Beyond Parameters: Locally-Guided Knowledge Distillation for Decentralized Federated Learning

Behnaz Soltani*, Yipeng Zhou*†, Saqr Thabet*, Elaf Alhazmi*, Lina Yao‡, and Quan Z. Sheng*
*School of Computing, Macquarie University, NSW, Australia
‡CSIRO's Data61, NSW, Australia
{behnaz.soltani, saqr.thabet, elaf.alhazmi}@hdr.mq.edu.au, {yipeng.zhou, michael.sheng}@mq.edu.au,
lina.yao@data61.csiro.au

2025 IEEE International Conference on Data Mining (ICDM) | 979-8-3315-9599-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICDM65498.2025.00080

***Abstract*—Federated Learning (FL) has revolutionized privacy-preserving machine learning by enabling collaborative model training across multiple clients without sharing raw data. However, server-based FL suffers from scalability issues, communication bottlenecks, and the risk of a single point of failure. As an alternative, Decentralized Federated Learning (DFL) eliminates the need for a central server through a peer-to-peer setup. Yet, it faces significant challenges due to non-IID (non-independent and identically distributed) data across clients, leading to unstable convergence and poor generalization. A fundamental dilemma in DFL arises from this non-IID nature: aggregating model parameters to promote global consistency can erode client-specific knowledge, while conducting local training exacerbates model drift, misaligning clients from shared objectives. To address this, we propose BlendDFL, a novel DFL framework that enhances generalization performance by integrating locally-guided knowledge distillation into local training. Specifically, rather than merely minimizing a local loss function, each client minimizes a synthetic loss, which combines the knowledge distillation loss (preserving useful information from neighboring models) and the local model loss. This approach mitigates the overwriting of peer knowledge during local updates. Besides, considering class imbalance and data heterogeneity due to non-IID data distributions, BlendDFL introduces adaptive per-sample weighting, emphasizing underrepresented classes in both types of loss computation. Importantly, BlendDFL incurs no additional communication overhead and does not require access to public data. Extensive experiments on benchmark datasets demonstrate that BlendDFL consistently outperforms the state-of-the-art DFL baselines, achieving faster convergence and better generalization across diverse non-IID settings. The source code of our work is available at https://github.com/behnazsoltani/BlendDFL.**

***Index Terms*—Decentralized Federated Learning, Knowledge Distillation, Generalization.**

I. Introduction

Due to increasing privacy concerns and regulatory restrictions, collecting raw data from distributed sources, such as smartphones and IoT devices, for centralized machine learning is becoming increasingly impractical. Federated Learning (FL) and Decentralized Federated Learning (DFL) have emerged as privacy-preserving paradigms that enable collaborative model training without sharing raw data [1]. In FL, a central server coordinates the training process by selecting clients, aggregating their locally trained model updates, and distributing a global model [2], [3]. However, this centralized architecture poses several limitations, including a single point of failure, scalability issues, communication bottlenecks, and susceptibility to server-side attacks [4], [5]. DFL eliminates the need of a central server by enabling direct peer-to-peer communication among clients [6]–[10]. Clients collaboratively train a shared model by exchanging information with neighboring peers, typically following either a fixed communication topology (e.g., a ring) or a time-varying one. This decentralized structure enhances robustness, scalability, and privacy, making DFL an attractive alternative for distributed learning in privacy-sensitive or large-scale environments.

A key and persistent challenge in FL, and even more pronounced in DFL, is data heterogeneity across clients. Unlike traditional machine learning, which typically assumes that data are independently and identically distributed (IID), FL operates under non-IID conditions where clients possess skewed, unbalanced, or incomplete representations of the global data distribution. This statistical heterogeneity significantly complicates collaborative model training. In DFL, where model aggregation occurs only through communication among immediate neighbors rather than a central coordinator, the adverse effects of non-IID data are further magnified. Moreover, model aggregation alone is often insufficient to preserve valuable local knowledge, especially under non-IID settings, leading to difficulties in achieving a coherent and generalizable global model. This often results in degraded performance on unseen or globally representative data [11], [12]. Additionally, local training can intensify overfitting to local distributions, causing model drift and catastrophic forgetting of previously learned global patterns [13], [14]. These challenges highlight the critical need for novel strategies that effectively balance local heterogeneity with global generalization, particularly within the context of DFL.

To address the challenge of balancing local heterogeneity and global generalization in DFL, we propose a novel application of knowledge distillation (KD). Traditionally, KD involves training a compact student model to mimic the output distribution of a more informative teacher model [15], thereby transferring critical knowledge. In DFL, we adapt this paradigm such that each client distills knowledge from its neighboring peers using only its own local data. This approach preserves data privacy while enabling clients to align their predictions with those of their peers, improving consistency and model generalization across the network while conducting

†The corresponding author.

2374-8486/25/$31.00 ©2025 IEEE
DOI 10.1109/ICDM65498.2025.00080

local training on clients.

Specifically, we propose BlendDFL (**B**eyond Parameters: **l**ocally-Guid**e**d K**n**owledge **d**istillation for **D**ecentralized **F**ederated **L**earning), a novel decentralized DFL framework designed to improve model generalization under data heterogeneity. Unlike most existing DFL methods that focus solely on minimizing local losses, BlendDFL introduces a synthetic loss function that integrates both KD loss and local loss during local training. The KD loss allows each client to transfer and preserve knowledge from neighboring models by computing soft targets based on their outputs evaluated on the client's own local data. In parallel, the local loss ensures that the model continues to learn client-specific knowledge from private data. To address class imbalance common in non-IID settings, BlendDFL incorporates adaptive per-sample weighting, where each sample's importance is progressively adjusted based on its inverse class frequency. This mechanism gradually emphasizes rare or underrepresented classes, increasing their influence in both local training and KD. Although KD is performed solely on local datasets, this adaptive weighting ensures that minority classes contribute more meaningfully as training progresses. Importantly, unlike prior distillation-based FL approaches that require publicly available proxy datasets or shared auxiliary data [16], [17], BlendDFL performs the entire distillation process on private local data, eliminating the need for external resources. This design enhances privacy preservation, reduces communication overhead, and makes BlendDFL a practical and robust solution for decentralized learning in non-IID environments.

The main contributions of this work are summarized as follows:

- We propose BlendDFL, a novel DFL framework that incorporates adaptive, locally-guided knowledge distillation to enhance model generalization in non-IID settings.
- We develop an efficient distillation mechanism that operates solely on clients' private local data, eliminating the need for public proxy datasets and introducing no additional communication overhead.
- We introduce an adaptive per-sample weighting strategy that progressively emphasizes underrepresented classes, improving resilience to class imbalance caused by statistical heterogeneity.
- We conduct comprehensive experiments on standard benchmark datasets, showing that BlendDFL consistently outperforms the state-of-the-art DFL methods in generalization performance, convergence speed, and robustness across varying levels of data heterogeneity.

II. Related Work

A. Decentralized Federated Learning

DFL eliminates the need of a central server in FL by allowing clients to communicate and collaborate directly during the training process [7], [10], [18]–[20]. This architecture enhances system scalability and resilience to single-point failures. However, the lack of a centralized coordinator also brings challenges related to model consistency, convergence stability, and generalization, primarily due to the absence of a global model that aligns the knowledge of distributed clients [21], [22].

The work in [7] introduces the Decentralized Parallel Stochastic Gradient Descent (D-PSGD) algorithm, demonstrating that decentralized optimization can match or even surpass centralized approaches in terms of convergence speed and scalability under communication constraints. This work has laid the foundation for numerous subsequent studies exploring decentralized learning frameworks.

Several research efforts aim to enhance convergence and robustness in decentralized learning, especially under non-IID data distributions common in federated settings. For instance, DFedAvgM [8] extends the classic FedAvg algorithm by incorporating momentum-based updates in decentralized settings, thereby improving training stability and convergence. QG-DSGDm [23] is a momentum-based decentralized optimization method that enhances convergence speed and stability by leveraging a decentralized approximation of global momentum. Dis-PFL [24] introduces a pruning-based decentralized sparse training framework that adapts personalized sparse masks over time, enhancing personalization while significantly reducing communication and computational costs. Similarly, TOPFL [25] improves training efficiency in DFL by adopting partial model aggregation and dynamic topology construction. Instead of sharing entire models, only a subset of model parameters is exchanged among neighbors, striking a balance between communication overhead and model performance. DFedPGP [20] improves both communication efficiency and model personalization by enabling clients to exchange only shared gradients of a common feature extractor while retaining private, personalized classifiers.

B. Knowledge Distillation in FL

Knowledge distillation (KD) is originally introduced to improve the performance of smaller models by transferring knowledge from larger, more expressive models [26]. This is typically achieved by having a student model that mimics the soft output distributions (logits) or internal representations of a teacher model [15], [27]–[29]. In the context of federated learning, KD has been extended to enable knowledge transfer across heterogeneous clients.

FedMD [16] tackles model heterogeneity by aligning client models through a shared public dataset, using KD to synchronize predictions. Similarly, FedDF [17] aggregates client predictions on unlabeled public data to train a global classifier on the server. However, both methods rely on a centralized server and assume access to a public dataset, which restricts their applicability in fully decentralized or privacy-sensitive settings. To address knowledge retention and personalization, CFeD [30] proposes a dual-level distillation framework that mitigates both intra-task and inter-task forgetting using surrogate data shared between clients and a central server. PHP-FL [31] integrates privacy-preserving clustering and distillation by allowing clients to share soft predictions instead of raw data. It

employs ring-based aggregation and cross-cluster distillation to improve clients' performance while preserving privacy.

Focusing on non-IID challenges, FedNTD [32] proposes "not-true distillation" by aligning local client predictions with the global model's logits on unseen (non-local) classes, helping preserve global knowledge and reduce forgetting. In decentralized settings, MHD [33] introduces Multi-Headed Distillation, where each agent uses multiple auxiliary heads to capture diverse views from peer models, facilitating robust decentralized learning. Despite their effectiveness, most prior works rely on public datasets or central coordination, making them unsuitable for fully decentralized federated learning with strict privacy constraints. In contrast, our approach enables local KD using only private data, without requiring shared data or centralized orchestration.

To summarize, prior research in DFL has primarily focused on improving communication efficiency and convergence through optimized parameter exchange, momentum strategies, and dynamic topology adaptation. Meanwhile, KD techniques in FL often depend on centralized coordination, shared public datasets, or global model availability that limit their applicability in privacy-sensitive and fully decentralized settings. Despite recent efforts, a practical, lightweight, and fully decentralized distillation mechanism remains underexplored. Our work addresses this gap by introducing a novel peer-to-peer (P2P) KD framework that operates solely on private local data and exchanged logits without requiring a central server or shared data.

III. Methodology

In this section, we first introduce the problem setting, then present the motivation, and finally describe the proposed BlendDFL framework in detail.

A. Problem Setting

We consider a DFL setup comprising a set of N clients, denoted by $\mathcal{C} = \{1, 2, \ldots, N\}$. Each client $i \in \mathcal{C}$ holds a private local dataset $\mathcal{D}_i$, which is never shared due to privacy constraints. The overarching goal is to collaboratively train a high-quality global model by leveraging the collective knowledge across clients without central coordination or direct data sharing.

Unlike traditional federated learning, which relies on a centralized server to aggregate client updates, DFL assumes a serverless topology. Clients are interconnected via a peer-to-peer communication graph $\mathcal{G} = (\mathcal{C}, \mathcal{E})$, where an edge $(i, j) \in \mathcal{E}$ indicates that clients i and j can exchange information [7], [8], [21], [34]. Each client maintains its own local model θ_i and can only communicate with its immediate neighbors, denoted by $\mathcal{N}_i = \{j \mid (i, j) \in \mathcal{E}\}$. Each client optimizes a local empirical objective:

$$\min_{\theta_i} F_i(\theta_i), \quad \text{where } F_i(\theta_i) = \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\left[\ell(\theta_i; x, y)\right], \quad (1)$$

where $\ell(\theta_i; x, y)$ denotes the local loss function (e.g., cross-entropy) of model θ_i on a particular sample (x, y).

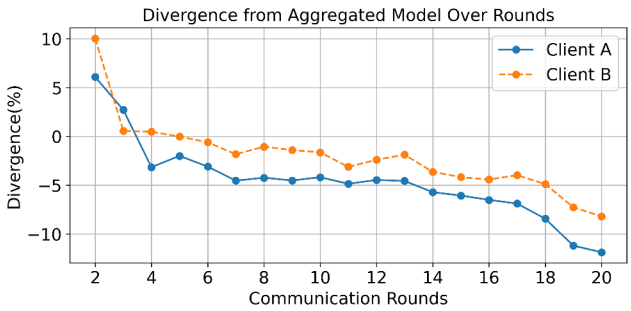


Fig. 1: Local Model Drift Over Communication Rounds.

The global learning objective is to find a consensus model θ that minimizes the average risk across all clients (i.e., generalization performance):

$$\min_{\theta} f(\theta) \coloneqq \frac{1}{N}\sum_{i=1}^{N} F_i(\theta). \quad (2)$$

Aligned with the existing frameworks [8], [21], [24], [34], DFL operates through multiple rounds of communications. During communication round t, the participating client i performs multiple local updates using stochastic gradient descent (SGD) as follows:

$$\theta_i^{t+1,k} \leftarrow \theta_i^{t,k} - \eta \nabla F_i(\theta_i^{t,k}), \quad \textit{for } k = 1, \ldots, K, \quad (3)$$

where η is the local learning rate, k denotes the local training iteration and K is the total number of local iterations per communication round. After local training, each client exchanges model parameters with its neighbors and aggregates the received neighboring models, including its own. Specifically, the aggregated model at client i is computed as:

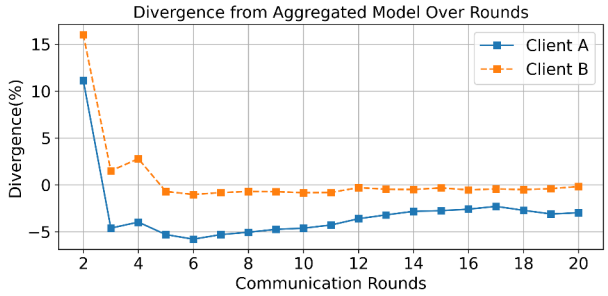
$$\bar{\theta}_i^{t+1} = \frac{1}{|\mathcal{N}_i| + 1} \sum_{j \in \mathcal{N}_i \cup \{i\}} \theta_j^{t,K}. \quad (4)$$

In the absence of a central server, DFL methods must carefully design decentralized protocols to enable effective knowledge exchange and model alignment. Our work addresses this challenge through a novel distillation-based approach, where clients leverage both their local data and soft predictions from neighbors to collaboratively improve local model generalization under non-IID data conditions.

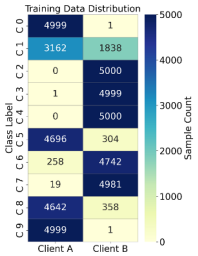
B. Motivation

In DFL, where clients collaboratively train models by exchanging model parameters directly with their peers, addressing statistical heterogeneity remains a major challenge. Statistical heterogeneity often leads to **model drift** where local models diverge from the global objective due to imbalanced and non-IID data. To quantify this effect, we define a divergence metric as the difference between a client's local model accuracy and the accuracy of the aggregated model on a global test set. We define the divergence of client i at round t as:

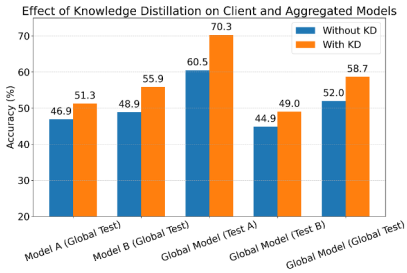
$$\text{Divergence}_i^{(t)} = \text{Acc}\left(\theta_i^{(t)}, \mathcal{D}_{\text{global}}\right) - \text{Acc}\left(\bar{\theta}^{(t-1)}, \mathcal{D}_{\text{global}}\right), \quad (5)$$



(a) Divergence between local models and the aggregated model.



(b) Training Data Distribution



(c) Accuracy of local and aggregated models

Fig. 2: The Effect of knowledge distillation in DFL

where $\theta_i^{(t)}$ denotes the local model parameters of client i at round t, and $\bar{\theta}(t-1)$ represents the global model obtained by averaging the local models from clients at communication round $t-1$. Client i employs $\bar{\theta}(t-1)$ to initialize its local training at round t. Both accuracies are evaluated on a global test dataset $\mathcal{D}_{\text{global}}$.

Intuitively speaking, a larger negative divergence implies that the local model $\theta_i^{(t)}$, after training on its local data, performs worse than the global model $\bar{\theta}(t-1)$ on $\mathcal{D}_{\text{global}}$. This indicates that the local model has become more specialized to its local data distribution, resulting in misalignment with the global model and compromised generalization performance.

Fig. 1 illustrates this drift with two clients using the CIFAR-10 data partitioned with a highly skewed Dirichlet distribution ($\alpha = 0.1$) to collaboratively train a model. At the initial communication rounds, the divergence values for both Client A and Client B are positive. This indicates that as the global model is still immature, the local models outperform the global model. As communication progresses, the divergence for both clients transitions into the negative range. This shift demonstrates that the global model becomes increasingly effective and better optimized for the overall data distribution. We observe that without mitigation, divergence not only persists but worsens over the communication rounds. This can be attributed to model drift, where local updates may overfit to non-IID local data and thus deviate from the globally beneficial direction.

To address this challenge, it is essential to retain global model information during local training. KD has been widely recognized as an effective and compact technique for knowledge transfer. Therefore, we employ KD to regularize the local model θ_i and preserve global knowledge throughout the training process. Instead of relying solely on hard labels, each client leverages soft predictions generated by its neighboring models, obtained by evaluating their models on its local data, as additional training signals. In this setup, the client augments its standard cross-entropy loss with a distillation loss that encourages alignment between its local model and the logits provided by its peers. Consequently, the local training objective becomes a combination of the original loss and the distillation loss. It should be noted that at this stage, we focus on illustrating the effectiveness of KD in preserving global knowledge. The specific strategy for combining the two loss terms will be detailed in our algorithm.

As illustrated in Fig. 2a, this approach significantly reduces the divergence between local models and the aggregated global model, fostering more stable and collaborative learning under non-IID data distributions. The training data distribution shown in Fig. 2b highlights the severe class imbalance between clients, which contributes to this divergence.

Moreover, we analyze how KD affects both the local and global performance after just five communication rounds. KD not only stabilizes the local training trajectory but also improves generalization performance. As shown in Fig. 2c, incorporating KD yields substantial improvements across both client-level and global test accuracy. For instance, Client A's accuracy increases from 46.9% to 51.3%, and the aggregated model evaluated on Client A's data improves from 60.5% to 70.3% when using KD. These results indicate that a naive model aggregation fails to preserve peer-specific information effectively, while distillation facilitates a communication-efficient mechanism for retaining global information, and hence enhancing generalization.

Overall, KD acts as an effective regularizer that mitigates local model drift and improves training stability in decentralized settings in our motivation examples. These insights motivate our broader investigation into distillation-based techniques as a principled approach for enabling robust collaboration under statistical heterogeneity and limited bandwidth.

C. BlendDFL Algorithm

We propose BlendDFL, a novel DFL framework that combines model aggregation with adaptive KD to improve generalization performance. BlendDFL comprises two main components: (i) locally-guided knowledge aggregation, and (ii) decentralized knowledge distillation. An overview of the framework is illustrated in Fig. 3, and the algorithmic workflow is provided in Algorithm 1.

1) Locally-guided Knowledge Aggregation: In decentralized federated learning, the inherent non-IID nature of client data leads to significant variation in local model performance across different classes. Clients tend to excel on frequently observed classes while underperforming on those that are rare or entirely absent. This imbalance causes naive model parameter averaging to suffer from model drift, overspecialization, and degraded generalization performance. Moreover, direct aggregation of model weights fails to capture the nuanced,

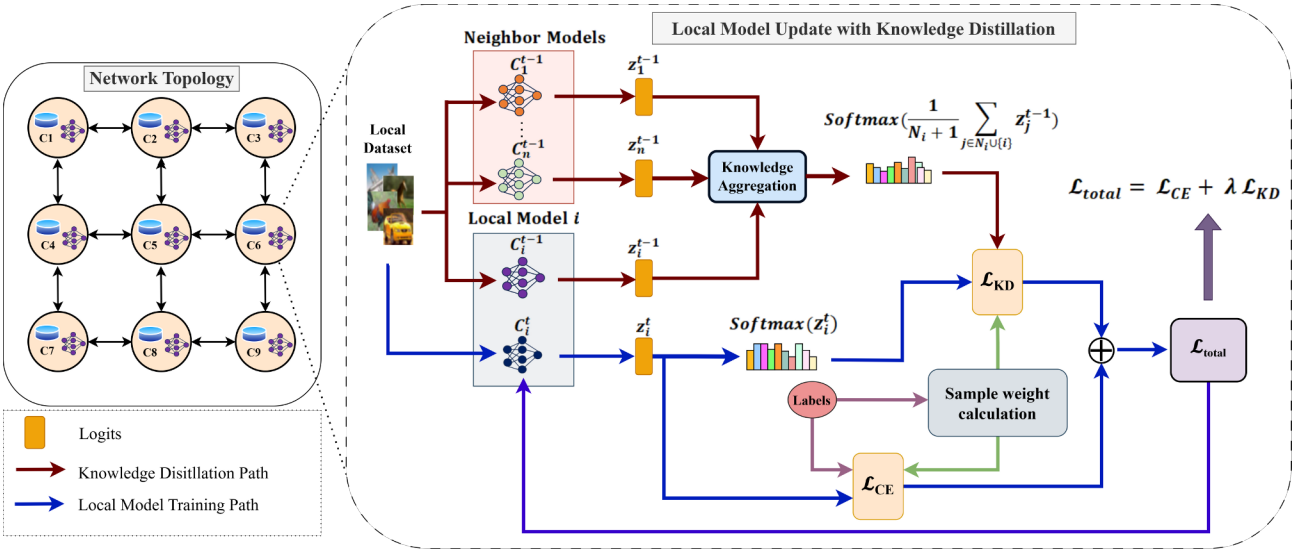


Fig. 3: Overview of the BlendDFL Framework. (1) Each client aggregates logits received from its neighbors and applies softmax to generate a soft teacher signal. (2) Local training is guided by a combination of weighted cross-entropy loss (using true labels) and KD loss (using the soft teacher signal). Sample weights are dynamically computed based on the current label distribution and communication round. The framework operates in a fully decentralized manner without any central coordination.

probabilistic relationships between classes that are essential for robust and fine-grained classification.

To address these limitations, we propose a KD-based mechanism that enables richer and more adaptive knowledge transfer across clients. Unlike centralized distillation frameworks that require access to a shared dataset, our method uses each client's own local data to distill knowledge from its neighbors. Specifically, each client collects the model outputs of its immediate neighbors using its private local data. These logits are then averaged to form a target distribution that encapsulates the collective knowledge of peers trained on diverse and potentially non-IID datasets.

We define the mini-batch $\mathcal{B}_i$ for client i as a set of labeled pairs $\{(x_s, y_s)\}$, where x_s is the input and y_s is the corresponding label for each sample x_s. Let z_j represent the logits generated by client j's model θ_j on a local sample $x_s \in \mathcal{D}_i$. The aggregated logits at communication round t for client i are computed as:

$$\tilde{z}_i^t(x_s) = \frac{1}{|\mathcal{N}_i| + 1} \sum_{j \in \mathcal{N}_i \cup \{i\}} z_j^{t-1}(x_s) \tag{6}$$

where $\mathcal{N}_i$ denotes the set of neighbors of client i. We include the client's own logits in the aggregation to ensure stability, preserve locally learned knowledge, and mitigate potential bias introduced by non-IID peer distributions. This also aligns with the spirit of collaborative learning, as in traditional FedAvg, where each client contributes equally to the global update. The aggregated logits $\tilde{z}_i^t(x_s)$ serve as targets for KD, guiding the client's local model θ_i during training.

By aligning its local predictions with the aggregated logits, each client incrementally integrates diverse semantic knowledge from its peers, while maintaining full data privacy and avoiding additional communication overhead. Importantly, this mechanism allows clients to benefit from complementary class representations even if individual neighbors provide imperfect predictions. The collective logit signal embeds cross-client and cross-class knowledge, enhancing generalization and mitigating overfitting to skewed local distributions. This adaptive and privacy-preserving knowledge exchange is particularly effective in decentralized setups where structural diversity among clients is high.

2) Knowledge Distillation in DFL: To incorporate peer knowledge in a decentralized manner, each client i aligns its local model θ_i with soft target distributions aggregated from its neighbors. This is achieved via a KD loss, which encourages the local model to mimic the ensemble of peer predictions using its own local data. Formally, for a mini-batch $\mathcal{B}_i = \{(x_s, y_s)\}$ sampled from client i's dataset $\mathcal{D}_i$, we define the KD loss as:

$$\mathcal{L}_{\text{KD}} = \frac{T^2}{\sum_s w_{y_s}(t)} \sum_s w_{y_s}(t) \cdot \mathcal{D}_{\text{KL}}^{(s)}, \tag{7}$$

$$\mathcal{D}_{\text{KL}}^{(s)} = \text{KL}\left(\sigma\left(\frac{\tilde{z}_i^t(x_s)}{T}\right) \middle\| \sigma\left(\frac{z_i^t(x_s)}{T}\right)\right) \tag{8}$$

where $z_i^t(x_s)$ is the local model's logits for sample x_s, $\tilde{z}_i^t(x_s)$ denotes the aggregated logits from neighbors $\mathcal{N}_i$ including its own logits, $\sigma(\cdot)$ is the softmax function, T is the distillation temperature controlling the softness of the output distribution, and $w_{y_s}(t)$ is a dynamic class-based weighting factor dependent on the label y_s and the communication round t.

$\mathcal{D}_{\text{KL}}^{(s)}$ is the Kullback–Leibler (KL) divergence that quantifies the discrepancy between two probability distributions. In this context, it measures how much the local model's output diverges from the softened aggregated logits. Given

two temperature-scaled softmax outputs $P = \sigma(\tilde{z}/T)$ and $Q = \sigma(z/T)$, the KL divergence is computed as:

$$\mathrm{KL}(P \,\|\, Q) = \sum_{c=1}^{C} P_c \log\left(\frac{P_c}{Q_c}\right), \tag{9}$$

where C is the number of classes, and P_c, Q_c denote the predicted probabilities for class c in the teacher (peer) and student (local) distributions, respectively.

3) Adaptive Per-Sample Weighting: Federated learning under non-IID conditions often suffers from significant class imbalance at the client level, where local datasets are skewed toward a limited subset of the global label space. This skew introduces a learning bias that disproportionately favors majority classes, limiting both local model performance and the effectiveness of inter-client knowledge transfer. In the context of BlendDFL, which leverages local data for KD, such imbalance hinders the effective utilization of knowledge from underrepresented classes.

To address this challenge, we propose an *adaptive per-sample weighting mechanism* that amplifies the contribution of minority-class samples during local training. Our approach dynamically adjusts class weights based on their inverse local class frequencies and incorporates a time-dependent scaling factor to progressively increase the influence of rare classes over communication rounds. This strategy encourages the model to attend more effectively to underrepresented concepts, enhances the diversity of transferred knowledge, and promotes fairness and generalization in highly skewed federated environments. By aligning the optimization objective with the underlying data heterogeneity, the proposed mechanism not only mitigates bias but also strengthens the collaborative learning dynamics of decentralized KD.

Let $y_s \in \{1, \ldots, C\}$ denote a class label for sample x_s, and $\mathcal{C}_{\text{local}} \subseteq \{1, \ldots, C\}$ be the set of classes observed locally. We first compute inverse-frequency weights to reduce the dominance of overrepresented classes:

$$\beta_{y_s} = \frac{1}{\text{count}(y_s)}, \quad \forall y_s \in \mathcal{C}_{\text{local}}, \tag{10}$$

where $\text{count}(y_s)$ is the number of samples from class y_s within the current mini-batch. The inverse-frequency weights are normalized such that their mean over locally observed classes equals 1. This rescaling ensures consistent loss magnitudes across clients and training rounds, promoting stability and fairness during optimization:

$$\beta_{y_s} \leftarrow \frac{\beta_{y_s}}{\sum_{c \in \mathcal{C}_{\text{local}}} \beta_c} \cdot |\mathcal{C}_{\text{local}}|. \tag{11}$$

To ensure stable optimization and prevent abrupt weight shifts early in training, we introduce a time-dependent class weighting that gradually amplifies the influence of minority classes as training progresses:

$$w_{y_s}(t) = 1 + \frac{t}{R} \cdot (\beta_{y_s} - 1), \tag{12}$$

where t denotes the current communication round and R is the total number of rounds. This annealing strategy allows the model to initially focus on stable patterns before gradually increasing attention to underrepresented classes, thereby improving generalization.

4) Local Training Objective in BlendDFL: These adaptive weights are applied to both the supervised cross-entropy loss and the KD loss. This ensures that each training sample, regardless of class frequency, contributes proportionally to

Algorithm 1: BlendDFL: Locally-Guided Knowledge Distillation in Decentralized Federated Learning

Input: Initial models $\{\theta_i^{0,0}\}_{i=1}^N$, datasets $\{\mathcal{D}_i\}$, neighbor sets $\{\mathcal{N}_i\}$, total rounds R, local epochs K, learning rate η

for $t \leftarrow 0$ **to** R **do**
  foreach *client* $i \in \{1, \ldots, N\}$ ***in parallel*** **do**
    if t == *0* **then**
      for $k \leftarrow 0$ **to** $K-1$ **do**
        Sample mini-batch $\mathcal{B}_i = \{(x_s, y_s)\}$ from $\mathcal{D}_i$;
        Compute CE loss $\mathcal{L}_{\text{CE}}$ using Eq. (13);
        Update model:
        $$\theta_i^{0,k+1} = \theta_i^{0,k} - \eta \nabla_{\theta_i} \mathcal{L}_{\text{CE}}$$
    else
      Initialize local model: $\theta_i^{t,0} \leftarrow \bar{\theta}_i^{t-1}$;
      for $k \leftarrow 0$ **to** $K-1$ **do**
        Sample mini-batch $\mathcal{B}_i = \{(x_s, y_s)\}$ from $\mathcal{D}_i$;
        foreach *sample* $(x_s, y_s) \in \mathcal{B}_i$ **do**
          Compute logits $z_j(x_s)$ from each neighbor model $\theta_j^{t-1,K}$;
          Aggregate logits:
          $$\tilde{z}_i^t(x_s) = \frac{1}{|\mathcal{N}_i| + 1} \sum_{j \in \mathcal{N}_i \cup \{i\}} z_j^{t-1}(x_s)$$
          Compute class-based weight $w_{y_s}(t)$ using Eqs. (10)–(12);
        Compute KD loss $\mathcal{L}_{\text{KD}}$ using Eq. (7);
        Compute CE loss $\mathcal{L}_{\text{CE}}$ using Eq. (13);
        Compute total loss:
        $$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{KD}}$$
        Update model:
        $$\theta_i^{t,k+1} = \theta_i^{t,k} - \eta \nabla_{\theta_i} \mathcal{L}_{\text{total}}$$
    Send updated model $\theta_i^{t,K}$ to neighbors;
    Receive neighbor models $\{\theta_j^{t,K}\}_{j \in \mathcal{N}_i}$;
    Aggregate model parameters:
    $$\bar{\theta}_i^t = \frac{1}{|\mathcal{N}_i| + 1} \sum_{j \in \mathcal{N}_i \cup \{i\}} \theta_j^{t,K}$$

both learning signals. By aligning the weighting schemes across the supervised and distillation objectives, we maintain consistency in optimization and encourage the model to learn meaningful representations for underrepresented classes in both loss components. Specifically, the supervised objective is computed using a weighted cross-entropy loss, where each training sample is scaled by its corresponding time-adaptive class weight:

$$\mathcal{L}_{\text{CE}} = \frac{1}{\sum_s w_{y_s}(t)} \sum_s w_{y_s}(t) \cdot \text{CE}(z_i^t(x_s), y_s), \tag{13}$$

where $\text{CE}(\cdot)$ denotes the standard cross-entropy loss and $z_i^t(x_s)$ is the model prediction at round t for sample x_s. The total local objective is then given by:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{KD}}, \tag{14}$$

where λ is a hyperparameter that adjusts the strength of the KD loss $\mathcal{L}_{\text{KD}}$ relative to the supervised loss $\mathcal{L}_{\text{CE}}$.

This locally-guided KD framework offers **two key advantages**. First, it respects data privacy by avoiding the need to share raw data or intermediate representations. Second, it enhances generalization by transferring nuanced inter-class relationships embedded in the soft predictions of neighboring models. By aligning with these soft targets, clients are able to refine their local decision boundaries in a semantically meaningful way, leading to improved performance under heterogeneous data distributions.

IV. Experiments

A. Experimental Setup

Datasets. We evaluate BlendDFL on two widely used real-world image classification datasets: CIFAR-10 and CIFAR-100 [35], which consist of 60,000 32×32 color images in 10 and 100 classes, respectively. Each dataset contains 50,000 training images and 10,000 test images. These datasets are commonly used in federated learning to evaluate performance under data heterogeneity. To simulate a realistic non-IID setting, we adopt a Dirichlet-based data partitioning strategy, where each client's label distribution is sampled from a Dirichlet distribution with concentration parameter α. Lower values of α indicate a higher degree of non-IIDness. In our experiments, we use $\alpha = 0.3$ and $\alpha = 1.0$ to model severe and moderate label distribution skew, respectively.

Compared Methods. To evaluate the effectiveness of our proposed framework, we compare it with several strong baselines in DFL settings. **D-PSGD** [7] (Decentralized Parallel Stochastic Gradient Descent) is a foundational decentralized optimization algorithm in which each client aggregates models received from its neighbors and performs a single-step local update via stochastic gradient descent. To adapt D-PSGD for FL, we extend the local training procedure from a single SGD step to multiple local epochs per communication round, aligning it with standard FL protocols. **DFedSAM** [21] incorporates Sharpness-Aware Minimization (SAM) into DFL to improve generalization by encouraging flat minima. Clients apply SAM locally and communicate with neighbors to collaboratively train a global model in a decentralized manner. **DFedAvgM** [8] introduces a momentum-based decentralized federated averaging approach that employs heavy-ball momentum to enhance convergence speed and stability.

Network Topology. To emulate realistic decentralized environments, we evaluate our method under two widely studied communication topologies: **grid** and **ring**. These topologies reflect common peer-to-peer communication structures in real-world applications such as edge computing, sensor networks, and wireless systems. In the grid topology, clients are arranged in a two-dimensional lattice. Each client communicates with its immediate neighbors (up, down, left, right), though the actual number of neighbors varies by location where corner clients have two, edge clients have three, and interior clients have four. In contrast, the ring topology connects each client to exactly two neighbors in a circular configuration, simulating highly constrained communication. Evaluating under both settings allows us to assess the robustness and adaptability of our method across different network sparsity levels and communication diameters.

Implementation Details. To reflect the limited computational resources typical of clients in federated learning scenarios, we adopt a lightweight convolutional neural network (CNN) architecture, similar to [36]. The model consists of two convolutional layers, each with 64 filters of size 5×5, followed by three fully connected layers with 384, 192, and 10 neurons, respectively. Group Normalization is applied after each convolutional layer to enhance generalization and training stability under non-IID data.

We use the SGD optimizer with a weight decay of 0.0005 for both D-PSGD and our proposed BlendDFL. All experiments are conducted with 50 clients. For the grid topology, clients are arranged in a 2D lattice with 10 rows and 5 columns. Each client performs 5 local training epochs before communication. The mini-batch size is set to 64. The learning rate is initialized at 0.01 and decays exponentially with a factor of 0.998 per communication round. All experiments are run for 1,000 communication rounds. Each client communicates with its neighbors in every round, simulating moderately sparse decentralized connectivity. We set the temperature to $T = 3$ to soften the aggregated logits and produce informative soft labels. We select the KD loss weight λ from the set $\{1, 5, 10, 15, 20, 25\}$ through empirical tuning. The optimal value of λ varies based on the dataset and the network topology. For the grid topology, we use $\lambda = 10$ for CIFAR-10 $\lambda = 1$ for CIFAR-100. For the ring topology, we use $\lambda = 25$ for CIFAR-10 $\lambda = 5$ for CIFAR-100. For DFedAvgM, we set the momentum parameter to 0.9, following the original implementation [8]. For DFedSAM, we set the perturbation radius ρ to 0.01, consistent with recommendations in [21].

Evaluation Metrics. To evaluate the effectiveness of our approach, we measure the *global test accuracy* of the aggregated model at each client, capturing its generalization ability

TABLE I: Average test accuracy (%) of different methods on CIFAR-10 and CIFAR-100 under the grid topology.

Methods	Grid Topology			
	CIFAR-10		CIFAR-100	
	Dir 0.3	Dir 1.0	Dir 0.3	Dir 1.0
D-PSGD [7]	55.56 ± 1.99	57.13 ± 1.19	44.33 ± 1.44	48.34 ± 0.93
DFedAvgM [8]	60.78 ± 1.21	65.21 ± 1.04	37.16 ± 1.18	38.82 ± 1.34
DFedSAM [21]	57.56 ± 1.79	62.66 ± 1.26	44.19 ± 1.36	49.41 ± 1.11
BlendDFL (ours)	**63.67 ± 0.88**	**68.43 ± 0.50**	**48.46 ± 0.75**	**51.26 ± 0.87**

TABLE II: Average test accuracy (%) of different methods on CIFAR-10 and CIFAR-100 under the ring topology.

Methods	Ring Topology			
	CIFAR-10		CIFAR-100	
	Dir 0.3	Dir 1.0	Dir 0.3	Dir 1.0
D-PSGD [7]	49.45 ± 3.32	53.12 ± 2.15	37.24 ± 1.58	41.42 ± 1.04
DFedAvgM [8]	54.32 ± 2.69	59.47 ± 1.36	28.91 ± 1.04	32.31 ± 1.25
DFedSAM [8]	50.06 ± 2.85	54.41 ± 1.85	36.32 ± 1.26	40.08 ± 0.90
BlendDFL (ours)	**57.63 ± 1.92**	**65.27 ± 1.53**	**41.96 ± 0.86**	**44.83 ± 1.20**

across the network. At each round, after local training, each client aggregates its locally updated model with those of its neighbors to form the final model used for inference. We evaluate this aggregated model on a subset of the global test dataset. The reported metric is the average of these per-client accuracies, reflecting the overall generalization performance across all clients.

B. Experimental Results

We evaluate the proposed BlendDFL framework on CIFAR-10 and CIFAR-100 under two degrees of data heterogeneity ($\alpha = 0.3$ and $\alpha = 1.0$) across the grid and ring network topologies. The experimental results assess both convergence behavior and final generalization performance. Table I and II summarize the final global test accuracy averaged over the last communication round under grid and ring topologies, respectively. BlendDFL consistently outperforms all baselines including D-PSGD [7], DFedAvgM [8], and DFedSAM [21] across all settings, achieving up to 5.8% improvement in average accuracy for CIFAR-10 and up to 4.7% improvement for CIFAR-100. In addition to superior average performance, BlendDFL also exhibits low standard deviation, indicating stable and consistent generalization across clients. This demonstrates the effectiveness of incorporating locally-guided KD in enhancing generalization.

Fig. 4 and Fig. 5 illustrate the learning curves of different methods over communication rounds for $\alpha = 0.3$ and $\alpha = 1.0$, respectively. BlendDFL demonstrates faster convergence and higher final accuracy under both non-IID settings. Specifically, BlendDFL reaches high accuracy within fewer communication rounds across all datasets and topologies, and achieves the best generalization performance in both moderate and severe data heterogeneity conditions. Notably, on the more challenging CIFAR-100 dataset, BlendDFL improves the performance gap introduced by data heterogeneity, while baselines such as DFedAvgM exhibits stagnation or slower progress. The consistent gains across topologies further confirm that BlendDFL is topology-agnostic and can generalize well under different network configurations.

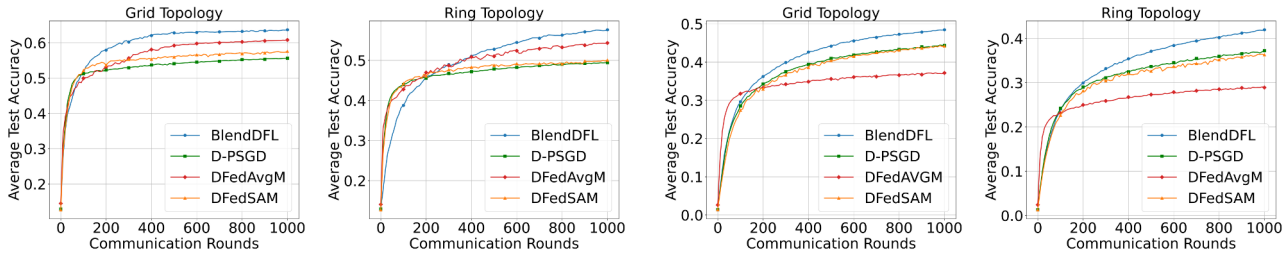
These results demonstrate that locally-guided KD from neighbors enables more effective and stable learning in decentralized federated settings, leading to improved global performance without centralized coordination.

C. Ablation Studies

To evaluate the contribution of each key component in BlendDFL, we conduct a systematic ablation study by disabling one component at a time while keeping the others active. We compute the average generalization accuracy on CIFAR-10 and CIFAR-100 with $\alpha = 0.3$ using grid topology, and consider the following variants:

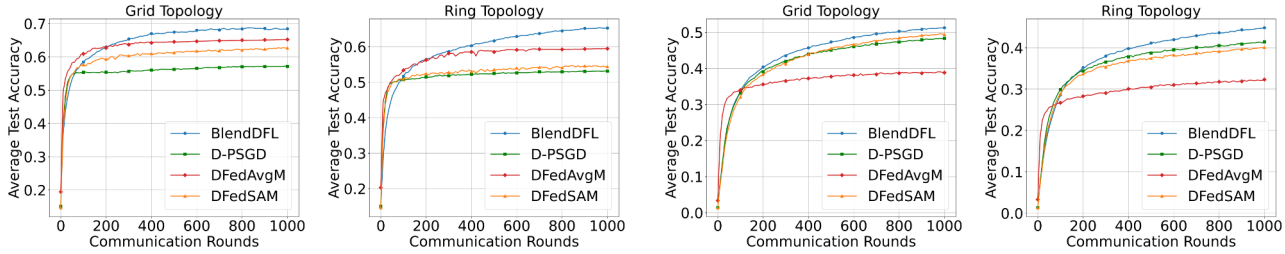
- **BlendDFL-NoKD:** This variant disables the KD component by setting $\lambda = 0$. Each client trains its local model independently without utilizing logits from neighboring clients.
- **BlendDFL-NoWeight:** This variant removes the adaptive weighting mechanism by setting $w_{y_s}(t) = 1$ for all samples, where each sample contributes equally to local training loss.
- **BlendDFL-FixedWeight:** This variant replaces adaptive weights with fixed inverse-frequency weights, where each sample is weighted by $w_{y_s}(t) = \beta_{y_s}$.

The results are reported in Table III. Across both CIFAR-10 and CIFAR-100, we observe that removing any single component from BlendDFL results in reduced performance, confirming that each module contributes meaningfully to overall generalization. Disabling KD (BlendDFL-NoKD) causes the largest drop in accuracy, underscoring the central role of peer-informed training. We find that for CIFAR-10, the fixed inverse-frequency weighting (BlendDFL-FixedWeight) performs slightly better than uniform weighting, whereas for CIFAR-100, the uniform approach (BlendDFL-NoWeight) is more effective than the fixed scheme. These results suggest that the effectiveness of weighting strategies may depend on dataset characteristics. Nevertheless, our adaptive weighting consistently delivers the best performance across both datasets, highlighting the benefit of dynamically adjusting sample weights based on training dynamics.



(a) CIFAR-10 (b) CIFAR-100

Fig. 4: Test accuracy of different algorithms on (a) CIFAR-10 and (b) CIFAR-100 datasets under the Grid and Ring topologies with data heterogeneity $\alpha = 0.3$.



(a) CIFAR-10 (b) CIFAR-100

Fig. 5: Test accuracy of different algorithms on (a) CIFAR-10 and (b) CIFAR-100 datasets under the Grid and Ring topologies with data heterogeneity $\alpha = 1.0$.

TABLE III: Ablation study: Effect of each BlendFL component on average test accuracy (%).

Method Variant	CIFAR-10	CIFAR-100
BlendDFL-NoKD	55.83	45.07
BlendDFL-NoWeight	61.99	47.84
BlendDFL-FixedWeight	62.78	46.92
BlendDFL (Ours)	**63.67**	**48.46**

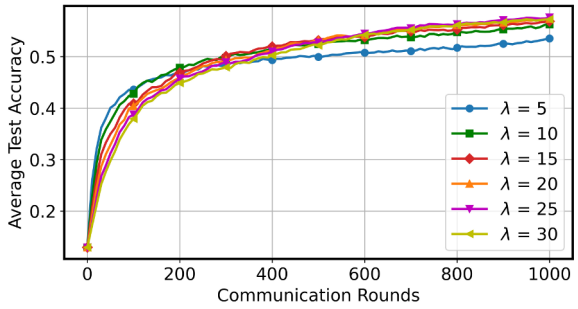


Fig. 6: Impact of Distillation Weight λ.

Impact of Distillation Weight λ. We evaluate the impact of varying the KD weight λ, which controls the relative contribution of the distillation loss to the standard cross-entropy loss. To examine this effect, we conduct experiments on CIFAR-10 with $\alpha = 0.3$ using a ring topology, exploring $\lambda \in \{5, 10, 15, 20, 25, 30\}$. As shown in Fig. 6, model performance is sensitive to the choice of λ. Smaller values such as $\lambda = 5$ lead to faster initial accuracy gains, likely due to stronger reliance on local supervision. However, their performance improves more slowly in later rounds. Moderate values like $\lambda = 25$ strike a more effective balance, facilitating better integration of peer knowledge and yielding higher final accuracy. In contrast, excessively large values such as $\lambda = 30$ can slightly degrade performance, suggesting that over-reliance on peer logits may suppress beneficial local learning. These results underscore the importance of carefully tuning λ to balance local and peer supervision.

V. Conclusion

In this paper, we have introduced BlendDFL, a novel decentralized federated learning framework that enhances generalization performance in non-IID environments. By employing locally-guided knowledge distillation, BlendDFL enables clients to assimilate rich knowledge from their neighbors while preserving their own data privacy and distribution-specific insights. Our design eliminates reliance on public proxy datasets and introduces adaptive per-sample weighting to further enhance robustness against data heterogeneity and imbalance. Extensive experiments across diverse benchmarks demonstrate that BlendDFL consistently outperforms the state-of-the-art decentralized FL methods, especially under non-IID conditions. For the future work, we aim to extend BlendDFL to heterogeneous-resource environments, allowing efficient adaptation to clients with diverse computational, memory, and communication capabilities. Moreover, we plan to explore asynchronous variants of BlendDFL to better accommodate practical decentralized systems characterized by client variability in speed and availability, further broadening the applicability of our approach.

ACKNOWLEDGMENT

This work was supported in part by the Australian Research Council (ARC) under grants DP230100233 and DE180100950, the Australia-Germany Joint Research Cooperation Scheme under grant 57702286, and Australian Commonwealth Government Digital Finance CRC (Cooperative Research Centre) via its PhD Top-Up Scholarship.

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. of the 20th International Conference on Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[2] B. Soltani, V. Haghighi, A. Mahmood, Q. Z. Sheng, and L. Yao, "A survey on participant selection for federated learning in mobile networks," in *Proc. of the 17th ACM Workshop on Mobility in the Evolving Internet Architecture*, 2022, pp. 19–24.

[3] F. Islam, A. Mahmood, N. Mukhtiar, K. E. Wijethilake, and Q. Z. Sheng, "Fairequityfl–a fair and equitable client selection in federated learning for heterogeneous iov networks," in *International Conference on Advanced Data Mining and Applications*. Springer, 2024, pp. 254–269.

[4] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[5] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[6] Z. Tang, S. Shi, B. Li, and X. Chu, "Gossipfl: A decentralized federated learning framework with sparsified and adaptive communication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 3, pp. 909–922, 2022.

[7] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[8] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4289–4301, 2022.

[9] S. K. S. Thabet, B. Soltani, Y. Zhou, Q. Z. Sheng, and S. Wen, "Towards efficient decentralized federated learning: A survey," in *International Conference on Advanced Data Mining and Applications*. Springer, 2024, pp. 208–222.

[10] E. T. M. Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2983–3013, 2023.

[11] B. Soltani, V. Haghighi, Y. Zhou, Q. Z. Sheng, and L. Yao, "Dflstar: A decentralized federated learning framework with self-knowledge distillation and participant selection," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 2108–2117.

[12] C.-Y. Huang, K. Srinivas, X. Zhang, and X. Li, "Overcoming data and model heterogeneities in decentralized federated learning via synthetic anchors," in *International Conference on Machine Learning*. PMLR, 2024, pp. 20 111–20 133.

[13] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*. Elsevier, 1989, vol. 24, pp. 109–165.

[14] F. Varno, M. Saghayi, L. Rafiee Sevyeri, S. Gupta, S. Matwin, and M. Havaei, "Adabest: Minimizing client drift in federated learning via adaptive bias estimation," in *European Conference on Computer Vision*. Springer, 2022, pp. 710–726.

[15] A. Mora, I. Tenison, P. Bellavista, and I. Rish, "Knowledge distillation in federated learning: a practical guide," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024, pp. 8188–8196.

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

[17] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 2351–2363.

[18] L. Yuan, Z. Wang, L. Sun, P. S. Yu, and C. G. Brinton, "Decentralized federated learning: A survey and perspective," *IEEE Internet of Things Journal*, vol. 11, no. 21, pp. 34 617–34 638, 2024.

[19] E. Hallaji, R. Razavi-Far, M. Saif, B. Wang, and Q. Yang, "Decentralized federated learning: A survey on security and privacy," *IEEE Transactions on Big Data*, vol. 10, no. 2, pp. 194–213, 2024.

[20] Y. Liu, Y. Shi, Q. Li, B. Wu, X. Wang, and L. Shen, "Decentralized directed collaboration for personalized federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 23 168–23 178.

[21] Y. Shi, L. Shen, K. Wei, Y. Sun, B. Yuan, X. Wang, and D. Tao, "Improving the model consistency of decentralized federated learning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 31 269–31 291.

[22] J. Liu, T. Che, Y. Zhou, R. Jin, H. Dai, D. Dou, and P. Valduriez, "Aedfl: efficient asynchronous decentralized federated learning with heterogeneous devices," in *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*. SIAM, 2024, pp. 833–841.

[23] T. Lin, S. P. Karimireddy, S. Stich, and M. Jaggi, "Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6654–6665.

[24] R. Dai, L. Shen, F. He, X. Tian, and D. Tao, "Dispfl: Towards communication-efficient personalized federated learning via decentralized sparse training," in *International Conference on Machine Learning*. PMLR, 2022, pp. 4587–4604.

[25] S. Chen, Y. Xu, H. Xu, Z. Ma, and Z. Wang, "Enhancing decentralized and personalized federated learning with topology construction," *IEEE Trans. on Mobile Computing*, vol. 23, no. 10, pp. 9692–9707, 2024.

[26] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[27] E. Zhu, C. Zhao, H. Yang, J. Li, Y. Wu, and R. Ding, "A comprehensive review of knowledge distillation-methods, applications, and future directions," *International Journal of Innovative Research in Computer Science & Technology*, vol. 12, no. 3, pp. 106–112, 2024.

[28] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan, "Fine-tuning global model via data-free knowledge distillation for non-iid federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 174–10 183.

[29] X. Gong, A. Sharma, S. Karanam, Z. Wu, T. Chen, D. Doermann, and A. Innanje, "Preserving privacy in federated learning with ensemble cross-domain knowledge distillation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 11, 2022, pp. 11 891–11 899.

[30] Y. Ma, Z. Xie, J. Wang, K. Chen, and L. Shou, "Continual federated learning based on knowledge distillation." in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 2022, pp. 2182–2188.

[31] Y. Pan, Z. Su, J. Ni, Y. Wang, and J. Zhou, "Privacy-preserving heterogeneous personalized federated learning with knowledge," *IEEE Trans. on Network Science and Engineering*, vol. 11, no. 6, pp. 5969–5982, 2024.

[32] G. Lee, M. Jeong, Y. Shin, S. Bae, and S.-Y. Yun, "Preservation of the global knowledge by not-true distillation in federated learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 461–38 474, 2022.

[33] A. Zhmoginov, M. Sandler, N. Miller, G. Kristiansen, and M. Vladymyrov, "Decentralized learning with multi-headed distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8053–8063.

[34] M. Bornstein, T. Rabbani, E. Z. Wang, A. Bedi, and F. Huang, "SWIFT: Rapid decentralized federated learning via wait-free model communication," in *Proc. of The Eleventh International Conference on Learning Representations*, 2023.

[35] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univerity of Toronto, Tech. Rep., 2009.

[36] A. E. Durmus, Z. Yue, M. Ramon, M. Matthew, W. Paul, and S. Venkatesh, "Federated learning based on dynamic regularization," in *International Conference on Learning Representations*, 2021.