Dual Class-Aware Contrastive Federated Semi-Supervised Learning

Qi Guo, Di Wu, Yong Qi, and Saiyu Qi

Abstract—Federated semi-supervised learning (FSSL) facilitates labeled clients and unlabeled clients jointly training a global model without sharing private data. Existing FSSL methods predominantly employ pseudo-labeling and consistency regularization to exploit the knowledge of unlabeled data, achieving notable success in raw data utilization. However, the effectiveness of these methods is challenged by large deviations between uploaded local models of labeled and unlabeled clients, as well as confirmation bias introduced by noisy pseudo-labels, both of which negatively affect the global model’s performance. In this paper, we present a novel FSSL method called Dual Class-aware Contrastive Federated Semi-Supervised Learning (DCCFSSL). This method considers both the local class-aware distribution of each client’s data and the global class-aware distribution of all clients’ data within the feature space. By implementing a dual class-aware contrastive module, DCCFSSL establishes a unified training objective for different clients to tackle large deviations and incorporates contrastive information in the feature space to mitigate confirmation bias. Additionally, DCCFSSL introduces an authentication-reweighted aggregation technique to improve the server’s aggregation robustness. Our comprehensive experiments show that DCCFSSL outperforms current state-of-the-art methods on three benchmark datasets and surpasses the FedAvg with relabeled unlabeled clients on CIFAR-10, CIFAR-100, and STL-10 datasets.

Index Terms—Distributed learning, federated learning, federated semi-supervised learning.

I. INTRODUCTION

Federated learning (FL) is an emerging distributed learning approach where isolated clients collaboratively train a global model without sharing their private data. Despite its advancement, FL confronts challenges like differences in data sources and high labeling costs, which contribute to statistical and annotation heterogeneity. Statistical heterogeneity refers to differences in data distribution across clients, while annotation heterogeneity means the presence of both labeled and unlabeled datasets. Existing FL methodologies, including FedAvg [1], Fedprox [2], SCAFFOLD [3], FedNova [4], and MOON [5], have notably addressed statistical heterogeneity. However, their reliance on supervised learning (SL) with fully labeled data from each client poses practical limitations.

To leverage widely-existing unlabeled datasets to further improve FL performance, we focus on federated semi-supervised learning (FSSL) with fully labeled and fully unlabeled clients in this work. This typical scenario has many practical applications. For example, in healthcare, some institutions might have well-annotated medical records, while others, especially in resource-constrained regions, might only have access to unlabeled data. FSSL in this context allows for collaborative model training across varied datasets, improving diagnostic models without compromising patient privacy. In the realm of autonomous driving, different companies might have varying levels of data annotation. Some might have extensively labeled datasets from test drives, while others might have large volumes of unlabeled sensory data. FSSL allows these entities to collaboratively enhance autonomous driving algorithms. Existing FSSL methods mainly employ pseudo-labeling and consistency regularization to utilize knowledge of unlabeled data [6], [7], [8]. For instance, FedMatch [9] considers inter-client consistency loss to encourage consistent outputs from different clients to improve the global model. FedConsist [6] aims to minimize the difference between pseudo-labels generated from unlabeled images and predictions from the same unlabeled images after augmentation. RSCFed [8] that performs random sub-sampling over clients to achieve consensus optimizes the labeled and unlabeled clients by supervised cross-entropy and mean-teacher-based consistency loss.

However, these methods suffer from two main limitations. One is the large deviation of uploaded local models from labeled clients and unlabeled clients. The other is confirmation bias [10] induced by noisy pseudo labels. This large deviation of uploaded local models not only stems from the non-independent and identically distributed distribution (i.e., NonIID) of clients’ data but also from the difference in training objective functions employed by labeled and unlabeled clients. Specifically, labeled clients generally use cross-entropy for training on labeled data, approximating the actual data distribution with the predicted data distribution through maximum likelihood estimation. Conversely, unlabeled clients typically adopt pseudo-label-based consistency regularization for training on unlabeled data, ensuring that the learned decision boundary lies within a low-density region and that similar data points yield similar outputs. These divergent training objectives lead to substantial disparities in
the local models uploaded from labeled and unlabeled clients, even in datasets without statistical heterogeneity. Additionally, confirmation bias emerges from overfitting to self-generated labels, as model-generated pseudo-labels often contain substantial noise due to the model’s inherent inaccuracies. Consequently, FSSL performance is substantially compromised by these two limitations. Note that RSCFed [8] alleviates the deviation to a certain extent through performing random sub-sampling over clients. However, reducing the deviation by aggregation of models without considering the difference of training objectives still suffers from large deviations. Moreover, to alleviate confirmation bias, FedConsist [6] employs a high confidence threshold to filter out inaccurately pseudo-labeled data, but relying solely on model output without integrating additional information remains vulnerable to confirmation bias.

To address the aforementioned limitations, we aim to uncover consistent knowledge at a deeper level, transcending the disparities between labeled and unlabeled clients in order to facilitate collaboration. While labeled and unlabeled clients exhibit both statistical and annotation heterogeneity, their data distributions can be viewed as components of the global data distribution. This implies that all clients share an implicit representation space corresponding to the global class distribution, irrespective of their data being labeled or not. In essence, the most valuable common knowledge beyond client differences is the global class distribution shared by all clients. To capture and utilize this shared knowledge effectively, we introduce the concept of class prototypes. A prototype, in this context, serves as a compact representation of the common features or characteristics of a particular class. Prototypes can be thought of as ‘average’ representations of samples within the same class, providing a straightforward and effective way to summarize class-level information. Given that prototypes can serve as a straightforward and effective representation for samples within the same class [11], [12], [13], we introduce class prototype contrastive learning to reduce large deviations by guiding all clients towards a common learning direction and mitigate confirmation bias by incorporating contrastive information.

Nonetheless, two main challenges persist: one is the mismatch between global and local class prototypes, and the other is the unreliability of class prototypes, particularly for unlabeled clients. Owing to the decentralized nature and statistical heterogeneity of clients’ data in FSSL, the class distribution of an individual client’s local data (i.e., local class-aware distribution) may diverge from the class distribution of the global data encompassing all clients (i.e., global class-aware distribution). Solely relying on the local class-aware distribution is inadequate to address the challenges posed by NonIID data in the context of data statistical heterogeneity. On the other hand, focusing exclusively on the global class-aware distribution may result in inconsistencies between the global class-aware distribution and each client’s potential feature space, leading to significant fluctuations in the global model and impairing the local performance of some clients during the training process. Hence, it is crucial to consider the impact of both global class-aware distribution and local class-aware distribution simultaneously, which entails conducting contrastive learning of global class prototypes and local class prototypes concurrently. Furthermore, due to the inherent inaccuracies in the model’s output, the generated prototypes may not be entirely reliable. For unlabeled clients, it is impossible to classify the generated prototypes based on data labels as we do for labeled clients, given the absence of data labels. To make use of the prototypes of unlabeled data, we can label and classify them using pseudo-labels. However, directly applying pseudo-labels to classify the generated prototypes can result in a considerable decline in model performance. This issue arises from the unreliability of pseudo-labels, which leads to the unreliability of the generated prototypes, ultimately causing the model to train in the wrong direction. In summary, addressing these two predominant challenges - the mismatch between global and local class prototypes and the unreliability of class prototypes, especially for unlabeled clients - is paramount. We must consider both global and local class-aware distributions simultaneously and enhance the robustness of the class prototypes, mitigating the limitations posed by NonIID data and the inherent inaccuracies of the model’s output.

To this end, we propose Dual Class-aware Contrastive Federated Semi-Supervised Learning (DCCFSSL), which mitigates large deviations of uploaded local models from significantly different training goals and alleviates confirmation bias induced by noisy pseudo-labels. On the one hand, to reduce large deviations from different clients, a unified training goal for different clients is introduced through the dual class-aware contrastive module. On the other hand, to alleviate the confirmation bias, the training process of each client is regularized by employing dual class-aware information in the feature space. As depicted in Fig. 1, DCCFSSL considers both local and global class-aware distributions within the feature space to tackle the mismatch between global class prototypes and local class prototypes. The local class prototypes reflect the local class data distribution under the current model, whereas the global class prototypes reflect the global class data distribution under the current model. At the start of training, there’s often a significant mismatch between the local and global prototypes due to the random nature of model parameter initialization. To address this, our training process incorporates a dual class-aware contrastive mechanism at the client level. This strategy enables the model to align with the global class data distribution (as represented by the global class prototypes), while also considering the local class data distribution. Subsequently, refined local class prototypes are derived from the local data, leveraging the newly aligned model. Then, new global class prototypes are formulated by aggregating these updated local class prototypes on the server. Through this iterative process, the model is finally aligned to the global data distribution while considering the local data distribution, thereby achieving consensus learning. Moreover, we propose the concept of authentication samples to handle the unreliability of class prototypes. Based on authentication samples, we present authentication-reweighted aggregation techniques to bolster the robustness of the global model and class prototypes. Extensive experiments demonstrate that our method surpasses state-of-the-art approaches on three benchmark datasets and outperforms FedAvg with relabeled unlabeled clients on the CIFAR-10, CIFAR-100, and STL10 datasets. These results suggest that
To our best knowledge, we are the first to identify and relate [1][23][21] is a rapidly evolving distributed adhering to the manifold assumption, which [4][15][8][26][24][5][27][18][25][1][20][24][16][2][14][2][38][313] reduces model volatility under both IID and NonIID settings. Moreover, DCCFSSL greatly enhances the model stability during the training process.

II. RELATED WORKS

A. Federated Learning

Federated learning (FL) [1] is a rapidly evolving distributed machine learning paradigm that enables clients to collaborate on training using individual local data without compromising their privacy. In practice, FL is extensively applied in various critical fields such as industrial Internet of Things (IoT) [14], 6G time-sensitive networks [15], and healthcare [6]. Numerous studies have been conducted to address various challenges faced by FL, including heterogeneity [16], [17], communication efficiency [18], [19], and robustness [20], [21]. Among these, statistical heterogeneity continues to be a significant obstacle, arising from the differences in clients’ data distributions due to their decentralized nature. FedAvg [1], the most prevalent FL baseline, has demonstrated its effectiveness in mitigating statistical heterogeneity. Subsequent research efforts in this domain can be categorized into two complementary perspectives: client-based training methods and server-based aggregation methods. Client-based training methods emphasize regulating the deviation between local models and the global model in the parameter space to stabilize the local training phase, such as FedProx [2], SCAFFOLD [3], MOON [5]. Server-based aggregation methods seek to enhance the efficacy of model aggregation, as exemplified by FedNova [4], FedMA [22], Fedbe [23]. Nevertheless, these methods necessitate supervised samples, while we have to consider the utilization of unlabeled data in FSSL. Most existing methods are not directly applicable to the FSSL setting, as the annotation heterogeneity in training optimization between labeled and unlabeled clients can result in uneven model reliability [8].

B. Semi-Supervised Learning

Pseudo-labeling and consistency regularization form the primary components of semi-supervised learning. Pseudo-labeling methods generate self-predictions for unlabeled data to produce pseudo-labels, allowing the model to self-validate. This self-training process is considered self-confirming. However, due to the risk of overfitting incorrect predictions during training, pseudo-labeling methods are prone to confirmation bias [10]. High-confidence predictions are typically employed to filter noisy unlabeled data [24], [25]. In contrast, consistency regularization methods [26] adhere to the manifold assumption, which posits that different views of the same image should occupy the same position in high-dimensional space, and consequently generate consistent predictions for various image perspectives. Weak and strong augmentations are used to simulate image perturbations and create distinct views of an image [24], [25]. Building upon these techniques, pseudo-labeling and consistency regularization training [24], [25], [27] have achieved remarkable performance.

DCCFSSL allows FSSL to achieve competitive performance even when compared to standard federated supervised learning using labeled data for all clients. Meanwhile, DCCFSSL not only substantially enhances model performance but also greatly reduces model volatility under both IID and NonIID settings.

Concretely, our contributions and novelty can be summarized as follows:

- To our best knowledge, we are the first to identify and address the large deviations between uploaded models from labeled clients and unlabeled clients, as well as the confirmation bias introduced by noisy pseudo-labels in federated semi-supervised learning.
- We propose the DCCFSSL, a novel FSSL method designed to tackle the large deviations of uploaded models and mitigate confirmation bias induced by noisy pseudo-labels. This method simultaneously considers both local and global class-aware distributions to address the mismatch between local class prototypes and global class prototypes. Additionally, we also introduce authentication-reweighted aggregation techniques to enhance the robustness of the global model and class prototypes.
- We conduct a theoretical analysis of the proposed DCCFSSL, establishing its convergence. To the best of our knowledge, this is the first theoretical analysis in our federated semi-supervised learning scenario.
- Comprehensive experiments demonstrate that our DCCFSSL significantly outperforms other state-of-the-art FSSL methods. Notably, DCCFSSL enables FSSL to achieve substantial improvements even when compared to standard federated supervised learning using labeled data for all clients. Furthermore, DCCFSSL greatly enhances the model stability during the training process.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
 Nonetheless, the aforementioned methods primarily concentrate on centralized data, while FSSL addresses the challenges of distributed data under privacy protection for both labeled and unlabeled clients. Instead of focusing on centralized data containing labeled and unlabeled images, this work presents a novel FSSL method that tackles distributed heterogeneous challenges arising from labeled and unlabeled clients.

C. Prototype in Federated Learning

The concept of prototype, defined as the mean of multiple features, has been extensively explored across various tasks, including image classification [11], action recognition [28], and natural language processing benchmarks [29]. FL also utilizes prototypes to enhance its performance. For instance, a federated prototype learning framework has been developed to improve communication efficiency in heterogeneous settings [30]. FedNH [31] combines class prototypes’ uniformity and semantics to boost local models in both personalization and generalization. The authors [32] propose federated prototype learning under domain shift, which involves constructing cluster prototypes and unbalanced prototypes to provide fruitful domain knowledge and a fair convergent target. Prototypes are also applied to specific tasks in FL, such as human activity recognition [33], streaming data [34], and remote sensing images [35]. Existing prototype research in FL is based on supervised learning and doesn’t address the issue of missing labels in client data. In our paper, we consider not only traditional labeled clients but also unlabeled clients, which can take full advantage of widely-existing unlabeled datasets to improve FL performance. To the best of our knowledge, we are the first to propose a prototype-based class-aware contrastive approach in FSSL. Additionally, this is also the first instance of addressing the unreliability of class prototypes in FSSL.

D. Federated Semi-Supervised Learning

FSSL scenarios can be categorized into three groups based on the distribution of labeled data among clients and the server [36]: (1) Labels-At-Server, where clients possess only unlabeled datasets and the server holds a labeled dataset [9], [37], [38]; (2) Labels-At-Clients, where clients own a hybrid dataset consisting of both labeled and unlabeled data [9], [39]; and (3) Labels-At-Partial-Clients, where some clients have labeled datasets while others have unlabeled datasets [6], [8], [40], [41]. FedMatch [9] employs inter-client consistency loss to apply FSSL in the first two scenarios. In the third scenario, various methods are used to enhance FL with unlabeled datasets. RSCFed [8] uses distillation on several sub-consensus models and distance-reweighted model aggregation to leverage unlabeled datasets. FedConsist [6] adopts pseudo-labeling consistency technology in unlabeled clients for a realistic COVID region segmentation. CBAD [40] offers class balanced adaptive pseudo-labeling to leverage unlabeled clients, focusing on a fixed strategy to avoid catastrophic forgetting and adaptive thresholds based on local training data distribution. The authors [41] propose a federated pseudo-labeling strategy in distributed medical image domains, where unlabeled clients learn from the knowledge embedded in labeled clients. Due to the scarcity and high cost of labels in reality, as well as the universality of unlabeled data, the practical application of FL is severely limited if only a small number of labeled datasets are used. Therefore, it is important to develop and leverage knowledge from a large amount of unlabeled datasets in FL. In this work, our focus lies on the third scenario, aiming to leverage widely-existing unlabeled datasets to further improve FL performance, where a dual class-aware contrastive module is introduced for this purpose.

III. METHODOLOGY

In this section, we first outline the problem setting and introduce relevant notations. Then, we present our Dual Class-Aware Contrastive Federated Semi-Supervised Learning (DCCFSSL) approach for FSSL, detailing the local training of clients and the aggregation process of the server. An overview of our DCCFSSL is provided in Fig. 2.

A. FSSL Setting

In this methodology, we consider the FSSL with fully-labeled and fully-unlabeled clients. DCCFSSL involves \( n \) labeled clients and \( m \) unlabeled clients. Labeled client \( k \) has its own local dataset \( D_k^l = \{ (x_i, y_i)_{i=1}^{|D_k^l|} \} \), where \( x_i \in \mathbb{R}^P \) is the \( P \)-dimensional feature vector of a sample, and \( y_i \in \{ 1, 2, \ldots, C \} \) represents its label, and \( |D_k^l| \) is the size of dataset \( D_k^l \). Similarly, unlabeled client \( k \) also has its own local dataset \( D_k^u = \{ (x_i)_{i=1}^{|D_k^u|} \} \), where \( x_i \in \mathbb{R}^P \) is a sample without the label, and \( |D_k^u| \) is the size of dataset \( D_k^u \).

B. Framework

Our proposed DCCFSSL introduces a dual class-aware contrastive module based on the fundamental modules of supervised learning (SL) and semi-supervised learning (SSL), as depicted in Fig. 2. Specifically, in each round, our DCCFSSL conducts the following steps: (1) The client performs local training after receiving a new global model as initialization, then sends the updated model and prototypes back to the server. (2) The server employs authentication-reweighted model aggregation (AMA) and authentication-reweighted prototype aggregation (APA) to obtain the next global model and next global class prototypes on the server; it then randomly samples local clients and sends the current global model and global class prototypes to the selected clients.

C. Local Training

The local training in FSSL consists of a local basic training module and a dual class-aware contrastive module.

**Local basic training module** performs the supervised learning of the weak augmentation in labeled clients, and pseudo-label-based consistency learning in unlabeled clients, respectively. For simplicity, we employ the widely used Fixmatch [25] for pseudo-label-based consistency learning in our method.

**Dual class-aware contrastive module** simultaneously considers both local class-aware distribution and global class-aware distribution. Class-aware distribution is a general term referring to the distribution of data points (or features) across different

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fig. 2. Framework of our proposed DCCFSSL.

classes in a dataset. It reflects how data is spread or clustered concerning their class labels. Local class-aware distribution refers to the class-aware distribution of data within a specific client in a FL setup. For a client \( k \) with dataset \( D_k \), we can define the local class-aware distribution \( P_D^k(y) \) as the set of the class label \( y \) in the client’s local dataset: \( P_D^k(y) = \{ (x, y) \mid x \in D_k : \text{label}(x) = y \} \). Since each client has its own dataset, the local class-aware distribution represents how data is distributed across classes within that individual dataset. It is crucial in FL, where data is not centrally stored but distributed across different clients, each with potentially unique data characteristics. In contrast to the local class-aware distribution, the global class-aware distribution represents the overall distribution of data across all clients in the FL system. The global class-aware distribution \( P_G^y(y) \) can be defined as the set of the class label \( y \) across all clients’ datasets: \( P_G^y(y) = \{ (x, y) \mid x \in \bigcup_k D_k : \text{label}(x) = y \} \). It is a comprehensive view of how data points are distributed across classes when considering the federated data from all clients. Therefore, the dual class-aware contrastive module contains the local and global class-aware contrastive loss functions. The local class-aware contrastive loss function is applied to the data within a single client in an FL environment. It aims to enhance the model’s ability to distinguish between classes based on the unique data characteristics present in the individual client’s dataset. Contrary to the local class-aware contrastive loss function, the global class-aware contrastive loss function addresses the data distribution across all clients in the FL system. It aims to optimize the model’s performance on a more comprehensive scale, taking into account the overall diversity and distribution of data across all clients. For local class-aware distribution, we treat the global class prototype from the same category as a positive pair and other global class prototypes as negative for a specific sample.

Note that DCCFSSL uses data augmentation and extracts representations of samples as follows [27].

Data augmentation presents multiple views of a single image. A weak augmentation \( \text{Aug}_w(z) \) and two strong augmentations \( \text{Aug}_s(z) \) are used for a labeled or unlabeled image \( x \). Two strong augmentations build upon the foundational experiences prevalent in classical contrastive learning [42], [43]. The application of two strong augmentations facilitates the creation of more intricate positive sample pairs, which in turn makes model learning more robust and generalized.

Encoder \( \mathcal{F}(\cdot) \) is used to extract representation \( z = \mathcal{F}(\text{Aug}(x)) \) for a given input \( x \).

1) Labeled Clients: In labeled clients, the basic training module employs a supervised loss \( L_{\text{basic}}^l \) based on \( \text{Aug}_w(\cdot) \) using the cross-entropy:

\[
L_{\text{basic}}^l = \frac{1}{N} \sum_{i=1}^{N} H(y_i, \text{Aug}_w(x_i)),
\]

where \( H \) is the cross entropy.

For the dual class-aware contrastive module in labeled clients, we consider the local class-aware contrastive loss \( L_{\text{lcc}}^l \) and the global class-aware contrastive loss \( L_{\text{gcc}}^l \).

As in [44], [45], we randomly sample a minibatch of \( N \) images, where encoder \( \mathcal{F}(\cdot) \) uses two strong augmentations \( \text{Aug}_s(\cdot) \) of each image \( x_i \) to extract features \( z \). The local class-aware
contrastive loss can be formulated as the following:

\[
\mathcal{L}_{\text{bcc}}^l = -\frac{1}{2N} \sum_{i=1}^{2N} \sum_{j=0}^{C-1} \frac{1}{1 + |S(i)|} \sum_{s \in S(i)} \log \frac{\exp(z_i \cdot z_s / \tau)}{\sum_{j=1}^{2N} \exp(z_i \cdot z_j / \tau)},
\]

where \( \tau \) denotes the temperature parameter and \( S(i) \) represents the indices of the views from other images of the same class. \(|S(i)|\) denotes its cardinality and \( |S(i)| + 1 \) represents all positive pairs.

The global class-aware contrastive loss can be formulated as:

\[
\mathcal{L}_{\text{gcc}}^l = -\frac{1}{2N} \sum_{i=1}^{2N} \sum_{j=0}^{C-1} \frac{1}{1 + |S(i)|} \sum_{s \in S(i)} \log \frac{\exp(z_i \cdot z_s / \tau)}{\exp(z_i \cdot z_j / \tau)},
\]

where \( C \) represents the number of classes in total, \( \bar{z}_j \) is the global prototype of the \( j \)-th class sent by the server for local latent feature alignment of samples belonging to class \( j \). The dual class-aware contrastive loss \( \mathcal{L}_{\text{dcc}}^l \) in labeled client can be expressed as:

\[
\mathcal{L}_{\text{dcc}}^l = \lambda_{\text{bcc}} \mathcal{L}_{\text{bcc}}^l + \lambda_{\text{gcc}} \mathcal{L}_{\text{gcc}}^l,
\]

where \( \lambda_{\text{bcc}} \) and \( \lambda_{\text{gcc}} \) are coefficient factors used to control the influence of the local class-aware distribution and the global class-aware distribution, respectively.

Finally, the total loss for the labeled client can be formulated as:

\[
\mathcal{L}_{\text{total}}^l = \mathcal{L}_{\text{bcc}} + \mathcal{L}_{\text{dcc}}.
\]

2) Unlabeled Clients: The basic training module in unlabeled clients employs pseudo-label-based consistency regularization as the unsupervised loss \( \mathcal{L}_{\text{basic}}^u \). Pseudo labels \( \hat{y}_i = \arg\max(p_i) \) on the weak view of image \( x_i \) are generated by the model’s prediction \( p_i = P_{\text{cls}}(\text{Aug}_{\text{uw}}(x_i)) \) for \( \mathcal{L}_{\text{basic}}^u \). But consistency training will only retain pseudo labels with high confidence \( q \geq T_{\text{thr}} \) where \( q = \max(p) \), and \( T_{\text{thr}} \) is the threshold to ensure the accuracy of pseudo labels. \( \mathcal{L}_{\text{basic}}^u \) can be formulated as:

\[
\mathcal{L}_{\text{basic}}^u = \frac{1}{N} \sum_{i=1}^{N} \log \left( \max(p_i) \geq T_{\text{thr}} \right) H \left( \hat{y}_i, P_{\text{cls}}(\text{Aug}_{\text{uw}}(u_i)) \right),
\]

where \( H \) means cross entropy. \( \mathcal{L}_{\text{basic}}^u \) adopts a pseudo-label consistency regularization approach, a commonly employed technique in semi-supervised learning. Initially, when the model is nascent and only a minority of high-threshold samples are identified, employing ‘N’ as a divisor results in a comparatively lower loss value. This effectively reduces the influence of consistency regularization, which aligns with the initial unreliability of the model. However, as the training progresses and the model begins to identify a larger proportion of high-threshold samples, nearing the total count ‘N’, it indicates an improvement in the model’s performance. Consequently, the effect of consistency regularization loss on each sample becomes more pronounced. This escalation is logical, reflecting the model’s increasing reliability and its ability to discern and learn from more complex patterns. To summarize, direct division by ‘N’ not only facilitates a holistic evaluation of the minibatch but also dynamically scales the impact of consistency regularization in tandem with the model’s evolution from a lower to a higher level of performance.

Similar to labeled clients, the dual class-aware contrastive module in unlabeled clients contains both the local class-aware contrastive loss and the global class-aware contrastive loss. Due to lacking the label of the sample, we assume the images have a high probability to be reliable representations and should be pulled closer with the same class and pushed away from the other classes.

The local class-aware contrastive loss in unlabeled clients can be formulated as:

\[
\mathcal{L}_{\text{bcc}}^u = -\frac{1}{2N} \sum_{i=1}^{2N} \sum_{j=0}^{C-1} \sum_{s \in S(i)} \log \frac{\exp(z_i \cdot z_s / \tau)}{\sum_{j=1}^{2N} \exp(z_i \cdot z_j / \tau)},
\]

where \( S(i) \) represents the indices of the views from other images that are assumed to be of the same class as the current image \( i \) and satisfy the condition \( \max(p) \geq T_{\text{thr}} \). Specifically, an image \( j \) is included in \( S(i) \) if and only if: (i) It belongs to the same predicted class as image \( i \), and (ii) Its maximum predicted probability is greater than or equal to \( T_{\text{thr}} \). We can explicitly define \( S(i) \) as: \( S(i) = \{ j \mid y_i = y_j, j \in \{1, 2, \ldots, 2N\}, j \neq i, \text{and} \max(p_j) \geq T_{\text{thr}}, \text{and} \max(p_i) \geq T_{\text{thr}} \} \). Here, \( \max(p) \) refers to the maximum probability in the model’s prediction for each unlabeled image, and \( T_{\text{thr}} \) is a predefined threshold. \( |S(i)| \) denotes the cardinality of \( S(i) \), and \( |S(i)| + 1 \) represents the total number of positive pairs, including the current image \( i \).

The global class-aware contrastive loss in unlabeled clients can be formulated as:

\[
\mathcal{L}_{\text{gcc}}^u = -\frac{1}{2N} \sum_{i=1}^{2N} \sum_{j=0}^{C-1} \sum_{s \in S(i)} \log \frac{\exp(z_i \cdot z_s / \tau)}{\sum_{j=1}^{2N} \exp(z_i \cdot z_j / \tau)},
\]

where \( C \) represents the indices of the views from other images that are assumed to be of the same class as the current image \( i \) and satisfy the condition \( \max(p) \geq T_{\text{thr}} \). Specifically, an image \( j \) is included in \( S(i) \) if and only if: (i) It belongs to the same predicted class as image \( i \), and (ii) Its maximum predicted probability is greater than or equal to \( T_{\text{thr}} \). We can explicitly define \( S(i) \) as: \( S(i) = \{ j \mid y_i = y_j, j \in \{1, 2, \ldots, 2N\}, j \neq i, \text{and} \max(p_j) \geq T_{\text{thr}}, \text{and} \max(p_i) \geq T_{\text{thr}} \} \). Here, \( \max(p) \) refers to the maximum probability in the model’s prediction for each unlabeled image, and \( T_{\text{thr}} \) is a predefined threshold. \( |S(i)| \) denotes the cardinality of \( S(i) \), and \( |S(i)| + 1 \) represents the total number of positive pairs, including the current image \( i \).

The dual class-aware contrastive loss \( \mathcal{L}_{\text{dcc}}^u \) in unlabeled clients can be expressed as:

\[
\mathcal{L}_{\text{dcc}}^u = \lambda_{\text{bcc}} \mathcal{L}_{\text{bcc}}^u + \lambda_{\text{gcc}} \mathcal{L}_{\text{gcc}}^u.
\]

Finally, the total loss in unlabeled clients is:

\[
\mathcal{L}_{\text{total}}^u = \mathcal{L}_{\text{basic}}^u + \mathcal{L}_{\text{dcc}}^u.
\]

D. Authentication-Reweighted Prototype Aggregation

To enhance the robustness of global class prototypes, we propose a new authentication-reweighted prototype aggregation (APA) method. In this context, authentication samples are proposed and defined as correctly classified samples for labeled clients and the samples with high-confidence pseudo labels for unlabeled clients, respectively. Specifically, for a
labeled client $k$, the authentication samples are defined as $AS_k = \{(x, y) \in \mathcal{D}_k \mid \arg\max(p(x)) = y\}$. For an unlabeled client $k$, the authentication samples are defined as $AS_k = \{(x, y) \in \mathcal{D}_k \mid \max(p(x)) \geq T_{thr}\}$. Here, $p(x)$ denotes the model’s prediction probability vector for the sample $x$, and $T_{thr}$ represents the confidence threshold. After completing the local training, each client would use local data to generate local class prototypes $\mathbf{o}^k = (o^k_0, o^k_1, \ldots, o^k_{C-1})$ by the new local model. The authentication samples of labeled clients are decided by using the training dataset. Note that only authentication samples within each client will be employed to produce the corresponding class prototype. Additionally, let $v^k_j$ represent the number of authentication samples corresponding to the prototype of $j$-th class in $k$-th client. Therefore, the authentication samples number vector is $\mathbf{v}^k = (v^k_0, v^k_1, \ldots, v^k_{C-1})$ in $k$-th client. Therefore, local class prototype $o^k_j$ of $j$-th class in $k$-th client can be formulated as:

$$o^k_j = \frac{1}{v^k_j} \sum_{i \in o^k_j} z_i,$$  

(11)

where $E^k_j$ represents the indices of authentication samples of $j$-th class in $k$-th client. These local class prototypes are derived from the learned representations of the local data and do not contain any raw data or direct loss information. The local class prototypes provide a summarized and anonymized view of the local data, making it difficult to reconstruct or infer sensitive information about individual data points.

On the server, we maintain up-to-date local class prototypes $\mathbf{o}^k$ with the vector $\mathbf{v}^k$ for each client. By aggregating all the latest local class prototypes, we aim to develop global prototypes that encompass the data distribution of all clients, not just a subset. This ensures a more comprehensive and representative prototype aggregation, reflecting the collective data distribution across all clients in FL. In each round, the new $\mathbf{o}^k$ and $\mathbf{v}^k$ are sent by selected clients to the server for updating their prototypes and the number vectors status. Considering the reliability of labeled client’s prototypes and the relative uncertainty of the unlabeled client’s prototypes, we adaptively expand the authentication samples number vector of labeled clients by a factor $\mu = m/n$ (i.e., the ratio of unlabeled clients to labeled clients) while maintaining that of unlabeled clients unchanged. This update of $v^k_j$ can be explicitly expressed as follows:

$$v^k_j = \begin{cases} \mu v^k_j, & \text{if labeled client} \\ v^k_j, & \text{if unlabeled client} \end{cases}$$

Then, server will update the next global class prototypes $O = (O_0, O_1, \ldots, O_{C-1})$ as the following:

$$O_j = \sum_{k=1}^{n+m} \frac{v^k_j}{v^{total}_j} o^k_j, \quad v^{total}_j = \sum_{k=1}^{n+m} v^k_j.$$  

(12)

After updating $O$ by APA, the server distributes the new global class prototypes to selected clients for the next round.

### E. Authentication-Reweighted Model Aggregation

To further enhance the robustness of the models, we also propose an authentication-reweighted model aggregation (AMA) method. This method dynamically adjusts the aggregation weight of local models according to the number of authentication samples from different clients.

In each round, the average model can be represented as:

$$w^{avg} = \sum_{k=1}^{K} \frac{A_k}{A_{total}} w^k, A_{total} = \sum_{k=1}^{K} A_k, A_k = \sum_{j=0}^{C-1} v^k_j,$$  

(13)

where $K$ is the number of selected clients, $C$ is the total number of data categories, and $A_k, k \in \{1, 2, \ldots, K\}$ is authentication samples count of client $k$. $A_k$ quantifies the number of samples in client $k$ that are either correctly classified (for labeled clients) or have high-confidence pseudo labels (for unlabeled clients).

The complete description of DCCFSSL is presented in Algorithm 1.

---

**Algorithm 1: Dual Class-Aware Contrastive Federated Semi-Supervised Learning (DCCFSSL).**

**Input:** the global model $w$, the global class prototypes $O$, the labeled dataset $\mathcal{D}_i$, the unlabeled dataset $\mathcal{D}_u$, maximum training round $R$, and number of subset $\mathcal{K}$.

**Output:** the final model $w_R$

1. **Server executes:**
   1. **initialize** $w^0$, $O^0$
   2. **for** $r = 0$ to $R - 1$ do
      1. **Randomly select** $\{c_{1i}, \ldots, c_{mi}\}$ from $n + m$ clients
      2. **for** $k \leftarrow i_1$ to $i_K$ in parallel do
         1. send the global model $w^r$ to $C_k$
         2. send the global class prototypes $O^r$ to $C_k$
         3. $w^{r+1}_k, o^k, v^k \leftarrow LocalTraining(r, k, w^r, O^r)$
      end
      4. **w**$^{r+1}$ $\leftarrow$ AMA($w_k^{r+1}$, $v^k$)
   end
   3. Return the final model $w^R$

4. **LocalTraining:**($r$, $k$, $w^r$, $O^r$):
   1. **for** epoch $= 1$ to $E$ do
      2. **for each batch** do
         1. **if Labeled client** is True then
            1. **compute** $L^{total} = L^{basic} + L^{dcc}$ by Eq.5
            2. $w^r \leftarrow w^r - \eta \nabla L^{total}$
         end
         3. else
            1. **compute** $L^{total} = L^{basic} + L^{dcc}$ by Eq.10
            2. $w^r \leftarrow w^r - \eta \nabla L^{total}$
         end
   end

5. **generate** local prototypes $o^k = (o^k_0, o^k_1, \ldots, o^k_{C-1})$ with the vector $v^k = (v^k_0, v^k_1, \ldots, v^k_{C-1})$ by $w_k^{r+1}$

6. Return $w^{r+1}_k, o^k$, and $v^k$. 

---
IV. THEORETICAL ANALYSIS

This section provides a theoretical analysis to better illustrate our method. We start by introducing necessary mathematical symbols and common assumptions in theoretical analysis. Then, we describe the problem and present the theorem’s results. A detailed mathematical proof is available in the supplementary material.

A. Assumptions

To facilitate theoretical analysis and enhance its comprehensibility, we present the following commonly used assumptions.

Assumption 1: (L-smooth) For all local objective functions $F_1, \ldots, F_n$ and all model parameters $w$ and $w'$ within the domain of $F_i$, and with a fixed prototype $O$, the following holds:

$$F_i(w', O) \leq F_i(w, O) + \langle w' - w, \nabla F_i(w, O) \rangle + \frac{L}{2} ||w' - w||^2.$$ 

This is equivalent to:

$$||\nabla F_i(w', O) - \nabla F_i(w, O)|| \leq L||w' - w||.$$ 

Assumption 2: (Unbiased local gradient estimator) Let $x_{i,k}^T$ be a random data sample in the $T$-th round for the $i$-th client at the $k$-th step. The local gradient estimator is unbiased:

$$E[\nabla F_i(w_{i,k}^T, O^T, x_{i,k}^T)] = \nabla F_i(w_{i,k}^T, O^T).$$

Assumption 3: (Bounded local variance) There exists a constant $\sigma_l$ such that the variance of each local gradient estimator is bounded by:

$$E[||\nabla F_i(w_{i,k}^T, O^T, x_{i,k}^T) - \nabla F_i(w_{i,k}^T, O^T)||^2] \leq \sigma_l^2$$

Assumption 4: (Bounded global variance) There exists a constant $\sigma_g$ such that the global variability of the local gradient of the objective function is bounded by:

$$E[||\nabla F_i(w^T, O^T) - \nabla F(w^T, O^T)||^2] \leq \sigma_g^2$$

Here, the global objective function $F$ is defined as the weighted average of the local objective functions $F_i$, as shown below:

$$F(w, O) = \sum_{i=1}^{M} \rho_i F_i(w, O), \sum \rho_i = 1,$$ (14)

where $w$ represents the model parameters, $O$ denotes the global prototype, $M$ is the total number of clients, and $\rho_i$ is a coefficient that satisfies $\sum \rho_i = 1$. $F$ and $F_i$ are described in detail in the next subsection.

Assumptions 1 and 2 are commonly used for convergence analysis. Assumptions 3 and 4 are frequently applied in FL, where $\sigma_l$ and $\sigma_g$ represent the sampling noise and the variability in clients’ data distributions, respectively [16], [20], [46]. Compared to traditional FL, FSSL introduces an additional deviation between the uploaded models of labeled and unlabeled clients. Although this deviation could increase the value of $\sigma_g^2$ relative to FL, $\sigma_g^2$ remains finite, thereby preserving the validity of assumption 4. Our proposed method introduces a dual class-aware contrastive module that aims to reduce the deviation between the uploaded models of labeled and unlabeled clients by providing a unified training objective in the feature space. By encouraging all clients to learn consistent class-aware representations, the module helps to mitigate the deviation between the models of different clients.

B. Problem Description

To analyze our proposed method theoretically, we establish an optimization problem in FSSL based on (14). Specifically, considering the FSSL as an extension of FL, our goal is to solve the following optimization problem:

$$\min_w F(w, O) = \sum_{i=1}^{M} \rho_i F_i(w, O), \sum \rho_i = 1,$$ (15)

where $w$ represents the model parameter, $O$ denotes the global prototype, $M$ is the total number of clients, $\rho_i$ is a coefficient greater than 0, and $F$ and $F_i$ represent the global and local objectives, respectively. The local objective’s definition is key to differentiating traditional FL from FSSL. We define a general form of local objective for both labeled and unlabeled clients:

$$F_i(w, O) = E_{\xi \sim D_i} (B(w, \xi) + R_l(w, \xi) + R_g(w, O, \xi)),$$ (16)

where $w$ and $O$ represent the model and the global prototype for calculating the local objective, $\xi$ is a random data point sampled from local dataset $D_i$. The function $B$, serving as the basic loss function, is generally formulated as cross entropy in labeled clients or as consistency loss in unlabeled clients. $R_l$ represents the local regularization which can be calculated by model $w$ and data $\xi$. In practise, it represents $L_{loc}$ in labeled clients’ local objective and $L_{gcc}$ in unlabeled clients’ local objective. $R_g$ is the global regularization which can be calculated by model $w$, data $\xi$ and global prototypes $O$. In practise, it represents $L_{gcc}$ in labeled clients’ local objective and $L_{gcc}$ in unlabeled clients’ local objective.

In each round of training, based on the given local objective, the client performs $K$ steps of Stochastic Gradient Descent (SGD) locally to train its local model. A single step of SGD can be described as:

$$w_{i,k}^T = w_{i,k-1}^T - \eta \nabla F_i(w_{i,k-1}^T, O^T, x_{i,k}^T)$$ (17)

After local training, the client calculates its local prototype and sends it back to the server along with the local model. After receiving the local models and local prototypes from the clients, the server aggregates them to construct the global model and prototypes for the subsequent training round. The aggregation of the global model can be simplified as follows:

$$w^T = \sum \rho_i w_{i,K}^T, \sum \rho_i = 1$$ (18)

Finally, the global model will be sent to clients for the next round of training. It is pertinent to note that we simplify the calculation of prototype $O$. Originally, it was calculated as the average of local prototypes uploaded by the clients. In our theoretical analysis, $O$ is directly estimated based on the global model.
C. Analysis

Lemma 1: (Relaxed triangle inequality) For model parameters $w$ and $w'$, with $a > 0$, then the following is true:

$$\|w + w'\|^2 \leq (1 + a) \|w\|^2 + \left(1 + \frac{1}{a}\right) \|w'\|^2.$$ 

Lemma 2: (Bound of variance among multiple local steps) With all assumptions hold, while $\eta \leq \frac{1}{\sqrt{4KL}}$, it follows that:

$$\mathbb{E} \left\| w^T_{i,k} - w^T \right\|^2 \leq 4K(\eta^2\sigma^2 + 3K\eta^2\sigma^2_g)
+ 3K\eta^2\mathbb{E} \left\| \nabla F(w^T) \right\|^2.$$ 

Theorem 1: With all assumptions hold, with full clients participation, let local learning rate satisfies $\eta \leq \frac{1}{\sqrt{LK}}$, we have the convergence results on proposed method:

$$\min_{t \in (1, T)} \mathbb{E} \left\| \nabla F(w^t, O^t) \right\|^2 \leq \frac{3}{c^2} \frac{K^2\eta^2L^2\sigma^2_t}{c_0T} + \frac{3}{c^2} \frac{K^2\eta^2L^2\sigma^2_t}{c_0T} + \frac{6}{c^2} \frac{K^2\eta^2L^2\sigma^2_g}{c_0T} + \frac{\eta}{2c} \frac{LK^2\sigma^2_g}{c_0T}, \quad (19)$$

where $c$ is a constant that satisfies condition $\frac{1}{\sqrt{T}}K - 6K^3\eta^2L^2 \geq c \geq 0$, $K$ represents the total count of local Stochastic Gradient Descent step, and $F^*$ denotes the optimal solution.

Remark: The convergence bound can be decomposed into two parts. Following the order in Theorem 1, the first part is dominated by the initialization error and can be decaying based on our definition of $\eta$. The second part is dominated by local variance $\sigma^2_t$ and global variance $\sigma^2_g$. If $\eta$ decays by a rate of $\frac{1}{\sqrt{T}}$, then the first part would be decaying by a rate of $O(\frac{1}{\sqrt{T}})$ approximately. The second part would decay by a rate of $O(\frac{1}{T} + \frac{1}{\sqrt{T}})$ approximately. With $T$ growing large, both two parts decay with a rate of $O(\frac{1}{\sqrt{T}})$.

V. Experiments

In this section, we first outline the experimental setup in detail. Next, we evaluate the effectiveness of DCCFSSL in comparison to state-of-the-art FSSL methods. Additionally, we conduct ablation experiments to assess the specific impact of various components. We also investigate the influence of the dual class-aware contrastive module on the stability of the training model. Then, we examine the influence of key hyperparameters on the performance of DCCFSSL. Furthermore, we analyze the additional communication cost introduced by DCCFSSL. Finally, we investigate the effect of different ratios of labeled and unlabeled clients on DCCFSSL.

A. Experimental Setup

1) Datasets and Network Architecture: We perform experiments on three datasets: CIFAR-10 [47], CIFAR-100 [47], and STL-10 [48], implementing both the IID and NonIID data partition settings for all three datasets.

CIFAR-10 [47], as an image classification dataset, consists of 60,000 images across 10 classes, with 6,000 images per class. Each image has a fixed resolution of $32 \times 32$. The training set contains 50,000 images (with 5,000 images per class). This test set contains 10,000 images (1,000 images per class).

CIFAR-100 [47] is also an image classification dataset consisting of 60,000 images in 100 classes, with 600 images for each class. All images have a fixed size of $32 \times 32$ pixels. The training set contains 50,000 images, with 500 images per class. This test set contains 10,000 images, with 100 images per class.

STL-10 [49] is an image recognition dataset, where the labeled examples across 10 classes with 1300 images per class are used in the experiments. Each image has a resolution of $96 \times 96$ pixels and is presented in color. The dataset is partitioned such that 80% of the data from each class is allocated to the training set, while the remaining 20% is reserved for the test set.

The CIFAR-100 dataset, compared to the CIFAR-10 dataset, maintains an equal total number of training samples while featuring a larger number of classes, consequently increasing the complexity of the task in terms of class diversity. In contrast, the STL-10 dataset preserves the same class count as CIFAR-10 but provides significantly fewer overall training samples, thereby implying a heightened difficulty regarding the volume of training data. According to the work [27], we employ the Wide ResNet WRN-16-2 [50] as the backbone network for the experiments on CIFAR-10 and CIFAR-100, while the Wide ResNet WRN-10-2 [50] is used as the backbone network for the experiments on STL-10. Subsequently, a fully connected layer is added for classification purposes. It should be noted that the representation output dimension of the backbone network in this study is set to 128. To guarantee a fair comparison, the same backbone and classification network are utilized for both labeled and unlabeled clients across all methods.

2) Federated Learning Setting: Each dataset’s training set is divided among a total of 50 clients. For the IID setting, we use random sampling to generate IID data partitions across the 50 clients. For the NonIID setting, we follow existing methods [5], [22] using Dirichlet splitting and adopt a Dirichlet distribution Dir($\gamma$) ($\gamma = 1$ for all benchmark datasets in this work) to produce NonIID data partition among the 50 clients. After the IID or NonIID data partition in clients, 10% of the clients are randomly selected as fully labeled clients, while the remaining 90% are unlabeled clients.

3) Baselines: We categorize six baselines into three groups: (1) The standard federated supervised learning method using only labeled clients: FedAvg-SL-Lower [1]. The standard federated supervised learning method using labeled clients and relabeled unlabeled clients: FedAvg-SL-Lower [1]. (2) The naive combination of FL with semi-supervised learning, for example, FedAvg-FixMatch [1], [25] and Fedprox-FixMatch [2], [25]. (3) The state-of-the-art FSSL methods corresponding to this work, e.g., FedConsist [6], RSCFed [8]. The details of the six baselines are presented as follows.

FedAvg-SL-Lower [1]: During the FL training, only labeled clients with labeled data are considered for supervised learning, while unlabeled clients are excluded. Labeled clients use the
The proposed DCCFSSL outperforms the other five base methods. All experiments are implemented using Python 3.9 and PyTorch 1.11 on an Intel(R) Xeon(R) Gold 6226R CPU with 256G memory, an NVIDIA A100 GPU, and Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-150-gener x86_64). We use the SGD optimizer for these experiments. The learning rates in labeled clients and unlabeled clients are set to 1 and 0.95 by default, respectively. To reduce the computational cost, we exclude unlabeled clients from the first half of the global round, while all clients participate as usual in the second half.

For our method, the default hyperparameters $\lambda_{dgc}$ and $\lambda_{gcc}$ in the dual class-aware contrastive loss are both set to 1. The temperature $\tau$ and the high confidence threshold $T_{	ext{thr}}$ are set to 1 and 0.95 by default, respectively. To reduce the computational cost, we exclude unlabeled clients from the first half of the global round, while all clients participate as usual in the second half.

For FedConsist [6], we follow the re-implement setting [8] that enlarges the total weight of selected labeled clients to about 50% and makes selected unlabeled clients share the remaining 50% weight in each global round.

Note that RSCFed [8] fails to directly generalize in this practical setting, so we re-implement RSCFed in experimental results. Specifically, considering that there is only one labeled client in the original setting of RSCFed and the weight adjustment proposal [8], we select all 5 labeled clients and 5 random unlabeled clients in each round, where the number of sub-sampling and the number of clients in each sub-sampling are 2 and 10, respectively. Then we search for the scaling factor hyperparameter $\beta$ from the set $\{1, 100, 10000\}$, which is used for $L_2$ norm of the model gradient between the local model and temporal averaged model within the subset. According to the experimental results, $\beta = 100$ achieves the best performance and is set as the default value. Meanwhile, we try to increase the aggregation weight of labeled clients from the set $\{10\%, 50\%, 70\%, 90\%\}$. Our experiments show that 90% achieve the best classification accuracy. Therefore, we empirically enlarge the total weight of selected labeled clients to about 90% and make selected unlabeled clients share the remaining 10% weight in each global round.

B. Overall Results

The comparative results on three datasets are presented in Table 1. We can make the following observations:

- The proposed DCCFSSL outperforms the other five base lines (FedAvg-SL-Lower, FedAvg-FixMatch, Fedprox-FixMatch, FedConsist, and RSCFed) with substantially higher accuracy across all three datasets. This superior performance across diverse data settings (including both the IID setting and the NonIID setting), datasets, and various benchmark methods convincingly demonstrates the broad generalizability and efficacy of our proposed DCCFSSL.
- DCCFSSL not only surpasses state-of-the-art methods but also demonstrates substantial performance improvements over FedAvg-SL-Upper in both IID and NonIID settings. For instance, on the CIFAR-10 dataset, DCCFSSL achieves accuracy improvements of 11.64% and 9.52% over FedAvg-SL-Upper in IID and NonIID settings, respectively. Likewise, on the CIFAR-100 dataset, DCCFSSL exhibits improvements of 7.25% and 3.18% in accuracy over FedAvg-SL-Upper for IID and NonIID settings, respectively. Moreover, on the STL-10 dataset, DCCFSSL attains 9.88% and 2.92% accuracy enhancements over FedAvg-SL-Upper for IID and NonIID settings, respectively. These results suggest that DCCFSSL, which can necessitate only 10% of clients to be labeled, outperforms conventional federated supervised learning methods that require all clients to be labeled. This highlights the potential of DCCFSSL to achieve competitive performance in federated semi-supervised learning relative to standard federated supervised learning utilizing fully labeled data. DCCFSSL outperforms FedAvg-SL-Upper, even when the latter uses labeled data from all clients, due to three key factors: (1) DCCFSSL’s dual class-aware contrastive module leverages local and global class-aware distributions to learn robust feature representations, mitigating data heterogeneity and improving generalization.
DCCFSSL effectively utilizes unlabeled data through pseudo-labeling and consistency regularization, capturing additional information about the underlying data distribution. (3) DCCFSSL’s authentication-reweighted aggregation assigns higher weights to reliable clients during model and prototype aggregation, mitigating the impact of noisy updates and stabilizing the global model. In contrast, FedAvg-SL-Upper depends exclusively on labeled data for supervised learning, which may overlook other valuable information and potentially result in suboptimal performance.

FedAvg-FixMatch and Fedprox-FixMatch exhibit modest performance improvements over the FedAvg-SL-Lower method on the CIFAR-10, CIFAR-100, and STL-10 datasets; however, these enhancements are limited and do not attain the performance level of FedAvg-SL-Upper. Although semi-supervised algorithms like FixMatch exhibit remarkable performance, it is essential to consider deeper collaborative characteristics in FSSL. As a result, we propose the dual class-aware contrastive module to improve FSSL in the feature space, emphasizing both local and global class-aware distributions.

RSCFed does not consider the differences between different training objectives (cross-entropy loss as the training objective for labeled clients, and consistency regularization loss as the training objective for unlabeled clients). The impact of different training objectives is reflected in the global model’s performance, which is implicitly demonstrated in the performance differences between RSCFed to FedAvg-SL-Lower and DCCFSSL. Taking the IID datasets on CIFAR-10 as an example, RSCFed achieves a slight performance improvement compared to FedAvg-SL-Lower (0.3% improvement in accuracy), with accuracies of 53.94% and 53.64%, respectively. RSCFed differs from FedAvg-SL-Lower in that it uses cross-entropy as the training objective on labeled datasets while adding consistency regularization as the training objective on unlabeled datasets, which means leveraging unlabeled datasets introduces differences in training objectives. The aggregation-based sub-sampling technique in RSCFed alleviates the deviation of locally uploaded models, but it does not address the fundamental differences in training objectives, resulting in only a slight performance improvement over FedAvg-SL-Lower. On the other hand, DCCFSSL considers the differences between the two training objectives and introduces a common learning direction for both objectives through the dual class-aware contrastive module. As a result, DCCFSSL achieves a significant performance improvement compared to RSCFed, reaching an accuracy of 86.62% (surpassing RSCFed by 32.68%).

The impact of different training objectives in RSCFed can be observed from the joint comparison of RSCFed with FedAvg-SL-Lower and DCCFSSL, where these three methods employ different training loss functions.

Table I: Results on CIFAR-10, CIFAR-100, and STL-10 datasets under the IID and NonIID settings (%)

| Labeling Strategy | Method | Client Number | IID | NonIID | | |
|-------------------|--------|---------------|-----|--------|---|---|
|                   |        | (labeled/0)   | (unlabeled/0) | Acc | AUC | Precision | F1 | Acc | AUC | Precision | F1 |
| Fully supervised  |        |               |               |     |     |           |    |     |     |           |    |
| FedAvg-SL-Upper   | [1]    | 50/0          |               | 74.98 | 96.43 | 75.81 | 74.85 | 73.65 | 95.96 | 73.56 | 73.4 |
| FedAvg-SL-Lower   | [1]    | 5/0           |               | 53.64 | 88.95 | 56.15 | 53.86 | 52.05 | 87.91 | 53.15 | 51.21 |
| Semi supervised   |        |               |               |     |     |           |    |     |     |           |    |
| FedAvg-FixMatch   | [1], [25] | 5/45       |               | 73.26 | 96.52 | 74.58 | 71.47 | 64.33 | 93.8  | 64.34 | 61.23 |
| Fedprox-FixMatch  | [2], [25] | 5/45       |               | 73.24 | 96.26 | 74.21 | 71.92 | 63.88 | 93.79 | 65.78 | 61.38 |
| FedConsist        | [6]    | 5/45          |               | 66.72 | 93.89 | 67.72 | 66.34 | 60.3  | 92.3  | 62.85 | 57.26 |
| RSCFed            | [8]    | 5/45          |               | 53.94 | 88.71 | 55.05 | 53.36 | 48.85 | 86.95 | 50.89 | 48.48 |
| DCCFSSL (ours)    | 5/45   |               |               | 86.62 | 98.7 | 86.53 | 86.48 | 83.17 | 98.02 | 83.08 | 82.86 |

CIFAR-100

| Fully supervised  |        |               |               |     |     |           |    |     |     |           |    |
| FedAvg-SL-Upper   | [1]    | 50/0          |               | 42.10 | 94.53 | 43.29 | 42.16 | 41.13 | 94.29 | 41.08 | 40.00 |
| FedAvg-SL-Lower   | [1]    | 5/0           |               | 15.89 | 81.71 | 17.09 | 15.88 | 15.73 | 81.79 | 15.54 | 14.36 |
| Semi supervised   |        |               |               |     |     |           |    |     |     |           |    |
| FedAvg-FixMatch   | [1], [25] | 5/45       |               | 27.23 | 91.26 | 29.04 | 24.92 | 24.37 | 90.75 | 25.98 | 20.89 |
| Fedprox-FixMatch  | [2], [25] | 5/45       |               | 27.33 | 91.17 | 29.18 | 24.99 | 24.3 | 90.71 | 25.93 | 20.87 |
| FedConsist        | [6]    | 5/45          |               | 12.44 | 80.81 | 21.98 | 12.69 | 13.2 | 81.88 | 21.5 | 10.66 |
| RSCFed            | [8]    | 5/45          |               | 17.05 | 82.96 | 17.51 | 15.77 | 15.65 | 82.3 | 16.68 | 13.34 |
| DCCFSSL (ours)    | 5/45   |               |               | 49.35 | 96.18 | 50.22 | 48.79 | 45.31 | 95.86 | 47.08 | 44.22 |

STL10

| Fully supervised  |        |               |               |     |     |           |    |     |     |           |    |
| FedAvg-SL-Upper   | [1]    | 50/0          |               | 65.58 | 93.23 | 66.12 | 65.65 | 62.96 | 92.63 | 63.07 | 63.17 |
| FedAvg-SL-Lower   | [1]    | 5/0           |               | 47.81 | 86.81 | 48.15 | 47.40 | 41.35 | 83.38 | 44.92 | 37.85 |
| Semi supervised   |        |               |               |     |     |           |    |     |     |           |    |
| FedAvg-FixMatch   | [1], [25] | 5/45       |               | 60.85 | 92.3 | 61.34 | 60.83 | 48.88 | 87.91 | 51.81 | 47.13 |
| Fedprox-FixMatch  | [2], [25] | 5/45       |               | 60.73 | 92.43 | 62.34 | 60.38 | 48.27 | 87.58 | 51.73 | 45.88 |
| FedConsist        | [6]    | 5/45          |               | 51.96 | 87.78 | 53.08 | 51.04 | 43.57 | 84.37 | 46.23 | 40.07 |
| RSCFed            | [8]    | 5/45          |               | 46.27 | 86.26 | 47.09 | 45.88 | 37.96 | 82.10 | 46.28 | 35.17 |
| DCCFSSL (ours)    | 5/45   |               |               | 75.46 | 96.31 | 75.36 | 75.33 | 65.88 | 93.58 | 66.07 | 65.86 |

Note that the best and second-best results are marked in bold and underlined, respectively.
• FSSL methods exhibit lower performance improvements in the NonIID setting compared to the IID setting, suggesting that the scarcity of labeled clients exacerbates the impact of data statistical heterogeneity in FSSL. In addition to our proposed DCCFSSL achieving the best performance, the second best performance is consistently achieved by FedAvg-SL-Upper. This underlines the significance of data labels in influencing model performance, while also demonstrating that our proposed approach can effectively leverage unlabeled datasets in the absence of data labels.

C. Ablation Studies

To gain further insights into DCCFSSL, we conduct ablation studies to assess the effectiveness of its various components. Note that the experimental settings are identical except for the specific variables of interest in each group of experiments. We construct four variants of DCCFSSL as follows:

1) DCCFSSL-loc: This variant removes the local class-aware component from the dual class-aware contrastive module, abandoning the local class-aware contrastive loss in both labeled and unlabeled clients.
2) DCCFSSL-gcc: This variant removes the global class-aware component from the dual class-aware contrastive module, abandoning the global class-aware contrastive loss in both labeled and unlabeled clients.
3) DCCFSSL-DCC: This variant removes the entire dual class-aware contrastive module, abandoning both local and global class-aware contrastive losses in labeled and unlabeled clients.
4) DCCFSSL-ARA: This variant removes the authentication-reweighted aggregation based on authentication samples, abandoning authentication samples and authentication-reweighted aggregation for labeled and unlabeled clients.

As illustrated in Table II, we compare the performance of DCCFSSL and its four variants in the IID and NonIID settings across CIFAR-10, CIFAR-100, and STL-10. It is evident that the four variants, DCCFSSL-loc, DCCFSSL-gcc, DCCFSSL-DCC, and DCCFSSL-ARA, exhibit varying degrees of performance degradation compared to DCCFSSL on all three datasets, thereby demonstrating the effectiveness of these components in DCCFSSL. Specifically, we observe that removing the entire dual class-aware contrastive module or its individual parts results in diminished FSSL performance. This suggests not only that the global class-aware component is vital for capturing interactions, but also that the local class-aware component still contributes to DCCFSSL for a better prediction. Moreover, eliminating the entire dual class-aware contrastive module leads to a substantial performance decline, as evidenced by a 4.66% and 4.03% reduction in accuracy for DCCFSSL-DCC on CIFAR-10 under the IID and NonIID settings, respectively. This finding underscores the importance of collaboration between the global and local class-aware components for FSSL performance, fostering representations from distinct isolated perspectives to bolster performance. Furthermore, we observe that, even with the presence of a dual class-aware contrastive module, the model’s performance significantly decreases without the authentication-reweighted aggregation based on authentication samples. For instance, DCCFSSL-ARA experiences a 13.54% and 16.21% reduction in accuracy compared to DCCFSSL on CIFAR-100 under IID and NonIID settings, respectively.

These ablation studies not only highlight the exceptional effectiveness of the dual class-aware contrastive module for FSSL, but also emphasize the individual contributions of the global and local class-aware components. Additionally, the results underscore the effectiveness of authentication-reweighted aggregation based on authentication samples.

D. Stability of Model Training

We further analyze the dual class-aware contrastive module by examining its impact on the stability of the model training process, as illustrated in Fig. 3. Specifically, we calculate the standard deviation (STD) of the global model’s test accuracy for the final 250 global training rounds. As can be observed from Fig. 3, in the IID setting, the local class-aware and the global class-aware components reduce the global model accuracy fluctuation by 17.08% (∆ = 6.85−5.68) and 22.04% (∆ = 6.85−5.34), respectively. Differently, under the NonIID setting, the global class-aware component increases the fluctuation by 66.72% (∆ = 11.93−19.89), while the local class-aware component reduces it by 52.89% (∆ = 11.93−5.62). These findings, together with the ablation studies, indicate that the local class-aware component can reduce the fluctuation of the global model while providing performance improvement to a certain extent. In comparison, the global class-aware component also enhances model performance but increases model volatility in the NonIID setting. In the IID setting, the global class-aware

| Components | CIFAR-10 | CIFAR-100 | STL-10 |
|------------|----------|-----------|--------|
|           | IID | NonIID | IID | NonIID | IID | NonIID |
| lec | gcc | ARA | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 |
| DCCFSSL | ✓ | ✓ | 86.62 | 85.48 | 83.17 | 82.86 | 50.23 | 49.79 | 45.80 | 45.00 | 75.46 | 75.33 | 65.88 | 65.86 |
| DCCFSSL-loc | ✓ | ✓ | 86.38 | 86.38 | 82.28 | 81.91 | 45.12 | 44.38 | 42.33 | 41.21 | 72.38 | 72.49 | 39.42 | 38.02 |
| DCCFSSL-gcc | ✓ | ✓ | 84.68 | 84.49 | 80.57 | 79.80 | 45.49 | 45.29 | 43.52 | 42.53 | 74.50 | 74.39 | 65.35 | 64.55 |
| DCCFSSL-DCC | ✓ | ✓ | 81.96 | 81.83 | 79.14 | 78.39 | 37.61 | 36.57 | 36.20 | 34.15 | 69.23 | 68.99 | 59.73 | 58.79 |
| DCCFSSL-ARA | ✓ | ✓ | 78.56 | 78.00 | 68.95 | 66.69 | 36.69 | 35.58 | 33.70 | 31.34 | 71.77 | 71.37 | 62.46 | 59.07 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
component reduces model volatility due to the homogeneous data distribution among clients. Furthermore, Fig. 3 demonstrates that the dual class-aware contrastive module significantly improves model performance while reducing model volatility in both IID and NonIID settings, effectively combining the dual strengths of the local class-aware and global class-aware components at the same time. Although DCCFSSL significantly mitigates model volatility, it may not entirely eradicate fluctuations, particularly in the IID scenario where the model inherently exhibits lower volatility. Several factors can still contribute to a certain degree of model volatility, including the presence of noise in sample data, inconsistencies in feature distribution within the same data category, and partial client participation in each communication round. Noise in the sample data can introduce random variations that affect model updates, while inconsistent feature distributions within a data category can lead to divergent model updates across clients. Furthermore, when only a subset of clients participate in each round, the model may not capture the full spectrum of data, potentially resulting in some volatility. Despite these challenges, DCCFSSL remains an effective approach to reducing model volatility in FL settings.

E. Impact of Hyperparameter

To examine the influence of hyperparameters on the performance of our proposed method, we conduct experiments focusing on key hyperparameters $\lambda_{lcc}$ and $\lambda_{gcc}$, as demonstrated in Table III.

From the experimental results with the hyperparameter constraint $\lambda_{lcc}=\lambda_{gcc}$, we observe that maintaining an appropriate ratio between the loss coefficients of the dual class-aware contrastive module and the basic training module is crucial. When this ratio is too small, the dual class-aware contrastive module cannot effectively contribute to performance improvement, limiting its impact. Increasing the ratio moderately can enhance the performance of FSSL. However, when the ratio is too large, it negatively impacts the data representation capabilities of the basic training module, resulting in performance degradation instead of further improvements.

Our experiments with fixed hyperparameters $\lambda_{lcc}=1$ and $\lambda_{gcc}=1$ indicate that an overly large or small ratio of $\lambda_{lcc}/\lambda_{gcc}$ can lead to performance deterioration. It is essential to maintain an appropriate balance between $\lambda_{lcc}$ and $\lambda_{gcc}$ to ensure effective collaboration of local and global class-aware components, yielding better performance improvements. Under the IID setting, the optimality is achieved with $\lambda_{lcc}=1$ and $\lambda_{gcc}=10$, yielding an optimal accuracy of 88.81% and an optimal F1 value of 88.74%. Under the NonIID setting, the optimality is obtained with both $\lambda_{lcc}$ and $\lambda_{gcc}$ set to 1, resulting in an optimal accuracy of 83.17% and an optimal F1 value of 82.86%. In the IID setting, where clients share a homogeneous data distribution, enhancing the global class-aware component while keeping the local class-aware component constant improves global collaboration among clients, thus boosting performance. In the NonIID setting, characterized by heterogeneous data distributions across...
TABLE IV
PARAMETER QUANTITY OF PROTOTYPE AND MODEL IN THIS WORK

| Dataset | Prototype | Model | Ratio  |
|---------|-----------|-------|--------|
| CIFAR-10 | 1280      | 692810 | 0.001848 |
| CIFAR-100 | 12800     | 704420 | 0.01817 |
| STL-10  | 1280      | 692810 | 0.001848 |

clients, excessively increasing the global class-aware component can exaggerate the mismatch between a client’s local data distribution and the global data distribution. This can negatively impact local training, leading to a decline in global model performance.

When implementing the dual class-aware contrastive module in other works, it is advisable to set both $\lambda_{ucc}$ and $\lambda_{gcc}$ to 1 as a simple and effective starting point for further hyperparameter adjustments.

F. Communication Cost and Computational Cost

In the proposed DCCFSSL method, we upload not only model parameters but also prototype data. We evaluate the added communication cost resulting from the uploading of prototype data. As demonstrated in Table IV, the incremental communication cost for the CIFAR-10 and STL-10 datasets amounts to a mere 0.1848% compared to the model’s communication cost, while for the CIFAR-100 dataset, the increase is only 1.817%. This modest rise in communication cost, ranging from 0.1848% to 1.817%, is deemed acceptable, particularly in cross-silo scenarios. Moreover, it is important to note that the additional communication cost is solely determined by the dimension of the prototype representation and does not increase proportionally with the growth of the model parameters.

In addition to evaluating communication cost, we have also evaluated computational cost. FSSL baselines, such as FedAvg-FixMatch, Fedprox-FixMatch, FedConsist, and RSCFed, typically use two augmented samples from a single sample and feed all of them to the encoder for computation. In contrast, our method uses three augmented samples from a single sample and feeds all of them to the encoder for computation. Consequently, based on the quantity of samples fed to the encoder for computation, our computational cost is 1.5 times that of these methods, representing a 50% increase. However, our method significantly boosts performance, outperforming the highest accuracy of these four methods by 9.31%-22.02% in the IID setting and by 17%-20.94% in the NonIID setting, across three datasets. Considering that FL participants in cross-silo scenarios are often large institutions with sufficient computational capacity, the additional computational cost incurred by our method is acceptable given the substantial performance gains. We also observe that FedAvg-FixMatch, Fedprox-FixMatch, FedConsist, and RSCFed do not provide a theoretical analysis of convergence speed. To our best knowledge, our work is the first to present a theoretical analysis of convergence speed in FSSL. Our proposed method achieves a convergence speed of $O(\frac{1}{\sqrt{T}})$, as demonstrated in the theoretical analysis section.

G. Effect of Labeled Client Ratio

In practical implementation, the ratio of labeled clients to unlabeled clients can vary. Hence, we investigate the impact of different ratios on DCCFSSL performance using the CIFAR-10 dataset. Specifically, we consider seven cases where the ratios of labeled clients to unlabeled clients are 1:49, 5:45, 10:40, 25:25, 40:10, 45:5, and 50:0, respectively. The results of seven cases are as presented in Table V. We observe that DCCFSSL performance improves as the proportion of labeled clients increases. It should be observed that relying exclusively on labeled clients does not produce the most favorable performance. This phenomenon arises because, when all clients are labeled, a slight decrease in their quantity has a minimal effect on the ultimate accuracy, given the ample availability of labeled data. In contrast, the integration of unlabeled clients contributes consistency information within the context of a high-dimensional space, which leads to a more substantial improvement in accuracy than the reduction associated with diminishing the number of labeled clients. Notably, DCCFSSL achieves 75.76% and 54.84% accuracy in the IID and NonIID settings, respectively, even with just one labeled client. This finding suggests that DCCFSSL can perform well even with an extremely limited number of labeled clients.

Moreover, we further compare our method with FedAvg-SL-Lower and FedAvg-SL-Upper, as illustrated in Fig. 4. For the IID setting, even with only one labeled client, our method achieves 34.81%, and 0.78% improvements over FedAvg-SL-Lower and FedAvg-SL-Upper. For the NonIID setting, our method achieves a 19.26% improvement over FedAvg-SL-Lower but does not suppress FedAvg-SL-Upper. This result stems from the severely uneven data distribution of a single client, where some classes may have little or no data. However, as the proportion of labeled clients increases, our method’s performance significantly improves, surpassing that of FedAvg-SL-Upper in the case of 5:45.

H. Impact of Dirichlet Distribution Coefficient $\gamma$

To further investigate the impact of Dirichlet distribution coefficient $\gamma$, we conducted experiments across three datasets—CIFAR10, CIFAR100, and STL-10—under different settings of...
coefficient $\gamma$. The results are presented in Table VI. It was observed that the accuracy of the experiments exhibited significant fluctuations when different Dirichlet distribution coefficients were applied to the same dataset. For instance, CIFAR10 showed a maximum fluctuation of 15.20%, CIFAR100 had up to 11.02%, and STL-10 experienced a maximum of 31.27%. Notably, the lowest accuracy was achieved when the Dirichlet distribution coefficients were at their minimum. Specifically, the lowest accuracies recorded were 71.42% for CIFAR10, 38.33% for CIFAR100, and 44.19% for STL-10. Notably, this decrease in model performance can be attributed to the increased heterogeneity among the datasets at lower coefficients. As the Dirichlet distribution coefficients increased, indicating a reduction in dataset heterogeneity, there was a gradual improvement in accuracy. When the coefficients were sufficiently large, surpassing 100, the heterogeneity of the datasets approached homogeneity, and accuracy stabilized. This resulted in the highest performance, with CIFAR10 reaching an accuracy of 86.62%, CIFAR100 at 49.35%, and STL-10 at 75.46%.

VI. CONCLUSION

In this paper, we propose DCCFSSL, a novel federated semi-supervised learning method that leverages unlabeled datasets to further improve FL performance. On the one hand, DCCFSSL builds a common training goal to reduce the large deviation from uploaded local models of labeled clients and unlabeled clients. On the other hand, it introduces dual class-aware contrastive information in the feature space to alleviate confirmation bias. Meanwhile, the proposed dual class-aware contrastive module of DCCFSSL addresses the mismatch between global and local class prototypes by considering both local and global class-aware distributions within the feature space. In addition, DCCFSSL presents authentication-reweighted aggregation based on authentication samples to improve the robustness of the global model and class prototypes. Extensive experiments demonstrate that DCCFSSL significantly outperforms other state-of-the-art FSSL methods. Meanwhile, DCCFSSL substantially improves the model stability of the training process. Remarkably, even when compared to the standard federated supervised learning with labeled data for all clients, DCCFSSL still achieves 2.92%~11.64% accuracy improvement on CIFAR-10, CIFAR-100, and STL-10 datasets.

ACKNOWLEDGMENT

The authors sincerely thank all anonymous reviewers and editor for their valuable and constructive comments.

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 Proc. Int. Conf. Artif. Intell. Statist., 2017, pp. 1273–1282.
[2] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” in Proc. Mach. Learn. Syst., vol. 2, pp. 429–450, 2020.
[3] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, “Scaffold: Stochastic controlled averaging for federated learning,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 5132–5143.
[4] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, “Tackling the objective inconsistency problem in heterogeneous federated optimization,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 7611–7623.
[5] Q. Li, B. He, and D. Song, “Model-contrastive federated learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog., 2021, pp. 10713–10722.
[6] D. Yang et al., “Federated semi-supervised learning for COVID region segmentation in chest CT using multi-national data from China, Italy, Japan,” Med. Image Anal., vol. 70, 2021, Art. no. 101992.
[7] Q. Liu, H. Yang, Q. Dou, and P.-A. Heng, “Federated semi-supervised medical image classification via inter-client relation matching,” in Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Intervention, 2021, pp. 325–335.
[8] X. Liang, Y. Lin, H. Fu, L. Zhu, and X. Li, “RSCFed: Random sampling consensus federated semi-supervised learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 1015–10163.

[9] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, “Federated semi-supervised learning with inter-client consistency & disjoint learning,” 2020, arXiv: 2009.12957.

[10] E. Araozo, D. Ortego, P. Albert, N. E. O’Connor, and K. McGuinness, “Pseudo-labeling and confirmation bias in deep semi-supervised learning,” in Proc. 2020 Int. Joint Conf. Neural Netw., 2020, pp. 1–8.

[11] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 4080–4090.

[12] F. Yang et al., “Class-aware contrastive semi-supervised learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 14421–14430.

[13] X. Mu et al., “FedProc: Prototypical contrastive federated learning on non-IID data,” 2021, arXiv:2109.12273.

[14] W. Zhang et al., “Optimizing federated learning in distributed industrial IoT: A multi-agent approach,” IEEE Syst. J., vol. 16, no. 1, pp. 6368–3703, Dec. 2021.

[15] D. Yang et al., “DetFed: Dynamic resource scheduling for deterministic federated learning over time-sensitive networks,” IEEE Trans. Mobile Computing, vol. 23, no. 5, pp. 5162–5178, May 2024.

[16] X. Li, J. Huang, Y. Wang, S. Wang, and Z. Zhang, “On the convergence of FedAvg on non-IID data,” 2019, arXiv:1907.02189.

[17] Y. Zhao, M. Li, L. Lai, N. Suda, D. Cavin, and V. Chandra, “Federated learning with non-IID data,” 2018, arXiv:1806.00582.

[18] Y. Guo, Y. Sun, R. Hu, and Y. Gong, “Hybrid local SGD for federated learning with heterogeneous communications,” in Proc. Int. Conf. Learn. Representations, 2021. [Online]. Available: https://openreview.net/forum?id=H0oaWl6THa

[19] W. Luping, W. Wei, and L. Bo, “CMFL: Mitigating communication overhead for federated learning,” in Proc. 2019 IEEE 39th Int. Conf. Distrib. Syst. Comput., 2019, pp. 954–964.

[20] T. Li, S. Hu, A. Beirami, and V. Smith, “Ditto: Fair and robust federated learning through personalization,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 6357–6368.

[21] L. Lu et al., “Privacy and robustness in federated learning: Attacks and defenses,” 2020, arXiv:2012.06337.

[22] H. Wang, M. Urochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, “Federated learning with matched averaging,” 2020, arXiv:2002.06440.

[23] H.-Y. Chen and W.-L. Chao, “FedBE: Making Bayesian model ensemble applicable to federated learning,” 2020, arXiv:2009.01974.

[24] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, “MixMatch: A holistic approach to semi-supervised learning,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 5050–5060.

[25] K. Sohn et al., “FixMatch: Simplifying semi-supervised learning with consistency and confidence,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 596–608.

[26] J. Jeong, S. Lee, J. Kim, and N. Kwak, “Consistency-based semi-supervised learning for object detection,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 10758–10767.

[27] B. Zhang et al., “FlexMatch: Boosting semi-supervised learning with curvature pseudo-labeling,” in Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 18408–18419.

[28] K. Simonoyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 568–576.

[29] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, “Towards universal paraphrastic sentence embeddings,” 2015, arXiv:1511.08198.

[30] Y. Tan et al., “FedProto: Federated prototype learning across heterogeneous clients,” in Proc. AAAI Conf. Artif. Intell., 2022, pp. 8432–8440.

[31] W. Huang, M. Ye, Z. Shi, H. Li, and B. Du, “Rethinking federated learning with domain shift: A prototype view,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2023, pp. 16312–16322.

[32] Y. Dai, Z. Chen, J. Li, S. Heinzecke, L. Sun, and R. Xu, “Tackling data heterogeneity in federated learning with class prototypes,” in Proc. AAAI Conf. Artif. Intell., 2023, pp. 7314–7322.

[33] D. Cheng, L. Zhang, C. Bu, X. Wang, H. Wu, and A. Song, “ProtoHAR: Prototype guided personalized federated learning for human activity recognition,” IEEE J. Biomed. Health Inform., vol. 27, no. 8, pp. 3900–3911, Aug. 2023.

[34] C. B. Mawuli et al., “FedStream: Prototype-based federated learning on distributed concept-drifting data streams,” IEEE Trans. Syst., Man, Cybern.: Syst., vol. 53, no. 11, pp. 7112–7124, Nov. 2023.
Yong Qi received the PhD degree from Xi’an Jiaotong University, China. He is a professor with the Department of Computer Science and Technology, Xi’an Jiaotong University and the director of the institute of computer software and theory. He is the deputy director with the system software committee and the deputy director of the information system committee of the China Computer Federation. He is a member of the expert review team of the National Natural Science Foundation of China. He has published many papers in important international journals IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Reliability, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Services Computing, IEEE Transactions on Smart Grid, IEEE Transactions on Computers, IEEE Transactions on Mobile Computing, Journal of Parallel and Distributed Computing, and international important conferences ACM VEE, USENIX ATC, IEEE ICDCC, INFOCOM, PerCom, ICNP, and ICPP. His research interests include distributed systems, cloud computing, mobile computing, and federated learning.

Saiyu Qi received the BS degree in computer science and technology from Xi’an Jiaotong University, Xi’an, China, in 2008, and the PhD degree in computer science and engineering from Hong Kong University of Science and Technology, Hong Kong, in 2014. He is currently an associate professor with the School of Computer Science and Technology, Xi’an Jiaotong University. He has published papers in IEEE Transactions on Mobile Computing, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, IJCAI, CCS, and so on. His research interests include applied cryptography, cloud security, and distributed systems.