FedCC: Towards Addressing Label Distribution Skews in Distillation-Based Federated Learning

Wenxuan Ye *†, Onur Ayan*, Xueli An*, Georg Carle†
* *Huawei Heisenberg Research Center, Huawei Technologies Duesseldorf GmbH*
† *TUM School of Computation, Information and Technology, Technical University of Munich*
wenxuan.ye@tum.de, {onur.ayan, xueli.an}@huawei.com, carle@net.in.tum.de

***Abstract*—Federated Learning (FL) enables distributed clients to collaboratively train models without sharing raw data, making it promising for leveraging massive devices in communication networks. In distillation-based FL, each client applies its local model on an *unlabeled public dataset*, and shares only prediction results with the server. While heterogeneous local data introduces *label distribution skew*, thus biasing client models toward majority classes and leading to potentially inaccurate predictions. The lack of ground-truth labels in the public dataset hampers the server's ability to calibrate predictions, which ultimately degrades overall performance. To address this, we propose *FedCC*, a simple and effective algorithm for mitigating client misclassification. Instead of being forced to classify and risking error propagation, clients are allowed to *tag ambiguous samples as 'unknown'*. This additional class, together with calibrated pseudo-labels on the public data, balances confidence in majority classes against uncertainty in under-represented ones. Extensive experiments demonstrate that FedCC significantly outperforms existing methods, especially under severe label skew. In the extreme scenario where each client holds samples from only one of ten classes, FedCC achieves 67.3% accuracy, while baselines collapse to near-random results.**

***Index Terms*—Federated learning, Distillation-based FL, Label distribution skew, Pseudo-labeling**

I. Introduction

Federated Learning (FL) [1] enables collaborative model training across decentralized clients without exposing their raw data. In the classical parameter-based approach, each client refines the global model with its private data, uploads the updated parameters to the server, and the server aggregates these updates to produce the next-round global model. While effective, this approach incurs heavy communication overhead on parameter transmission, and assumes a homogeneous model architecture, overlooking the diverse bandwidth, computation, and data distributions across real-world devices [2]–[4].

Drawing on Knowledge Distillation (KD) [5], Distillation-based FL offers an alternative for client collaboration [6]. Each client runs its local model on a public dataset and uploads only the resulting prediction vectors (i.e., class-score logits); the server then aggregates these vectors into a global prediction vector and redistributes it, reducing communication overhead by orders of magnitude. In addition, because logits are architecture-agnostic, the approach natively supports heterogeneous models, enabling flexible deployment across varied networked devices. Regarding the public dataset, various strategies have been employed, including the use of a labeled dataset [7], or the generation of synthetic data derived from local datasets or local model parameters [8]. Recent studies [9] have relaxed the requirement by adopting unlabeled ones, avoiding costly labeling processes and alleviating privacy concerns. Accordingly, this paper adopts *an unlabeled public dataset* as an integral component of the scenario setting.

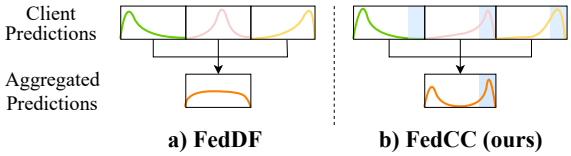


Fig. 1: Comparison of distillation-based FL under label distribution skews. a) FedDF [9] averages biased local predictions, leading to suboptimal outcomes. b) Our FedCC introduces an 'unknown' class (highlighted in blue shadow), allowing clients to acknowledge their unknownness and thereby yielding more informative aggregation.

Despite its promise, distillation-based FL is vulnerable to *label distribution skew*: heterogeneous client data cause local models to overfit the majority classes, and the resulting biases are reflected in the prediction vectors [2]. The challenge is further compounded by the absence of ground-truth labels in the public dataset, which prevents the server from calibrating logits and results in distorted supervision that degrades global performance. Existing approaches in parameter-based FL (e.g., weighted model aggregation [10], model regularization [11]) presume access to client model weights. This assumption breaks down in distillation-based FL, where only logit vectors are exchanged, providing far sparser signals. Recent efforts focus on weighting client updates by estimated reliability during server-side aggregation [12], moving beyond naive averaging as in FedDF [9]. Others aim at amplifying minority-class information and aligning local models with the global objective [13]. Nevertheless, label-skew bias still creeps in, and existing schemes cannot consistently filter out misleading updates during aggregation. Empirical evidence [14] demonstrates that, under label-distribution skew, local models overwhelmingly predict their majority classes, thereby injecting errors into the aggregated prediction (as in Fig. 1a).

In response, we propose *FedCC*, a simple and effective algorithm designed to mitigate *C*lient mis*C*lassification. Treating the mismatch between a client's skewed data and the ideal balanced distribution as an open-set problem [15], we introduce an 'unknown' class to absorb minority and missing classes.

Instead of being forced to issue a class prediction and risking error propagation, clients are allowed to acknowledge their limitations and mark ambiguous inputs as 'unknown'. Illustrated in Fig. 1b, this 'unknown' class (shown in blue shadow) enables clients to express uncertainty, yielding more informative aggregated predictions. To implement this, we incorporate the unlabeled dataset into the local objective function and employ a calibrated pseudo-label generation method, to weigh confidence in majority classes with the uncertainty hidden in unknown classes. This weight is adaptively adjusted per client and per sample, optimizing the model's generalization under diverse data conditions.

To evaluate performance, we conduct experiments on three widely used datasets: CIFAR-10, CIFAR-100, and TinyImageNet, across various label skew scenarios. FedCC consistently outperforms state-of-the-art FL methods by at least 3% points in most scenarios, and the accuracy gap grows as label skew becomes more severe. In the extreme scenario where each client holds samples from only one of ten classes, FedCC achieves an accuracy of 67.3%, while other baselines fail to generate meaningful predictions. Beyond delivering strong global generalization, FedCC enhances local model performance on its minority classes by leveraging uncertainty-aware predictions.

To summarize, our key contributions are as follows:

- We introduce FedCC, a simple and effective method for mitigating client misclassification. We frame label skew as an open-set recognition problem, and extend the existing class framework with an 'unknown' class to absorb minority and missing classes.
- FedCC incorporates the unlabeled dataset into the local objective function, and proposes a novel pseudo-label generation method, enabling the model to recognize and account for unknownness. This method can be interpreted as adding a regularizer, mitigating overfitting.
- Extensive experiments demonstrate that FedCC significantly outperforms current FL methods, and the performance gap increases as label skew becomes more severe.

II. Related Works

Parameter-based FL with label distribution skews: While FL safeguards privacy by keeping data local, the skewed label distributions across clients undermine model generalization. To stabilize heterogeneous training, methods like FedProx [11] and FedSAM [16] employ regularization, variance reduction, or gradient filtering. Other approaches target class imbalance, e.g., via rescaling logits (FedRS [17]), or calibrating predictions (FedLC [18], FedMR [13], FedVLS [19]). Alternatively, FedConcat [20] departs from model averaging by concatenating local encoders to collaboratively train a global classifier. While FedOV [14] uses open-set recognition to alleviate label skew, its reliance on synthesizing outlier samples with implicit labels limits its applicability to unlabeled data.

Distillation-based FL with label distribution skews: To curb label skew under distillation-based FL, FedNed [21] emphasizes distilling minority-class knowledge, LightTS [22] re-weights clients by strength, and Co-Boosting [8] alternates synthetic data generation with ensemble updates. However, their reliance on labeled public datasets or large-scale synthesis raises annotation and privacy concerns. When public data is unlabeled, the lack of ground-truth labels complicates bias correction. Existing solutions address this by retaining high-confidence predictions [7], clustering clients [23], or communicating only hard labels [24]. While emphasizing reliable local knowledge, these methods struggle to filter misinformation stemming from inherent local biases (As shown in Fig. 1a). Thus, effective solutions for fair client-side classification remain an open challenge.

III. Scenario Setting

Consider an FL scenario with K clients and C classes (label set $\mathbb{C} = \{1, \dots, C\}$). Each client k holds a private local dataset $\mathbb{D}_l^{(k)} = \{(x, y)\}$. Additionally, a global unlabeled public dataset $\mathbb{D}_u = \{x_u\}$ is accessible to all clients and the server. Due to label skew, client label priors $P^{(k)}(y)$ vary, though we assume the conditional distribution $P(y|x)$ is consistent across clients [2]. We partition $\mathbb{C}$ into three mutually exclusive subsets ($\mathbb{C}_J^{(k)} \cup \mathbb{C}_N^{(k)} \cup \mathbb{C}_S^{(k)} = \mathbb{C}$) for each client k: majority classes $\mathbb{C}_J^{(k)}$ (sufficiently represented), minority classes $\mathbb{C}_N^{(k)}$ (sparse, noisy estimates), and missing classes $\mathbb{C}_S^{(k)}$ (absent).

Collaborative distillation protocol: Client k deploys a heterogeneous classifier $f^{(k)}(\cdot; \theta^{(k)})$. In each communication round, client k computes logits $f^{(k)}(x_u) \in \mathbb{R}^C$ for public samples $x_u \in \mathbb{D}_u$. These are converted to probabilities $\hat{\mathbf{y}}^{(k)}(x_u) := \sigma^{(k)}(x_u)$ via a temperature-scaled softmax (with temperature $\tau = 1$ omitted for brevity).

The server aggregates these probabilities into an ensemble teacher:

$$\hat{\mathbf{y}}^{(s)}(x_u) := \sum\nolimits_k \mu_k \, \hat{\mathbf{y}}^{(k)}(x_u), \quad \mu_k \geq 0, \; \sum\nolimits_k \mu_k = 1. \quad (1)$$

Clients then synchronize their models by minimizing the Kullback-Leibler divergence with the teacher distribution:

$$\mathbb{E}_{x_u \sim \mathbb{D}_u} \left[\mathrm{D}_{\mathrm{KL}} \left(\hat{\mathbf{y}}^{(s)}(x_u) \parallel \sigma^{(k)}(x_u) \right) \right]. \quad (2)$$

Post-synchronization, clients refine their models on local private data before sharing predictions in the next round.

IV. Approach

FedCC targets client-side misclassification, and by lowering local errors, it ultimately improves global generalization. Sections IV-A-IV-B detail the proposed client-level algorithm, and Section IV-C outlines the overall federated training pipeline. In Sections IV-A–IV-B we focus on a single client, and therefore omit the client index for notational brevity (e.g., f, θ and σ instead of $f^{(k)}$, $\theta^{(k)}$ and $\sigma^{(k)}$).

A. Local learning objective

Semi-supervised learning: We begin by reviewing Semi-Supervised Learning (SSL), a paradigm that leverages both labeled and unlabeled data to build effective models. A classical SSL strategy is self-training [25], which bootstraps a model

with its own high-confidence predictions. Firstly, a classifier $f(\cdot;\theta_0)$ is fitted on labeled data $\mathbb{D}_l = \{(x, y)\}$. The trained model θ_0 then assigns pseudo-labels to the unlabeled data $x_u \in \mathbb{D}_u$ by selecting the class of maximum posterior probability: $\hat{y}(x_u) = \arg\max_{c\in\mathbb{C}} f_c(x_u;\theta_0)$. Merging the original and pseudo-labeled examples yields an enlarged dataset on which the model is re-optimized. The overall training objective is to minimize the following loss function:

$$\begin{aligned}\mathcal{L}(\theta) := \mathbb{E}_{(x,y)\sim\mathbb{D}_l}\left[\ell_{\text{CE}}(f(x;\theta), y)\right] \\ + \lambda\, \mathbb{E}_{x_u\sim\mathbb{D}_u}\left[\ell_{\text{CE}}(f(x_u;\theta), \hat{y}(x_u))\right]\end{aligned} \tag{3}$$

where the cross-entropy loss $\ell_{\text{CE}}(f(x), y) = -\log \sigma_y(x)$ quantifies the discrepancy between the label y and prediction logits $f(x)$, and $\sigma_y(x)$ is given as the probability after the softmax operation; $\lambda > 0$ balances the labeled and unlabeled terms.

Our method: As described in the scenario setting, client k maintains a labeled dataset $\mathbb{D}_l^{(k)}$ and unlabeled dataset $\mathbb{D}_u$, mirroring the typical SSL structure. To optimally utilize both datasets, we design the objective function based on Eq. 3 as:

$$\begin{aligned}\mathcal{L}(\theta) := \underbrace{\mathbb{E}_{(x,y)\sim\mathbb{D}_l^{(k)}}\left[\ell_{\text{CE}}(f(x;\theta), y)\right]}_{:\mathcal{L}_l(\theta)} \\ + \lambda\, \underbrace{\mathbb{E}_{x_u\sim\mathbb{D}_u}\left[\ell_{\text{CE}}(f(x_u;\theta), \hat{y}(x_u))\right]}_{:\mathcal{L}_u(\theta)}\end{aligned} \tag{4}$$

where $\mathcal{L}_l(\theta)$ and $\mathcal{L}_u(\theta)$ denote the losses on the labeled and unlabeled datasets, respectively.

As introduced before, client k first fits a model θ_0 on its labeled data, detailed as $\theta_0 := \arg\min_\theta \mathcal{L}_l(\theta)$. Ideally, θ_0 would master classes it has encountered, i.e., $\mathbb{C}_J^{(k)} \bigcup \mathbb{C}_N^{(k)}$. However, in practice, the severe under-representation of minority classes $\mathbb{C}_N^{(k)}$ leads to poorly calibrated posterior probabilities, which often drift toward majority classes [18]. Furthermore, θ_0 lacks the ability to recognize samples from missing classes as it has never learned them during training.

To mitigate misclassification of minority and missing classes, we enlarge the label space with an auxiliary 'unknown' class u, which acts as a catch-all for samples with unreliable predictions. With the label space augmented to $\mathbb{C} \cup \{u\}$, the model then satisfies $\sum_{y\in\mathbb{C}} \sigma_y(x_u) + \sigma_u(x_u) = 1$.

Building on this, we ideally partition the unlabeled dataset $\mathbb{D}_u$ into samples from majority classes $\mathbb{C}_J^{(k)}$, and all others. If the sample's true class belongs to client k's majority set $y(x_u) \in \mathbb{C}_J^{(k)}$, the pseudo-label is assigned to: $m(x_u) := \arg\max_{c\in\mathbb{C}_J^{(k)}}(f_c(x_u;\theta_0))$, as abundant local examples enable the client to identify these classes reliably [26]; if $y(x_u) \notin \mathbb{C}_J^{(k)}$, the sample is intended to be classified as u. Then, the ideal pseudo-label assignment is

$$\hat{y}(x_u) = m(x_u) \;\text{ if }\; y(x_u) \in \mathbb{C}_J^{(k)} \;\text{ else }\; u \tag{5}$$

The unlabeled loss $\mathcal{L}_u(\theta)$ is therefore reformulated as:

$$\begin{aligned}\mathbb{E}_{x_u\sim\mathbb{D}_u}\big[\,\mathbf{1}[y(x_u) \in \mathbb{C}_J^{(k)}]\,\ell_{\text{CE}}(f(x_u;\theta), m(x_u)) \\ + (1 - \mathbf{1}[y(x_u) \in \mathbb{C}_J^{(k)}])\,\ell_{\text{CE}}(f(x_u;\theta), u)\,\big]\end{aligned} \tag{6}$$

where the indicator $\mathbf{1}[P]$ equals 1 if the event P is true and 0 otherwise. Since the truth labels $y(x_u)$ are unavailable, we approximate this indicator with a soft weight $\alpha(x_u) \in [0, 1]$:

$$\begin{aligned}\mathcal{L}_u(\theta) \approx \mathbb{E}_{x_u\sim\mathbb{D}_u}\big[\,\alpha(x_u)\,\ell_{\text{CE}}(f(x_u;\theta), m(x_u)) \\ + (1 - \alpha(x_u))\,\ell_{\text{CE}}(f(x_u;\theta), u)\,\big]\end{aligned} \tag{7}$$

The comparison between Eq. 4 and Eq. 7 yields the pseudo-label in probabilistic-vector form:

$$\hat{\mathbf{y}}(x_u) := \alpha(x_u)\cdot\mathbf{e}_{m(x_u)} + (1 - \alpha(x_u))\cdot\mathbf{e}_u \tag{8}$$

where $\mathbf{e}_{m(x_u)}$ is the one-hot vector for the predicted class $m(x_u)$, and $\mathbf{e}_u$ represents that of class u. This pseudo-label distributes probability mass over only two classes: the predicted class $m(x_u)$ and the unknown class u. We detail the concrete design of α in the following subsection.

B. Weight design

The weight $\alpha(x_u)$ re-weighs the local loss on unlabeled data so that learning is guided by global rather than local class prevalence. Instead of relying on class probabilities, which risk privacy leakage, we derive the weights solely from entropies, since they are inexpensive to compute and reflect uncertainty.

Algorithm 1 Algorithm of FedCC pipeline

Require: K clients, global rounds T, local labeled dataset $\{\mathbb{D}_l^{(k)}\}_{k=1}^K$, public unlabeled dataset $\mathbb{D}_u$.

Training:

for $t = 1$ to T **do**

Client k **(in parallel):**

Obtain model $\theta_{sync}^{(k),t}$ by solving the optimization problem specified in Eq. 2. (Omit when $t = 1$)

$\theta_0^{(k),t} \leftarrow \arg\min_{\theta^{(k)}} \mathcal{L}_l(\theta^k)$ ▷ Initialize at $\theta_{sync}^{(k),t}$

$\theta^{(k),t} \leftarrow$ FedCC_Local$(\theta_0^{(k),t}, \mathbb{D}_l^{(k)}, \mathbb{D}_u)$

Compute pseudo-predictions: $\hat{\mathbf{y}}^{(k),t}(x_u) \leftarrow \sigma_\tau^{(k)}(x_u;\theta^{(k),t})\ \forall\, x_u \in \mathbb{D}_u$

Upload $\{\hat{\mathbf{y}}^{(k),t}(x_u)\}$ to server

Server:

$\mu^{(k),t}(x_u) \leftarrow$ normalize $\{1 - \hat{y}_u^{(k),t}(x_u)\}_k$

$\tilde{\mathbf{y}}^{(s),t}(x_u) \leftarrow \sum_k \mu^{(k),t}(x_u)\,\hat{\mathbf{y}}^{(k),t}(x_u)$

Set the unknown-class entry of $\tilde{\mathbf{y}}^{(s),t}(x_u)$ to zero, and renormalize the remaining entries to get $\hat{\mathbf{y}}^{(s),t}(x_u) \in \mathbb{R}^C$

Broadcast $\{\hat{\mathbf{y}}^{(s),t}(x_u)\}$ back to all clients

end for

return $\{\theta^{(k),T}\}_{k=1}^K$

procedure FedCC_Local$(\theta_0^{(k),t}, \mathbb{D}_l^{(k)}, \mathbb{D}_u)$

Generate pseudo-labels via Eq. 8 with $\alpha^{(k),t}(x_u)$ from Eq. 10

Obtain model $\theta^{(k),t}$ by optimizing Eq. 4

return $\theta^{(k),t}$

end procedure

Inference: Given a new input x

Generate $\tilde{\mathbf{y}}^{(s),T}(x_u)$ via Eq. 11

$\hat{y} = \arg\max_{c\in\mathbb{C}} \tilde{\mathbf{y}}_c^{(s),T}(x_u)$

Weight towards majority classes (w_l): Following [27], we regard a solid-color image x_0 as featureless, and probe the classifier's intrinsic bias by feeding it into the pre-trained model, recording $\mathbf{p}^{(k)} := \sigma(x_0; \theta_0)$. To keep our entropy-based weighting scheme responsive even for near-certain predictions, we apply label smoothing [28]:

$$\tilde{\mathbf{p}}^{(k)} = (1-\delta)\,\mathbf{p}^{(k)} + \delta\frac{\mathbf{1}}{|\mathbb{C}|} \tag{9}$$

The weight is then $w_l := H\big(\tilde{\mathbf{p}}^{(k)}\big)$, where $H(\cdot)$ denotes entropy. The smoothed entropy quantifies how strongly the model already favors its local label set and stays strictly positive even under a one-class bias.

Weight over unknown classes (w_u): Classes under-represented by client k are collected in $\mathbb{C}_N^{(k)} \bigcup \mathbb{C}_S^{(k)}$ with number being $|\mathbb{C}| - |\mathbb{C}_J^{(k)}|$. Since the client lacks prior knowledge for those classes, we assign them the maximum uncertainty, defined as the entropy of a uniform distribution. This yields $w_u := \log(|\mathbb{C}| - |\mathbb{C}_J^{(k)}| + 1)$. Here, we add one virtual category to account for residual ambiguity introduced by the majority classes, reflecting that the client remains uncertain even with only one under-represented class.

Since w_l quantifies the client's concentration of probability over majority classes and w_u captures its uncertainty about unknown classes, a natural way to trade off these two effects is through their ratio: $\alpha_{base} = \frac{w_l}{w_l + w_u} \in [0, 1]$. Note that w_l and w_u are client-specific and vary across clients.

To further account for **sample-specific prediction uncertainty**, we quantify the pretrained model's confidence for each unlabeled instance x_u by $w_{x_u} := \min\big(w_l, H(\sigma(x_u; \theta_0))\big)$. As high entropy indicates low confidence in assigning the sample to a class in $\mathbb{C}$, we penalize α_{base} with w_{x_u}, leading to the final $\alpha(x_u)$ expression:

$$\alpha(x_u) := \frac{w_l - w_{x_u}}{w_l + w_u} \tag{10}$$

This weighting factor α limits the extent to which a local model's bias affects pseudo-labeling. α reserves a non-zero probability for the 'unknown' class, allowing the model to consider an alternative label rather than forcing a majority-class assignment. By deferring these uncertain cases, α curbs early mislabelling of minority samples and prevents such errors from cascading through subsequent training rounds.

After generating pseudo-labels, each client updates its local model via Eq. 4 to obtain θ, which is then used to generate predictions shared with the server.

C. Federated training pipeline

The training process runs over T communication rounds. For notational clarity, we use the superscript (k) and (s) to denote the client index and the server-aggregated information, respectively, and t to indicate the communication round.

At the beginning of round t, each client k downloads the aggregated soft targets from the previous round, denoted as $\hat{y}^{(s),t-1}(x_u), \forall x_u \in \mathbb{D}_u$, and then use Eq. 2 to synchronize its local model. Then each client adapts its model to its private labeled dataset $\mathbb{D}_l^{(k)}$, resulting in the intermediate model $\theta_0^{(k),t}$. Unlike prior distillation-based FL schemes that share predictions immediately, FedCC inserts an extra refinement step. Each client generates the pseudo-label via Eq. 8 and obtains the final round-specific parameters $\theta^{(k),t}$ by optimizing the objective in Eq. 4. Each client then evaluates all samples in $\mathbb{D}_u$ and produces temperature-scaled prediction vectors, $\hat{\mathbf{y}}^{(k),t}(x_u) = \sigma_\tau^{(k)}(x_u; \theta^{(k),t}) \in \mathbb{R}^{C+1}$, where the last entry corresponds to the unknown class. These prediction vectors are uploaded to the server. Clients' contributions are weighted by known-class confidence: we take $1 - \hat{y}_u^{(k),t}(x_u)$ as a preliminary weight, then normalize it across clients to produce $\mu^{(k),t}(x_u)$. The server then aggregates the received predictions as

$$\tilde{\mathbf{y}}^{(s),t}(x_u) = \sum\nolimits_k \mu^{(k),t}(x_u)\,\hat{\mathbf{y}}^{(k),t}(x_u) \tag{11}$$

It then zeroes out the unknown-class component and renormalizes the remaining C entries to yield a valid probability vector $\hat{\mathbf{y}}^{(s),t}(x_u) \in \mathbb{R}^C$. The resulting soft targets are then redistributed to all clients to initiate round $t+1$.

During inference, we adopt the ensemble strategy used in FedMD [29] and FedOV [14]. After completing all T rounds, clients transmit their model $f^{(k)}$ with parameters $\theta^{(k),T}$ to the server. The server then aggregates the predictions from all client models using Eq. 11 and selecting the most probable label among the original classes $\mathbb{C}$. Formally, the predicted label is given by $\hat{y} = \arg\max_{c \in \mathbb{C}} \tilde{\mathbf{y}}_c^{(s),T}(x_u)$.

Algorithm 1 details the complete FedCC procedure.

V. Experiments

A. Experimental details

Dataset settings: Following the previous work [18], [19], we conduct classification tasks using CIFAR-10, CIFAR-100 and TinyImageNet datasets. We randomly divide each dataset into two parts: a public dataset and a private dataset. The public dataset, which comprises 10% of the total training samples, has all labels removed and is shared among all participants. To simulate different label skews in the private dataset, we use two partition methods: (1) Dirichlet distribution $p \sim \mathrm{Dir}(\eta)$: for each class c, we sample $p_c \sim \mathrm{Dir}_K(\eta)$, then assign client k a fraction $p_{c,k}$ of the class-c samples to its private set. A lower η value indicates greater heterogeneity. (2) Pathological partition $|\mathbb{C}_J^{(k)}| = N$: each client has data from a fixed number of classes N, with an equal number of samples for each class distributed among the assigned clients.

Baseline: We benchmark FedCC against nine parameter-based methods (FedAvg [1], FedProx [11], FedNova [10], FedRS [17], FedSAM [16], FedLC [18], FedMR [13], FedConcat [20], FedVLS [19]) and six distillation-based methods (FedMD [29], FedDF [9], LSR [28], FedET [30], FedHKT [23], FedCT [24]). For fairness, we exclude methods requiring a labeled public dataset or those based on generating labeled data. For details, please refer to the corresponding paper.

Experiment settings: Inspired by [23], parameter-based FL uses ResNet20 on all nodes; in distillation-based FL, clients split evenly between ResNet14 and ResNet20. The setup

TABLE I: Performance comparison under various label skews. FedCC demonstrates superior performance across almost all scenarios, and an increasing performance gap as label skew intensifies.

Method	TinyImageNet				CIFAR-100				CIFAR-10			
	$p \sim \text{Dir}(\eta)$		$\lvert\mathbb{C}_J^{(k)}\rvert = N$		$p \sim \text{Dir}(\eta)$		$\lvert\mathbb{C}_J^{(k)}\rvert = N$		$p \sim \text{Dir}(\eta)$		$\lvert\mathbb{C}_J^{(k)}\rvert = N$	
	0.5	0.05	40	20	0.5	0.05	20	10	0.5	0.05	2	1
FedAvg	$36.2_{\pm 1.6}$	$24.1_{\pm 1.3}$	$31.1_{\pm 1.9}$	$14.3_{\pm 0.8}$	$49.3_{\pm 1.2}$	$34.2_{\pm 2.1}$	$33.5_{\pm 1.3}$	$18.2_{\pm 1.3}$	$72.6_{\pm 2.6}$	$44.3_{\pm 4.2}$	$37.5_{\pm 1.0}$	$11.0_{\pm 1.1}$
FedProx	$35.0_{\pm 3.0}$	$26.4_{\pm 1.0}$	$28.4_{\pm 0.5}$	$13.1_{\pm 2.0}$	$48.9_{\pm 0.9}$	$29.3_{\pm 0.7}$	$30.6_{\pm 1.8}$	$16.1_{\pm 0.6}$	$70.1_{\pm 2.3}$	$47.2_{\pm 2.0}$	$41.0_{\pm 1.3}$	$10.6_{\pm 0.4}$
FedNova	$34.9_{\pm 1.1}$	$25.7_{\pm 0.6}$	$27.2_{\pm 1.3}$	$14.1_{\pm 1.0}$	$47.8_{\pm 1.7}$	$33.7_{\pm 1.7}$	$37.9_{\pm 1.5}$	$20.9_{\pm 1.1}$	$75.3_{\pm 1.1}$	$42.5_{\pm 0.7}$	$35.2_{\pm 0.8}$	$10.0_{\pm 0.0}$
FedRS	$32.5_{\pm 1.3}$	$22.1_{\pm 1.0}$	$27.7_{\pm 1.4}$	$12.9_{\pm 1.5}$	$46.3_{\pm 0.8}$	$27.8_{\pm 0.9}$	$34.2_{\pm 1.4}$	$17.7_{\pm 1.2}$	$76.6_{\pm 1.9}$	$53.0_{\pm 1.0}$	$35.4_{\pm 1.5}$	$10.0_{\pm 0.0}$
FedSAM	$35.5_{\pm 0.8}$	$30.3_{\pm 1.1}$	$29.4_{\pm 1.2}$	$14.1_{\pm 0.9}$	$48.7_{\pm 1.3}$	$33.7_{\pm 1.3}$	$35.3_{\pm 1.7}$	$16.5_{\pm 0.8}$	$74.6_{\pm 0.6}$	$43.8_{\pm 1.6}$	$32.8_{\pm 1.3}$	$10.0_{\pm 0.0}$
FedLC	$39.2_{\pm 0.9}$	$28.0_{\pm 1.0}$	$28.8_{\pm 1.0}$	$12.5_{\pm 1.3}$	$48.5_{\pm 1.1}$	$34.6_{\pm 0.7}$	$39.4_{\pm 1.1}$	$21.4_{\pm 0.4}$	$76.0_{\pm 1.2}$	$41.5_{\pm 0.6}$	$36.2_{\pm 1.6}$	$10.0_{\pm 0.0}$
FedMR	$34.8_{\pm 0.9}$	$24.7_{\pm 0.5}$	$29.2_{\pm 0.8}$	$15.1_{\pm 1.3}$	$50.1_{\pm 1.2}$	$34.6_{\pm 0.9}$	$38.6_{\pm 1.5}$	$21.8_{\pm 1.0}$	$76.2_{\pm 0.9}$	$57.8_{\pm 1.0}$	$38.4_{\pm 0.9}$	$11.7_{\pm 0.2}$
FedConcat	$28.3_{\pm 2.5}$	$18.7_{\pm 3.9}$	$22.3_{\pm 2.3}$	$9.9_{\pm 1.2}$	$40.3_{\pm 2.4}$	$21.2_{\pm 1.7}$	$20.6_{\pm 0.4}$	$14.7_{\pm 1.7}$	$61.3_{\pm 1.0}$	$34.0_{\pm 2.1}$	$28.5_{\pm 1.2}$	$12.7_{\pm 0.7}$
FedVLS	$26.4_{\pm 1.4}$	$20.7_{\pm 0.4}$	$22.4_{\pm 0.7}$	$15.3_{\pm 2.6}$	$43.5_{\pm 1.6}$	$27.3_{\pm 1.2}$	$27.1_{\pm 3.5}$	$18.3_{\pm 0.3}$	$66.0_{\pm 2.8}$	$39.5_{\pm 0.8}$	$32.9_{\pm 3.6}$	$10.0_{\pm 0.0}$
FedMD	$30.1_{\pm 1.7}$	$24.8_{\pm 1.5}$	$30.2_{\pm 0.8}$	$23.1_{\pm 1.6}$	$36.4_{\pm 0.7}$	$28.1_{\pm 0.7}$	$34.9_{\pm 1.2}$	$26.4_{\pm 1.6}$	$69.0_{\pm 0.7}$	$34.2_{\pm 2.6}$	$54.5_{\pm 1.0}$	$10.7_{\pm 0.9}$
FedDF	$26.8_{\pm 2.0}$	$16.9_{\pm 2.7}$	$20.7_{\pm 0.6}$	$10.3_{\pm 2.8}$	$33.6_{\pm 1.2}$	$22.1_{\pm 0.8}$	$27.6_{\pm 3.1}$	$14.1_{\pm 1.1}$	$63.9_{\pm 1.9}$	$39.9_{\pm 3.7}$	$50.8_{\pm 1.4}$	$11.7_{\pm 1.0}$
LSR	$26.2_{\pm 0.9}$	$16.7_{\pm 1.1}$	$23.8_{\pm 1.2}$	$10.2_{\pm 0.6}$	$31.8_{\pm 1.6}$	$21.3_{\pm 0.8}$	$27.8_{\pm 1.4}$	$16.5_{\pm 0.3}$	$58.6_{\pm 1.0}$	$31.4_{\pm 0.7}$	$27.2_{\pm 1.4}$	$10.0_{\pm 0.0}$
FedET	$23.8_{\pm 3.8}$	$18.5_{\pm 2.0}$	$21.3_{\pm 0.8}$	$11.6_{\pm 1.0}$	$27.2_{\pm 0.9}$	$22.3_{\pm 1.7}$	$28.0_{\pm 1.1}$	$22.1_{\pm 0.8}$	$62.1_{\pm 2.0}$	$38.4_{\pm 2.7}$	$51.2_{\pm 0.6}$	$10.0_{\pm 0.0}$
FedHKT	$27.7_{\pm 1.0}$	$21.3_{\pm 0.9}$	$26.9_{\pm 1.7}$	$14.6_{\pm 0.4}$	$34.4_{\pm 1.9}$	$25.6_{\pm 0.5}$	$31.8_{\pm 3.6}$	$21.3_{\pm 0.9}$	$67.6_{\pm 1.4}$	$44.9_{\pm 0.9}$	$49.1_{\pm 0.4}$	$11.1_{\pm 1.2}$
FedCT	$25.9_{\pm 0.5}$	$17.7_{\pm 1.3}$	$21.2_{\pm 0.7}$	$9.9_{\pm 2.6}$	$33.2_{\pm 0.4}$	$21.8_{\pm 1.6}$	$26.7_{\pm 0.8}$	$12.2_{\pm 2.3}$	$65.0_{\pm 1.2}$	$29.7_{\pm 1.7}$	$24.1_{\pm 0.2}$	$10.0_{\pm 0.0}$
FedCC	$40.6_{\pm 0.8}$	$34.8_{\pm 1.0}$	$36.9_{\pm 2.0}$	$34.3_{\pm 1.3}$	$48.9_{\pm 1.3}$	$40.5_{\pm 1.8}$	$45.3_{\pm 1.4}$	$42.6_{\pm 1.3}$	$74.7_{\pm 1.6}$	$60.9_{\pm 2.2}$	$68.2_{\pm 1.1}$	$67.3_{\pm 1.8}$

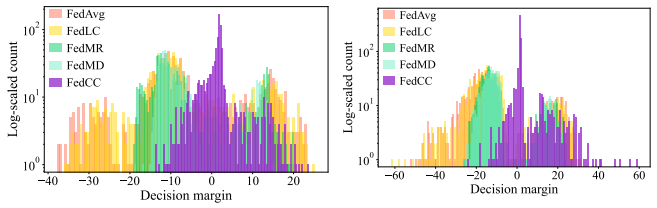
TABLE II: Local model performance over local and global test data, where FedCC leads in almost every case. (CIFAR-100 subset)

Test data	Local model	$p \sim \text{Dir}(\eta)$ 0.5	$p \sim \text{Dir}(\eta)$ 0.05	$\lvert\mathbb{C}_J^{(k)}\rvert = N$ 20	$\lvert\mathbb{C}_J^{(k)}\rvert = N$ 10
Local	FedAvg	$37.9_{\pm 2.2}$	$60.0_{\pm 2.7}$	$57.4_{\pm 3.1}$	$72.3_{\pm 4.9}$
	FedLC	$36.1_{\pm 1.4}$	$60.5_{\pm 4.4}$	$56.9_{\pm 5.1}$	$72.2_{\pm 6.0}$
	FedMR	$36.7_{\pm 2.2}$	$60.2_{\pm 5.1}$	$56.7_{\pm 4.7}$	$70.6_{\pm 5.3}$
	SSL	$34.6_{\pm 2.0}$	$58.0_{\pm 2.9}$	$54.6_{\pm 2.0}$	$70.5_{\pm 5.2}$
	FedCC	$38.2_{\pm 2.3}$	$64.4_{\pm 2.6}$	$60.0_{\pm 4.6}$	$76.6_{\pm 3.3}$
Global	FedAvg	$17.6_{\pm 1.4}$	$10.1_{\pm 0.9}$	$11.3_{\pm 0.8}$	$7.2_{\pm 0.6}$
	FedLC	$19.0_{\pm 1.0}$	$11.0_{\pm 1.6}$	$11.8_{\pm 1.0}$	$7.0_{\pm 0.8}$
	FedMR	$19.3_{\pm 1.5}$	$10.7_{\pm 1.7}$	$11.6_{\pm 1.1}$	$7.0_{\pm 0.8}$
	SSL	$15.4_{\pm 1.1}$	$10.2_{\pm 1.3}$	$11.4_{\pm 0.9}$	$7.3_{\pm 0.5}$
	FedCC	$71.1_{\pm 0.3}$	$74.7_{\pm 1.2}$	$71.1_{\pm 2.4}$	$77.9_{\pm 3.8}$

includes 10 clients, 15 global epochs, and 20 local epochs per step in client model training. We adopt Adam optimizer with batch size 64, learning rate 0.001. Temperature τ is set to 1, tradeoff λ in Eq. 4 to 0.2, δ in $\tilde{\mathbf{p}}^{(k)}$ (Eq. 9) to 10^{-3}. We categorize classes by client k's bias $\mathbf{p}^{(k)}$ (Eq. 9): a class c is a majority if $p_c^{(k)} > 2\%$, minority if $0 < p_c^{(k)} \leq 2\%$, and missing if $p_c^{(k)} = 0$. We report the results based on three experiments conducted with different random seeds, presenting model accuracy with the percent sign omitted from the reported values.

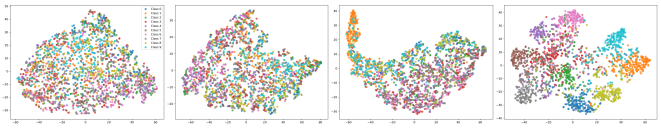
B. An overall comparison

Experiment results, as summarized in Table I, show that FedCC outperforms current FL methods across almost all tested scenarios, with its advantage widening as label skew intensifies. On heavy skew ($p \sim \text{Dir}(0.05)$ or pathological partition), it is ahead by at least 3% points and often by more than 10%. In the harshest scenario where each client has access to only one class of CIFAR-10, baselines collapse to near-random guessing ($\leq$ 12.7%), yet FedCC still delivers a robust 67.3%. This robustness stems from permitting clients to only predict familiar data and to label uncertain samples as 'unknown', thereby abstaining from unreliable predictions and reducing misleading information.



(a) $p \sim \text{Dir}(0.05)$ (b) $\lvert\mathbb{C}_J^{(k)}\rvert = 2$

Fig. 2: Distribution of decision margins for local model on CIFAR-10.



(a) FedAvg local (b) FedAvg global (c) FedCC local (d) FedCC global

Fig. 3: T-SNE visualizations of learned representations on CIFAR-10 where the local distribution follows $\lvert\mathbb{C}_J^{(k)} = 1\rvert$.

Comparison of local model performance: We evaluate the performance of local models on both local and global test data, as shown in Table II. The models are generated using FedAvg [1], FedLC [18], FedMR [13], standard SSL [25], and our FedCC. For the global test data, if the true label does not belong to the majority classes, classifying it as the 'unknown' class is considered valid. As expected, FedCC shows significant improvement on the global test data. On the local test set, FedCC leverages uncertainty to supply richer signals for minority classes and increases their likelihood of being correctly learned, consistently outperforming all baselines.

C. Performance visualization

Decision margin: The decision margin measures the gap between the model's score for the correct class and its highest incorrect guess. Evaluating under two challenging data partitions, Figure 2 shows margin histograms with a logarithmic y-axis. FedCC markedly shifts the distribution rightward to achieve a positive mean, whereas the best competing method

remains negative. This rightward shift confirms that FedCC effectively ensures clear class separation and suppresses client-side misclassification, even under severe label skew.

T-SNE analysis of learned representations: Figure 3 visualizes the learned feature representations where each client holds data from a single class ($|\mathbb{C}_J^{(k)}| = 1$) on CIFAR-10, where each color denotes a distinct class. The embeddings are obtained by applying PCA whitening followed by t-SNE in a unified pipeline. *Baseline*: each client trains a local model using standard cross-entropy on its data, with the embedding of a representative client shown in Fig.3a, and that of the FedAvg global model in Fig.3b. Due to the lack of inter-class variation, local representations collapse along class-specific axes, and averaging these misaligned feature spaces yields weak global performance. *FedCC*: we next train local models with FedCC's objective and aggregate them by ensembling their soft predictions. The corresponding client-side embedding (Fig.3c) cleanly isolates the client's own class from the "unknown" region, and the aggregated embedding (Fig.3d) recovers distinct clusters for all classes. FedCC thus curbs error propagation by explicitly separating learned versus unseen classes at the client level, allowing the server to combine information without blurring minority categories.

VI. Conclusion

We present FedCC, a novel distillation-based FL algorithm that mitigates client misclassification by incorporating an auxiliary 'unknown' class. By enabling clients to defer ambiguous predictions to this class, FedCC acts as an implicit regularizer, preventing overconfidence on majority classes and suppressing noisy updates. Extensive empirical evaluation confirms that it consistently outperforms state-of-the-art baselines in both local and global performance. Crucially, FedCC's lightweight reliance on logit exchanges and robust performance make it well suited for deployment in real-world communication networks, which involves large numbers of devices with heterogeneous compute power, bandwidth, and data distributions.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017.

[2] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji *et al.*, "Advances and open problems in federated learning," *Foundations and trends® in machine learning*, 2021.

[3] W. Ye, C. Qian, X. An, X. Yan, and G. Carle, "Advancing federated learning in 6g: A trusted architecture with graph-based analysis," in *GLOBECOM 2023-2023 IEEE Global Communications Conference*. IEEE, 2023, pp. 56–61.

[4] B. Liu, W. Y. Poe, R. Trivisonno, and G. Caire, "Foundation models for generalizable semantic and goal-oriented communication," in *ICC 2026 - IEEE International Conference on Communications*, 2026, pp. 1–6.

[5] G. Hinton, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[6] W. Ye, Y. Zhang, X. An, G. Carle, and Y. Ma, "Select to think: Unlocking SLM potential with local sufficiency," in *Forty-third International Conference on Machine Learning*, 2026.

[7] Z. Li, X. Wang, D. Hu, N. M. Robertson, D. A. Clifton, C. Meinel, and H. Yang, "Not all knowledge is created equal: Mutual distillation of confident knowledge," in *Workshop on trustworthy and socially responsible machine learning, NeurIPS 2022*, 2022.

[8] R. Dai, Y. Zhang, A. Li, T. Liu, X. Yang, and B. Han, "Enhancing one-shot federated learning through data and ensemble co-boosting," *arXiv preprint arXiv:2402.15070*, 2024.

[9] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," *Advances in neural information processing systems*, 2020.

[10] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, 2020.

[11] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of machine learning and systems*, 2020.

[12] X. Gong, A. Sharma, S. Karanam, Z. Wu, T. Chen, D. Doermann, and A. Innanje, "Ensemble attention distillation for privacy-preserving federated learning," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

[13] Z. Fan, J. Yao, R. Zhang, L. Lyu, Y. Zhang, and Y. Wang, "Federated learning under partially class-disjoint data via manifold reshaping," *arXiv preprint arXiv:2405.18983*, 2024.

[14] Y. Diao, Q. Li, and B. He, "Towards addressing label skews in one-shot federated learning," in *International conference on learning representations*, 2023.

[15] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE transactions on pattern analysis and machine intelligence*, 2012.

[16] Z. Qu, X. Li, R. Duan, Y. Liu, B. Tang, and Z. Lu, "Generalized federated learning via sharpness aware minimization," in *International conference on machine learning*, 2022.

[17] X. Li and D. Zhan, "Fedrs: Federated learning with restricted softmax for label distribution non-iid data," in *Proceedings of the ACM SIGKDD conference on knowledge discovery & data mining*, 2021.

[18] J. Zhang, Z. Li, B. Li, J. Xu, S. Wu, S. Ding, and C. Wu, "Federated learning with label distribution skew via logits calibration," in *International Conference on Machine Learning*, 2022.

[19] K. Guo, Y. Ding, J. Liang, Z. Wang, R. He, and T. Tan, "Exploring vacant classes in label-skewed federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[20] Y. Diao, Q. Li, and B. He, "Exploiting label skews in federated learning with model concatenation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[21] J. Lu, S. Li, K. Bao, P. Wang, Z. Qian, and S. Ge, "Federated learning with label-masking distillation," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023.

[22] D. Campos, M. Zhang, B. Yang, T. Kieu, C. Guo, and C. S. Jensen, "Lightts: Lightweight time series classification with adaptive ensemble distillation," *Proceedings of the ACM on management of data*, 2023.

[23] Y. Deng, J. Ren, C. Tang, F. Lyu, Y. Liu, and Y. Zhang, "A hierarchical knowledge transfer framework for heterogeneous federated learning," in *IEEE conference on computer communications*. IEEE, 2023.

[24] A. Abourayya, J. Kleesiek, K. Rao, E. Ayday, B. Rao, G. I. Webb, and M. Kamp, "Little is enough: Boosting privacy by sharing only hard labels in federated semi-supervised learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[25] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Annual meeting of the association for computational linguistics*, 1995.

[26] Y. Zou, Z. Yu, B. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proceedings of the European conference on computer vision*, 2018.

[27] H. Lee and H. Kim, "Cdmad: Class-distribution-mismatch-aware debiasing for class-imbalanced semi-supervised learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024.

[28] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng, "Revisiting knowledge distillation via label smoothing regularization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[29] D. Li and J. Wang, "Fedmd: Heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[30] Y. J. Cho, A. Manoel, G. Joshi, R. Sim, and D. Dimitriadis, "Heterogeneous ensemble knowledge transfer for training large models in federated learning," *International joint conferences on artificial intelligence*, 2022.