DSAC-T agent that continuously interacts with the learning environment. At each communication round, the server observes the system state, which reflects the global model condition and the behavior of participating clients. Based on this state, the agent determines an action specifying how much data each client should use for local training. The selected actions are then broadcast to clients, enabling adaptive data usage in every round. In the following, we describe the state space, action space, and reward function used by the agent.

State: The state at round t captures both the current global model and the training dynamics across clients:

$$s_t = \langle w_t, o_t^1, o_t^2, \ldots, o_t^K \rangle. \tag{2}$$

Here w_t denotes the global model parameters, while $o_t^k = \|w_t^k - w_t\|$ measures the parameter drift of client k relative to the global model. The deviation o_t^k reflects the combined effect of statistical heterogeneity and the strength of local updates, indicating how well each client's update aligns with the global optimization direction.

Action: Given the state s_t, the agent outputs a continuous action vector

$$a_t = \langle d_t^1, d_t^2, \ldots, d_t^K \rangle, \tag{3}$$

where $d_t^k \in [0, 1]$ represents the proportion of local data used by client k in round t. This action directly controls the construction of training subsets $\mathcal{S}_t^k$ and implicitly adjusts the participation intensity of each client. The continuous nature of the action space allows the agent to finely tune the workload distribution across clients, enabling adaptive and nuanced control over the federated training process.

To keep the state practical and reproducible, FedDO does not feed raw global weights into the RL networks. Instead, it converts the global model into a compact global summary and combines it with client-wise drift features that measure how far each client is from the current global model. These client drift features are consistently normalized within each round so they are comparable across clients and over training rounds. As a result, the actor and critics use a fixed-size, stable state vector rather than an infeasible high-dimensional weight input.

Reward: To reflect the effectiveness of each action on global learning progress, we design a reward based on the change of model accuracy between consecutive rounds. Let ϕ_t denote the accuracy of the global model evaluated on a held-out server validation set after round t. The reward is defined as

$$r_t = \begin{cases} \Xi^{(\phi_t - \phi_{t-1})}, & \phi_t \geq \phi_{t-1} \\ -\Xi^{(\phi_{t-1} - \phi_t)}, & \phi_t < \phi_{t-1} \end{cases} \tag{4}$$

where Ξ is a shaping coefficient that controls the sensitivity to accuracy changes, and set to 64 in our experiments. For the first round, no historical accuracy is

available, the reward is initialized directly from the current accuracy as $r_1 = \Xi^{\phi_1}$, which provides a consistent baseline for subsequent reward shaping. The reward design encourages the agent to promote steady improvements in model performance while penalizing degradation, guiding the training toward faster and more stable convergence.

3.3 Workflow of FedDO

Fig. 1 illustrates the overall workflow of FedDO. The entire process consists of one initialization step, a local update performed on all clients, and a repeated sequence of two steps in each communication round:

Step 1: All K clients register with the FL server. The server initializes the global model w_{init} and prepares the DSAC-T agent. Each client downloads the initialized global model w_{init}. All clients then perform local updates and upload the updated models $\{w_1^k \mid k \in [K]\}$ to the server.

Step 2: The server aggregates the uploaded models to obtain the new global model w_1 and computes the deviations o_1^k for all clients $k \in [K]$. The DSAC-T agent constructs the system state s_1 and outputs the data-usage ratios a_1 for the next round.

Step 3: In round t, where $t = 1, 2, \ldots$, clients with positive data-usage ratios perform local training on their sampled subsets and upload the updated local models $\{w_{t+1}^k \mid k \in \mathcal{K}_t\}$ to the server.

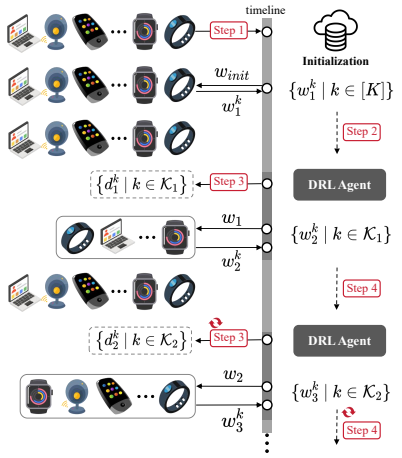


Fig. 1: The operational workflow of FedDO.

Step 4: The server aggregates the received models to obtain w_{t+1} and computes the deviations o_{t+1}^k. The DSAC-T agent encodes these deviations into the new system state s_{t+1} and produces the action a_{t+1} representing the data-usage ratios.

Steps 3–4 are repeated until the target accuracy or the communication budget is reached. In this loop, each round t proceeds as follows: clients upload local models, and the server aggregates them to obtain w_t. The server then forms the state $s_t = \langle w_t, o_t^1, \ldots, o_t^K \rangle$ with $o_t^k = \|w_t^k - w_t\|$. The actor outputs a continuous action $a_t = \langle d_t^1, \ldots, d_t^K \rangle$, clients with $d_t^k > 0$ participate in local training, and the returned models are aggregated into w_{t+1}.

3.4 Training the Agent

As illustrated in Fig. 2, the DSAC-T agent in FedDO forms a closed training loop between the actor and twin critics, where the critics evaluate the value distribution of federated actions and the actor refines the data usage policy accordingly.

During training, the DSAC-T agent continuously interacts with the federated environment to refine its policy toward maximizing long term global accuracy gains. The agent seeks a stochastic policy $\pi_\psi(a|s)$ that maximizes the expected soft return, defined as the cumulative reward regularized by entropy. This objective encourages exploration while preventing the policy from collapsing into deterministic patterns, ensuring robustness in heterogeneous federated settings. Formally, the underlying maximum-entropy RL objective can be written as

$$J(\pi_\psi) = \mathbb{E}_{\pi_\psi}\Big[\sum_{t=0}^{T}\gamma^t\big(r_t + \alpha\mathcal{H}(\pi_\psi(\cdot|s_t))\big)\Big], \tag{5}$$

where γ is the discount factor and α controls the entropy regularization strength. This objective characterizes the desired behavior of the agent, and the subsequent DSAC-T updates provide a tractable way to optimize it from replayed transitions.

To stabilize target generation, a pair of slowly updated target networks, $\bar{\theta}_i$ for critics and $\bar{\psi}$ for the actor, are maintained alongside their online counterparts. These target networks are softly synchronized with a moving average rate τ after each update step.

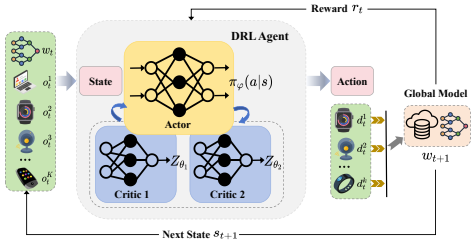


Fig. 2: The DRL agent training process.

In each training iteration, the agent samples transitions (s_t, a_t, r_t, s_{t+1}) from the replay buffer $\mathcal{B}$. For each pair (s_t, a_t), two critic networks output independent value distributions $Z_{\theta_i}(s_t, a_t) = \mathcal{N}(Q_{\theta_i}(s_t, a_t), \sigma^2_{\theta_i}(s_t, a_t))$, $i \in \{1, 2\}$, where Q_{θ_i} estimates the expected soft return and σ_{θ_i} reflects its uncertainty. Given the next state s_{t+1} and action $a_{t+1} \sim \pi_{\bar{\psi}}(\cdot|s_{t+1})$, the critic with the smaller mean value provides the target for both networks. The twin Value Distribution Learning suppresses over-optimistic estimation and stabilizes training. The expected-value substitution further reduces gradient noise by coupling distributional and expected targets:

$$y_q^{\min} = r_t + \gamma\big(Q_{\bar{\theta}_{\bar{i}}}(s_{t+1}, a_{t+1}) - \alpha\log\pi_{\bar{\psi}}(a_{t+1}|s_{t+1})\big)\,, \tag{6}$$

$$y_z^{\min} = r_t + \gamma\big(Z_{\bar{\theta}_{\bar{i}}}(s_{t+1}, a_{t+1}) - \alpha\log\pi_{\bar{\psi}}(a_{t+1}|s_{t+1})\big)\,, \tag{7}$$

with $\bar{i} = \arg\min_i Q_{\bar{\theta}_i}(s_{t+1}, a_{t+1})$. Each critic minimizes the distributional Bellman error using $y_q^{\min}$ to anchor the mean and $y_z^{\min}$ to preserve the variance structure of the target distribution. A variance-aware clipping boundary $C(y_z^{\min}; b) = \mathrm{clip}(y_z^{\min}, Q_{\theta_i} - b, Q_{\theta_i} + b)$ is applied to suppress unstable deviations, where $b = \xi\,\mathbb{E}_{\mathcal{B}}[\sigma_{\theta_i}]$ tracks uncertainty through a moving average.

To compensate for different reward magnitudes, the critic objective is weighted by a variance-based factor ω_i. We define $\omega_i = \mathbb{E}_{\mathcal{B}}\big[\sigma^2_{\theta_i}\big]$, yielding the

scaled distributional loss

$$J_{\mathcal{Z}}^{\text{scale}}(\theta_i) \;=\; \omega_i \, \mathbb{E}_{(s,a)\sim\mathcal{B}}\big[D_{\text{KL}}\big(y_z^{\min} \,\|\, Z_{\theta_i}(s,a)\big)\big] \,. \tag{8}$$

Weighting ω_i aligns the critic update magnitude with the return variance, making the learning process inherently insensitive to the absolute scale of the reward. Consequently, in FedDO, the exponential base Ξ in the reward function merely modulates the relative sensitivity to accuracy changes, while its exact value has negligible influence on training stability.

Once the critics converge, the actor is updated to maximize a soft value surrogate that is consistent with the maximum-entropy objective in Eq. (5). Following the DSAC-T formulation, the actor parameters are optimized by

$$J_\pi(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{B}}\Big[\min_{i=1,2} Q_{\theta_i}(s,a) \;-\; \alpha \, \log \pi_\psi(a|s)\Big] \,, \tag{9}$$

which serves as a sample-based approximation of $J(\pi_\psi)$ and encourages actions yielding higher soft returns while preserving adequate stochasticity.

Through the iterative process, the DSAC-T agent in FedDO learns an adaptive data-usage strategy across clients, achieving steady convergence of the global model while minimizing redundant communication.

4 Experiments

4.1 Experimental Settings

We implement and integrate FedDO on top of the standard FedAvg framework. Unless otherwise specified, all experiments assume a total of 100 clients. To ensure fair comparison, all participating clients use the same batch size and perform an identical number of local training epochs. For all federated learning methods involved, we adopt a unified optimization setup using stochastic gradient descent (SGD) with a learning rate of 0.01, momentum of 0.9, and a weight decay of 1×10^{-5}. All experiments are conducted on an Ubuntu multi-GPU workstation equipped with an Intel Xeon CPU, 256 GB RAM, and NVIDIA RTX 4090 GPUs.

Dataset. We evaluate our method on four widely used visual classification datasets: CIFAR-10 [17], CIFAR-100 [17], FashionMNIST [34], and TinyImageNet [5]. To systematically examine the behavior of FedDO under different levels of data heterogeneity, we adopt three distribution settings: IID and non-IID partitions generated by a Dirichlet distribution, denoted as $\text{Dir}(\alpha)$ with $\alpha \in \{0.1, 0.5\}$. Smaller values of α correspond to stronger label and sample skew (higher heterogeneity); and the IID case serves as the uniform baseline.

Model. We evaluate three representative convolutional architectures for image classification: a lightweight CNN, the mid–depth residual network ResNet-18 [14], and the deeper sequential model VGG-11 [30]. The CNN follows the classical baseline commonly adopted in FedAvg, consisting of two convolution–pooling blocks followed by fully connected layers. Both ResNet-18 and

VGG-11 use their standard configurations from the official PyTorch model repository, with only the final classification head replaced to match the number of classes for each dataset.

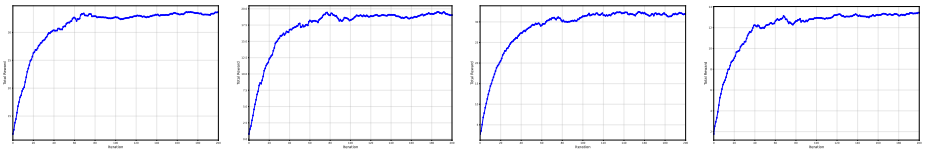
Baseline. We compare FedDO with seven representative federated learning baselines covering the main techniques for mitigating heterogeneity, including classical aggregation, contrastive regularization, dynamic regularization, label-shift calibration, knowledge distillation, cross-model collaboration, and adaptive aggregation. The configurations of all baselines are summarized as follows.

- **FedAvg** [24] is the classical federated averaging framework, where the server aggregates local model updates via weighted averaging in each communication round.
- **MOON** [20] introduces a contrastive objective between the global model and each client's historical model to alleviate representation drift.
- **Oort** [18] adopts an exploration-exploitation based client selection strategy to balance statistical utility and system efficiency.
- **FedLC** [36] mitigates label distribution skew by calibrating model logits according to client-specific class-frequency statistics.
- **FedNTD** [19] adopts a not-true distillation strategy to reduce inter-client model discrepancies through knowledge distillation.
- **FedCross** [15] is a cross-model collaborative aggregation method that leverages pairwise model similarity for collaborative optimization.
- **FedAA** [13] employs deep reinforcement learning to adaptively select participating clients and optimize aggregation weights for robust and fair federated optimization.

For all baseline methods, we follow the recommended hyperparameter settings reported in their original papers. We implement all federated learning methods using the PFLlib Benchmark [35].

4.2 Experiment Results

Training the DRL Agent We train a separate DRL agent on each of the four datasets, allowing the policy to specialize to the distinct statistical characteristics and optimization trajectories of different FL tasks. To ensure computational efficiency and maintain a fair basis for cross-dataset comparison, all agents are implemented with an identical network architecture.

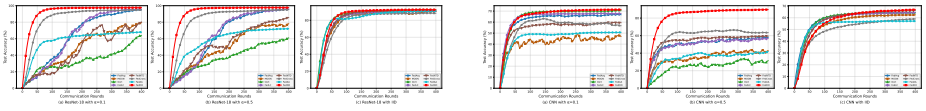


(a) FashionMNIST **(b)** CIFAR-10 **(c)** CIFAR-100 **(d)** TinyImageNet

Fig. 3: Training convergence curves of the DRL agents on four datasets.

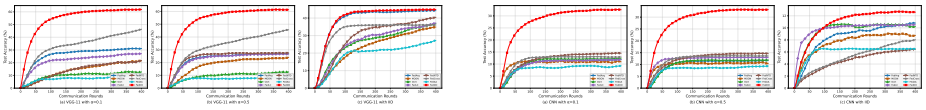
Although lightweight, this architecture incorporates representative design elements from CNN, ResNet-18, and VGG-11 to balance expressive capacity and training stability. Specifically, it adopts VGG-style double convolutional blocks with max pooling to capture local spatial patterns, introduces residual connections at strategically selected layers to alleviate gradient degradation in deeper stages, and applies adaptive average pooling followed by a compact fully connected classification head to preserve feature consistency across datasets of varying input dimensions.

Fig. 3 illustrates the training dynamics of the DRL agent on three representative FL tasks. Each agent is trained for 200 iterations, where each iteration corresponds to a complete federated learning cycle—from initializing the global model, performing round-by-round client participation and data-usage control, to terminating once the predefined target accuracy is achieved. This formulation enables the agent to observe the entire trajectory of a federated optimization process and learn policies that generalize across different levels of data heterogeneity. The target accuracies for the four datasets are set as follows: FashionMNIST: 0.95, CIFAR-10: 0.55, CIFAR-100: 0.45, and TinyImageNet: 0.30. These thresholds reflect the typical achievable performance under heterogeneous client settings and define a unified stopping criterion for evaluating DRL-guided FL optimization.



(a) FashionMNIST with ResNet-18 **(b)** CIFAR-10 with CNN

Fig. 4: Test Accuracy vs. Communication Rounds on FashionMNIST and CIFAR-10



(a) CIFAR-100 with VGG-11 **(b)** TinyImagenet with CNN

Fig. 5: Test Accuracy vs. Communication Rounds on CIFAR-100 and TinyImagenet

Performance Comparison FedDO consistently outperforms all baseline methods across various models, datasets, and heterogeneity settings. As shown in Tab. 1, in non-IID conditions, FedDO achieves a significant and stable accuracy improvement over all baselines. For example, on TinyImageNet, it outperforms the best baseline by more than 20–30 percentage points. In more challenging datasets like CIFAR-10/100, FedDO maintains its superiority across different values of α, with the advantage becoming more prominent as heterogeneity increases, demonstrating its strong adaptability to client distribution skew. On relatively simpler datasets like FashionMNIST, FedDO approaches saturation while still maintaining a lead over all methods. Even in IID environ-

Table 1: Top-1 accuracy of baseline methods and FedDO under non-IID and IID data partitions.

Model	Dataset	Heterogeneity Setting	Acc. (%)							
			FedAvg	MOON	Oort	FedLC	FedNTD	FedCross	FedAA	FedDO
CNN	FashionMNIST	$\alpha = 0.1$	83.66 ± 1.49	74.00 ± 10.10	84.29 ± 0.60	82.96 ± 1.83	83.78 ± 1.32	84.75 ± 0.31	75.53 ± 1.86	$\mathbf{87.49 \pm 0.07}$
		$\alpha = 0.5$	83.52 ± 2.76	80.48 ± 1.95	82.99 ± 1.67	83.48 ± 2.31	84.53 ± 1.22	84.57 ± 0.69	76.58 ± 1.48	$\mathbf{97.34 \pm 0.01}$
		IID	90.80 ± 0.20	90.77 ± 0.14	91.06 ± 0.09	90.16 ± 0.21	90.74 ± 0.23	89.35 ± 0.03	89.88 ± 0.03	$\mathbf{90.99 \pm 0.02}$
	CIFAR-10	$\alpha = 0.1$	66.56 ± 0.72	46.19 ± 3.81	70.78 ± 0.27	67.83 ± 0.08	59.51 ± 0.43	55.68 ± 1.27	50.76 ± 0.10	$\mathbf{71.36 \pm 0.10}$
		$\alpha = 0.5$	56.88 ± 4.60	46.89 ± 4.04	34.08 ± 3.26	56.23 ± 1.93	59.52 ± 0.69	63.28 ± 0.67	43.39 ± 2.11	$\mathbf{89.70 \pm 0.05}$
		IID	64.31 ± 0.06	62.76 ± 0.24	63.98 ± 0.09	64.51 ± 0.15	66.13 ± 0.07	59.25 ± 0.07	57.10 ± 0.05	$\mathbf{67.03 \pm 0.08}$
	CIFAR-100	$\alpha = 0.1$	28.92 ± 0.06	26.41 ± 0.35	25.71 ± 0.32	27.98 ± 0.18	28.93 ± 0.12	49.81 ± 0.05	20.22 ± 0.21	$\mathbf{50.14 \pm 0.15}$
		$\alpha = 0.5$	26.65 ± 0.18	24.82 ± 0.47	25.17 ± 0.53	27.02 ± 0.32	27.86 ± 0.19	50.92 ± 0.06	19.12 ± 0.23	$\mathbf{53.44 \pm 0.21}$
		IID	23.44 ± 0.16	22.50 ± 0.15	22.88 ± 0.16	24.29 ± 0.03	25.91 ± 0.06	31.44 ± 0.03	19.61 ± 0.06	$\mathbf{32.39 \pm 0.14}$
	TinyImageNet	$\alpha = 0.1$	12.01 ± 0.18	10.69 ± 0.43	12.49 ± 0.05	11.96 ± 0.15	14.44 ± 0.06	12.98 ± 0.02	9.47 ± 0.10	$\mathbf{32.45 \pm 0.04}$
		$\alpha = 0.5$	11.81 ± 0.13	10.11 ± 0.39	11.94 ± 0.18	12.98 ± 0.03	14.57 ± 0.05	13.54 ± 0.05	9.42 ± 0.13	$\mathbf{33.17 \pm 0.16}$
		IID	10.81 ± 0.08	8.85 ± 0.09	10.22 ± 0.13	10.52 ± 0.03	6.50 ± 0.08	8.21 ± 0.05	6.54 ± 0.05	$\mathbf{12.59 \pm 0.04}$
ResNet-18	FashionMNIST	$\alpha = 0.1$	95.30 ± 0.32	80.87 ± 3.79	65.34 ± 2.01	95.79 ± 0.22	83.51 ± 0.77	94.90 ± 0.02	67.87 ± 0.61	$\mathbf{97.68 \pm 0.03}$
		$\alpha = 0.5$	95.83 ± 0.07	82.35 ± 4.39	63.85 ± 0.42	96.26 ± 0.16	87.28 ± 1.21	95.34 ± 0.04	72.21 ± 0.58	$\mathbf{97.69 \pm 0.01}$
		IID	90.69 ± 0.09	90.61 ± 0.06	92.11 ± 0.03	90.77 ± 0.05	91.10 ± 0.10	88.23 ± 0.04	91.00 ± 0.06	$\mathbf{92.11 \pm 0.03}$
	CIFAR-10	$\alpha = 0.1$	55.75 ± 4.16	54.18 ± 1.27	59.25 ± 0.25	66.42 ± 0.04	53.54 ± 1.19	82.50 ± 0.17	42.89 ± 1.92	$\mathbf{83.24 \pm 0.10}$
		$\alpha = 0.5$	87.08 ± 0.36	65.92 ± 4.89	22.75 ± 4.48	69.99 ± 3.44	72.79 ± 2.86	85.27 ± 0.46	54.09 ± 0.61	$\mathbf{88.26 \pm 0.10}$
		IID	66.07 ± 0.13	67.03 ± 0.08	66.24 ± 0.21	68.19 ± 0.06	68.38 ± 0.13	75.06 ± 0.03	57.70 ± 0.33	$\mathbf{77.03 \pm 0.08}$
	CIFAR-100	$\alpha = 0.1$	23.89 ± 0.29	24.22 ± 0.20	23.48 ± 0.26	27.62 ± 0.25	27.30 ± 0.14	26.25 ± 0.09	19.20 ± 0.16	$\mathbf{31.65 \pm 0.04}$
		$\alpha = 0.5$	19.74 ± 0.59	22.09 ± 0.45	24.09 ± 0.34	19.83 ± 0.88	20.65 ± 0.50	17.15 ± 0.11	19.16 ± 0.15	$\mathbf{26.61 \pm 0.20}$
		IID	32.37 ± 0.32	31.32 ± 0.09	33.21 ± 0.06	33.08 ± 0.05	36.03 ± 0.10	32.65 ± 0.06	27.65 ± 0.14	$\mathbf{37.87 \pm 0.01}$
	TinyImageNet	$\alpha = 0.1$	11.11 ± 0.15	11.90 ± 0.11	11.34 ± 0.10	10.84 ± 0.27	18.49 ± 0.13	13.89 ± 0.06	8.30 ± 0.07	$\mathbf{22.02 \pm 0.23}$
		$\alpha = 0.5$	10.42 ± 0.05	12.36 ± 0.09	11.01 ± 0.16	20.47 ± 0.02	15.17 ± 0.49	30.48 ± 0.10	8.12 ± 0.07	$\mathbf{31.22 \pm 0.11}$
		IID	17.15 ± 0.07	15.54 ± 0.10	15.82 ± 0.12	18.29 ± 0.05	19.73 ± 0.17	16.65 ± 0.10	8.95 ± 0.12	$\mathbf{20.01 \pm 0.06}$
VGG-11	FashionMNIST	$\alpha = 0.1$	77.96 ± 1.38	40.52 ± 4.52	38.05 ± 1.57	74.84 ± 4.29	76.11 ± 1.60	74.06 ± 0.09	60.17 ± 2.16	$\mathbf{90.65 \pm 0.02}$
		$\alpha = 0.5$	72.29 ± 2.11	55.08 ± 1.94	40.20 ± 2.24	81.81 ± 2.47	72.53 ± 3.94	74.13 ± 0.36	51.92 ± 1.56	$\mathbf{94.63 \pm 0.01}$
		IID	91.11 ± 0.06	90.79 ± 0.10	92.06 ± 0.17	91.00 ± 0.05	90.60 ± 0.04	89.67 ± 0.03	89.40 ± 0.21	$\mathbf{92.49 \pm 0.05}$
	CIFAR-10	$\alpha = 0.1$	75.11 ± 1.77	69.58 ± 2.66	64.44 ± 0.59	88.55 ± 0.03	76.72 ± 0.54	71.90 ± 0.36	46.05 ± 1.95	$\mathbf{81.76 \pm 0.08}$
		$\alpha = 0.5$	78.67 ± 1.30	35.74 ± 3.24	14.66 ± 2.96	76.28 ± 2.15	65.05 ± 1.54	71.75 ± 0.11	29.38 ± 1.53	$\mathbf{90.35 \pm 0.14}$
		IID	78.54 ± 0.20	79.69 ± 0.19	80.27 ± 0.29	78.73 ± 0.30	81.80 ± 0.10	73.06 ± 0.04	69.35 ± 0.25	$\mathbf{82.37 \pm 0.07}$
	CIFAR-100	$\alpha = 0.1$	31.37 ± 0.20	22.07 ± 0.52	12.98 ± 0.61	27.27 ± 0.35	22.55 ± 0.30	47.15 ± 0.09	11.29 ± 0.59	$\mathbf{61.92 \pm 0.15}$
		$\alpha = 0.5$	26.86 ± 0.07	24.66 ± 0.33	12.89 ± 0.42	26.78 ± 0.15	27.87 ± 0.19	46.12 ± 0.10	9.19 ± 1.48	$\mathbf{61.34 \pm 0.09}$
		IID	44.18 ± 0.12	35.18 ± 0.24	37.30 ± 0.38	37.71 ± 0.66	40.91 ± 0.55	38.85 ± 0.15	27.98 ± 0.27	$\mathbf{44.84 \pm 0.06}$
	TinyImageNet	$\alpha = 0.1$	6.92 ± 0.54	10.41 ± 0.12	3.75 ± 0.23	6.62 ± 0.34	5.97 ± 0.41	11.61 ± 0.17	5.91 ± 0.05	$\mathbf{14.73 \pm 0.09}$
		$\alpha = 0.5$	6.55 ± 0.36	8.35 ± 0.22	4.14 ± 0.22	7.02 ± 0.21	6.62 ± 0.24	13.91 ± 0.06	4.13 ± 0.22	$\mathbf{15.10 \pm 0.29}$
		IID	21.85 ± 0.19	20.28 ± 0.64	13.27 ± 0.34	18.93 ± 0.55	19.24 ± 0.46	18.71 ± 0.12	7.62 ± 0.50	$\mathbf{24.14 \pm 0.07}$

ments, where the performance gap naturally narrows, FedDO remains the best or tied for the best, indicating that its benefits arise not solely from compensating for heterogeneity, but from a fundamentally more stable and efficient global optimization mechanism.

As shown in Figs. 4 and 5, FedDO consistently accelerates global convergence and achieves higher final accuracy across all datasets and model architectures. In the early communication rounds, the FedDO curves climb much more steeply than those of all baselines, indicating that the learned control policy quickly identifies effective participation and data-usage strategies, especially under the highly non-IID Dir(0.1) setting where several competing methods exhibit slow progress or even visible stagnation. As training proceeds, FedDO maintains a remarkably smooth and steadily increasing trajectory, while other methods often show pronounced oscillations or premature plateaus, reflecting their sensitivity to gradient drift and client imbalance. This advantage becomes particularly evident on the more challenging CIFAR-100 and Tiny-ImageNet tasks, where FedDO continues to make substantial gains after baselines have largely saturated, but it also persists in the mildly non-IID and IID cases, where FedDO

reaches the accuracy ceiling earlier and with fewer fluctuations. Taken together, these learning curves demonstrate that FedDO not only improves the ultimate accuracy but also stabilizes the entire optimization process, providing robust and adaptable behavior under varying degrees of data heterogeneity.

Tab. 2 summarizes the communication rounds and accuracies required to reach the target accuracy on CIFAR-10 with ResNet-18, revealing a clear and consistent advantage of FedDO across all data-partition settings. Under the most heterogeneous configuration, Dir(0.1), FedDO attains the target accuracy in only 344 rounds, whereas every baseline except FedCross fails to reach the target within 1000 rounds, demonstrating that FedDO accelerates convergence by more than a factor of two while preserving accuracy. Under Dir(0.5), FedDO again shows a pronounced advantage, among the baselines that eventually converge, only FedAvg reaches the target, taking almost twice as many rounds as FedDO and still offering no accuracy benefit. In the IID setting, where most methods typically behave more similarly, FedDO still maintains a meaningful lead. It reaches the target accuracy in 476 rounds and achieves 67.4%, outperforming all baselines in both efficiency and final accuracy. The results in Tab. 2 demonstrate that FedDO not only converges substantially faster but also attains consistently higher or comparable accuracies. These improvements underscore the robustness and optimization stability of FedDO, particularly in challenging non-IID regimes where classical FL algorithms struggle to maintain reliable progress.

Table 2: Communication rounds R and accuracy (Acc, %) needed to reach the target accuracy on CIFAR-10 with ResNet. Each cell shows $(R,\ Acc)$; $>$ denotes the method failed to reach the target within the 1000 rounds and use the best accuracy achieved.

Method	Dir(0.1)	Dir(0.5)	IID
Target	71.4	87.6	66.6
FedAvg	(>, 55.7)	(784, 87.7)	(>, 66.2)
MOON	(>, 14.5)	(>, 37.3)	(773, 67.1)
Oort	(>, 59.35)	(>, 24.1)	(>, 66.4)
FedLC	(>, 66.42)	(>, 71.7)	(639, 68.1)
FedNTD	(>, 54.6)	(>, 73.9)	(625, 68.0)
FedCross	(729, 82.4)	(>, 85.7)	(821, 75.0)
FedAA	(>, 42.1)	(>, 54.7)	(>, 57.1)
FedDO	(**344**, **71.5**)	(**429**, **87.8**)	(**476**, **67.4**)

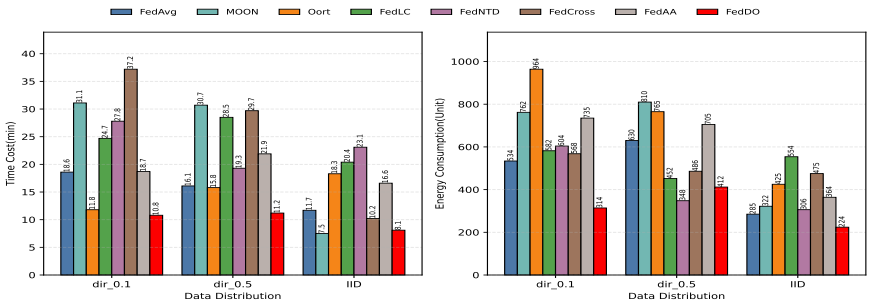
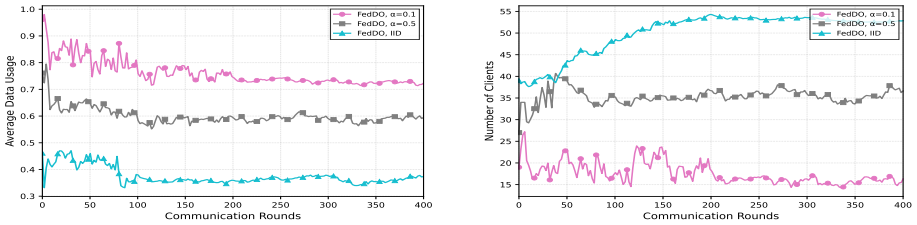


Fig. 6: Time and Energy Cost on CIFAR-10 with ResNet

Under the settings in Tab. 2, we further evaluate the deployment efficiency of different methods by measuring the wall-clock time and total energy consump-

tion during online federated optimization required to reach the target accuracy. For methods that fail to reach the target, we instead report the time and energy consumed until convergence. For FedDO, the reported results exclude the one-time offline DRL policy learning cost and only account for the online deployment stage. The total energy consumption is estimated following [23], including both local computation energy and communication energy. Specifically, the computation energy depends on the amount of local data processed and the computing frequency, while the communication energy is determined by the transmission power and upload duration. As shown in Fig. 6, the reported results focus on the online deployment stage. While FedDO incurs a one-time offline policy learning cost, this cost is independent of subsequent federated optimization and can be amortized across repeated deployments. During online execution, the learned policy adaptively controls the client-side data usage ratio, reducing redundant computation and unnecessary communication. Consequently, FedDO achieves competitive time and energy efficiency across all data distributions, with particularly significant gains under the highly heterogeneous Dir(0.1) setting. Together with the substantial reduction in communication rounds shown in Table 2, these results demonstrate the practical deployment efficiency of FedDO.

Exploration on DRL Agent As shown in Fig. 7, the DRL agent in FedDO quickly transitions from early-stage exploration to a stable, dataset-aware decision pattern. For data usage, all three curves exhibit only brief fluctuations in the first few rounds before converging to distinct steady levels that reflect the degree of statistical heterogeneity: Dir(0.1) consistently maintains the highest average data usage, followed by Dir(0.5), while the IID case converges to the lowest level. This suggests that under severe distribution shift, the agent allocates more local data to each selected client to mitigate gradient drift and improve the stability of global updates, whereas more homogeneous settings require less local computation.



(a) Average data usage. **(b)** Number of participating clients.

Fig. 7: DRL Agent Exploration on CIFAR-10 with VGG.

The number of participating clients exhibits the opposite yet complementary trend. The agent stabilizes at a relatively small client subset under Dir(0.1), a moderate participation level under Dir(0.5), and the largest and most stable client set under IID. Together with the data-usage trends, these results show that FedDO automatically balances aggregation quality, communication efficiency,

and client representativeness: stronger heterogeneity favors fewer but more "intensive" clients with larger data budgets, whereas the IID setting involves more clients while assigning less data to each. Overall, the learned policy effectively captures the trade-off between client participation and data usage under different heterogeneity levels, leading to stable and efficient control.

Scalability In real-world federated learning systems, it is common to concurrently involve hundreds or even thousands of clients. Under such settings, the Actor in DSAC-T is required to output an action vector whose dimension equals the number of clients (K), while the Critic must process a high-dimensional concatenated input. As K increases, the parameters of the Actor's final layer and the Critic's first layer grow approximately linearly with K, leading to substantial computational overhead and optimization instability. This inherent scaling behavior constitutes the primary bottleneck in terms of scalability.

To address this issue, we adopt a parameterized restructuring strategy to reduce the model size. Specifically, we apply low-rank projection to compress the high-dimensional input representation and perform low-rank factorization on the output mapping. The modification does not alter the DSAC-T architecture itself; on the contrary, it introduces a more compact parameterization of the high-dimensional mappings. As shown in Tab. 3, evaluated on FashionMNIST with VGG-11 under the IID setting, the total number of parameters (Para) is significantly reduced. A slight decrease in accuracy is observed, which can be attributed to the low-rank constraint limiting the expressive capacity of the model to some extent by reducing its degrees of freedom.

Table 3: Impact of parameterized restructuring on model size and accuracy across varying numbers of clients.

	Original		Low-Rank	
K	Acc	Para	Acc	Para
100	90.62	0.93M	90.29	0.74M (↓20.4%)
500	91.03	2.26M	89.27	1.28M (↓43.4%)
1000	89.78	3.96M	89.55	1.95M (↓50.8%)
2000	88.51	7.25M	85.96	3.47M (↓52.1%)

5 Conclusion

In this paper, we proposed FedDO, an adaptive federated learning framework that employs a DSAC-T RL agent to mitigate statistical and scale heterogeneity among clients. By assigning continuous data-usage ratios to determine optimal client participation in each round, FedDO dynamically reduces gradient conflicts and stabilizes global model convergence. To ensure scalability to thousands of clients, we further integrate a low-rank projection module to significantly compress the agent's parameters. Comprehensive experiments demonstrate that FedDO consistently outperforms existing baselines, achieving faster convergence and higher accuracy across varying degrees of data heterogeneity. Future work will explore extending the FedDO framework to asynchronous federated learning environments and integrating advanced privacy-preserving mechanisms.

References

1. Acar, D.A.E., Zhao, Y., Navarro, M., Mattina, M., Whatmough, P., Matni, N.: Federated learning based on dynamic regularization. In: ICLR (2021)
2. Chen, H.Y., Chao, W.L.: Fedbe: Making bayesian model ensemble applicable to federated learning. In: ICLR (2021)
3. Collins, L., Hassani, H., Mokhtari, A., Shakkottai, S.: Exploiting shared representations for personalized federated learning. In: ICML. pp. 2089–2099 (2021)
4. Dai, Y., Chen, Z., Li, J., Heinecke, S., Sun, L., Jin, R.: Tackling data heterogeneity in federated learning with class prototypes. In: AAAI. pp. 7314–7322 (2023)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255 (2009)
6. Duan, J., Wang, W., Xiao, L., Gao, J., Li, S.E., Liu, C., Zhang, Y.Q., Cheng, B., Li, K.: Distributional soft actor-critic with three refinements. IEEE TPAMI **47**(5), 3935–3946 (2025)
7. Ezzeldin, Y., Yan, S., He, C., Ferrara, E., Avestimehr, S.: Fairfed: Enabling group fairness in federated learning. In: AAAI. pp. 7494–7502 (2023)
8. Fallah, A., Mokhtari, A., Ozdaglar, A.: Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In: NeurIPS. pp. 3557–3568 (2020)
9. Gao, L., Fu, H., Li, L., Chen, Y., Xu, M., Xu, C.Z.: Feddc: Federated learning with non-iid data via local drift decoupling and correction. In: CVPR. pp. 10102–10111 (2022)
10. Ghosh, A., Chung, J., Yin, D., Ramchandran, K.: An efficient framework for clustered federated learning. In: NeurIPS. pp. 19586–19597 (2020)
11. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: ICML. pp. 1856–1865 (2018)
12. Hamman, F., Dutta, S.: Demystifying local and global fairness trade-offs in federated learning using partial information decomposition. In: ICLR (2024)
13. He, J., Chen, W., Zhang, X.: Fedaa: a reinforcement learning perspective on adaptive aggregation for fair and robust federated learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 17085–17093 (2025)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
15. Hu, M., Zhou, P., Yue, Z., Ling, Z., Huang, Y., Li, A., Liu, Y., Lian, X., Chen, M.: Fedcross: Towards accurate federated learning via multi-model cross-aggregation. In: ICDE. pp. 2137–2150 (2024)
16. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. In: ICML. pp. 5132–5143 (2020)
17. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009)
18. Lai, F., Zhu, X., Madhyastha, H., Chowdhury, M.: Oort: Efficient federated learning via guided participant selection. In: OSDI. pp. 19–35 (2021)
19. Lee, G., Jeong, M., Shin, Y., Bae, S., Yun, S.Y.: Preservation of the global knowledge by not-true distillation in federated learning. In: NeurIPS (2022)

20. Li, Q., He, B., Song, D.: Model-contrastive federated learning. In: CVPR. pp. 10713–10722 (2021)
21. Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z.: On the convergence of fedavg on non-iid data. In: ICLR. pp. 1–26 (2020)
22. Liu, J., Jia, J., Che, T., Huo, C., Ren, J., Zhou, Y., Dai, H.: Fedasmu: Efficient asynchronous federated learning with dynamic staleness-aware model update. In: AAAI. pp. 13900–13908 (2024)
23. Mao, W., Lu, X., Jiang, Y., Zheng, H.: Joint client selection and bandwidth allocation of wireless federated learning by deep reinforcement learning. IEEE Transactions on Services Computing **17**(1), 336–348 (2024)
24. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: AISTATS. pp. 1273–1282 (2017)
25. Mohri, M., Sivek, G., Suresh, A.T.: Agnostic federated learning. In: ICML. pp. 4615–4625 (2019)
26. Nguyen, N.H., Nguyen, P.L., Nguyen, T.T., Nguyen, T.T.T., Nguyen, D.L., Nguyen, D.T., Gonçalves, L.M., Pereira, V.: Feddrl: Deep reinforcement learning-based adaptive aggregation for non-iid data in federated learning. In: ICPP. pp. 73:1–73:11 (2022)
27. Pan, Z., Li, C., Yu, F., Wang, S., Wang, H., Tang, X., Zhao, J.: Fedlf: Layer-wise fair federated learning. In: AAAI. pp. 14527–14535 (2024)
28. Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konecny, J., Kumar, S., McMahan, B.: Adaptive federated optimization. In: ICLR (2021)
29. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
31. Sun, B., Song, X., Tu, Y., Liu, M.: Fedagent: Federated learning on non-iid data via reinforcement learning and knowledge distillation. Expert Syst. Appl. **285**, 127973 (2025)
32. Wang, J., Li, Y., Shao, Y., Xue, Z., Guan, Z., Li, A., Ye, G.: Reinforcement active client selection for federated heterogeneous graph learning. In: AAAI. pp. 21117–21125 (2025)
33. Wu, N., Kuang, Z., Yan, Z., Yu, L.: From optimization to generalization: Fair federated learning against quality shift via inter-client sharpness matching. In: IJCAI. pp. 5199–5207 (2024)
34. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
35. Zhang, J., Liu, Y., Hua, Y., Wang, H., Song, T., Xue, Z., Ma, R., Cao, J.: Pfllib: A beginner-friendly and comprehensive personalized federated learning library and benchmark. Journal of Machine Learning Research **26**(50), 1–10 (2025)
36. Zhang, J., Li, Z., Li, B., Xu, J., Wu, S., Ding, S., Wu, C.: Federated learning with label distribution skew via logits calibration. In: ICML. pp. 26311–26329 (2022)
37. Zhang, J., Wang, H., Li, F., Li, J., Zhou, N., Li, Q.: Contrastive reinforcement learning for adaptive client selection in federated learning. Procedia Computer Science **264**, 262–269 (2025)
38. Zhang, S.Q., Lin, J., Zhang, Q.: A multi-agent reinforcement learning approach for efficient client selection in federated learning. In: AAAI. pp. 9091–9099 (2022)