FedEvoQ: Evolutionary Diversity Probing for Data-Quality-Aware Federated Aggregation

Leming Wu[1], **Yaochu Jin**[2*], **Han Yu**[3] and **Riquan Zhang**[1*]
[1]School of Statistics and Data Science, Shanghai University of International Business and Economics, Shanghai, China
[2]School of Engineering, Westlake University, Hangzhou, China
[3]College of Computing and Data Science, Nanyang Technological University (NTU), Singapore
lemingwu@suibe.edu.cn, jinyaochu@westlake.edu.cn, han.yu@ntu.edu.sg, rqzhang@suibe.edu.cn

Abstract

Federated learning (FL) enables collaborative training of deep models while keeping raw data on local devices. A critical factor affecting the global performance is server-side aggregation. The most common strategy weights client updates by the amount of local data, which is often unreliable as the server cannot inspect data. More importantly, it ignores substantial variations in data quality (e.g., label noise) across heterogeneous clients. To address this issue, we propose a data-Quality-aware aggregation framework by introducing an Evolutionary-computation-inspired design into Federated learning (`FedEvoQ`), with a lightweight dual-branch architecture. Each client trains a main-task branch as usual, while a frozen auxiliary branch (EvoQ) performs evolutionary-style controlled sampling with crossover and mutation at the input and representation level to probe data diversity and reliability. EvoQ outputs a compact quality score that jointly reflects learnability (loss-based signal), geometric consistency (stability under perturbations), representation diversity, and out-of-distribution (OOD) deviation, enabling the server to compute quality-only aggregation weights. Extensive experiments on CIFAR10, CIFAR100, and PathMNIST under noisy and Non-independent and identically distributed (Non-IID) settings demonstrate that `FedEvoQ` consistently outperforms 10 state-of-the-art (SOTA) federated aggregation methods, improving the average test accuracy of the global model by over 2% with modest overhead.

1 Introduction

In federated learning (FL), collaborative training mainly involves two steps: clients train local models using their private data and upload the learned model parameters to the server, and the server performs a weighted aggregation of the received client parameters to obtain a global model [Jin *et al.*, 2023]. Since the final effectiveness of collaboration depends on the performance of the global model, server-side weighted aggregation is critical to federated training [Fan *et al.*, 2025].

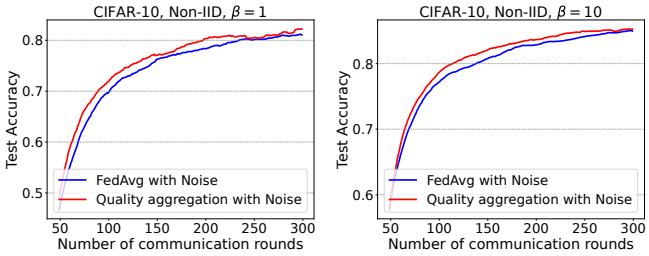


Figure 1: Global model test accuracy with noisy data: aggregation by noise-free data quantity vs. total data volume. Here, quality aggregation means aggregating according to the amount of noise-free data.

The most common weighted aggregation strategy, FedAvg [McMahan *et al.*, 2017; Li *et al.*, 2020; Wu *et al.*, 2023; Wu *et al.*, 2024], assigns weights according to the amount of data on each client. This strategy has several advantages: it can minimize the global empirical risk and prevent clients with small datasets from dominating the optimization due to high estimation bias. Additionally, it can reduce aggregation variance, improve convergence speed and generalization performance, and maintain solution stability under heterogeneous data distributions. However, weighting aggregation only by the amount of training data while ignoring its quality can cause serious issues [Wu *et al.*, 2025; Deng *et al.*, 2022; Gao *et al.*, 2023]. Our experiments show that when noisy data exists, aggregating by the total data size yields lower performance than aggregating by the amount of noise-free data, and the latter significantly improves the accuracy of the global model, as shown in Figure 1. Therefore, a fairer weighted aggregation strategy is to weight clients according to the size of their effective training data.

Existing data quality-aware aggregation methods (e.g., FedAW [Tang, 2024], FAIR [Deng *et al.*, 2021a], FedFAIM [Shi *et al.*, 2022], and other quality-driven auto-weighting approaches) introduce quality signals to suppress the negative impact of low-quality or noisy clients on the global model. However, they typically suffer from several limitations, including insufficient comparability of quality measurements

*Corresponding author.

(metric bias under Non-IID data [Li *et al.*, 2022]), additional evaluation cost and system complexity, and potential manipulability and fairness degradation (where long-tail or disadvantaged clients may be overly down-weighted). In particular, methods that use proxy signals such as training loss [Yi *et al.*, 2022] or validation-set performance to measure client data quality often require extra evaluation mechanisms or additional evaluation data, are sensitive to noisy or anomalous data, and can be vulnerable to strategic manipulation.

To address these limitations, we introduce evolutionary-computation principles into FL and propose `FedEvoQ` with three key components:

- **Lightweight dual-branch client design.** Each client maintains a dual-branch architecture, where the main branch is optimized for the target task, while a frozen auxiliary branch conducts evolutionary-style probing through controlled sampling, crossover, and mutation.
- **Compact data-quality scoring.** We compute a compact data-quality score Q_k from the probing outputs to summarize the reliability of each client update—capturing learnability, stability, representation diversity, and OOD deviation—without accessing any raw data.
- **Server-side quality-only weighted aggregation.** On the server, we perform quality-only weighted aggregation by mapping $\{Q_k\}$ to aggregation weights, thereby down-weighting low-quality or noisy client updates. Across ten SOTA aggregation baselines, `FedEvoQ` improves the average global accuracy by more than 2%.

2 Related Work

2.1 Data quality-based aggregation

Sample-size averaging is efficient but cannot separate clean information from Non-IID drift, label noise, or poisoning. Recent work therefore injects quality and credibility into aggregation: malicious label corruption can be exposed by recovering latent feature distributions and flagging abnormal clients [Jiang *et al.*, 2023], or by multi-level quality regularization that down-weights low-quality instances, features, or clients during training [Zhang *et al.*, 2024]. Loss and performance proxies are also used to infer contribution and support quality-weighted aggregation [Shyn *et al.*, 2021; Shi *et al.*, 2022; Liu *et al.*, 2022], while relevance-aware data selection improves global generalization when local data are partly unrelated to the target task [Nagalapatti *et al.*, 2022]. Distributional valuation further aggregates via principled distances (e.g., Wasserstein barycenter) to obtain transparent contribution-aware weights [Li *et al.*, 2024]. However, quality inference may still be possible under secure aggregation, creating privacy risk and making "quality-aware" aggregation harder to deploy safely [Pejó and Biczók, 2023].

2.2 Selection, incentives, and robust aggregation

Quality-aware FL is often coupled with participation control: incentive and recruitment mechanisms integrate estimated learning quality into payments and aggregation to attract useful clients [Deng *et al.*, 2021a; Deng *et al.*, 2022], and automated selection optimizes quality under budget and efficiency constraints [Deng *et al.*, 2021b; Saha *et al.*, 2022]. Robust and heterogeneity-aware policies suppress unreliable clients via confidence re-weighting, robust aggregation, or clustering-based weighting when quality and distribution heterogeneity coexist [Fang and Ye, 2022; Fang and Ye, 2024; Li *et al.*, 2025; Ma *et al.*, 2025]. Other lines correct label noise explicitly [Lin *et al.*, 2025] or reduce long-term harm via server-side filtering and federated unlearning [Wang *et al.*, 2023].

Despite progress, many methods require extra supervision or auxiliary data, rely on assumed attack ratios, or implicitly trust reported quality; some even leak relative quality under secure aggregation [Huang *et al.*, 2024]. To address these issues, `FedEvoQ` employs the EvoQ branch with evolutionary sampling, crossover, and mutation to enrich representation signals and evaluate client data quality.

3 The Proposed `FedEvoQ` Method

This section systematically describes the proposed algorithm `FedEvoQ`, with reference to the schematic framework illustrated in Figure 2.

3.1 Problem Description

In heterogeneous Non-IID FL, client data quality (e.g., diversity, coverage, label noise, and OOD ratio) strongly affects the global model's generalization. However, the server cannot access raw data and only observes locally trained parameters, making quality estimation and utilization challenging. To address this, we equip each client with a dual-branch architecture: a primary model f_θ for standard local task training and an evolutionary model g_ϕ that performs evolutionary-style sampling and augmentation (crossover and mutation) to construct a set for quality assessment and output privacy-preserving statistics that quantify local data quality. The g_ϕ is pre-trained on a small synthetic dataset. The server then uses these statistics to perform quality-weighted aggregation, aiming to reduce global risk.

Let $\mathcal{K} = \{1, \dots, K\}$ be the client set and $\mathcal{S}^t \subseteq \mathcal{K}$ the sampled clients at round t. Client k holds $D_k = \{(x_i, y_i)\}_{i=1}^{N_k}$ with $|D_k| = N_k$, and optimizes the local empirical risk

$$\mathcal{L}_k(\theta) = \frac{1}{N_k} \sum_{(x,y)\in D_k} \ell\big(f_\theta(x), y\big). \tag{1}$$

The global objective is

$$\min_\theta \sum_{k=1}^{K} p_k \, \mathbb{E}_{(x,y)\sim D_k}\big[\ell(f_\theta(x), y)\big], \quad \sum_k p_k = 1, \tag{2}$$

and the server aggregates local updates as

$$\theta^{t+1} = \sum_{k\in\mathcal{S}^t} p_k^t \, \theta_k^t. \tag{3}$$

Since p_k critically determines global performance, volume-based weighting is often suboptimal under Non-IID data because it ignores quality. The core problem is thus to learn data-quality-dependent aggregation weights from client-reported statistics, without exposing raw data.

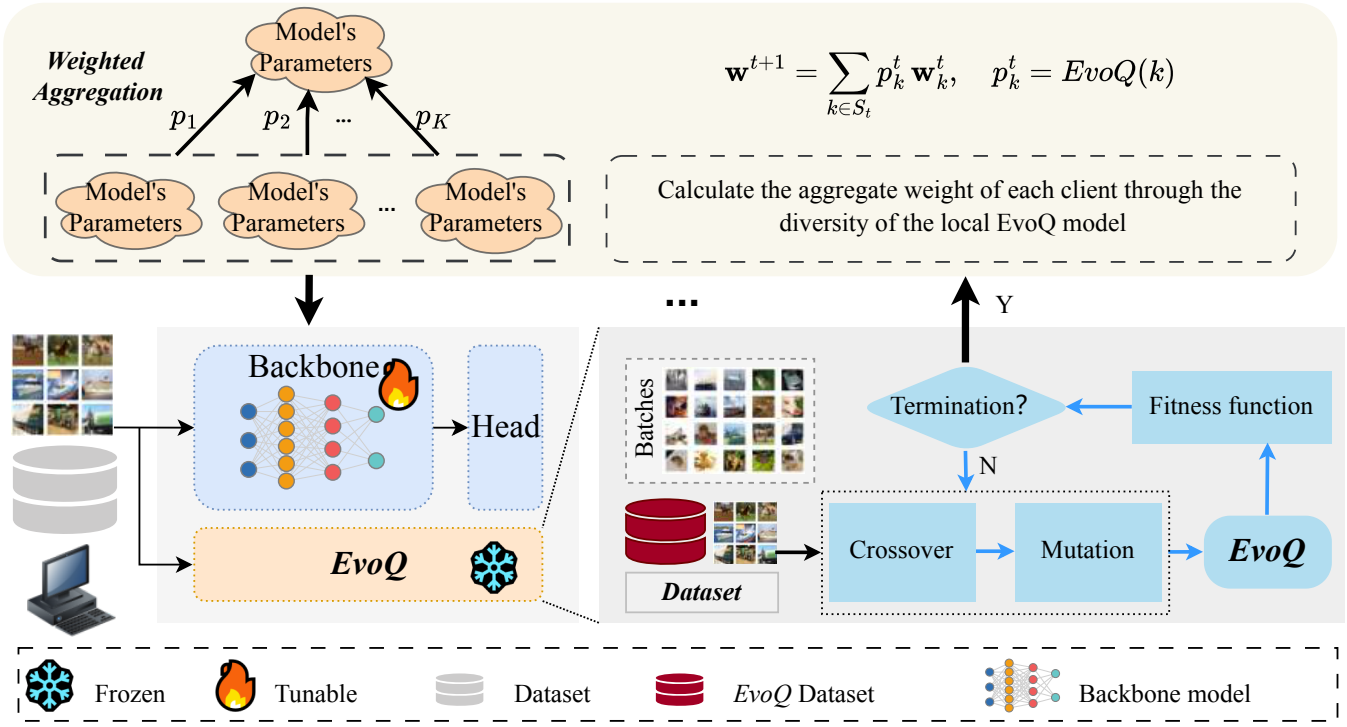


Figure 2: Framework diagram of the proposed FedEvoQ. We design a dual-branch network architecture. The Backbone main branch is used to accomplish the client's primary task. The EvoQ branch estimates representation diversity as a proxy signal of data quality and sends this signal back to the server, enabling a data-quality-aware weighted aggregation scheme based on each client's training data quality.

3.2 Proposed FedEvoQ Framework

In the proposed FedEvoQ, the model designed to assess training-data quality mainly consists of two components. One component is the backbone network, i.e., the main-task branch, which handles the primary task. The other component is EvoQ, which is responsible for evaluating the quality of the training data.

The main-task branch enables each client to accomplish the primary objective, such as classification or prediction. The data-quality evaluation branch, EvoQ, integrates evolutionary computation and uses representation diversity as a proxy signal of data quality. This signal is sent back to the server to support a data-quality-aware weighted aggregation scheme. In this deep learning network, we further define hyperparameters in the EvoQ branch, including the sampling ratio of client training data C_{Ev} and the crossover and mutation probabilities P_C and P_M, to mitigate fairness issues when evaluating data quality via diversity.

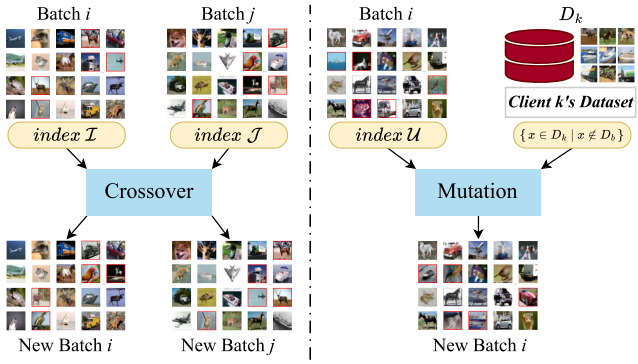


Figure 3: Illustration of the crossover and mutation operations in the EvoQ branch for client training-data quality assessment in the proposed method. The images highlighted by red boxes indicate the samples selected for crossover and mutation.

3.3 EvoQ Branch Network

Evolutionary algorithms [Jin *et al.*, 2018; Zhang *et al.*, 2025; Liu *et al.*, 2025; Zhao *et al.*, 2025] maintain the diversity of candidate solutions in a population, which encourages broad exploration of the search space and helps avoid premature convergence to local optima. Through the combined effects of selection, crossover, and mutation, high-quality and diverse solutions are continuously recombined and refined, gradually approaching the global optimum. This diversity-driven principle of evolutionary algorithms inspires our proposed EvoQ model.

Both EvoQ and the main-task branch require input samples to train the model. The difference is that the main-task branch feeds the entire local dataset into the network in a standard mini-batch manner, whereas the EvoQ branch samples the data before input. Based on the sampled inputs, EvoQ further increases the diversity of the solution space (i.e., the data space) by applying operations analogous to crossover and mutation in evolutionary algorithms. Specifically, EvoQ first randomly samples several mini-batches from the client's

Algorithm 1 FedEvoQ

Require: Total rounds T, client set $\mathcal{K}$, initial global model θ^0; local epochs E, batch size B; sampling ratio C_{Ev}, crossover prob. P_C, mutation prob. P_M; temperature τ, weight floor ϵ_p.

for $t = 0$ **to** $T-1$ **do**
  Server: sample participating clients $\mathcal{S}^t \subseteq \mathcal{K}$ and broadcast θ^t
  for all clients $k \in \mathcal{S}^t$ **in parallel do**
    Client k**:** $\theta \leftarrow \theta^t$ {Main-task local training}
    for $e = 1$ **to** E **do**
      for each mini-batch $\mathcal{B} \subset D_k$ of size B **do**
        $\theta \leftarrow \theta - \eta \nabla_\theta \ell\big(f_\theta(\mathcal{B})\big)$
      end for
    end for
    {Frozen EvoQ forward evaluation for quality}
    Sample N_b mini-batches from D_k with ratio C_{Ev} to form $\mathcal{D}_b = \{(X_i, y_i)\}_{i=1}^{N_b}$
    With prob. P_C, apply crossover across (X_i, y_i) and (X_j, y_j) by swapping indexed subsets
    With prob. P_M, apply mutation by replacing selected indices with samples re-drawn from $D_k \notin \mathcal{D}_b$
    Compute $\tilde{\mathcal{L}}^t_{\mathrm{Evo}}(k)$, $\tilde{S}^t_{\mathrm{mix}}(k)$, $\tilde{R}^t_{\mathrm{stab}}(k)$, $\tilde{D}^t_{\mathrm{Rep}}(k)$, $\tilde{O}^t_{\mathrm{odd}}(k)$ from Eq. 5, 6, 7, 8 and 9
    Upload $(\theta^t_k = \theta,\ Q^t_k)$ to the server
  end for
  Server: compute quality-only weights by Equation 10 and 11
  Server: aggregate $\theta^{t+1} \leftarrow \sum_{k\in\mathcal{S}^t} p^t_k\, \theta^t_k$
end for
return θ^T

local training data and constructs

$$\mathcal{D}_b = \{(X_i, y_i)\}_{i=1}^{N_b}, \quad X_i \in \mathbb{R}^{B\times d}, \quad y_i \in \mathbb{R}^B \tag{4}$$

where N_b denotes the number of sampled mini-batches, B is the batch size, and d is the sample dimension. These mini-batches are then fed into the frozen EvoQ for forward evaluation. During this process, crossover is applied to the sampled mini-batches with probability P_C to construct mixed batches for quality assessment. For example, samples indexed by $\mathcal{I}$ in X_i are swapped with samples indexed by $\mathcal{J}$ in X_j. Meanwhile, EvoQ performs mutation with probability P_M, i.e., it samples a subset of data from outside $\mathcal{D}_b$ to update the current mini-batch. The sampling, crossover, and mutation operations of the EvoQ branch are illustrated in Figure 3.

We define the quality score of client k, denoted by Q_k, as a monotonic mapping of the EvoQ fitness function. We first construct the fitness:

$$F_k = a\left(1 - \tilde{\mathcal{L}}_{\mathrm{Evo}}\right) + b\, C_k + c\, I_k \tag{5}$$

where $\tilde{\mathcal{L}}_{\mathrm{Evo}}$ is the averaged EvoQ loss after client-side controlled sampling, crossover and mutation. The term

$$C_k = \zeta_1 \tilde{S}_{\mathrm{mix}} + \zeta_2 \tilde{R}_{\mathrm{stab}} \tag{6}$$

is the structural-consistency component, where $\tilde{S}_{\mathrm{mix}}$ is a linear-consistency metric for feature interpolation:

$$\tilde{S}_{\mathrm{mix}} \propto \mathbb{E}\big[\big\| g_\phi(\tilde{x}_{i\leftarrow j}) - \big(\rho\, g_\phi(x_i) + (1-\rho)\, g_\phi(x_j)\big)\big\|_2\big]^{-1}, \tag{7}$$

It measures the model's linear interpolability along the connected path between samples. $\tilde{R}_{\mathrm{stab}}$ denotes prediction stability, defined as the negative variance of predictions under data augmentation or small perturbations, after normalization, a larger value indicates higher stability. The term

$$I_k = \xi_1 \tilde{D}_{\mathrm{Rep}} - \xi_2 \tilde{O}_{\mathrm{odd}} \tag{8}$$

is the information–compatibility component. Here, $\tilde{D}_{\mathrm{Rep}}$ is the representation diversity. It can be instantiated by the spectral dispersion of the feature covariance at an intermediate layer (e.g., the entropy of covariance eigenvalues or the log-determinant), or by the mean inter-cluster distance based on prototypes, reflecting the information content and coverage of the local data. $\tilde{O}_{\mathrm{odd}}$ is an OOD deviation measure, which can be determined by thresholding the Mahalanobis distance in the feature space with respect to global prototypes or a reference distribution; a larger value indicates a stronger deviation from the global distribution and thus lower quality. The coefficients $a, b, c, \zeta_1, \zeta_2, \xi_1, \xi_2 \geq 0$ can be selected using a validation set or updated via meta-learning. Finally, we compress the fitness into $(0, 1)$ by

$$Q_k = \sigma(F_k) \tag{9}$$

where $\sigma(\cdot)$ is the Sigmoid function. The Q_k is then used by the server as the basis for quality-aware aggregation weights.

$\tilde{\mathcal{L}}_{\mathrm{Evo}}$, C_k, and I_k measure learnability, geometric consistency and effective informativeness under distribution compatibility, respectively. In Non-IID and noisy scenarios, they are linked to: 1) lower variance of local gradients and better optimizability, 2) stability-enhanced generalization with smoother decision boundaries, and 3) stronger compatibility with the global distribution, improved transferability, and increased Fisher information. Hence, Q_k serves as an approximate reliability indicator of client updates toward the global optimum. For quality-aware aggregation, under mild assumptions, our weighting admits an inverse-variance-style reliability interpretation (Appendix), improving convergence stability and global accuracy against noise and a small fraction of malicious clients. Moreover, $\tilde{O}_{\mathrm{odd}}$ in I_k downweights samples that are diverse yet incompatible, mitigating bias toward external distributions and preserving the robustness–coverage trade-off.

Given the quality score $Q^t_k \in (0, 1)$, we define the aggregation weights solely based on quality by

$$p^t_k = \frac{\exp(\tau Q^t_k)}{\sum_{j\in\mathcal{S}^t} \exp(\tau Q^t_j)} \tag{10}$$

, where $\tau > 0$ is a temperature controlling how sharply the server focuses on high-quality clients. To avoid vanishing weights and improve stability, one may further apply a floor $\epsilon_p > 0$:

$$\tilde{p}^t_k = \max(p^t_k, \epsilon_p), \qquad p^t_k \leftarrow \frac{\tilde{p}^t_k}{\sum_{j\in\mathcal{S}^t} \tilde{p}^t_j}. \tag{11}$$

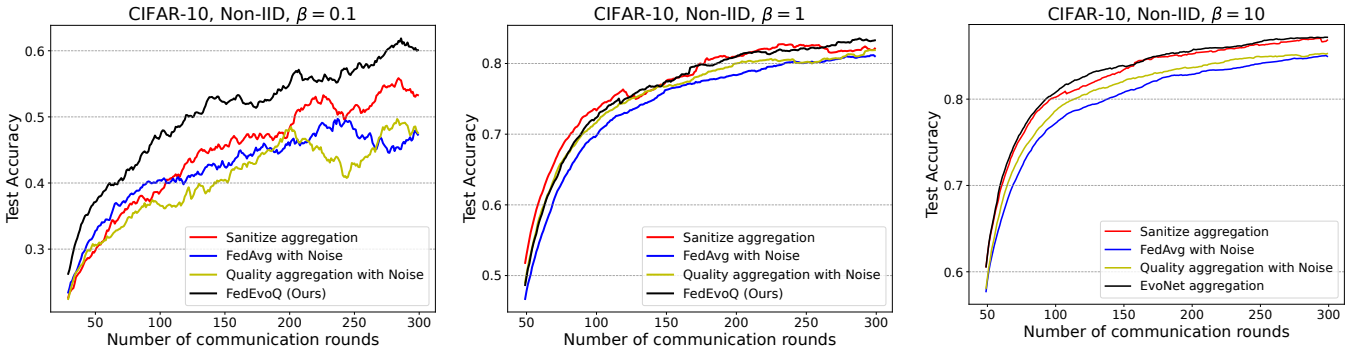


Figure 4: Comparison of test accuracy on CIFAR-10 with Gaussian noise: standard data-volume aggregation, data-volume aggregation after removing noisy samples, FedAvg on noise-free CIFAR-10 (Sanitize aggregation), and the proposed FedEvoQ.

Variant	CIFAR10						CIFAR100						PathMNIST					
	0.1	0.5	1	2	5	10	0.1	0.5	1	2	5	10	0.1	0.5	1	2	5	10
FedAvg (w/o quality)	48.90 ±0.84	81.30 ±0.06	83.21 ±0.04	86.01 ±0.01	86.16 ±0.02	86.56 ±0.01	48.94 ±0.01	53.54 ±0.01	53.64 ±0.01	52.88 ±0.01	51.12 ±0.01	51.21 ±0.01	24.71 ±0.91	78.39 ±0.03	78.66 ±0.01	78.56 ±0.06	81.22 ±0.01	80.41 ±0.01
Random sampling only	53.81 ±0.57	82.16 ±0.09	83.24 ±0.03	85.97 ±0.05	86.61 ±0.04	86.97 ±0.02	49.27 ±0.03	53.78 ±0.04	54.86 ±0.02	55.56 ±0.03	55.78 ±0.05	55.22 ±0.02	46.51 ±0.73	78.59 ±0.04	78.96 ±0.03	79.31 ±0.04	81.26 ±0.01	82.94 ±0.03
w/o crossover (P_C=0)	54.09 ±0.67	82.24 ±0.06	83.66 ±0.02	86.14 ±0.03	86.78 ±0.03	87.93 ±0.02	50.33 ±0.04	54.83 ±0.03	55.73 ±0.01	55.85 ±0.02	56.49 ±0.04	56.67 ±0.03	47.32 ±0.41	78.82 ±0.03	79.39 ±0.01	80.71 ±0.03	81.79 ±0.02	84.03 ±0.06
w/o mutation (P_M=0)	54.67 ±0.34	82.54 ±0.10	83.82 ±0.01	86.42 ±0.01	86.39 ±0.02	88.03 ±0.04	49.49 ±0.01	55.09 ±0.03	55.69 ±0.01	55.89 ±0.02	56.29 ±0.06	56.59 ±0.03	48.87 ±0.82	78.85 ±0.01	79.37 ±0.01	81.19 ±0.03	82.17 ±0.02	83.70 ±0.04
only $(1-\tilde{\mathcal{L}}_{\text{Evo}})$	54.32 ±0.28	82.09 ±0.07	83.44 ±0.03	86.14 ±0.02	86.45 ±0.05	87.34 ±0.01	48.97 ±0.02	54.32 ±0.04	55.17 ±0.03	54.52 ±0.01	55.76 ±0.02	55.43 ±0.04	46.29 ±0.96	78.93 ±0.05	79.14 ±0.03	80.48 ±0.02	81.68 ±0.04	83.30 ±0.05
remove OOD penalty (ξ_2=0)	54.67 ±0.47	82.48 ±0.08	83.79 ±0.02	86.76 ±0.4	87.01 ±0.03	88.16 ±0.03	49.96 ±0.05	54.66 ±0.03	55.46 ±0.02	56.05 ±0.01	56.92 ±0.05	56.96 ±0.02	45.84 ±0.93	77.93 ±0.07	79.45 ±0.02	80.88 ±0.03	82.12 ±0.01	84.75 ±0.06
FedEvoQ (Full)	**55.44 ±0.53**	**83.44 ±0.04**	**84.50 ±0.02**	**87.53 ±0.01**	**87.29 ±0.04**	**88.81 ±0.01**	**51.02 ±0.01**	**55.98 ±0.04**	**56.29 ±0.01**	**56.93 ±0.01**	**57.29 ±0.01**	**57.65 ±0.07**	**49.97 ±0.33**	**79.38 ±0.01**	**79.98 ±0.02**	**81.87 ±0.01**	**83.32 ±0.01**	**84.96 ±0.03**

Table 1: Ablation study of FedEvoQ (Test Acc., %). Results are reported under different Non-IID levels $\beta \in \{0.1, 0.5, 1, 2, 5, 10\}$ on CIFAR10, CIFAR100, and PathMNIST.

3.4 Secure and Efficient Dual-Branch Training Mechanism

To estimate client data quality more accurately without extra privacy risk or communication overhead, we employ a dual-branch design. The Main branch is trained locally and aggregated as in standard FL. The EvoQ branch is fully frozen (no updates and no parameter transmission) and is used only for forward inference to compute the quality score Q_k. In each round, the client uploads only $\nabla_{\theta_k^{\text{Main}}}$, Q_k, and a small set of statistical summaries.

This design offers two benefits. Privacy: gradient inversion attacks typically exploit exposed gradients and a traceable computation graph. Freezing EvoQ removes its backward path and eliminates multi-branch gradient exposure, leaving only main-branch updates and a few scalars, which substantially reduces reconstruction risk. Communication: it avoids transmitting EvoQ-related parameters, reducing the per-round cost from

$$O\left(|\theta_{\text{Main}}| + |\theta_{\text{EvoQ}}|\right), \tag{12}$$

to $O\left(|\theta_{\text{Main}}|\right) + O(1)$. The server then performs quality-aware aggregation using weights derived from Q_k, balancing measurable quality, controlled risk, and low cost.

4 Experimental Evaluation

4.1 Experiment Settings

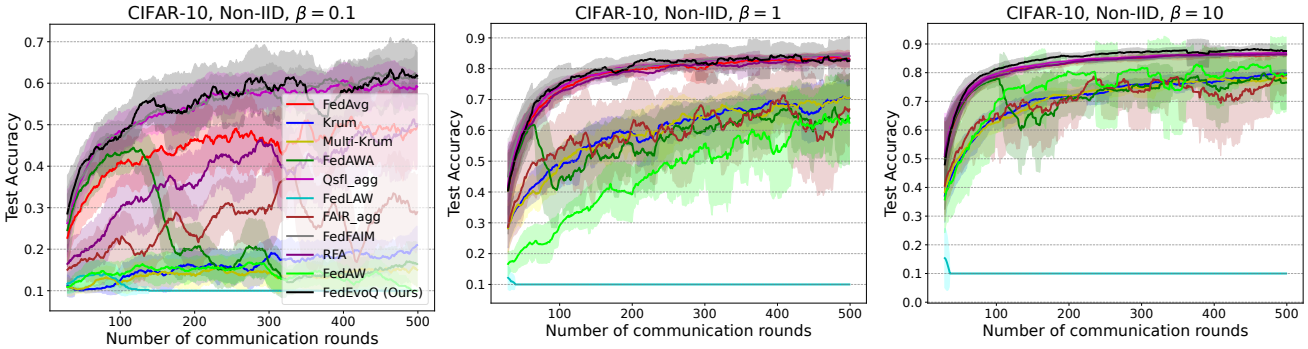
Datasets and Models. We select three datasets: CIFAR-10 [Krizhevsky *et al.*, 2009], CIFAR-100, and PathMNIST [Kather *et al.*, 2019]. The first two are widely used benchmark RGB image datasets for classification, containing 10 and 100 classes, respectively. PathMNIST is a pathology image classification subset from the MedMNIST [Yang *et al.*, 2023] collection, which provides labels for 9 classes. For the above three datasets, we adopt VGG16 [Simonyan and Zisserman, 2014], ResNet34, and ResNet18 [He *et al.*, 2016] as the model architectures in our experiments, respectively.

Implementation Details. In the FL setup, we set the total number of clients to 100. In each communication round, 10 clients are randomly selected to participate in training, corresponding to a client participation rate of 10%. All experiments use a default configuration: $C_{Ev} = 0.2$, $N_b = 10$, $B = 64$, $P_C = 0.5$, $P_M = 0.3$, $r_M = 0.2$, and $\rho \sim \text{Beta}(1, 1)$. We set $a = b = c = 1$, $\zeta_1 = \zeta_2 = 0.5$, and $\xi_1 = \xi_2 = 0.5$, with per-round normalization across participating clients. For aggregation, $\tau = 5$ and $\epsilon_p = 0.01$. We set the OOD threshold to the 95th percentile. Local training uses $E = 5$ epochs with SGD (lr= 0.01, momentum= 0.9) or Adam (lr= 0.001). Non-IID partitions are generated by a Dirichlet distribution, where β controls heterogeneity and a smaller β indicates higher Non-IID. All experiments are implemented in PyTorch and conducted on NVIDIA H20 GPUs.

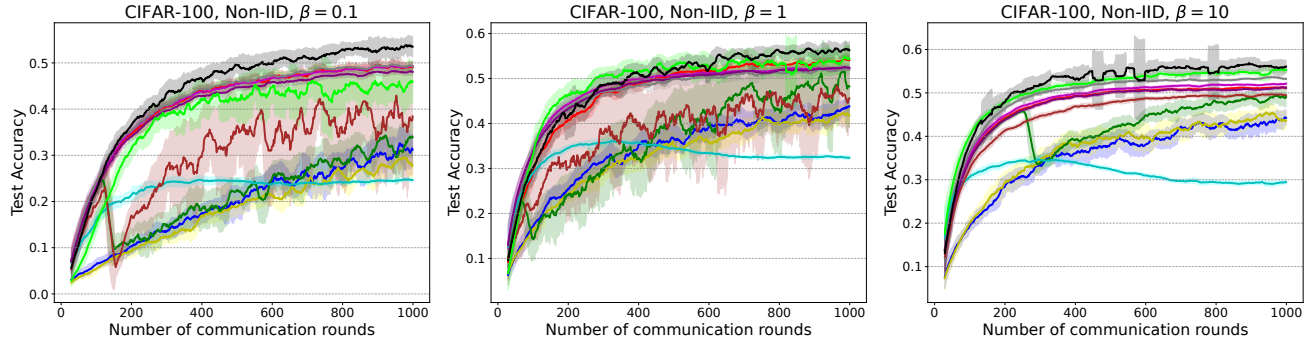
To simulate realistic data conditions, we add Gaussian noise to the training data. The noise level is governed by three factors: the proportion of noisy clients, the fraction of noisy samples on each noisy client, and the noise magnitude (variance). In our experiments, 50% of clients are designated

Algorithms	Aggregation Rule	Additional Assumptions	Server Cost	Non-IID	Incentive
FedAvg [McMahan *et al.*, 2017]	Sample-size-weighted averaging	Client sample counts known	Low	Medium	None
Krum [Blanchard *et al.*, 2017]	Majority-consistent update selection	Byzantine bound, pairwise distances	High ($O(n^2)$)	Medium	None
Multi-Krum [Blanchard *et al.*, 2017]	Multi-update selection averaging	Byzantine bound, pairwise distances	High	Medium	None
RFA [Pillutla *et al.*, 2022]	Geometric-median aggregation	Iterative median optimization	Med–High	Medium	None
FedAWA [Shi *et al.*, 2025]	Alignment-aware dynamic reweighting	Update-direction metrics required	Medium	Strong	None
FedLAW [Li *et al.*, 2023]	Learned aggregation weights and shrinkage	Weight learning, consistency checks	Medium	Strong	None
FedAW [Tang, 2024]	Quality-aware weighting	Quality and reliability signals required	Medium	Strong	Weak
QSFL [Yi *et al.*, 2022]	Eligibility screening, uplink sparsification	Eligibility criteria, compression settings	Medium	Medium	None
FAIR [Deng *et al.*, 2021a]	Loss-aware upweighting for tail fairness	Client-side loss reporting	Medium	Med–Strong	Strong
FedFAIM [Shi *et al.*, 2022]	Contribution-aware aggregation, incentive allocation	Contribution eval, incentive mechanism	Med–High	Medium	Strong
`FedEvoQ`	Quality-aware weighting	Local EvoQ-based quality score	Low	Strong	Strong

Table 2: Comparison of aggregation methods in FL.



(a) CIFAR-10



(b) CIFAR-100

Figure 5: Test accuracy convergence of `FedEvoQ` and ten federated aggregation strategies on CIFAR-10 and CIFAR-100 under different Non-IID settings.

as noisy, 30% of samples on each noisy client are perturbed, and the injected noise follows a Gaussian distribution with variance 0.5.

Comparison Methods. To comprehensively evaluate the robustness and fairness of the proposed method under distribution heterogeneity and quality heterogeneity, we select ten representative FL aggregation frameworks from recent years as baselines. All methods are evaluated under the same training setting. The selected baselines are listed in Table 2.

4.2 Performance Comparison

First, we evaluate the proposed `FedEvoQ` on CIFAR-10 under the classical FedAvg setting. Under three Non-IID levels ($\beta = \{0.1, 1, 10\}$), we compare four settings: (i) FedAvg with Gaussian noise ($\sigma = 0.5$) and data quantity weighting, (ii) FedAvg with the same noise but weighted by the post-cleaning effective quantity of data, (iii) sanitize aggregation (training on cleaned data and aggregating by the quantity of data), and (iv) `FedEvoQ` under the same noise. The results are reported in Fig. 4. As shown in Fig. 4, when noisy data exist, weighting FedAvg by the effective sample size after noise removal consistently outperforms standard data quantity weighting, because the effective sample size serves as a credibility proxy and reduces the impact of noisy clients. More importantly, `FedEvoQ` achieves the best accuracy across all Non-IID levels, regardless of

	CIFAR10		CIFAR100		PathMNIST	
Variant	0.1	10	0.1	10	0.1	10
Full ($\tau = \tau_0$, $\epsilon_p = 0$)	**55.44** ±**0.53**	**88.81** ±**0.01**	**51.02** ±**0.01**	57.65 ±0.07	**49.97** ±**0.33**	84.96 ±0.03
$\tau = 1$ (w/ floor)	54.47 ±0.71	88.57 ±0.02	49.17 ±0.03	56.86 ±0.06	47.64 ±0.52	84.33 ±0.05
$\tau = 5$ (w/ floor)	54.43 ±0.49	88.61 ±0.01	49.82 ±0.02	**57.96** ±**0.04**	49.58 ±0.45	**85.11** ±**0.03**
$\tau = 10$ (w/ floor)	53.72 ±0.86	88.54 ±0.03	50.26 ±0.03	56.98 ±0.04	46.62 ±0.61	84.72 ±0.03
w/o floor ($\epsilon_p = 0$)	51.47 ±0.52	86.94 ±0.02	49.38 ±0.04	55.89 ±0.03	46.67 ±0.76	83.26 ±0.02

Table 3: Sensitivity of quality-only weighting: temperature τ and weight floor ϵ_p. Results are averaged over $\beta \in \{0.1, 10\}$.

whether noisy samples are removed. For $\beta = 1$, after 300 rounds, the converged accuracies are 81.02% (noisy FedAvg), 82.17% (effective-sample weighting), 83.46% (sanitize aggregation), and 84.50% (FedEvoQ), corresponding to gains of 3.48%, 2.33%, and 1.04%, respectively. This improvement comes from assigning smaller aggregation weights to low-quality (high-noise) clients, which suppresses contaminated updates and drives the solution closer to the clean-data optimum. Compared with sanitize aggregation, FedEvoQ further avoids the bias and information loss caused by discarding samples under Non-IID data, since it suppresses noise via client-level reliability weighting.

To further validate FedEvoQ, we compare it with 10 SOTA aggregation methods in Table 2. As shown in Figure 5, on CIFAR-10 and CIFAR-100 with $\beta = 0.1, 1, 10$, FedEvoQ consistently achieves the highest test accuracy, with larger margins under severe heterogeneity ($\beta = 0.1$). For example, on CIFAR-100 with $\beta = 0.1$, FedEvoQ reaches 51.02%, outperforming the strongest baselines FedFAIM (47.25%), FedAvg (47.04%), QSFL (46.97%), and RFA (44.89%). Overall, FedEvoQ shows a clear and robust advantage over all compared methods.

4.3 Ablation Study

Table 1 demonstrates that the full model consistently achieves the best performance on CIFAR10, CIFAR100 and PathMNIST across all Non-IID levels $\beta \in \{0.1, 0.5, 1, 2, 5, 10\}$, validating the effectiveness of quality-aware aggregation. Even without evolutionary recombination, Random sampling only markedly outperforms FedAvg, indicating that quality-based weighting is the primary source of gains (e.g., PathMNIST at $\beta = 0.1$: 46.51% vs. 24.71%). Disabling either crossover or mutation reduces accuracy, while using both yields the highest results, confirming their complementary contribution. Moreover, using only the learnability term $1 - \tilde{\mathcal{L}}_{\text{Evo}}$ underperforms the full score, and removing the OOD penalty by setting $\xi_2 = 0$ causes a clear drop in challenging regimes (e.g., PathMNIST at $\beta = 0.1$: 49.97% to 45.84%), supporting the necessity of penalizing diverse-but-incompatible clients.

Table 3 studies the sensitivity of the quality-only weighting to the temperature τ and the weight floor ϵ_p under representative $\beta \in \{0.1, 10\}$. With ϵ_p>0, performance is relatively stable across τ, and moderate τ is generally preferable; in contrast, removing the floor ($\epsilon_p = 0$) consistently degrades accuracy (e.g., CIFAR10: 55.44% to 51.47% at β=0.1 and 88.81% to 86.94% at β=10), indicating that the floor is crucial to prevent weight collapse and stabilize aggregation. Overall, these ablations confirm that the proposed score composition and the bounded, temperature-controlled weighting are both necessary for robust aggregation.

4.4 Computation and Communication Costs

Table 4 reports the average time per communication round under identical settings. FedEvoQ adds a lightweight, forward-only EvoQ evaluation for quality scoring, hence the overhead over FedAvg is small on CIFAR10 (e.g., β=0.1: 25.40s vs. 24.16s) and remains competitive with most baselines across datasets, while being far cheaper than heavy methods such as FedLAW (e.g., CIFAR100: 39.2s vs. 80-115s). Communication overhead is negligible since only the main-branch update is transmitted and EvoQ is frozen; each client additionally uploads only a scalar quality score Q_k (and a few summaries), so the uplink complexity remains $O(|\theta_{\text{Main}}|) + O(1)$.

Times (s)	CIFAR10			CIFAR100			PathMNIST		
	0.1	1	10	0.1	1	10	0.1	1	10
FedAvg	24.16	20.23	19.85	26.59	30.98	32.62	14.91	15.16	12.01
Krum	20.22	23.05	25.82	28.17	22.47	22.60	11.11	11.08	15.04
Multi-Krum	19.39	22.96	22.82	22.54	20.00	34.03	15.26	11.73	9.74
FedAWA	21.32	23.77	23.64	32.58	37.38	38.97	14.85	14.95	11.73
QSFL	24.17	24.04	23.58	28.24	31.73	45.85	14.62	15.10	11.89
FedLAW	122.33	120.20	104.52	115.13	81.48	79.96	57.57	49.93	49.55
FAIR	20.26	25.86	26.09	28.52	34.52	34.85	15.09	15.89	12.35
FedFAIM	21.35	23.77	23.54	29.54	34.70	34.68	32.97	22.39	22.08
RFA	22.49	28.41	25.67	29.22	34.52	34.89	15.19	15.05	11.74
FedAW	11.40	16.04	16.72	21.51	21.46	25.38	10.26	11.09	8.70
FedEvoQ (ours)	25.40	25.53	24.28	39.56	39.28	39.21	19.86	13.73	18.72

Table 4: Running time (s) per communication round (average across algorithms under identical experimental settings) under different Non-IID levels ($\beta = \{0.1, 1, 10\}$) on CIFAR10, CIFAR100, and PathMNIST.

5 Conclusions and Future Work

In this paper, we address the issue of unreliable aggregation in FL under heterogeneous and noisy Non-IID data. We propose FedEvoQ, a data-quality-aware aggregation framework that weights clients using quality signals only. The key idea is a dual-branch client design: the main branch performs local training, while a frozen auxiliary branch uses evolutionary-style probing to compute a compact quality score measuring learnability, geometric consistency, representation diversity, and OOD compatibility. The server then conducts quality-only weighted aggregation based on these scores. Experimental results demonstrate that FedEvoQ consistently improves global generalization, with moderate overhead and negligible extra communication.

In subsequent research, we will study robustness against strategic manipulation of quality signals and extend FedEvoQ to larger models and more diverse FL scenarios.

Acknowledgements

This work is funded in part by an International Collaboration Fund for Creative Research of National Science Foundation of China (NSFC ICFCRT) under the Grant no. W2441019; National Natural Science Foundation of China (62136003, 12531013 and 12371272); the Ministry of Education, Singapore, under its Academic Research Fund Tier 1 (RG101/24); the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR, as well as supported by Alibaba Group and NTU Singapore through Alibaba-NTU Global e-Sustainability CorpLab (ANGEL).

References

[Blanchard *et al.*, 2017] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017.

[Deng *et al.*, 2021a] Yongheng Deng, Feng Lyu, Ju Ren, Yi-Chao Chen, Peng Yang, Yuezhi Zhou, and Yaoxue Zhang. Fair: Quality-aware federated learning with precise user incentive and model aggregation. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021.

[Deng *et al.*, 2021b] Yongheng Deng, Feng Lyu, Ju Ren, Huaqing Wu, Yuezhi Zhou, Yaoxue Zhang, and Xuemin Shen. Auction: Automated and quality-aware client selection framework for efficient federated learning. *IEEE Transactions on Parallel and Distributed Systems*, 33(8):1996–2009, 2021.

[Deng *et al.*, 2022] Yongheng Deng, Feng Lyu, Ju Ren, Yi-Chao Chen, Peng Yang, Yuezhi Zhou, and Yaoxue Zhang. Improving federated learning with quality-aware user incentive and auto-weighted model aggregation. *IEEE Transactions on Parallel and Distributed Systems*, 33(12):4515–4529, 2022.

[Fan *et al.*, 2025] Tao Fan, Hanlin Gu, Xuemei Cao, Chee Seng Chan, Qian Chen, Yiqiang Chen, Yihui Feng, Yang Gu, Jiaxiang Geng, Bing Luo, et al. Ten challenging problems in federated foundation models. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[Fang and Ye, 2022] Xiuwen Fang and Mang Ye. Robust federated learning with noisy and heterogeneous clients. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10072–10081, 2022.

[Fang and Ye, 2024] Xiuwen Fang and Mang Ye. Noise-robust federated learning with model heterogeneous clients. *IEEE Transactions on Mobile Computing*, 2024.

[Gao *et al.*, 2023] Yuanyuan Gao, Lei Zhang, Lulu Wang, Kim-Kwang Raymond Choo, and Rui Zhang. Privacy-preserving and reliable decentralized federated learning. *IEEE Transactions on Services Computing*, 16(4):2879–2891, 2023.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Huang *et al.*, 2024] Wenke Huang, Mang Ye, Zekun Shi, Guancheng Wan, He Li, Bo Du, and Qiang Yang. Federated learning for generalization, robustness, fairness: A survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9387–9406, 2024.

[Jiang *et al.*, 2023] Yifeng Jiang, Weiwen Zhang, and Yanxi Chen. Data quality detection mechanism against label flipping attacks in federated learning. *IEEE Transactions on Information Forensics and Security*, 18:1625–1637, 2023.

[Jin *et al.*, 2018] Yaochu Jin, Handing Wang, Tinkle Chugh, Dan Guo, and Kaisa Miettinen. Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation*, 23(3):442–458, 2018.

[Jin *et al.*, 2023] Yaochu Jin, Hangyu Zhu, Jinjin Xu, and Yang Chen. *Federated Learning*. Springer, 2023.

[Kather *et al.*, 2019] Jakob Nikolas Kather, Johannes Krisam, Pornpimol Charoentong, Tom Luedde, Esther Herpel, Cleo-Aron Weis, Timo Gaiser, Alexander Marx, Nektarios A Valous, Dyke Ferber, et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS medicine*, 16(1):e1002730, 2019.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Li *et al.*, 2020] 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.

[Li *et al.*, 2022] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 965–978. IEEE, 2022.

[Li *et al.*, 2023] Zexi Li, Tao Lin, Xinyi Shang, and Chao Wu. Revisiting weighted aggregation in federated learning with neural networks. In *International Conference on Machine Learning*, pages 19767–19788. PMLR, 2023.

[Li *et al.*, 2024] Wenqian Li, Shuran Fu, Fengrui Zhang, and Yan Pang. Data valuation and detections in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12027–12036, 2024.

[Li *et al.*, 2025] Zihan Li, Shuai Yuan, and Zhitao Guan. Robust and scalable federated learning framework for client data heterogeneity based on optimal clustering. *Journal of Parallel and Distributed Computing*, 195:104990, 2025.

[Lin *et al.*, 2025] Ziqian Lin, Xuefeng Jiang, Kun Zhang, Chongjun Fan, and Yaya Liu. Feddshar: A dual-strategy federated learning approach for human activity recognition amid noise label user. *Future Generation Computer Systems*, 166:107724, 2025.

[Liu *et al.*, 2022] Zelei Liu, Yuanyuan Chen, Yansong Zhao, Han Yu, Yang Liu, Renyi Bao, Jinpeng Jiang, Zaiqing Nie, Qian Xu, and Qiang Yang. Contribution-aware federated learning for smart healthcare. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12396–12404, 2022.

[Liu *et al.*, 2025] Qiqi Liu, Yaochu Jin, and Guodong Chen. Optimization of an implicit acquisition function for federated bayesian many-task optimization. *IEEE Transactions on Evolutionary Computation*, 2025.

[Ma *et al.*, 2025] Yanbiao Ma, Wei Dai, Wenke Huang, and Jiayi Chen. Geometric knowledge-guided localized global distribution alignment for federated learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20958–20968, 2025.

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

[Nagalapatti *et al.*, 2022] Lokesh Nagalapatti, Ruhi Sharma Mittal, and Ramasuri Narayanam. Is your data relevant?: Dynamic selection of relevant data for federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7859–7867, 2022.

[Pejó and Biczók, 2023] Balázs Pejó and Gergely Biczók. Quality inference in federated learning with secure aggregation. *IEEE Transactions on Big Data*, 9(5):1430–1437, 2023.

[Pillutla *et al.*, 2022] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022.

[Saha *et al.*, 2022] Rituparna Saha, Sudip Misra, Aishwariya Chakraborty, Chandranath Chatterjee, and Pallav Kumar Deb. Data-centric client selection for federated learning over distributed edge networks. *IEEE Transactions on Parallel and Distributed Systems*, 34(2):675–686, 2022.

[Shi *et al.*, 2022] Zhuan Shi, Lan Zhang, Zhenyu Yao, Lingjuan Lyu, Cen Chen, Li Wang, Junhao Wang, and Xiang-Yang Li. Fedfaim: A model performance-based fair incentive mechanism for federated learning. *IEEE Transactions on Big Data*, 10(6):1038–1050, 2022.

[Shi *et al.*, 2025] Changlong Shi, He Zhao, Bingjie Zhang, Mingyuan Zhou, Dandan Guo, and Yi Chang. Fedawa: Adaptive optimization of aggregation weights in federated learning using client vectors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 30651–30660, 2025.

[Shyn *et al.*, 2021] Sung Kuk Shyn, Donghee Kim, and Kwangsu Kim. Fedccea: A practical approach of client contribution evaluation for federated learning. *arXiv preprint arXiv:2106.02310*, 2021.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Tang, 2024] Yitong Tang. Adapted weighted aggregation in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23763–23765, 2024.

[Wang *et al.*, 2023] Pengfei Wang, Zongzheng Wei, Heng Qi, Shaohua Wan, Yunming Xiao, Geng Sun, and Qiang Zhang. Mitigating poor data quality impact with federated unlearning for human-centric metaverse. *IEEE Journal on Selected Areas in Communications*, 42(4):832–849, 2023.

[Wu *et al.*, 2023] Leming Wu, Yaochu Jin, and Kuangrong Hao. Optimized compressed sensing for communication efficient federated learning. *Knowledge-Based Systems*, 278:110805, 2023.

[Wu *et al.*, 2024] Leming Wu, Yaochu Jin, Yuping Yan, and Kuangrong Hao. Fl-otcsenc: Towards secure federated learning with deep compressed sensing. *Knowledge-Based Systems*, 291:111534, 2024.

[Wu *et al.*, 2025] Leming Wu, Yaochu Jin, Kuangrong Hao, and Han Yu. Local data quantity-aware weighted averaging for federated learning with dishonest clients. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2025.

[Yang *et al.*, 2023] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

[Yi *et al.*, 2022] Liping Yi, Wang Gang, and Liu Xiaoguang. Qsfl: A two-level uplink communication optimization framework for federated learning. In *International Conference on Machine Learning*, pages 25501–25513. PMLR, 2022.

[Zhang *et al.*, 2024] Zongxiang Zhang, Gang Chen, Yunjie Xu, Lihua Huang, Chenghong Zhang, and Shuaiyong Xiao. Feddqa: A novel regularization-based deep learning method for data quality assessment in federated learning. *Decision Support Systems*, 180:114183, 2024.

[Zhang *et al.*, 2025] Haoyu Zhang, Zhihao Yu, Yaochu Jin, Xiufeng Liu, Weiguo Sheng, Ruyu Liu, Xiumei Li, Qiqi Liu, and Ran Cheng. Efficient evolutionary neural architecture search with hierarchical parameter mapping for monocular depth estimation. *IEEE Transactions on Evolutionary Computation*, 2025.

[Zhao *et al.*, 2025] Jie Zhao, Kang Hao Cheong, and Yaochu Jin. Multidomain evolutionary optimization on combinatorial problems in complex networks. *IEEE Transactions on Cybernetics*, 2025.