Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.081065

ARTICLE

# FKD-RTM: Heterogeneous Federated Knowledge Distillation Method Based on Residual-Enhanced Tree-to-MLP Transfer

**Sheyun Zhang, Ruichun Gu*, Chaofeng Li, Zhijian Dong and Hefei Wang**

School of Digital and Intelligence Industry, Inner Mongolia University of Science and Technology, Baotou, China

*Corresponding Author: Ruichun Gu. Email: reachcool@imust.edu.cn

Received: 22 February 2026; Accepted: 24 April 2026; Published: 15 June 2026

**ABSTRACT:** Federated learning (FL) enables collaborative model training without sharing raw data. However, in real-world applications, clients often exhibit statistical heterogeneity, missing classes, and long-tailed distributions, which can substantially degrade the generalization performance of conventional parameter aggregation and some personalization approaches. Moreover, distillation or alignment-based methods may suffer from unstable supervision and difficult optimization under highly heterogeneous settings. To this end, this paper proposes a novel method called FKD-RTM (Heterogeneous Federated Knowledge Distillation Based on Residual-Enhanced Tree-to-MLP Knowledge Transfer). The key idea is to decouple local teaching from globally aggregatable student learning: we introduce a Gradient Boosting Decision Tree (GBDT) as a local teacher at each client, providing more reliable soft supervision based on shared feature representations for a Multi-Layer Perceptron (MLP) student that supports efficient global aggregation and adaptation. To further correct the prediction bias left after first-stage distillation, we introduce a residual enhancement mechanism. It learns complementary knowledge in the pre-normalization score domain and enables second-stage corrective learning. In addition, FKD-RTM performs partial parameter fine-tuning of the feature extractor and student model for personalized local adaptation. Personalized updates are excluded from global aggregation to avoid contaminating the global model. Experiments on multiple datasets, including CIFAR-100, demonstrate that the proposed FKD-RTM method consistently improves accuracy and generalization under diverse complex data settings and achieves a better trade-off between global and personalized performance.

**KEYWORDS:** Federated learning; complex data; heterogeneous distillation; GBDT; residual enhancement

## 1 Introduction

Deep learning typically relies on centralized training, which becomes increasingly challenging under distributed data ownership and privacy constraints. Federated learning (FL) enables collaborative model training without centralizing raw data, effectively alleviating data silos and facilitating collaborative training while protecting the privacy of participating parties [1–3]. In recent years, FL has been widely adopted in finance, healthcare, and personalized recommendation [3]. However, practical FL often operates under complex data conditions induced by nonlinear feature structures, feature heterogeneity, and limited samples, where statistical heterogeneity [4–6] across clients is prevalent. Such heterogeneity significantly slows convergence and degrades generalization, thereby limiting the effectiveness of existing FL methods in real deployments [4,6].

To mitigate performance degradation under highly non-independent and identically distributed (Non-IID) data, personalized federated learning approaches [7–9] have shown advantages. Nevertheless, they

remain limited when data are extremely scarce, classes are severely missing, or client feature spaces exhibit strong nonlinear discrepancies. Robust aggregation [10] and regularization methods [11] reduce global model drift by constraining excessive local updates and adjusting server-side aggregation weights. Yet, these methods often depend on stable local optimization trajectories or gradient statistics, which can be unreliable in extremely heterogeneous, sparse, or class-missing clients, still leading to global performance drops. Prototype-based methods [12], global distillation [13,14], and heterogeneous-model [15], FL leverage knowledge distillation (KD) at the server to compensate for client drift and to extract valuable representations and decision boundaries from the global system. Recent studies have further shown that the performance of federated distillation is affected not only by the degree of data heterogeneity across clients, but also by the reliability of the distilled knowledge itself. DKDR [16] improves the stability and adaptability of knowledge transfer in heterogeneous client settings by dynamically adjusting the distillation objective and incorporating domain expert knowledge. This study indicates that, under complex data distributions, traditional static and unified distillation mechanisms cannot fully capture the differences among clients, which limits the effectiveness of knowledge transfer. Furthermore, their effectiveness typically relies on a feature-space alignment assumption. Under strong heterogeneity or highly nonlinear feature structures, prototypes may fail to reflect true class centers; global distillation [17] may miss local neighborhood variations and break under local class-missing or noisy settings; and heterogeneous architectures may be hard to align, making the distillation signal difficult to absorb by weaker student models. Therefore, achieving effective, robust, and scalable FL under real-world conditions with high heterogeneity, limited samples, and local class missing remains an open challenge.

To address these limitations, this paper proposes FKD-RTM, a federated knowledge distillation method for complex data scenarios. Different from conventional federated distillation methods that mainly rely on homogeneous neural teachers or server-side soft-label fusion, FKD-RTM follows a role-decoupled and two-stage distillation principle. Specifically, the method assigns distinct roles to different models. A tree-based model is used as a local teacher to capture client-specific nonlinear decision structures under few-shot, class-missing, and highly heterogeneous conditions. A lightweight MLP is used as a globally aggregatable student to absorb transferable knowledge and support efficient personalization, together enabling stronger modeling capacity [18] under complex data conditions. On top of this, FKD-RTM introduces a second-stage residual correction mechanism. It explicitly learns the complementary knowledge left unabsorbed after first-stage distillation. This improves discrimination on hard samples and long-tail categories. Concretely, each client first trains a local GBDT teacher on shared feature representations. The teacher then distills its knowledge into a MLP student. The student learns the principal knowledge in the first stage, while a residual model is further introduced to learn the discrepancy between teacher and student in the pre-normalization score domain, enabling second-stage corrective learning [19]. FKD-RTM uses a globally shared feature extractor to preserve global consistency. It performs lightweight client-side partial fine-tuning only in the personalization stage to improve local adaptability. Unlike conventional server-side distillation pipelines, FKD-RTM performs teacher-to-student distillation on each client. Only the student and residual model updates are uploaded. This helps avoid cross-client alignment failure and supports learning under highly Non-IID conditions.

The main contributions are summarized as follows:

- We propose FKD-RTM, a principle-driven heterogeneous federated knowledge distillation method. The key idea is to decouple client-specific local GBDT teaching from globally aggregatable MLP student learning.

- We develop a two-stage residual-enhanced distillation mechanism. The first stage learns principal teacher knowledge, whereas the second stage explicitly models residual discrepancy in the pre-normalization score domain to correct prediction bias.
- We design a shared-representation and lightweight personalization strategy. By introducing a globally shared feature extractor and client-side partial fine-tuning, FKD-RTM improves local adaptation while preserving global consistency and avoiding contamination of the global model.
- Extensive experiments on multiple standard benchmark datasets demonstrate that FKD-RTM consistently improves accuracy and robustness, while ablation studies further verify the effectiveness of the proposed design.

## 2 Related Works

### *2.1 Federated Learning*

FL is a distributed machine learning paradigm that enables clients to collaboratively train a global model without sharing raw data, thereby balancing privacy protection and model performance. FL has been widely applied in financial risk control, medical diagnosis, and personalized recommendation. With ongoing research progress, efforts have focused on key techniques such as optimization algorithms, communication compression, and privacy enhancement, and have introduced contrastive learning, transfer learning, and data augmentation to cope with complex data scenarios. For example, Ref. [20] improves system robustness and interpretability via visual digital twins, but it relies on complicated data assimilation and simulation algorithms and is difficult to generalize to cross-domain and few-shot settings. Communication-compressed FL methods [21] reduce communication by structured pruning. However, when dealing with multi-source heterogeneous images, sparsification may discard a small amount of critical discriminative features, limiting final classification accuracy. Image clustering and joint representation learning frameworks [22] alleviate the impact of Non-IID data on classifier training to some extent, but they lack explicit identification of key discriminative subspaces, which constrains their ability to distinguish scarce classes.

To address these issues, a more robust form of locally discriminative yet shareable knowledge is needed. This paper proposes the FKD-RTM method, which adopts an intuitive architectural design and focuses on learning local features under complex data conditions caused by feature heterogeneity, scarce classes, and few-shot data, where distilled knowledge serves as additional guidance. Therefore, the FKD-RTM is expected to achieve stronger fitting ability in complex-data scenarios and better generalization performance for classification tasks.

### *2.2 Gradient Boosting Decision Trees*

GBDTs are a strongly nonlinear ensemble model composed of multiple decision trees, which improves overall predictive capability by continuously fitting the errors (residuals) from previous iterations. In FL, the presence of few samples and missing classes often leads to limited and inconsistent client datasets, making conventional neural networks prone to oscillation and overfitting during local training, thereby restricting performance. For instance, a global-attention method [23] improves few-shot classification focus through global attention and key localization, but it often ignores fine-grained local neighborhood relationships and neighborhood consistency, resulting in limited performance in local or noisy scenarios. A multi-channel local-description and context-enhancement network [24] improves few-shot discrimination by strengthening joint local–context representations, yet it remains limited in explicitly selecting key discriminative subspaces and in efficient computation, making it difficult to balance accurate neighborhood modeling and inference overhead. Although lightweight models [25,26] reduce parameters and computation via model compression and multi-scale fusion, they often sacrifice neighborhood information fusion and cross-scale

consistency modeling, which limits classification performance in application scenarios with missing classes or subtle inter-class differences.

To alleviate these issues, this paper introduces GBDT in the FKD-RTM method and use it as the local teacher model. Compared with conventional neural networks, GBDT can maintain stable discriminative capability under few-shot, class-missing, and distribution-biased conditions (i.e., complex data scenarios). As a result, the method adopts GBDT as the teacher because it can still preserve strong fitting ability in complex environments, improving model adaptability and generalization, and enhancing practicality for deployment.

### *2.3 Multi-Layer Perceptron*

MLP is a lightweight neural network model composed only of linear layers, activation layers, and normalization layers. In FL, to reduce communication, researchers introduce knowledge distillation and typically adopt neural networks as teacher and student models. Knowledge distillation uses soft labels to extend supervision from hard labels to soft labels, enhancing learnability under few-shot settings; meanwhile, communication still relies on parameter aggregation, which can greatly reduce communication cost with only minor loss of performance. However, in complex data scenarios, distillation with neural-network students has clear limitations. For example, a distillation-based method [27] enhances generalization under few-shot classification and class imbalance by aligning the probability distributions of the teacher and student. Yet, when the feature space is sparse or local classes are missing, the student may fail to learn the teacher's distribution and local neighborhood relationships in these sparse regions, leading to alignment failure and degraded performance.

Compared with conventional convolutional neural network (CNN) students, MLPs are lighter, train faster, and have stronger fitting ability. More importantly, MLPs are easier to integrate with partial fine-tuning, and are thus more suitable for cross-domain knowledge distillation in the feature space. GBDT is a piecewise strong decision model, whereas the convolutional structure of CNNs tends to be locally smooth and spatially biased, making it less compatible with fixed feature representations in this setting. Accordingly, FKD-RTM adopts an MLP as the student: the mapping of an MLP can approximate the piecewise function of GBDT and mimic its logical decision process, enabling the student to better match the teacher's discriminative capability and alleviating possible alignment failure and performance degradation during distillation.

### *2.4 Feature Extractor and Partial Fine-Tuning*

The feature extractor is responsible for transforming image data from all clients into a shared representation space. However, in FL, client data often exhibit complex conditions such as missing classes and limited samples. If the feature extractor is fully frozen, the shared representation may not fit extreme clients, preventing the student model from adequately fitting local data. Therefore, we freeze most parameters of the feature extractor and only allow the last few layers to be fine-tuned, updating them with each client's local data to adapt to its own distribution. The shared representation mitigates cross-client representation misalignment, while partial fine-tuning addresses the mismatch for extreme clients [7]. Recent studies on personalized federated learning have also shown that relying only on a unified shared representation is often not enough to handle distribution differences across clients. Targeted local adaptation based on client feature distributions is still needed. For example, pFedFDA [28] combines shared feature representation with personalized classifier adaptation from the perspective of feature distribution adaptation. It achieves strong performance under covariate shift and local data scarcity. This further suggests that combining shared feature representation with local personalized adaptation is an effective direction. Accordingly, the FKD-RTM adopt

this design so that partial fine-tuning of the feature extractor remains a core advantage of the distillation method and can achieve stronger adaptability and higher personalization performance while preserving communication efficiency and privacy protection [29].

### *2.5 Residual Enhancement*

Residual enhancement incorporates a model's own residual information during prediction, enabling the model to maintain stability and plasticity during knowledge distillation and feature adaptation [30]. During distillation, the global MLP student can only passively smooth toward soft labels. In addition, partial fine-tuning of the feature extractor can easily introduce shifts, which may cause large fluctuations in client features or soft labels and lead to overfitting to local distributions, thereby undermining the advantages brought by GBDT-based distillation. To further strengthen client-side local adaptation while preserving global consistency, we introduce residual enhancement to prevent overfitting and improve local adaptability [31]. For feature representations after partial fine-tuning or for student logits (pre-normalization scores), the model does not directly replace the global output; instead, it treats them as residual supplement to the global model. Accordingly, the FKD-RTM design is highly compatible with the GBDT-to-MLP distillation structure and the partial fine-tuning strategy, and serves as an important component of our method.

## 3 Methodology

In distributed FL for image classification, it is usually necessary to properly partition the model architecture to meet the computation requirements of clients. Motivated by this, this paper focuses on complex multi-class image classification via cross-model distillation, and investigates key issues including feature extraction under distributed settings, the distillation architecture design, and model aggregation strategies. To address these issues, FKD-RTM follows a role-decoupled and two-stage distillation principle: robust local teaching is assigned to GBDT, globally aggregatable learning is assigned to the MLP student, and complementary corrective knowledge is learned through second-stage residual enhancement. The FKD-RTM goal is to validate the effectiveness and advantages of cross-model distillation when handling highly heterogeneous data, few-shot data, and even locally class-missing data.

We consider a FL system with $K$ clients. Each client $K \in \{1, \cdots, k\}$ holds a private local training dataset $D_k$, where samples are denoted by $\left\{\left(x_i^{(k)}, y_i^{(k)}\right)\right\}_{i=1}^{n_k}$, with input $x_i^{(k)} \in X$ and label $y_i^{(k)} \in \{1, \ldots, C\}$. A separate validation $D_k^{val}$ or test dataset $D_k^{test}$ is used for evaluation. All datasets involve $C$ classes. Due to privacy constraints, the raw data remain on local devices and are never transmitted. The data distributions across clients can differ substantially, resulting in a Non-IID setting, and may further exhibit complex phenomena such as missing classes and long-tailed distributions. Let $f_\theta(\bullet): X \in \mathbb{R}^d$ denote the feature extractor, which maps an inpu $x$ to a $d$ dimensional embedding vector $h \in \mathbb{R}^d$; we denote this embedding by $h_i$.

We denote the client-side teacher model as $T_k(\bullet): R_d - R_c$, and calibrate its output soft targets $t_k$ using a probability calibration function $C_k(\bullet)$. The student model is denoted by $S_\phi(\bullet): R_d - R_c$, whose distillation output is $S_k$. The student model parameters $w_k$ are uploaded to the server for weighted aggregation. In addition, we compute the residual $r_k(x) = t_k(x) - s_k(x)$ on each client and use it to train a client-specific residual model $R_k$. In FKD-RTM, the local residual model $g_k(\bullet)$ remains a GBDT-based residual learner that fits the discrepancy between teacher scores and student scores in the shared feature space. Since the parameters of tree-based residual models are not directly suitable for FedAvg-style averaging, clients do not upload residual tree parameters to the server. Instead, in each communication round, clients upload the residual predictions of their local GBDT residual learners on a small public anchor set $A = \{x_{k,i}\}_{i=1}^{m}$.

The server performs weighted aggregation of these anchor-based residual predictions in the function space, thereby forming a global residual knowledge representation $R_g$, which is then broadcast back to clients together with the updated global student model $W_g$. The residual branch provides an additive compensation to the student logits $\tilde{s}_k$, and the final prediction is obtained by applying a normalization function (e.g., softmax) to the residual-enhanced logits produced by $\tilde{s}_k = s_k + R_g(x)$. The server then broadcasts the updated student model $W_g$ and the residual-enhancement model $R_g$ back to all participating clients.

### *3.1 Client-Side Heterogeneous Distillation (GBDT→MLP)*

By combining tree models and MLPs, we integrate the advantages of tree ensembles in structured modeling with the strong nonlinear expressiveness of MLPs into the federated knowledge distillation method. The key principle is to decouple robust local teaching from globally aggregatable student learning. The FKD-RTM design can significantly improve the training performance of cross-model distillation. It reduces the reliance on homogeneous teacher–student architectures and provides an efficient solution for heterogeneous knowledge distillation.

Specifically, under limited and heterogeneous client data, tree models can quickly fit local data and provide stable probability estimates even in few-shot and class-missing conditions. Their nonlinear partitioning in the feature space captures local decision boundaries, producing informative and stable soft labels for the student. Meanwhile, MLPs can reliably align to the soft labels produced by an ensemble teacher, remain lightweight for efficient communication and local fine-tuning, and approximate the piecewise decision boundaries of gradient-boosted trees in a shared feature space. With partial fine-tuning, the MLP student can further adapt to local distributions. Overall, this framework improves generalization under complex data conditions and supports robust FL.

First, we train a GBDT on the shared feature representations to better handle few-shot settings and locally missing classes. GBDT is trained in a stage-wise manner by iteratively fitting the residuals from the previous iteration. On client $k$, we learn a local GBDT ensemble model $T_k$ using the embedded local data. After training, the teacher outputs a per-class score (or probability) for each sample:

$$T_k(h_i) = \sum_{m}^{M} R_m T_m(h), \tag{1}$$

Here, for a given embedded sample $h$ on client $k$, $T_m: R_d \to R_c$ denotes the vector output of the $m$-th tree (typically producing one score for each class), and $R_m$ is the learning rate. During training, we apply a probability transformation function $f(\bullet) = softmax(\bullet)$ to convert the raw tree outputs in Eq. (1) into a probability distribution in Eq. (2):

$$p_k(h_i) = softmax(T_k(h_i)) \in \Delta^{C-1}, p_k(h_i) \in R^C, \tag{2}$$

Although GBDT provides more robust local supervision than neural teachers under complex heterogeneous settings, its output probabilities may still be miscalibrated and over-confident. Calibration aims to align teacher confidence with empirical correctness, making the predicted probabilities more reliable, i.e., a prediction with confidence p should be correct approximately p proportion of the time. If the probabilities produced by the tree model in Eq. (2) are used without calibration, they may become over-confident and skew toward extreme values, which can amplify training discrepancies and lead to poor convergence or even training failure. Therefore, probability calibration is necessary for the teacher outputs. Each client fits a calibration function $C_k(\bullet)$ using its local validation set $D_k^{val}$, implemented as Platt scaling in our method,

and the resulting calibrated probabilities are further transformed as in Eq. (3). We further compare FKD-RTM with and without calibration in the ablation study, showing that local calibration improves the stability and effectiveness of teacher supervision under heterogeneous data settings. Temperature scaling, in contrast, controls the sharpness of the calibrated distribution and exposes richer inter-class relations, so that the student can better absorb dark knowledge from the teacher. To obtain a smoother and more informative target distribution, we further apply temperature scaling to the calibrated teacher outputs by transforming the original predicted probabilities and then performing temperature-based rescaling. Therefore, in FKD-RTM, calibration improves the correctness of the teacher probabilities, whereas temperature scaling improves their learnability for distillation. This yields the temperature-smoothed probabilities in Eq. (3):

$$\tilde{p}_k\left(h_i\right)=\frac{\exp\left(\log\left(\frac{\hat{p}_k\left(h_i\right)}{T}\right)\right)}{\sum_c \exp\left(\log\left(\frac{\hat{p}_c\left(h_i\right)}{T}\right)\right)}, \hat{p}_k\left(h_i\right)=C_k\left(p_k\left(h_i\right)\right), \hat{p}_c\left(h_i\right)=C_c\left(p_c\left(h_i\right)\right), \tag{3}$$

Second, on each client we obtain teacher soft targets from the trained teacher model and distill them into a lightweight MLP student model $S_\phi(\bullet)$. The student network consists of multiple fully connected layers. Given the feature representations of local training samples, the student is trained to minimize the Kullback-Leibler (KL) divergence between the student predictive distribution and the tree teacher's soft targets, thereby measuring and maximizing their distributional similarity. In this way, the student can efficiently align to the ensemble teacher's soft-label distribution, as formulated in Eq. (4).

$$L_{KD}\left(\phi\right)=\frac{1}{M}\sum_m^M KL\left(softmax\left(\frac{S_{\phi_k}\left(h_m\right)}{T}\right)\middle\|\sum_k \tilde{p}_k\left(h_m\right)\right), \tag{4}$$

Here, $\tilde{p}_k(h)$ denotes the teacher's temperature-smoothed soft targets, and $T$ is the temperature hyperparameter. Moreover, $S_{\phi_k}(h)$ represents the student's local predictive distribution on client $k$. To ensure that training remains faithful to the ground-truth labels, we further incorporate a cross-entropy loss term:

$$L_{CE}\left(\phi\right)=-\frac{1}{M}\sum_{m=1}^M\sum_{c=1}^C y_{m,c}\log\left(softmax\left(S_{\phi_k}\left(h_m\right)\right)\right)_c, \tag{5}$$

The overall objective for training the student model is a weighted combination of the hard-label loss in Eq. (5) and the soft-label KL-divergence loss in Eq. (4), resulting in the total loss in Eq. (6). Here, $\alpha$ denotes a weight coefficient that is linearly adjusted with respect to local training rounds, and is used to balance the contributions of the hard-label supervision and the temperature-smoothed distillation term.

$$L\left(\phi\right)=\left(1-\alpha\right)L_{CE}\left(\phi\right)+\alpha L_{KD}\left(\phi\right), \tag{6}$$

Finally, we apply a channel compression strategy in the last layer to reduce the feature dimensionality to the target output dimension, thereby lowering model complexity and improving inference efficiency.

**Comparison with Conventional Distillation-Based Approaches:** Unlike conventional federated distillation methods that align soft targets at the server side, the proposed FKD-RTM approach performs the entire distillation process locally on each client and only uploads the parameters of the student model and the residual model. As a result, the server only needs to conduct parameter-level aggregation, which preserves the benefits of distillation while avoiding the additional communication overhead and potential privacy risks associated with transmitting predictive distributions. Conventional distillation typically assumes identical or similar neural network architectures for the teacher and student. In such settings, the teacher essentially

learns a continuous and differentiable distribution optimized via backpropagation, which often requires substantial data to fit high-quality decision boundaries and is less robust under few-shot or class-imbalanced conditions. Moreover, the teacher's soft targets can be over-confident and become unstable in Non-IID or noisy scenarios. In contrast, tree-based models are naturally effective for few-shot learning, class imbalance, and other forms of complex data, and can provide more robust probability estimates than convolutional neural network (CNN) teachers, yielding more reliable distillation signals for the student. The outputs of tree teachers exhibit an inherent piecewise decision structure, making them well suited as a source of soft supervision in federated environments. Meanwhile, MLPs have strong nonlinear expressive power and can efficiently approximate the piecewise decision boundaries of gradient-boosted trees, leading to high compatibility with the soft targets produced by tree ensembles.

**Novelty and Advantages:** The proposed cross-architecture distillation framework FKD-RTM, which employs a tree-based teacher and an MLP student, constitutes a form of heterogeneous knowledge distillation under a cross-model paradigm. The piecewise partitioning structure of tree models complements the continuously differentiable representation capacity of MLPs, making the distillation signal more expressive and more amenable to learning. Moreover, the inherent properties of GBDT enable each client to generate complementary and diverse teacher knowledge. Through knowledge distillation, when student models are aggregated at the server, the global student can capture global decision structures that are difficult to obtain via conventional Federated Averaging (FedAvg) [1] or conventional neural-teacher distillation. In addition, by aligning predictive probabilities on a shared feature representation space, our method alleviates failures of feature-space alignment, while benefiting from more stable teacher outputs and more easily absorbable supervision for the student.

### *3.2 Shared Feature Extractor and Partial Fine-Tuning*

The feature extractor serves as the shared backbone network of the overall distillation framework, mapping the raw image input $x$ into a stable feature representation space $h$ that is shared across all clients. Neither the tree teacher nor the MLP student operates directly on raw images; instead, both models rely on this shared representation space for training and inference.

First, given a client sample $x$, we extract its representation using the feature extractor $f_\theta(\bullet)$ with shared parameters $\theta$, the feature extractor is globally shared across all clients. obtaining an embedding that can be directly used to train the tree model, as shown in Eq. (7). During the first-stage distillation, the feature extractor is fully frozen to preserve a consistent representation space and to avoid additional communication of extractor updates; it does not participate in client-side local optimization. Although client data distributions differ, all clients share the same feature extractor, which ensures that they operate within a consistent semantic embedding space:

$$h_{k,i} = f_\theta\left(x_{k,i}\right), h_{k,i} \in \mathbb{R}^d, \nabla_\theta, \tag{7}$$

The tree models trained on clients can fit their local data within a consistent representation space, producing teacher soft targets with the same dimensionality across clients. Accordingly, the student model can be trained directly in this representation space under the supervision of the teacher soft targets, enabling knowledge sharing after aligning client representations.

Local tree training is performed on the feature representations $T_k: R_d \to R_c$. This allows the piecewise, nonlinear, and locally adaptive strengths of tree ensembles to be more effectively exploited in the representation space. The stability of the learned representations also prevents the tree model from collapsing due to noise in raw pixel inputs.

During client-side distillation of the student model, the same feature representations extracted by the shared extractor are used as inputs. Since the student does not need to process raw images, the computational cost of knowledge distillation is substantially reduced, as indicated in Eq. (8):

$$S_\phi\left(h_{k,i}\right) = S_\phi\left(f_\theta\left(x_{k,i}\right)\right), \tag{8}$$

Second, in federated settings with highly Non-IID data, missing classes, or long-tailed samples, single-stage distillation often exhibits systematic bias in regions containing hard examples. To further compensate for the limitations of the student model, we introduce a residual enhancement module. During the residual and personalization stage, a partial fine-tuning module is applied when the global residual model is broadcast to clients and used to assist student training. This enables lightweight local adaptation of the residual signals, making the model better fit each client's complex data environment and improving local performance. To adapt the client-personalized model to the local distribution, partial fine-tuning unfreezes the last several layers of the feature extractor, only the last few high-level layers are unfrozen for local adaptation, while the lower layers remain fixed. These locally fine-tuned parameters are not uploaded to the server, which enables lightweight personalization without affecting the global model.

We unfreeze the last several layers of the feature extractor and the MLP, while freezing all remaining layers. This can be expressed as $\theta = \left\{\theta_{frozen}, \theta_{adapt}\right\}$, where $\theta_{frozen}$ denotes the lower-level feature parameters that remain fixed, and $\theta_{adapt}$ denotes the trainable high-level feature layers together with the classification head. Partial fine-tuning is performed using local ground-truth labels via the supervised objective in Eq. (9), without requiring an additional distillation step:

$$L_{FT} = -\sum_i y_{k,i} \log\left(softmax\left(S_\phi\left(f_\theta\left(x_{k,i}\right)\right)\right)\right), \tag{9}$$

To prevent personalization updates from contaminating the global model, partial fine-tuning is performed only on the client side, and the fine-tuned parameters are excluded from federated aggregation. Without altering the global model, this strategy allows different clients to update in different directions for personalized adaptation, while also reducing computational overhead.

**Novelty and Advantages:** Unlike conventional federated distillation methods that directly rely on raw features or raw model outputs, the proposed FKD-RTM approach introduces a globally shared feature extractor that maps heterogeneous client data into a unified representation space, as shown in Eq. (7). This alleviates a core issue in conventional federated distillation—namely, the misalignment of representations produced by different client teachers—thereby ensuring global consistency. The feature extractor remains fully frozen during most of the training process, which avoids synchronizing extractor parameters and thus reduces communication. Partial fine-tuning preserves globally shared semantic information while substantially lowering local training cost and preventing the global model from being dominated by highly heterogeneous clients. As a finer-grained and more stable personalization strategy, it is more lightweight than existing approaches such as [7,32]. Finally, partial fine-tuning is conducted entirely on-device, which improves security and further reduces communication overhead.

### *3.3 Residual Enhancement*

During the client-side heterogeneous distillation stage, the student model $S_\phi(\bullet)$ has already learned the major decision boundaries under the supervision of the tree teacher $T_k(\bullet)$ via its soft targets. However, under highly Non-IID data, missing classes, or long-tailed distributions, single-stage distillation often exhibits systematic bias in regions containing hard examples. To further capture the complementary knowledge not yet absorbed by the student, we introduce a second-stage complementary knowledge correction mechanism,

namely a residual enhancement module $R_g$ implemented by class-wise local GBDT learners. It explicitly models the teacher-student discrepancy in the form of logit residuals, enabling residual-based compensation and local personalization in the second stage.

Using the features extracted in the representation space in Eq. (7), the teacher produces soft targets, which are further converted into probabilistic outputs as in Eqs. (1)–(3). After the first-stage distillation, both the teacher soft targets and the student outputs are mapped into the logit-residual form. The residual in FKD-RTM is defined in the pre-normalization score domain rather than directly in the probability space. In the probability space, vectors are constrained on the simplex, so changes in one class are strongly coupled with changes in others. This makes additive correction less stable. By contrast, the pre-normalization score domain is more flexible for linear residual compensation. It allows the residual model to explicitly capture the teacher–student discrepancy before normalization. Specifically, the teacher soft targets are obtained from Eq. (3), while the student outputs logits on the shared representations and applies a softmax function to obtain the predictive distribution $p_s\left(h_{k,i}\right)$ as in Eq. (4). Based on these quantities, we define a global residual learning objective to capture the remaining knowledge that the student has not absorbed:

$$r\left(h_{k,i}\right) = \log\left(\tilde{p}_k\left(h_{k,i}\right) + \varepsilon\right) - \left(\left(S_\phi\left(h_{k,i}\right)\right) + \varepsilon\right), \tag{10}$$

To improve training stability, we apply temperature scaling to the raw residual in Eq. (10) and introduce a small stabilization constant $\varepsilon$ to avoid numerical issues caused by extremely small probabilities when transforming teacher outputs. We analyze the parameter sensitivity of the stabilization constant $\varepsilon$. The results show only minor fluctuations, indicating that FKD-RTM is not sensitive to $\varepsilon$ within a reasonable range. This yields the temperature-smoothed residual in Eq. (11). This temperature-based modeling makes residual learning more stable and controllable:

$$\tilde{r}\left(h_{k,i}\right) = r\left(h_{k,i}\right)/T_r, \tag{11}$$

In FKD-RTM, the local residual model is implemented as a class-wise GBDT residual learner. Specifically, for each client, one residual regressor is trained for each class in the pre-normalization score domain, and these regressors jointly form a multi-output correction module. Therefore, the residual learner does not replace the first-stage teacher, but instead captures the complementary corrective knowledge that remains after the student has absorbed the principal teacher knowledge. Since the local residual learner is tree-based, its parameters are not directly compatible with weighted parameter averaging. Therefore, FKD-RTM aggregates residual knowledge $R_g\left(h_{k,i}\right) \approx r\left(h_{k,i}\right)$ in the function space rather than in the parameter space. Let $\mathrm{A} = \left\{x_{k,i}\right\}_{j=1}^{m}$ denote a small public anchor set maintained at the server, and let $h_{k,i} = f_\phi\left(x_{k,i}\right)$ be the shared representation of anchor sample $x_{k,i}$. Each client evaluates its local residual learner $g_k\left(h_{k,i}\right) \approx \tilde{r}\left(h_{k.i}\right)$ on these anchor features and uploads the corresponding residual predictions $\hat{R}_{k,i} = g_k\left(h_{k,i}\right)$. The server then performs weighted fusion of the uploaded residual predictions:

$$\overline{R}_g = \sum_{k-1}^{k} \omega_k \hat{R}_{k,i}, \omega_k = \frac{n_k}{\sum_{l=1}^{k} n_l}, \tag{12}$$

where $n_k$ denotes the sample size of client $k$, and $\omega_k$ denotes the aggregation weight defined based on the sample size in Eq. (12). In this way, FKD-RTM does not directly average heterogeneous tree parameters, but instead fuses client-specific residual knowledge in the shared function space. The aggregated residual prediction $\overline{R}$ serves as the global residual knowledge for anchor sample.

By combining conventional distillation learning regime $S_\phi\left(h_{k,i}\right) \approx p_T\left(h_{k,i}\right)$ with residual enhancement regime $\Delta S_\phi\left(h_{k,i}\right) \approx \overline{R}_g$, the resulting model can be viewed as learning in a two-stage manner. The two

stages in FKD-RTM play different but complementary roles. Specifically, stage-1 distillation is responsible for learning the principal knowledge of the local teacher, i.e., the major decision structure that can be transferred reliably to the global student. Stage-2 residual learning is designed to capture complementary corrective knowledge, namely the prediction bias that remains after Stage-1, especially in hard regions induced by class imbalance, local class missing, or severe distribution shift. This two-stage learning process yields a model with stronger expressive capability, as described in Eq. (13):

$$\tilde{S}_{\phi}^{new}\left(h_{k,i}\right) = S_{\phi}\left(h_{k,i}\right) + \Delta S_{\phi}\left(h_{k,i}\right), \tag{13}$$

The aggregated global residual knowledge is then broadcast back to all clients and used as the second-stage corrective signal after local distillation. It is combined with partial fine-tuning of the feature extractor and the student model to enhance local learning on difficult regions.

**Novelty and Advantages:** Conventional distillation typically performs knowledge aggregation only once. In contrast, this paper proposes FKD-RTM method aggregates the residual model parameters on the server and broadcasts the resulting model to clients, introducing it as an auxiliary learning signal in the second stage. By explicitly constructing the residual and applying temperature smoothing (as in the corresponding equation), the proposed residual enhancement strengthens the student's ability to learn the portion of teacher knowledge that remains unknown after the first-stage distillation. This design fundamentally changes the distillation process: the student not only imitates the teacher but also actively compensates for its own deficiencies, focusing more on hard regions. Moreover, residuals are computed from each client's teacher soft targets, yielding discrepancy signals that reflect complementary gaps across clients. Aggregating these cross-client discrepancy signals provides an explicit training objective, enabling the student to capture shared regularities across multiple data sources as well as complex structures that are missed in the first-stage distillation, including complementary knowledge among client teachers. Overall, this module implements an incremental distillation strategy in which the student learns principal knowledge first and then absorbs the remaining knowledge, constituting a novel and effective distillation paradigm.

### *3.4 Dual-Model Aggregation*

In the FKD-RTM method, we introduce tree-based models and MLPs, employ a unified feature-extraction space together with a partial fine-tuning module, and further optimize the learning process via residual enhancement. These designs enable strong fitting ability and high privacy preservation under complex data conditions caused by nonlinear feature structures, feature heterogeneity, and limited samples, while improving generalization under Non-IID settings without sacrificing global consistency. Fig. 1 illustrates the overall federated distillation framework.

First, before the $t$-th communication round starts, the server maintains a globally shared feature extractor $f_{\theta}(\bullet)$. For client $k$, each local sample $x$ is mapped into the shared representation space as in Eq. (7). This ensures that samples from different clients are represented in a unified semantic space and provides an aligned input basis for subsequent tree-teacher training and student distillation, thereby substantially alleviating representation inconsistency caused by Non-IID data.

Subsequently, each client $k$ trains a local GBDT teacher model on its feature representations $\{(h_{k,i}, y_{k,i})\}$ in the shared space, as in Eq. (1), and outputs a class-probability distribution that is further calibrated. The tree teacher exhibits stronger local fitting ability under few-shot and Non-IID conditions. Moreover, it does not rely on gradient-based optimization and does not require parameter synchronization, making it well suited for federated settings. In addition, it can fit client-specific patterns and thus provides diverse and complementary knowledge across clients.

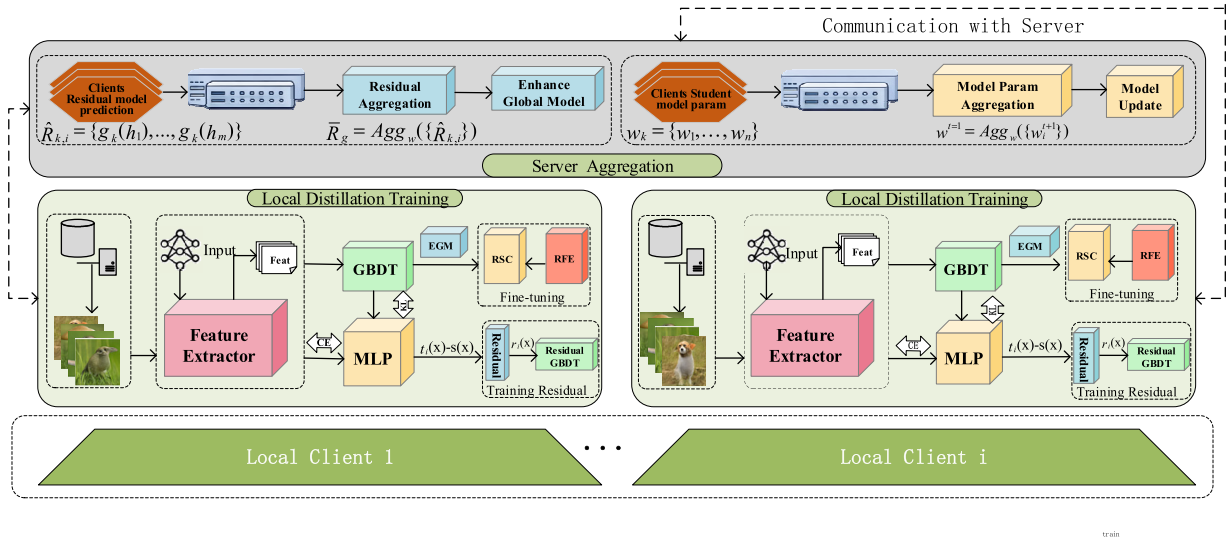


**Figure 1:** The FKD-RTM method and workflow of the proposed method. Here, enhance global model (EGM) denotes the global residual enhancement module; residual-aware student classifier (RSC) denotes the MLP student fine-tuning module; and residual-aware feature encoder (RFE) denotes the feature-extractor fine-tuning module.

In FKD-RTM, the student and residual models are aggregated through different operators: the student uses weighted parameter averaging, while the residual GBDT branch is aggregated in the function space via weighted fusion of anchor-based residual predictions. The first stage, the server collects the student model parameters $w_i$ uploaded by clients and performs global weighted aggregation across clients, as in Eq. (14). Note that Eq. (14) is used only for aggregating student parameters. The residual branch is not aggregated in the parameter space because the local residual model is GBDT-based; instead, it is aggregated through weighted fusion of anchor-based residual predictions, as defined in Eq. (12). The updated global model $W_g$ is then broadcast back to all clients. Under the supervision of the teacher's soft targets, the student is iteratively optimized to learn the principal knowledge from clients and to approximate the teacher's decision structure in the shared representation space, thereby forming a unified global model:

$$W^{t+1} = \sum_{k \in s_t} \alpha_k W_i^{t+1}, \tag{14}$$

Here, $\alpha_k$ denotes the weight determined by the client's data volume and or reliability proportion. On the server, $\alpha_k$ is computed based on an estimated teacher reliability score and is used to perform weighted aggregation for student parameters in Eq. (15). This weighting scheme enables more personalized aggregation and simultaneously provides reliability scores to filter out abnormal clients, thereby achieving robust aggregation.

To improve robustness under heterogeneous client quality, FKD-RTM adopts a reliability-aware aggregation strategy, where $n_k$ denote the number of local training samples on client $k$ and $r_k$ denote the reliability score of the local teacher. To jointly consider data size and teacher reliability, we define the pre-normalized aggregation weight as $\tilde{\alpha}_k$, and the final normalized aggregation weight is computed as:

$$r_k = Acc\left(T_k, D_k^{val}\right), \tilde{\alpha}_k = n_k \bullet \max\left(r_k, \tau_r\right), \alpha_k = \frac{\tilde{\alpha}_k}{\sum_{j \in S_t} \tilde{\alpha}_j}, \tag{15}$$

where Acc(·) denotes validation accuracy of the local teacher model. Here, $\tau_r$ is a lower clipping threshold used to avoid extremely small or unstable weights. We further include an ablation comparison with and without reliability-aware weighting aggregation in the experimental section. Second, after the first-stage distillation, we further enhance client-side adaptability while maintaining global consistency. The second stage, we compute the residual $r_i$ between the student and teacher soft targets, and train a residual model according to Eqs. (10) and (11). Unlike the student model, the residual learner is tree-based and therefore is not aggregated by directly averaging model parameters in Eq. (14). Instead, each client evaluates its residual learner on a server-side public anchor set and uploads the corresponding anchor-based residual predictions. The server then performs weighted aggregation of these residual predictions in Eq. (12), thereby constructing a global residual knowledge representation in the function space. The resulting global residual knowledge is broadcast to clients and used as a second-stage corrective signal. Combined with partial fine-tuning of the feature extractor and the student model, this mechanism strengthens the student's performance in disadvantaged regions and improves its ability to fit complex decision boundaries and long-tail classes. In this way, the server explicitly models such discrepancies, constructs a global residual enhancement model, and applies temperature smoothing. The resulting residual model is $\overline{R}_g$ then broadcast to clients, where it is combined with partial fine-tuning of the feature extractor and the student model to conduct second-stage learning. This module strengthens the student's performance in its disadvantaged regions and improves its ability to fit complex decision boundaries and long-tail classes. On each client, during the residual and personalization stage, partial fine-tuning of the feature extractor and the student facilitates enhanced distillation between the residual model and the student, and further helps absorb complementary knowledge across clients, thereby improving the overall capacity for knowledge extraction.

The core workflow of this FKD-RTM mechanism is summarized as follows. First, we construct a unified feature space to enable robust local teacher training. Next, the student learns the global principal knowledge and incorporates residual enhancement to absorb the remaining knowledge. Finally, the server aggregates the student parameters and residual model parameters and broadcasts the updates; on each client, feature-level fine-tuning is performed for personalized adaptation. This procedure is repeated in every federated communication round, enabling the student model to progressively approach an optimal solution corresponding to the multi-client ensemble. The overall federated algorithm is summarized in Algorithm 1.

**Algorithm 1:** FKD-RTM

//Server:
initialize global student $W_0$ and global anchor-based residual knowledge $R_0$.
//Client:
**for** each communication round t = 0, . . ., T − 1:
Server broadcasts $(W_t, R_t)$ to selected clients $k \in S_t$.
**for** each client k **in parallel do:**
    // Shared feature extraction
    Extract features: $h_i = f(x_i) \in \mathbb{R}^d$
    // Train local teacher
    Train teacher model $T_k$ on $\{(h_k, y)\}$;
    Obtain teacher soft label $\tilde{p}_{T_k}(\bullet|x)$.
    //Stage-1 Principal KD:
    Compute MLP student logits $s_k(x)$.
    Update $W_i$ by min KD loss (CE + KL) with teacher soft label.
    //Residual target construction

(Continued)

**Algorithm 1 (continued)**

Convert teacher soft targets to teacher logits: $t_k(x) = \log(\tilde{p}_{T_k}(x) + \varepsilon)$
Compute local residual signals $r_k(x) = t_k(x) - s_k(x)$.
// Fit Local Residual Learner
Train a class-wise local GBDT residual learner $g_k(\bullet)$ using pairs $\{(h_k, r_k(x))\}$.
The residual model output is $\hat{R}_{k,i}(h) = [g_k(h_1), g_k(h_m)]$
// Anchor-based residual upload
Evaluate local residual learner $g_k(\bullet)$ on shared anchor features $A = \{(x_{k,i})\}_{j=1}^{m}$.
Obtain anchor residual predictions $\hat{r}_{k,i} = g_k(h_{k,i})$.
//Stage-2 Residual Corrective Learning:
Use aggregated global residual knowledge $R_g$ to construct residual-enhanced.
Compute student(RFE + RSC) logits $\tilde{s}_i(x) = s_i(x) + R_g(x)$ using $\overline{R}_g(h)$.
Further update $W_i$ with the same KD loss form.
//Upload to server:
Student update $W_i$ and anchor residual predictions $\hat{R}_{k,i} \approx \hat{r}_{k,i}$.

**end**

//Server aggregation:

Aggregate student parameters via weighted averaging: $W^{t+1} \leftarrow Agg_w\left(\left\{W_i^{t+1}\right\}\right)$,
Aggregate anchor-based residual predictions via reliability-aware weighted averaging: $\overline{R}_g = Agg_w\left(\left\{\hat{R}_{k,i}\right\}\right)$
Broadcast updated global student and aggregated global residual knowledge.

**end**

Return: $W^T$ and $\overline{R}_g$.

## 4 Experiments

### *4.1 Experimental Setup*

**Training setups.** All experiments are conducted on a computing platform equipped with an NVIDIA P100 GPU. On all evaluation datasets, we use the AdamW optimizer with a learning rate of 0.001 and an L2 regularization coefficient of $3 \times 10^{-4}$. In the federated learning setting, each client performs 5 local steps per communication round, and the total number of communication rounds is 100. The batch size is fixed to 64 for all experiments. With the total data volume fixed, the default number of clients participating per round is 10. In the number of clients study, we set the total number of clients to 20, 40, and 60, respectively. To ensure the statistical reliability of the reported results, all experiments are repeated over three independent runs with different random seeds, and the reported results are averaged across these runs, and all values are reported as mean ± standard deviation, with standard deviations shown in parentheses. In all result tables, the best value in each column is highlighted in bold. The distillation temperature is set to 2 by default, and all methods use the same client sampling ratio C = 0.7. Similar to other distillation-based methods, we simulate data heterogeneity using a Dirichlet distribution over labels, $\mathrm{Dir}(\alpha)$. We allocate all training samples to user models (clients) and evaluate performance using all test samples. For the classifier in all methods, ResNet18 is adopted as a strong feature extractor module. The depth of the gradient-boosted decision trees is 6 with a learning rate of 0.05. The student network contains four linear layers; each hidden layer is followed by a fully connected layer with BatchNorm and GELU. The dropout rate is initialized to 0.4 and then decreases layer by layer. More detailed implementation settings and reproducibility-related configurations are provided in Appendix A.

**Experimental datasets.** The FKD-RTM study uses four widely adopted real-world image classification datasets, each following its official split: CIFAR-10 (a natural image classification benchmark) and CIFAR-100 (a more challenging multi-class dataset with more categories). To evaluate performance under high-resolution and few-shot settings, we choose the semi-supervised dataset STL-10. To evaluate performance on specialized medical imaging, we use PathMNIST. These datasets cover different difficulty levels and validate adaptability under different data scales or resolutions as well as cross-domain generalization, ensuring that experimental conclusions in federated learning are comprehensive and credible.

**Baselines.** We compare the proposed FKD-RTM method with the following mainstream baselines: FedAvg [1] update the global model by weighted averaging of clients' local model parameters without sharing raw data; FedProx [10] additionally introduces a proximal regularization term to constrain local updates from drifting too far from the global model; FedMD [13] exchanges clients' predictions on a shared public or reference dataset to perform inter-model knowledge distillation; FedDF [18] first obtains an initial global model via parameter aggregation, then further improves it by distillation using client models as teachers; FedMKD [33] combines knowledge from multiple teachers and uses confidence-based weighting to make distillation more robust; FedTKD [34] designed for heterogeneous model architectures and introduces an adaptive weighting mechanism; DaFKD [35] introduces domain awareness by considering distribution differences across clients and reweighting soft labels during distillation. FedGKD [14] and FedICT [36] are also closely related to our work. FedGKD [14] studies heterogeneous federated learning through distillation from historical global models. Our method instead relies on client-specific GBDT teachers and second-stage residual correction. FedICT [36] is developed for bidirectional distillation in multi-task edge-computing scenarios. In contrast, FKD-RTM focuses on complex image classification with shared feature representations and tree-to-MLP heterogeneous distillation. Therefore, we discuss them as relevant recent methods rather than direct baselines.

### *4.2 Experimental Results*

#### *4.2.1 Performance Comparison*

**Comparison with baselines across different datasets:** This subsection evaluates the proposed FKD-RTM method in terms of training accuracy and robustness, and shows its advantages over existing methods across multiple metrics. We test each method on four datasets under Dirichlet-distributed data partitions in both Non-IID ($\alpha = 0.1$) and IID ($\alpha = 10$) settings. We run multiple independent trials and report the mean of the last 10 rounds to reduce random fluctuations. Under Non-IID ($\alpha = 0.1$), Table 1 shows that FKD-RTM method achieves the best performance on most datasets and remains highly competitive on the remaining one, consistently outperforming baselines such as FedAvg and FedProx. Compared with the strongest baseline FedMD, the proposed FKD-RTM method improves by +0.61 on average; on CIFAR-100, it improves over FedMD by +1.78, showing strong performance on the more challenging dataset. For comparison, the corresponding IID results are summarized in Table 2. In addition to accuracy, Table 3 summarizes F1 scores on CIFAR-100 and PathMNIST across settings, indicating that FKD-RTM method not only improves overall accuracy but also maintains more consistent discriminative capability under class imbalance. These results confirm that the proposed FKD-RTM method stably outperforms baselines across all datasets and settings, highlighting its robustness in Non-IID scenarios. This advantage mainly comes from the method's ability to fit complex-data scenarios effectively. Through the cooperative design of tree teachers and MLP students, it exhibits stronger capability in capturing nonlinear local decision boundaries under heterogeneous data. In particular, on STL-10 (a semi-supervised dataset), the FKD-RTM method further demonstrates its effectiveness at capturing key features in small-sample data, significantly improving classification accuracy and robustness. As shown in Fig. 2a,b, on CIFAR-10 the model exhibits

smooth and stable convergence under both Non-IID and IID distributions as training rounds increase. In contrast, FedAvg and FedProx may suffer from compromise/underfitting in Non-IID settings; The classic distillation models FedMD and FedDF, as well as the recent models FedMKD and FedTKD may suffer from inconsistent distillation objectives and negative transfer under extreme heterogeneity; and DaFKD may fail when alignment assumptions break under extreme multi-domain differences or conflict with discriminativeness, resulting in degraded modeling ability. These observations further demonstrate the proposed model's optimization robustness and generalization under extreme data structures, validating its reliability and practicality in distributed heterogeneous environments.

**Model relative performance and statistical significance analysis:** Table 2 summarizes results under IID. From the accuracy perspective, on IID PathMNIST our FKD-RTM method improves over FedAvg by 0.73 percentage points. Although its average accuracy is not the highest among all distillation-based methods, it has the lowest standard deviation (±0.07), indicating higher stability across runs and better suitability for real-world deployment. To further verify the reliability of the improvements, we perform paired $t$-tests against the strongest baseline, FedMD, under the Non-IID setting ($\alpha = 0.1$) across three random seeds. The results indicate that FKD-RTM significantly outperforms FedMD on both CIFAR-100 and STL-10. Specifically, it achieves 0.6036 ± 0.12 on CIFAR-100 and 0.9420 ± 0.03 on STL-10, with statistically significant improvements $p = 0.000116$ and $p = 0.0181$, respectively. Given that the evaluation is conducted over only three random seeds, the corresponding 95% confidence intervals are relatively wide. This confirms that the observed gains are unlikely to be caused by random variation.

**Table 1:** Performance comparison of different models under the Non-IID setting ($\alpha = 0.1$).

| Method | CIFAR-10 | CIFAR-100 | STL-10 | PathMNIST | Avg. |
|---|---|---|---|---|---|
| FedAvg | 0.7799 (3.11) | 0.5818 (0.27) | 0.8284 (8.99) | 0.7383 (2.07) | 0.7321 |
| FedProx | 0.8119 (0.19) | 0.6063 (0.18) | 0.8601 (4.03) | 0.8125 (1.52) | 0.7727 |
| FedDF | 0.7895 (2.20) | 0.5983 (1.73) | 0.8671 (4.19) | 0.7936 (1.87) | 0.7621 |
| FedMD | 0.8308 (0.21) | 0.6114 (0.21) | 0.9381 (0.06) | 0.8215 (1.09) | 0.8030 |
| FedMKD | 0.8094 (1.68) | 0.5485 (0.64) | 0.6418 (2.08) | 0.8085 (0.54) | 0.7020 |
| FedTKD | 0.8315 (0.19) | 0.5472 (0.36) | 0.8603 (5.36) | 0.8091 (1.19) | 0.7620 |
| DaFKD | 0.7924 (1.08) | 0.5155 (1.35) | 0.8561 (3.12) | 0.7540 (1.03) | 0.7295 |
| FKD-RTM | **0.8354 (0.07)** | **0.6306 (0.12)** | **0.9420 (0.03)** | **0.8282 (0.17)** | **0.8091** |

**Table 2:** Performance comparison of different models under the IID setting ($\alpha = 10$).

| Method | CIFAR-10 | CIFAR-100 | STL-10 | PathMNIST | Avg. |
|---|---|---|---|---|---|
| FedAvg | 0.8672 (0.68) | 0.6649 (0.09) | 0.9353 (0.07) | 0.8284 (0.14) | 0.8240 |
| FedProx | 0.8532 (1.15) | 0.6660 (0.20) | 0.9374 (0.09) | **0.8509 (0.14)** | 0.8269 |
| FedDF | 0.8735 (0.17) | 0.6661 (0.05) | 0.9339 (0.16) | 0.8400 (0.11) | 0.8284 |
| FedMD | 0.8512 (0.20) | 0.6293 (0.14) | 0.9407 (0.16) | 0.8334 (0.13) | 0.8136 |
| FedMKD | 0.8626 (0.15) | 0.6274 (0.17) | 0.9257 (0.02) | 0.8430 (0.31) | 0.8147 |
| FedTKD | 0.8605 (0.06) | 0.6153 (0.25) | 0.9267 (0.21) | 0.8371 (0.25) | 0.8099 |
| DaFKD | 0.8726 (0.05) | 0.6202 (0.04) | 0.9308 (0.11) | 0.8459 (0.08) | 0.8174 |
| FKD-RTM | **0.8780 (0.04)** | **0.6807 (0.11)** | **0.9428 (0.12)** | 0.8357 (0.07) | **0.8343** |

**Table 3:** Macro-F1 performance comparison of different models under IID and Non-IID settings.

| Method | CIFAR-100 | | PathMNIST | |
|---|---|---|---|---|
| | α = 0.1 | α = 10 | α = 0.1 | α = 10 |
| FedAvg | 0.5450 (2.90) | 0.6621 (0.29) | 0.6835 (4.44) | 0.7791 (0.52) |
| FedProx | 0.5329 (1.68) | 0.6375 (0.24) | 0.7401 (3.86) | 0.8052 (0.34) |
| FedDF | 0.5935 (1.87) | 0.6575 (0.22) | 0.7417 (2.10) | 0.7876 (0.37) |
| FedMD | 0.6288 (0.65) | 0.6266 (0.08) | 0.7881 (0.49) | 0.7875 (0.31) |
| FedMKD | 0.5543 (0.56) | 0.6331 (0.14) | 0.7153 (5.05) | 0.7848 (0.26) |
| FedTKD | 0.5611 (0.48) | 0.6283 (0.35) | 0.7291 (3.49) | 0.7779 (1.00) |
| DaFKD | 0.5612 (2.32) | 0.6473 (0.09) | 0.7065 (5.54) | 0.7693 (0.30) |
| FKD-RTM | **0.6364 (0.17)** | **0.6762 (0.28)** | **0.8073 (0.14)** | **0.8270 (0.29)** |

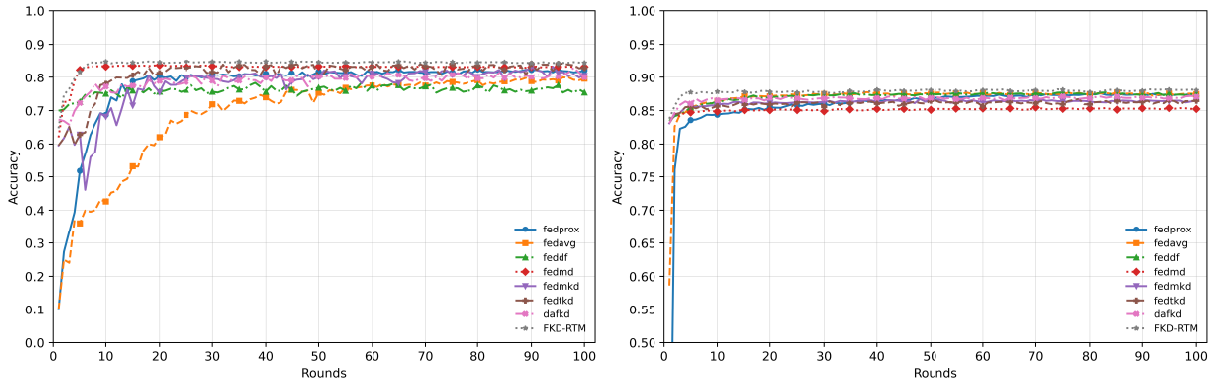


**Figure 2:** Accuracy trends of different models on CIFAR-10 under (**a**) the Non-IID setting and (**b**) the IID setting.

*4.2.2 Robustness under Extreme Heterogeneity*

To further evaluate robustness under more challenging conditions, we additionally test FKD-RTM under more extreme heterogeneous settings ($\alpha = 0.01$), including a smaller Dirichlet concentration parameter and a severe class-missing partition. Table 4 reports the results compared with the standard Non-IID setting, the performance gap between baselines becomes larger under these harder configurations, while FKD-RTM remains relatively stable. This result further demonstrates that the proposed role-decoupled distillation and residual correction mechanisms are particularly beneficial when client distributions become extremely sparse and imbalanced.

*4.2.3 Joint Comparison of AUC Curves and Training-Round Curves*

To characterize discriminative capability and training dynamics for multi-class federated image classification, the FKD-RTM method uses one-vs.-rest macro-AUC together with training-round curves to evaluate convergence speed and stability. Macro-AUC is more robust when classes are imbalanced or differ substantially in difficulty, reflecting overall separability across classes. From Fig. 3a,b, on CIFAR-10 FKD-RTM method achieves higher macro-AUC under both Non-IID and IID conditions, with smoother growth and smaller gain fluctuations, indicating consistent and repeatable improvements. Combined with the

round curves in Fig. 2a,b, FKD-RTM method reaches the high-performance region earlier under Non-IID and maintains low-variance convergence, effectively suppressing oscillations caused by complex data; under IID it also converges faster with less jitter, reflecting optimization efficiency. In both scenarios, FKD-RTM method consistently outperforms the compared methods. By contrast, baselines show lower and less stable AUC and round curves, failing to form consistent decision boundaries, indicating limited capability in fitting complex data and adapting to heterogeneity—thus restricting their generalization in complex federated image settings. The advantage of FKD-RTM method mainly stems from the novel design of heterogeneous distillation and residual enhancement, which effectively alleviates nonlinear fitting issues under heterogeneous data and shows strong modeling ability under extremely heterogeneous image data, achieving faster, steadier, and higher performance evolution while improving robustness and generalization.

**Table 4:** Performance comparison of different models under the Non-IID setting ($\alpha$ = 0.01).

| Method | CIFAR-10 | CIFAR-100 | STL-10 | PathMNIST | Avg. |
|---|---|---|---|---|---|
| FedAvg | 0.6043 (1.02) | 0.5333 (0.19) | 0.7807 (1.43) | 0.6238 (2.21) | 0.6355 |
| FedProx | 0.6623 (0.50) | 0.5416 (0.16) | 0.7878 (2.61) | 0.7465 (3.06) | 0.6846 |
| FedDF | 0.5977 (2.15) | 0.5318 (0.34) | 0.8080 (0.79) | 0.6804 (1.73) | 0.6545 |
| FedMD | 0.6946 (0.25) | 0.5521 (0.10) | 0.8861 (0.03) | 0.7531 (0.22) | 0.7215 |
| FedMKD | 0.6504 (0.41) | 0.5096 (0.15) | 0.7215 (7.05) | 0.7146 (6.66) | 0.6490 |
| FedTKD | 0.7061 (1.14) | 0.5065 (0.32) | 0.8078 (4.99) | 0.6990 (2.53) | 0.6799 |
| DaFKD | 0.6329 (0.47) | 0.4438 (0.05) | 0.7918 (0.42) | 0.6459 (0.41) | 0.6286 |
| FKD-RTM | **0.7658 (0.12)** | **0.5827 (0.22)** | **0.9068 (0.03)** | **0.7746 (0.01)** | **0.7575** |

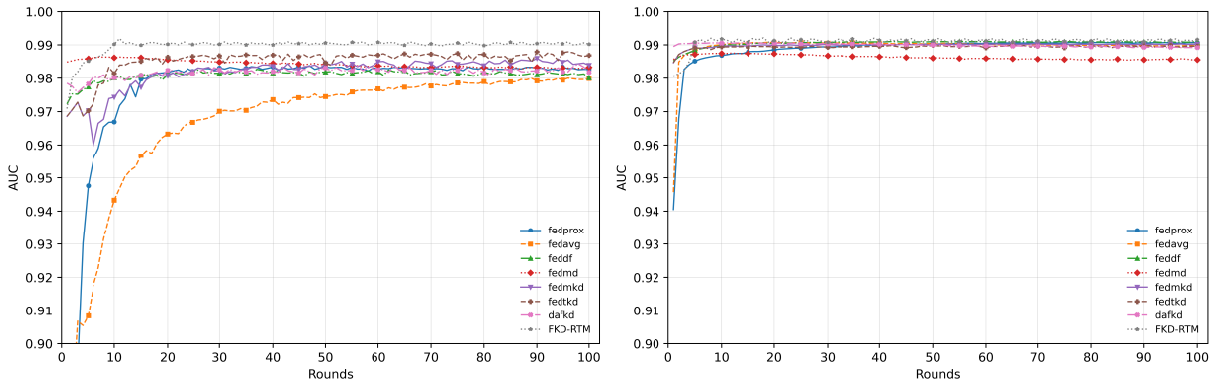


**Figure 3:** AUC trends of different models on CIFAR-10 under (**a**) the Non-IID setting and (**b**) the IID setting.

### *4.2.4 Varying Numbers of Clients*

To comprehensively evaluate performance in large-scale distributed environments, as shown in Fig. 4a,b, the FKD-RTM method gradually increases the number of clients from 20 to 60 under both Non-IID and IID settings on CIFAR-10, to examine how different models behave at different scales. Results show that as the number of clients increases, overall performance becomes more stable. This advantage comes from the proposed FKD-RTM method's ability to capture local nonlinear features, maintaining robust performance in distributed environments. On CIFAR-10, the model can accurately capture nonlinear local features, ensuring that as the number of clients increases—while per-client data decreases and Non-IID

effects become more pronounced—stability still improves. In contrast, other models are much more sensitive to changes in client number, especially in Non-IID settings, making them harder to scale and highlighting their limitations in handling heterogeneous data. Overall, as shown in Fig. 4a,b, FKD-RTM model also performs well in accuracy: with more clients, accuracy remains stable and higher than other methods, further validating its applicability in complex federated learning scenarios.

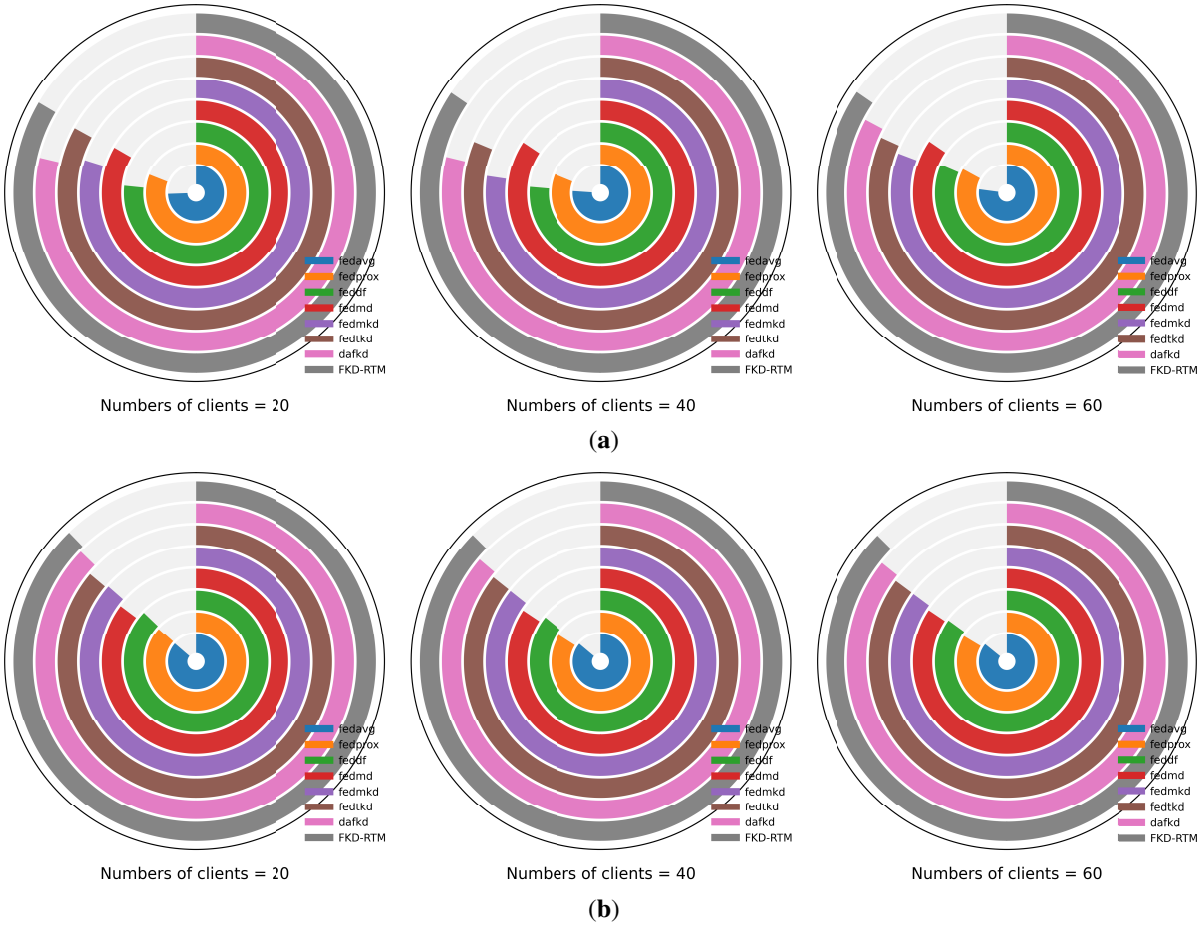


**Figure 4:** Accuracy comparison of different models on CIFAR-10 as the number of clients varies under (**a**) the Non-IID setting and (**b**) the IID setting.

### *4.2.5 Ablation Study*

**Module-wise ablation:** To further evaluate the contribution of each module of the FKD-RTM method, we conduct ablation experiments. For clarity, we compare settings using a strong feature extractor and a weak feature extractor, systematically analyzing the role of core modules. Specifically, we remove the proposed FKD-RTM modules Feature Fine-Tuning (FFT = RSC + RFE) and Enhance Global Model (EGM) one at a time, and compare them with the full framework. As shown in Table 5, the full method performs best in all tested scenarios, significantly surpassing any variant missing a single module. Variants with any module removed exhibit reduced stability under heterogeneous data, highlighting the rationality and completeness of the method design.

**Table 5:** Ablation study of the framework with strong and weak feature extractors.

| Feature Encoder | Model/Datasets | CIFAR-10 | CIFAR-100 | STL-10 | PathMNIST | Avg. |
|---|---|---|---|---|---|---|
| Week | baseline | 0.3985 | 0.2847 | 0.5343 | 0.7306 | 0.4870 |
| | +(FFT) | 0.5463 | 0.3362 | 0.5591 | 0.7676 | 0.5523 |
| | +(EGM) | 0.5370 | 0.3415 | 0.5776 | 0.7774 | 0.5584 |
| | Full model (ours) | **0.5502** | **0.3510** | **0.5815** | **0.7867** | **0.5674** |
| | CNN Teacher | 0.5199 | 0.2706 | 0.5803 | 0.7657 | 0.5341 |
| Strong | baseline | 0.8624 | 0.6660 | 0.9275 | 0.8273 | 0.8208 |
| | +(FFT) | 0.8758 | 0.6783 | 0.9369 | 0.8348 | 0.8315 |
| | +(EGM) | 0.8751 | 0.6707 | 0.9363 | 0.8311 | 0.8283 |
| | Full model (ours) | **0.8780** | **0.6807** | **0.9428** | **0.8357** | **0.8343** |
| | CNN Teacher | 0.8477 | 0.6003 | 0.9416 | 0.8147 | 0.8011 |

**Long-tail ablation study:** We further construct long-tail class distributions via Dirichlet partitioning, setting the front/back threshold to 30% of classes. As shown in Table 6, under IID the accuracy of low-frequency classes is close to that of high-frequency classes; under Non-IID, low-frequency classes still achieve relatively higher test accuracy and remain stable across different random seeds. These properties show strong potential in complex data scenarios and provide an innovative solution for optimizing distributed image learning.

**Table 6:** Ablation study on long-tailed class distribution accuracy under IID and Non-IID settings. Here, ↓ indicates that lower values are better.

| α | Acc. (%) | Low_Freq (%) | High_Freq (%) | Gap↓ |
|---|---|---|---|---|
| 10 | 87.23 ± 0.17 | 86.72 ± 0.31 | 88.19 ± 0.25 | 1.47 ± 0.36 |
| 0.1 | 83.28 ± 0.17 | 82.16 ± 0.39 | 88.99 ± 0.36 | 6.83 ± 0.54 |

**Teacher-type comparison:** To validate the choice of GBDT as the local teacher [37], we compare FKD-RTM with variants that use a CNN teacher instead under the same shared feature space and training setting. The results in Table 5 show that the GBDT teacher provides more reliable supervision. This is because GBDT does not rely on backpropagation and is therefore more stable to train in complex federated settings. In addition, under few-shot, class-missing, and highly heterogeneous conditions, its piecewise partitioning bias is more likely to form effective decision boundaries and produce robust local partitions in the shared embedding space. These results further support our view that, in complex federated environments, tree-based teachers are more robust than neural teachers.

**Reliability-weighting and calibration ablations:** We further validate the robustness of the proposed aggregation and distillation design. Specifically, we compare FKD-RTM with and without reliability-aware weighting, and with and without local calibration. The results in Table 7 show that both components improve performance under heterogeneous settings. Reliability-aware weighting stabilizes aggregation by down-weighting low-quality local teachers. Calibration improves the correctness of local teacher probabilities and leads to more effective soft supervision.

**Table 7:** Ablation of teacher probability calibration and reliability-aware weighting on PathMNIST under the Non-IID setting ($\alpha = 0.1$). Here, RW denotes reliability-aware weighting and baseline is no calibration and reliability-aware weighting. ↑ indicates that higher values are better, and ↓ indicates that lower values are better.

| Method | Acc↑ | Macro-F1↑ | LogLoss↓ | Brier↓ | ECE↓ |
|---|---|---|---|---|---|
| Baseline | 0.8023 | 0.7475 | 2.2812 | 0.7525 | 0.3073 |
| Only Calibration | 0.8125 | 0.7896 | 2.2140 | 0.7304 | 0.2397 |
| Calibration + RW (FKD-RTM) | **0.8282** | **0.8073** | **2.0422** | **0.7187** | **0.2295** |

### *4.2.6 Representation and Residual Mechanism Analysis*

**Representation visualization analysis:** To further understand why FKD-RTM achieves better performance, we visualize the learned shared representations from the same layer using t-SNE on CIFAR-100 under the Non-IID setting. Fig. 5 shows that the representations learned by FedAvg are highly mixed across classes, indicating weak class discrimination under heterogeneous local updates. FedMD improves local grouping to some extent, but the class boundaries remain ambiguous and several categories still overlap substantially. In contrast, FKD-RTM produces more compact intra-class clusters and clearer inter-class separation. This indicates that the proposed tree-guided heterogeneous distillation provides more stable supervision in the shared representation space. It also suggests that the residual enhancement mechanism helps refine difficult regions that are not sufficiently learned in the first stage. As a result, FKD-RTM learns more discriminative and transferable representations under complex Non-IID conditions.

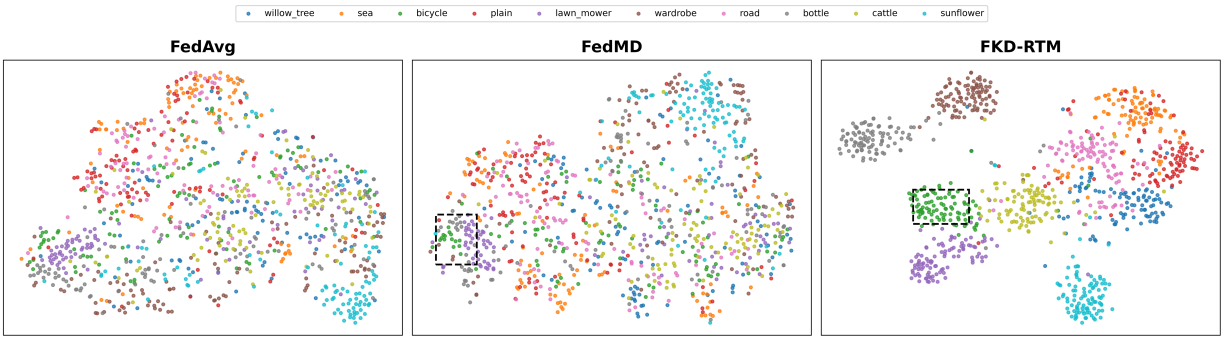


**Figure 5:** t-SNE visualization of shared representations on CIFAR-100 under the Non-IID setting.

**Residual magnitude analysis:** To better understand why the residual branch improves performance, we further analyze the residual magnitude across different sample groups. Fig. 6b shows that tail classes consistently exhibit relatively larger residual magnitudes than head classes. This indicates that the residual module mainly contributes corrective knowledge to data-scarce and imbalanced categories. Further class-wise analysis in Fig. 6c shows that different classes have different average residual magnitudes, implying that the difficulty of teacher-to-student transfer is class dependent. The observation is consistent with the long-tail results in Table 6. It further suggests that residual enhancement is especially useful for classes that are not sufficiently learned in the first-stage distillation. As shown in Fig. 6a, misclassified samples exhibit substantially larger residual magnitudes than correctly classified samples. This suggests that the residual branch mainly targets hard regions where the student still shows prediction bias after Stage-1 distillation. In contrast, it contributes little to samples that have already been well learned. Therefore, the second-stage residual learning in FKD-RTM can be viewed as an explicit corrective mechanism for

capturing and compensating for the complementary knowledge that has not yet been absorbed during first-stage distillation.

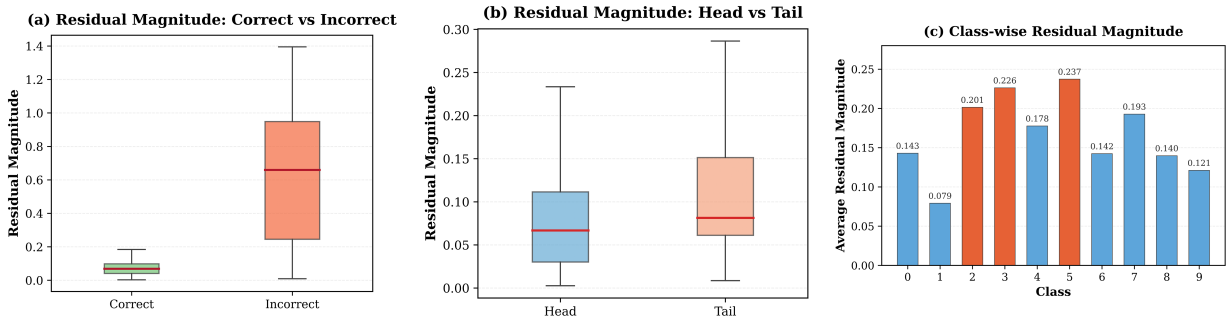


**Figure 6:** Residual magnitude analysis of FKD-RTM on CIFAR-10 under the Non-IID setting.

*4.2.7 Computational Overhead Analysis*

**Runtime and memory analysis:** To assess the FKD-RTM method computational efficiency, we measure peak memory usage and average runtime per round on CIFAR-10. Table 8 summarizes the results. Because the method requires training tree models separately, its runtime per round is about 50% higher than other baseline models. However, it per-round allocated memory decreases by about 55% and reserved memory decreases by about 36% compared with baselines. Compared with traditional federated learning methods such as FedAvg, the proposed model requires more runtime but achieves higher accuracy, indicating a reasonable trade-off for complex image classification tasks.

**Table 8:** Computational overhead comparison on CIFAR-10 (10 clients, Non-IID). Here, P denotes peak, ↓ indicates that lower values are better.

| Model | P_Allocated (MiB)↓ | P_Reserved (MiB)↓ | Comm/Round (MB)↓ | Runtime/Round(s)↓ |
|---|---|---|---|---|
| FedAvg | 643.40 | 786 | 169.02 | 74.61 |
| FedProx | 652.46 | 790 | 169.02 | 76.12 |
| FedDF | 725.03 | 854 | 169 | 104.24 |
| FedMD | 718.8 | 792 | 3.81 | 88.96 |
| FedMKD | 704 | 852 | 42.24 | 82.64 |
| FedTKD | 704 | 850 | 3.82 | 82.82 |
| DaFKD | 697.4 | 860 | 4.0 | 72.19 |
| FKD-RTM | 289.9 | 502 | 53.33 | 200.03 |

**Communication efficiency analysis:** Besides computational overhead, communication efficiency is also a key concern in federated learning. In FKD-RTM, only the parameters of the student model and the anchor-based residual predictions need to be uploaded. The globally shared feature extractor remains frozen during most of training. Therefore, it introduces no additional communication overhead. We further compare FKD-RTM with representative baselines in terms of the total communication volume per round, including both uploaded and downloaded model-related messages. For parameter-sharing methods, the communication cost is computed based on transmitted model parameters, while for prediction-sharing methods such as FedMD, it is computed based on transmitted logits or probability tensors rather than full model parameters. The results in Table 8 show that, although FKD-RTM introduces an additional residual

branch, its overall communication cost remains low and controllable due to the frozen shared feature extractor and the lightweight student model.

## 5 Conclusion

This paper addresses key challenges of FL in real-world complex data scenarios, including statistical heterogeneity, missing classes, and long-tailed distributions. This paper proposes FKD-RTM, a heterogeneous federated knowledge distillation method for complex data scenarios. Specifically, FKD-RTM follows a two-stage role-decoupled distillation principle. We introduce a GBDT as a local teacher on each client to provide more reliable soft supervision for a unified MLP student model, thereby improving training stability and knowledge transferability under highly heterogeneous settings. To further correct the systematic bias left after first-stage distillation, FKD-RTM introduces a residual enhancement mechanism in which the residual branch is modeled by class-wise local GBDT residual learners that learn complementary knowledge in the pre-normalization score domain. In addition, a globally shared feature extractor together with client-side partial fine-tuning enables lightweight personalization without contaminating the global model. Experimental results on multiple datasets demonstrate that FKD-RTM consistently improves classification accuracy, robustness, and global–personalized trade-off under diverse complex Non-IID settings. Despite these benefits, several limitations remain. First, training tree teachers on clients increases local computation and per-round latency. Future work may explore incremental tree training and early stopping, as well as tree pruning and compression, combined with resource-aware client scheduling to better balance accuracy and efficiency. Second, the current reliability estimation and aggregation-weight design can be further improved by incorporating more robust client quality assessment and anomaly update detection, strengthening stability under extreme heterogeneity and noisy labels. Finally, while this work is validated on image classification tasks; future research will extend the method to broader tasks and application scenarios and investigate integration with stronger representation learning backbones. Overall, FKD-RTM provides an effective and extensible solution for federated learning under complex heterogeneous data conditions.

**Acknowledgement:** Not applicable.

**Funding Statement:** The authors received no specific funding for this study.

**Author Contributions:** The authors confirm contribution to the paper as follows: Conceptualization: Sheyun Zhang, Ruichun Gu; methodology: Sheyun Zhang, Ruichun Gu; Model implementation: Sheyun Zhang, Ruichun Gu; Formal analysis and investigation: Sheyun Zhang, Ruichun Gu, Chaofeng Li, Zhijian Dong, Hefei Wang; Writing—original draft preparation: Sheyun Zhang, Ruichun Gu, Chaofeng Li, Zhijian Dong, Hefei Wang; Visualization: Sheyun Zhang, Ruichun Gu; Supervision: Sheyun Zhang, Ruichun Gu; Project administration: Sheyun Zhang, Ruichun Gu. All authors reviewed and approved the final version of the manuscript.

**Availability of Data and Materials:** Data available on request from the authors.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A Implementation Details and Reproducibility

To improve reproducibility, we summarize the implementation details of FKD-RTM as follows.

### *Appendix A.1 Feature Extractor and Student Architecture*

The shared feature extractor is based on ResNet18 and is initialized with ImageNet-pretrained weights. The output feature dimension is 512. A dropout rate of 0.3 is used, and L2 normalization is applied to the extracted features. The backbone contains four stages.

The student-side MLP consists of four fully connected layers with widths [1024, 512, 256, C], where $C$denotes the number of classes. Except for the final output layer, each hidden layer is followed by BatchNorm and GELU. The dropout rates are progressively decreased across layers and set to [0.4, 0.3, 0.2].

### *Appendix A.2 GBDT Teacher and Residual Learner*

The local GBDT teacher is implemented using XGBoost. The tree hyperparameters are set as follows: n_estimators = 200, max_depth = 6, and learning_rate = 0.05.

The local residual GBDT learner is also implemented using XGBoost. Its tree hyperparameters are set to n_estimators = 100, max_depth = 4, and learning_rate = 0.05.

### *Appendix A.3 Calibration and Training Schedule*

The calibration function $C_k(\bullet)$ is implemented using Platt scaling (sigmoid) and is fitted on the local validation set. The balancing coefficient in Eq. (6) is linearly scheduled over local training epochs according to Eq. (A1):

$$\alpha = \alpha_{start} + (\alpha_{end} - \alpha_{start}) \frac{epoch}{epoch - 1}, \tag{A1}$$

where $\alpha_{start} = 0.6$ and $\alpha_{end} = 0.2$

### *Appendix A.4 Anchor-Based Residual Aggregation*

For the residual branch, each client trains a class-wise local residual GBDT learner and evaluates it on a small server-side anchor set $A = \{(x_{k,i})\}_{j=1}^{m}$. The anchor set is constructed with 5–10 anchor samples per class, yielding a total size of approximately $(5-10) \times C$, where $C$ is the number of classes in the dataset. Each client uploads the residual predictions on this anchor set to the server, and the server performs weighted aggregation in the function space.

### *Appendix A.5 Reliability-Aware Aggregation*

The reliability-aware aggregation weight is computed according to Eq. (15), where the teacher reliability score is measured by local validation accuracy. To prevent extremely small reliability scores from causing unstable aggregation, a clipping threshold is applied. The reliability score is then combined with the local sample size and normalized as in Eq. (15) to obtain the final aggregation weight.

All experiments are repeated over three independent runs with random seeds [1, 10, 11], and the reported results are given as the mean and standard deviation over these three runs. Detailed implementation settings are provided in the appendix to facilitate reproducibility.

## References

1. McMahan B, Moore E, Ramage D, Hampson S, yArcas BA. Communication-efficient learning of deep networks from decentralized data. Artif Intell Stat. 2017;54:1273–82.
2. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and applications. ACM Trans Intell Syst Technol. 2019;10(2):1–9. doi:10.1145/3298981.

3. Kairouz P, McMahan HB. Advances and open problems in federated learning. Found Trends Mach Learn. 2021;14(1–2):1–210. doi:10.1561/2200000083.
4. Li T, Sahu AK, Talwalkar A, Smith V. Federated learning: challenges, methods, and future directions. IEEE Signal Process Mag. 2020;37(3):50–60. doi:10.1109/MSP.2020.2975749.
5. Zhao Y, Li M, Lai L, Suda N, Civin D, Chandra V. Federated learning with non-IID data. arXiv:1806.00582. 2018. doi:10.48550/arXiv.1806.00582.
6. Xu H, Li J, Wu W, Ren H. Federated learning with sample-level client drift mitigation. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2025 Feb 25–Mar 4; Philadelphia, PA, USA. p. 21752–60. doi:10.1609/aaai.v39i20.35480.
7. Collins L, Hassani H, Mokhtari A, Shakkottai S. Exploiting shared representations for personalized federated learning. In: Proceedings of the 38th International Conference on Machine Learning; 2021 Jul 18–24. p. 2089–99.
8. Fallah A, Mokhtari A, Ozdaglar A. Personalized federated learning with theoretical guarantees: a model-agnostic meta-learning approach. Adv Neural Inf Process Syst. 2020;33:3557–68.
9. Dinh CT, Tran NH, Nguyen TD. Personalized federated learning with Moreau envelopes (pFedMe). arXiv:2006.08848. 2020.
10. Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V. Federated optimization in heterogeneous networks. Proc Mach Learn Syst. 2020;2:429–50.
11. Karimireddy SP, Kale S, Mohri M, Reddi S, Stich S, Suresh AT. Scaffold: stochastic controlled averaging for federated learning. In: Proceedings of the 37th International Conference on Machine Learning; 2020 Nov 21; Online. p. 5132–43.
12. Wu S, Chen J, Nie X, Wang Y, Zhou X, Lu L, et al. Global prototype distillation for heterogeneous federated learning. Sci Rep. 2024;14(1):12057. doi:10.1038/s41598-024-62908-0.
13. Li D, Wang J. FedMD: heterogenous federated learning via model distillation. arXiv:1910;03581. 2019.
14. Yao D, Pan W, Dai Y, Wan Y, Ding X, Yu C, et al. FedGKD: toward heterogeneous federated learning via global knowledge distillation. IEEE Trans Comput. 2023;73(1):3–17. doi:10.1109/TC.2023.3315066.
15. Song W, Yan M, Li X, Han L. Bidirectional decoupled distillation for heterogeneous federated learning. Entropy. 2024;26(9):762. doi:10.3390/e26090762.
16. Yuan Y, Huang W, Wan G, Guan K, Li H, Ye M. DKDR: dynamic knowledge distillation for reliability in federated learning. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS); 2025 Nov 30–Dec 7; San Diego, CA, USA.
17. Lin T, Kong L, Stich SU, Jaggi M. Ensemble distillation for robust model fusion in federated learning. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS); 2020 Dec 6–12; Vancouver, BC, Canada.
18. Saberian M, Delgado P, Raimond Y. Gradient boosted decision tree neural network. arXiv:1910.09340. 2019.
19. Gao M, Shen Y, Li Q, Loy CC. Residual knowledge distillation. arXiv:2002.09168. 2020.
20. Su L, Wang D, Zhu J. DKD-pFed: a novel framework for personalized federated learning via decoupling knowledge distillation and feature decorrelation. Expert Syst Appl. 2025;259(15):125336. doi:10.1016/j.eswa.2024.125336.
21. Kim M, Saad W, Debbah M, Hong CS. SpaFL: communication-efficient federated learning with sparse models and low computational overhead. Adv Neural Inf Process Syst. 2024;37:86500–27. doi:10.52202/079017-2747.
22. Miao R, Koyuncu E. Federated momentum contrastive clustering. ACM Trans Intell Syst Technol. 2024;15(4):1–9. doi:10.1145/3653981.
23. Huang W, Ye M, Du B, Gao X. Few-shot model agnostic federated learning. In: Proceedings of the 30th ACM International Conference on Multimedia; 2022 Oct 10–14; Lisboa, Portugal, p. 7309–16. doi:10.1145/3503161.3548764.
24. Tian J, Yao C, He T, Shen Y, Zhang J, Zhong X. A federated learning model for small-sample scenarios. Appl Sci. 2024;14(9):3919. doi:10.3390/app14093919.
25. Wang S, Fu X, Ding K, Chen C, Chen H, Li J. Federated few-shot learning. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2023 Aug 6–10; Long Beach, CA, USA. p. 2374–85. doi:10.1145/3580305.3599347.

26. Wu C, Wu F, Lyu L, Huang Y, Xie X. Communication-efficient federated learning via knowledge distillation. Nat Commun. 2022;13(1):2032. doi:10.1038/s41467-022-29763-x.
27. Yu T, Zhao X, An Y, Tang M, Wang J. Knowledge distillation dealing with sample-wise long-tail problem. In: Proceedings of the Asian Conference on Computer Vision; 2024 Dec 8–12; Hanoi, Vietnam. p. 2354–70. doi:10.1007/978-981-96-0972-7_24.
28. McLaughlin CJ, Su L. Personalized federated learning via feature distribution adaptation. Adv Neural Inf Process Syst. 2024;37:77038–59. doi:10.52202/079017-2451.
29. Oh J, Kim S, Yun SY. FedBABU: towards enhanced representation for federated image classification. arXiv:2106.06042. 2022.
30. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Jun 27–30; Las Vegas, NV, USA. p. 770–8. doi:10.1109/CVPR.2016.90.
31. Li Z, Zhong Z, Zuo P, Zhao H. A personalized federated learning method based on the residual multi-head attention mechanism. J King Saud Univ Comput Inf Sci. 2024;36(4):102043. doi:10.1016/j.jksuci.2024.102043.
32. Arivazhagan MG, Aggarwal V, Singh AK, Choudhary S. Federated learning with personalization layers. arXiv:1912.00818. 2019.
33. Sun H, Pei J, Xu X, Xue R, Zhao L, Sun Q, et al. FedMKD: multi-teacher knowledge distillation for communication-efficient federated learning. Clust Comput. 2025;28(10):638. doi:10.1007/s10586-025-05392-z.
34. Chen L, Zhang W, Dong C, Zhao D, Zeng X, Qiao S, et al. FedTKD: a trustworthy heterogeneous federated learning based on adaptive knowledge distillation. Entropy. 2024;26(1):96. doi:10.3390/e26010096.
35. Wang H, Li Y, Xu W, Li R, Zhan Y, Zeng Z. DaFKD: domain-aware federated knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023 Jun 18–24; Vancouver, BC, Canada. p. 20412–21. doi:10.1109/CVPR52729.2023.01955.
36. Wu Z, Sun S, Wang Y, Liu M, Pan Q, Jiang X, et al. FedICT: federated multi-task distillation for multi-access edge computing. IEEE Trans Parallel Distrib Syst. 2024;35(6):1107–21. doi:10.1109/TPDS.2023.3289444.
37. Grinsztajn L, Oyallon E, Varoquaux G. Why do tree-based models still outperform deep learning on typical tabular data? Adv Neural Inf Process Syst. 2022;35:507–20. doi:10.52202/068431-0037.