Dynamic Topology Optimization for Non-IID Data in Decentralized Learning

Robust Decentralized Learning

Antreas Ioannou
Supervisors: Bart Cox, Jérémie Decouchant
EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to the EEMCS Faculty of Delft University of Technology,
In Partial Fulfilment of the Requirements
For the **Bachelor of Computer Science and Engineering**
June 8, 2025

Name of the student: Antreas Ioannou
Final project course: CSE3000 Research Project
Thesis committee: Bart Cox, Jérémie Decouchant, Anna Lukina

Dynamic Topology Optimization for Non-IID Data in Decentralized Learning

Antreas Ioannou, Bart Cox, Jérémie Decouchant
Delft University of Technology

***Abstract*—Decentralized learning (DL) enables a set of nodes to train a model collaboratively without central coordination, offering benefits for privacy and scalability. However, DL struggles to train a high accuracy model when the data distribution is non-independent and identically distributed (non-IID) and when the communication topology is static. To address these issues, we propose DissDL, a fully decentralized topology optimization algorithm for DL where nodes select the peers they exchange models with based on local model dissimilarity. DissDL maintains a fixed in-degree while dynamically adapting the communication graph via gossip-based peer discovery and diversity-driven neighbor selection, improving robustness to data heterogeneity. Experiments on CIFAR-10 and FEMNIST show that DissDL achieves faster convergence, more stable learning, as quantified by lower inter-node variance across nodes, and higher final accuracy than with static topologies and state-of-the-art baselines. For example, on CIFAR-10, DissDL reaches the best accuracy achieved by its most competitive baseline using $1.34\times$ fewer communication rounds and $1.34\times$ less communication cost, while maintaining comparable overhead.**

I. Introduction

Federated Learning (FL) has emerged as an alternative to traditional centralized machine learning, where data is aggregated in a central location, to reduce reliance on central data storage. FL is a common distributed learning paradigm where a central coordinator orchestrates the training process by aggregating model updates from participating clients [1–3]. In addition, FL addresses privacy concerns related to sensitive data being pooled on a central server [4, 5]. While several variants of FL have been described, e.g., using several servers [6] or asynchronous client-server interactions [7], FL always require central coordination, which can introduce scalability limitations [8–10] and create a performance bottleneck [11, 12]. Other challenges that FL face include privacy attacks [13–15] and robustness against poisoning attacks [16].

Decentralized Learning (DL) is a distributed learning scheme that has been proposed to eliminate FL's need for central coordinators. In DL, nodes discover each other and communicate through peer-to-peer (P2P) or gossip-based protocols [17, 18]. While DL mitigates many performance-related FL limitations, it also introduces challenges in communication efficiency. In particular, fully connected topologies are impractical in large-scale networks [19], which force DL to rely on sparsely connected communication topologies.

The communication topology used in a DL system significantly affects its communication cost, convergence rate, scalability, and final accuracy [20], especially under non-independent and identically distributed (non-IID) data conditions [21–24], where nodes possess diverse local datasets. Many studies attempt to address the non-IID challenge using static topologies and decentralized optimization methods such as decentralized parallel stochastic gradient descent (D-PSGD) [10]. However, such static-topology methods often struggle to effectively handle non-IID data when the network structure lacks sufficient connectivity or exposes nodes to overly similar local data, limiting global knowledge exchange [23].

To overcome this, recent research has explored adaptive topologies, demonstrating the benefits of dynamically adjusting the communication graph during training [25–27]. However, many such methods require some form of global knowledge or lack mechanisms for dynamic adaptation, limiting their scalability and robustness in heterogeneous settings. It is therefore still an open issue to design a fully decentralized approach that explicitly accounts for non-IID data while enabling intelligent dynamic peer selection.

This work introduces a fully decentralized method, named **Dissimilarity-guided Decentralized Learning (DissDL)**, which enables nodes to select communication partners based on local model dissimilarity, without relying on any form of global knowledge or central orchestration. Each node dynamically evaluates and adjusts its incoming connections, from which it receives others' models to update its own, while maintaining a fixed in-degree. Maintaining a fixed in-degree ensures that every node consistently receives model updates, preventing any node from becoming isolated or starved of information. Additionally, DissDL enables nodes to progressively discover new peers over time, expanding their local view of the network.

As a summary, this work makes the following contributions:

- We propose DissDL, a novel fully decentralized algorithm that dynamically adjusts the communication topology based on local model dissimilarity. This allows nodes to optimize their incoming connections in a fully distributed manner, without global information or centralized coordination. DissDL maintains a fixed in-degree per node by probabilistically selecting diverse peers for incoming, rather than outgoing, connections. This guarantees that every node is exposed to external information in every round, mitigating local overfitting under non-IID data. To enable peer discovery, nodes exchange information about their known neighbors during model updates, progressively expanding their local view of the network. This gossip-based mechanism, combined with a lightweight asyn-

chronous handshake protocol for establishing connections, aligns with practical decentralized environments where nodes lack global awareness of all peers. DissDL further allows nodes to infer model similarity with unknown peers via gossip, enabling informed peer selection even under partial network knowledge. This enhances adaptability in sparse and evolving topologies.
- We evaluate DissDL on the CIFAR-10 and FEMNIST datasets using realistic non-IID data distributions. DissDL outperforms static and state-of-the-art baselines by achieving their best top-1 accuracy (i.e., standard accuracy, where the model's most confident prediction must match the label) with up to $1.34\times$ fewer communication rounds and communication cost on CIFAR-10, and $1.17\times$ fewer on FEMNIST, while also achieving higher final accuracy and more consistent performance across nodes, as quantified by lower inter-node variance.

II. Background

A. System Model

A decentralized learning (DL) system consists of a set of distributed computational nodes, denoted by $\mathcal{N} = \{1, 2, \ldots, n\}$, which collaborate to train a shared model. Each node maintains access to its own local data and performs computations independently, without reliance on a central server. The communication among nodes occurs over a network topology represented by a graph $G = (V, E)$, where each node corresponds to a vertex $v \in V$, and an edge $(j, i) \in E$ indicates that node j can send information directly to node i.

In fully decentralized settings, such as the one assumed in this work, the communication model is inspired by classical peer-to-peer (P2P) systems, in which nodes operate as equal participants, both consuming and supplying information [28, 29]. Randomized gossip protocols are often used to propagate information efficiently without centralized scheduling [30–32]. In our system, a P2P peer discovery service periodically provides each node with a set of new potential neighbors, enabling continuous exploration of the network.

In this work, we consider directed graphs with evolving topologies, where communication links may change over time. The out-degree of a node is the number of other nodes it transmits information to, while its in-degree is the number of nodes from which it receives information. We assume that the initial communication graph is connected in the undirected sense, that is, if edge directions are ignored, there exists a path between any pair of nodes. While each node initially communicates only with a subset of neighbors, we assume that nodes can, in principle, establish connections with any other node, provided they become aware of its existence (e.g., via peer exchanges or broadcasts). These assumptions, or relaxed variations of them, are standard in prior work and reflect practical decentralized environments, where partial initial connectivity and the ability to communicate with any node over time are common.

B. Decentralized Learning

We consider the standard decentralized learning objective in which a group of n nodes seeks to collaboratively minimize a global loss function by performing local updates and exchanging information with neighbors. Each node $i \in [1, n]$ holds private data that follows a distribution $\mathcal{D}^{(i)}$, which may differ from the one of other nodes, over a data space $\mathcal{Z}$. Let $f : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}$ be a loss function that evaluates model performance on a data point. The local loss function at node i is defined as the expectation over its local distribution:

$$f^{(i)}(x) := \mathbb{E}_{\xi \sim \mathcal{D}^{(i)}}[f(x, \xi)]. \tag{1}$$

The goal of the system is to minimize the average loss over all nodes:

$$\min_{x \in \mathbb{R}^d} F(x) := \frac{1}{n} \sum_{i=1}^{n} f^{(i)}(x). \tag{2}$$

The training process proceeds in rounds, each consisting of local computation, communication with neighbors, possible topology adaptation, and aggregation of received model updates.

III. Related Work

Table I provides a comparative overview of recent topology aware distributed methods, highlighting their core characteristics in terms of decentralization, information requirements, adaptation strategy, and topological flexibility.

A. Fixed Topology Algorithms

Several methods have been proposed to improve the performance of decentralized learning (DL), typically assuming a fixed topology. These approaches either focus on enhancing the learning algorithm or on designing more effective fixed topologies to improve convergence under non-IID conditions.

Aketi et al. introduce two related methods tailored to non-IID environments: NGC and NGM. Neighborhood Gradient Clustering (NGC) [41] augments local updates by incorporating self-gradients along with model-variant and data-variant gradients from neighboring nodes. These are clustered based on similarity to better align updates with the global objective. Neighborhood Gradient Mean (NGM) [42] simplifies the approach by averaging local and cross-gradients, making it more suitable for bandwidth- or memory-constrained scenarios.

Gao et al. [43] employ a pre-trained Graph Neural Network (GNN) to guide model aggregation over a fixed topology. Esfandiari et al. [44] propose Cross-Gradient Aggregation (CGA), which improves model updates under non-IID data by incorporating gradient information from neighbors into a constrained quadratic programming (QP) framework. In a different approach, Song et al. [39] introduce EquiStatic, a family of communication-efficient topologies with network-size-independent consensus rates, designed to improve convergence under non-IID data. However, these approaches do not address the evolution of the communication topology itself and are bounded by the initial graph structure.

TABLE I

COMPARISON OF TOPOLOGY AWARE DISTRIBUTED ALGORITHMS. A METHOD IS CONSIDERED DECENTRALIZED IF IT REQUIRES NO CENTRAL COORDINATOR; NO GLOBAL INFO MEANS IT DOES NOT RELY ON KNOWLEDGE SUCH AS NODE IDENTITIES OR GLOBAL TOPOLOGY; GUIDED ADAPTATION USES HEURISTICS TO ADAPT THE TOPOLOGY RATHER THAN RANDOM UPDATES; FLEXIBLE TOPOLOGY MEANS IT CAN EVOLVE BEYOND A FIXED GRAPH.

Method	Decentralized	No Global Info	Guided Adaptation	Flexible Topology
Menegatti et al. [27], Lin et al. [25], Wang et al. (CoCo) [33], Zhou et al. [34], Tuan et al. (GNN+PSO) [35]	✗	✗	✓	✓
Behera et al. (PFedGame) [36]	✗	✗	✗	✓
Li et al. (L2C/meta-L2C) [37]	✓	✓	✓	✗
Assran et al. (SGP) [38]	✓	✓	✗	✗
Song et al. (EquiDyn) [39]	✓	✗	✗	✗
De Vos et al. (EL-Local) [26]	✓	✗	✗	✓
Bars et al. [22]	✗	✗	✓	✗
Dandi et al. [40]	✓	✓	✗	✗
Ours (DissDL)	✓	✓	✓	✓

B. Topology-Aware Algorithms with Global Coordination

Recent research has increasingly focused on topology-aware algorithms, often leveraging some form of global knowledge, such as the overall network structure or global connectivity patterns, to guide communication or adaptation strategies. Menegatti et al. [27] propose a centralized energy-aware topology optimization algorithm that maximizes algebraic connectivity to accelerate decentralized federated learning. While effective, their method requires full global knowledge and static pre-computation.

Behera et al. [36] frame peer interactions as a two-player cooperative game to determine aggregation strategies. While their method performs well under heterogeneous data, it is designed to operate within a given dynamic topology rather than actively adapting it, and it under-performs in more homogeneous scenarios.

A more structured approach that adapts the topology is proposed by Lin et al. [25], who use reinforcement learning to optimize peer selection based on network properties and data distributions. However, their method requires a centralized controller both to construct the initial topology and to operate a centralized deep reinforcement learning (DRL) agent for ongoing adaptation.

Likewise, relying on centralized coordination, Wang et al. [33] introduce CoCo, a topology- and compression-aware algorithm for decentralized federated learning. CoCo uses a central coordinator to gather local model and bandwidth information from all nodes, and then solves a linear program to assign peer connections and model compression levels tailored to each node. While the local training remains decentralized, the topology evolution is centrally controlled each round.

Zhou et al. [34] propose a heuristic algorithm that accelerates learning by strategically adding a limited number of edges to the communication graph. Their method aims to maximize the second smallest eigenvalue of the graph Laplacian, a measure of connectivity shown to correlate with convergence speed. However, their approach relies on algebraic connectivity, a global graph property that cannot be computed locally. Other strategies, such as combining particle swarm optimization with graph neural networks to predict globally optimal topologies [35], also depend on full knowledge of the global graph. While effective in theory, these methods are unsuitable for fully decentralized environments where nodes have only access to partial local information.

C. Decentralized Dynamic Topology Algorithms

Several studies propose methods that leverage dynamic topologies, using fully decentralized approaches or limited global knowledge. While these methods are generally decentralized, some assume minimal global information, such as unique node identities. This implies knowledge of other nodes' existence, a non-trivial assumption that, in practice, requires centralized coordination during initialization.

Koloskova et al. [45] establish convergence guarantees for decentralized stochastic optimization under time-varying topologies and heterogeneous data. Their results demonstrate the theoretical feasibility of using dynamic topologies in decentralized learning by showing that convergence is still attainable under mild conditions on graph connectivity and data smoothness.

Even though they do not actively change the topology throughout the entire training process, Li et al. [37] propose L2C and meta-L2C, two fully decentralized algorithms that learn personalized model aggregation weights to enhance collaboration among peers. Starting from a fully connected network, they prune connections early based on validation loss, resulting in a fixed sparse topology. While effective for personalization and reducing communication, full initial connectivity is impractical at scale, and the lack of ongoing adaptation limits responsiveness to changing conditions.

Another fully decentralized algorithm is introduced by Assran et al. [38], which, employs a dynamic schedule of pairwise exchanges within a one-peer exponential graph. They propose decomposing an initial static exponential graph into a sequence of one-peer graphs, where each node cycles through its neighbors, communicating with one at each round. Ying et al. [46] demonstrate that this approach achieves a convergence rate comparable to that of the static exponential graph, which has been shown to be empirically effective [19, 38, 47], while significantly reducing communication overhead. Nevertheless, this method requires control over the initial network topology, and its primary benefit lies in minimizing communication, as

its performance is ultimately bounded by that of the static exponential graph.

In a similar approach, Song et al. [39] propose EquiDyn, a dynamic variant of the EquiStatic family. In EquiDyn, each node communicates with only one neighbor per iteration, selected randomly from a predefined static graph. This approach enables EquiDyn to achieve a consensus rate that is independent of the network size while requiring only a single communication per node per round. Although the communication pattern varies over time, it is constrained to this fixed initial topology, meaning no new connections can be established. Moreover, the algorithm assumes consistent node indexing and knowledge of the total network size.

De Vos et al. [26] propose EL-Local, a decentralized topology adaptation algorithm designed for non-IID settings. At each iteration, every node selects a random subset of peers from the full network and broadcasts its model update, promoting information mixing. While this approach improves convergence, it lacks guided or adaptive neighbor selection, which may limit efficiency and early convergence. Additionally, the method assumes that each node is aware of all other nodes from the start, enabling peer selection over the full network, which is a practical limitation in fully decentralized environments.

D. Peer Dissimilarity as a Topology Signal

An important open question in decentralized learning is how to select communication partners effectively, especially when data distributions differ significantly across nodes. Recent work has begun to explore data-aware topology construction strategies, highlighting the importance of designing topologies that facilitate information exchange between heterogeneous nodes. Bars et al. [22] show that communication with dissimilar nodes, those whose local data distributions differ, helps ensure that each node's neighborhood better approximates the global distribution. Similarly, Dandi et al. [40] report that convergence improves when communication weights are aligned with the complementarity of local data, such that nodes with more diverse data distributions are more strongly connected. These findings suggest that, particularly under non-IID conditions, it is advantageous for nodes to communicate with others that have different data characteristics.

IV. DissDL

We propose a fully decentralized topology adaptation mechanism that dynamically updates each node's communication neighborhood based on model dissimilarity. Built on the standard decentralized parallel SGD (D-PSGD) framework provided by the DecentralizePy library [48], our implementation extends it to support dissimilarity-guided neighbor selection over evolving topologies, without requiring central coordination or global network information.

A. Connection Management and Peer Discovery

To support dynamic topologies, we extend DecentralizePy with two decentralized communication mechanisms: passive metadata exchange for membership discovery, and a decentralized three-way handshake for establishing new connections.

Algorithm 1: DissDL at node i

Input: Local model w_0, initial neighbors $\mathcal{N}_i$, total rounds T, evaluation frequency k

Initialization: Set known peers $\mathcal{P}_i \leftarrow \mathcal{N}_i$;
Initialize `wanted_senders` $\leftarrow$ outgoing neighbors in $\mathcal{N}_i$;
for $t \leftarrow 1$ **to** T **do**
  Process incoming connection requests and complete handshakes;
  Perform local training on private data;
  if $t \bmod k = 0$ **then**
    Call `UpdateWantedSenders()` to revise `wanted_senders` based on model dissimilarity and potentially initialize a new connection;
  Send `DPSGD_REQ` messages to all known peers: mark interest if peer $\in$ `wanted_senders`, otherwise indicate no interest;
  Receive `DPSGD_REQ` messages from peers;
  Send local model and peer information to peers that expressed interest;
  Receive models and peer information from peers in `wanted_senders`;
  Store up to the last 5 similarity estimates for each peer without a received model, using information shared by `wanted_senders` about their known peers;
  Update $\mathcal{P}_i$ using new peer information received from `wanted_senders`;
  Aggregate received models with local model via uniform averaging;

Peer Discovery via Metadata Sharing. During each communication round, nodes include the identifiers of their known peers in the metadata they share. This implicit, gossip-like mechanism enables each node to gradually expand its local view of the network without incurring significant communication or memory overhead.

Decentralized Connection Establishment. To initiate communication with a newly discovered peer, a node performs an asynchronous three-way handshake inspired by TCP. It sends a `SYN`, receives a `SYN-ACK`, and completes the connection with an `ACK`. This ensures both parties agree on the connection before any model exchange occurs. To maintain consistency despite iteration mismatches, nodes defer completing the handshake until they reach the required round, not earlier than the round indicated by the peer that initiated the connection.

B. System Overview

Each node in the system is initialized with its own private dataset, a local model, and a set of outgoing and incoming neighbors defined by a randomly generated initial graph G_0.

At initialization, a node knows only the identifiers of its direct neighbors; it has no access to their data or any global topology information. Communication occurs over a dynamic peer-to-peer network, and each node independently adapts its neighborhood over time.

The algorithm has two key parameters: k, which controls how frequently each node reevaluates its neighbors, and β, which governs the stochasticity of the peer selection process via a softmax over model similarity scores.

At each communication round t, the node executes the steps shown in Algorithm 1. First, it processes any incoming handshake messages to complete connection requests that were initiated in earlier rounds by other nodes. These are not new requests, but finalizations of pending connections.

Next, the node performs local training on its private dataset. If the round is a multiple of k, it invokes the `UpdateWantedSenders()` routine to revise its preferred incoming neighbors based on model dissimilarity, as detailed in Section IV-C. Only at this step are new connection requests initiated.

Crucially, unlike many traditional decentralized learning algorithms, where nodes decide whom to send updates to, our approach gives each node control over from whom it receives updates. This design ensures that each node's in-degree remains fixed over time, preventing situations where a node receives no updates and consequently overfits to its local data.

The node then sends `DPSGD_REQ` messages to all known peers, not just to its wanted senders, indicating, for each, whether it wishes to receive model updates. This is necessary due to the synchronized communication model offered by DecentralizePy, which requires mutual awareness of interest to coordinate message exchanges.

Each node sends its local model to peers that have expressed interest and receives models from those it has selected as wanted senders. Along with the models, it also receives metadata from these senders, specifically: the identifiers of each sender's known peers and similarity scores between the sender and those peers. The former enables the node to discover previously unknown peers and expand its known peer set $\mathcal{P}_i$, while the latter supports long-term peer discovery without incurring high memory overhead. Each node retains up to five recent similarity scores for peers from whom it has not yet received model updates, a limit chosen to balance estimation quality with memory cost, using them to estimate their potential utility. Finally, the node aggregates all received models with its own via uniform averaging.

C. Topology Adaptation via Diversity-Driven Peer Selection

We formulate the neighbor selection process as a local optimization problem, where each node independently determines the set of peers from which it will receive model updates. To maintain a fixed in-degree for the communication graph, each node adds one new peer and removes one existing peer during each adaptation round. The decision is guided by model dissimilarity, using the peer models' parameters to evaluate diversity. This topology adaptation occurs every k iterations according to Algorithm 2.

Algorithm 2: `UpdateWantedSenders()` at node i

Input : Current `wanted_senders`, known peers $\mathcal{P}_i$, known similarity scores, peer similarity estimates, dissimilarity temperature β

Define current senders $S \leftarrow$ `wanted_senders`;
Let candidates for addition: $A \leftarrow \mathcal{P}_i \setminus S$;
Let candidates for removal: $R \leftarrow S$;
if $|A| = 0$ *or* $|R| \leq 1$ **then**
  return
`// Fallback for unknown peers`
$f_{\text{default}} \leftarrow$ mean of known similarity scores;
foreach *peer* $j \in A$ **do**
  if *model received from* j **then**
    $d_j \leftarrow -\texttt{CosineSim}(w^i, w^j)$;
  else
    $d_j \leftarrow -\texttt{EstimateSimilarity}(j, f_{\text{default}})$;
Sample peer j^+ from A using softmax over $\{d_j\}$ with temperature β;
foreach *peer* $j \in R$ **do**
  if *model received from* j **then**
    $s_j \leftarrow \texttt{CosineSim}(w^i, w^j)$;
  else
    $s_j \leftarrow \texttt{EstimateFromHistory}(j, f_{\text{default}})$;
Sample peer j^- from R using softmax over $\{s_j\}$ with temperature β;
`wanted_senders` $\leftarrow (S \cup \{j^+\}) \setminus \{j^-\}$;
if j^+ *not in connected peers* **then**
  Send `CONNECT` request to j^+;

Peer Evaluation. Each node considers all known peers (connected and unconnected) as candidates to its incoming connections.

Score Computation. To quantify model diversity, each node computes a cosine similarity score between its local model w_i and the model of each candidate peer w_j. Instead of flattening the entire model into a single vector, we compute cosine similarity separately for each layer, then average the results. Let $\theta_l^{(i)}$ and $\theta_l^{(j)}$ denote the parameter tensors for the l-th layer of node i and peer j, respectively. The similarity for each layer is computed as:

$$\text{sim}_l = \frac{\theta_l^{(i)} \cdot \theta_l^{(j)}}{\|\theta_l^{(i)}\|_2 \cdot \|\theta_l^{(j)}\|_2} \tag{3}$$

The overall model similarity is then given by averaging across all L layers:

$$\text{sim}(w_i, w_j) = \frac{1}{L} \sum_{l=1}^{L} \text{sim}_l \tag{4}$$

Layer-wise averaging ensures balanced contributions across the model, while cosine similarity captures directional alignment efficiently, is robust to scale, and communication-efficient making it well-suited for decentralized settings [49].

When a node i has not directly received a peer z's model, it estimates similarity through transitive inference. Specifically, for each intermediate peer y from which node i has both the model and a reported similarity to z, the estimate is computed as:

$$\hat{\text{sim}}(w_i, w_z) := \frac{1}{|\mathcal{H}_z|} \sum_{(t,y,\sigma_{yz}) \in \mathcal{H}_z} \text{sim}(w_i, w_y) \cdot \sigma_{yz} \quad (5)$$

where $\mathcal{H}_z$ denotes the set of the 5 most recent similarity reports for peer z. This chaining approach yields grounded estimates from partial information.

Although cosine similarity is not transitive, it satisfies the following angular inequality [50]:

$$\begin{aligned} \arccos(\text{sim}(w_i, w_k)) \leq\ & \arccos(\text{sim}(w_i, w_j)) \\ & + \arccos(\text{sim}(w_j, w_k)) \end{aligned} \quad (6)$$

This inequality serves as a one-sided constraint, bounding the maximum angular difference between models through an intermediate peer. It provides theoretical justification for indirect estimation. Furthermore, quasi-transitive reasoning has been empirically shown to improve selection in noisy environments [51].

In rare cases where neither the peer's model nor similarity reports are available, the similarity is defaulted to the mean of all available similarity values observed so far. This fallback enables for peers whose existence is known but whose utility cannot yet be estimated to be considered in the selection process, supporting early-stage exploration.

Probabilistic Selection. After calculating or estimating the similarities scores, nodes use softmax sampling over the scores to select a peer with low similarity to add, from among peers that currently send updates, and a peer with high similarity to remove, from among peers that currently do not send updates. The probability of selecting peer j to add is given by:

$$p_j = \frac{\exp(-\beta \cdot \text{sim}(w, w_j))}{\sum_{k \in \mathcal{C}_a} \exp(-\beta \cdot \text{sim}(w, w_k))} \quad (7)$$

where w is the local model, w_j is the model of peer j, $\mathcal{C}_a$ is the candidate add set, and β is a positive temperature-like parameter that controls the sharpness of the distribution.

The probability of selecting peer j to remove is similarly defined as:

$$p_j = \frac{\exp(\beta \cdot \text{sim}(w, w_j))}{\sum_{k \in \mathcal{C}_r} \exp(\beta \cdot \text{sim}(w, w_k))} \quad (8)$$

where $\mathcal{C}_r$ is the candidate remove set.

This encourages connections with peers that hold models most probably trained on different or complementary data distributions, thereby promoting greater representation of global data during aggregation. We use softmax for guided selection, since it offers controlled stochasticity, which reduces the risk of getting stuck in local optima.

Topology Update. The selected peer to add is incorporated into the `wanted_senders` set, while the selected peer to remove is dropped. If the added peer is not already connected, a handshake (SYN, SYN-ACK, ACK) is initiated to establish communication.

While each node adapts its neighborhood independently, the use of model dissimilarity as a selection signal creates a self-reinforcing mechanism across the network. Nodes with complementary data distributions become more connected over time, enhancing the diversity of updates each node receives. This indirectly aligns local model spaces with the global data distribution, leading to more representative averaging during training [22, 40]. As a result, even without centralized coordination, the overall network topology evolves to better support convergence under data heterogeneity.

We further support our intuition with two experiments in Appendix B. The first tracks similarity between each node's model and the global model over time. The second, more indicative experiment, measures how each node's effective training distribution, i.e., the combination of its local data and peer updates, aligns with the true global distribution throughout rounds. Both indicate that adaptive selection of dissimilar peers, leads to more diverse updates and improved global alignment.

V. Evaluation

A. Datasets and Partitioning

CIFAR-10 is a widely-used image classification dataset consisting of 60,000 32×32 color images across 10 classes [52]. To simulate a non-IID data distribution, we partition the dataset across clients using a Dirichlet distribution [53] with a concentration parameter $\alpha = 0.1$. This results in each client having a different class distribution.

FEMNIST is a federated version of the Extended MNIST dataset, containing handwritten characters from 62 classes written by 3,550 users [54]. We report results on FEMNIST in Appendix A-A and focus on CIFAR-10 in this section, as the performance trends on FEMNIST are qualitatively similar.

B. Experimental and Implementation Setup

All experiments are conducted on a virtual machine hosted on the Google Cloud Platform (GCP), configured with 8 virtual CPUs and 30 GB of system memory (machine type `c4-standard-8`, Intel Emerald Rapids platform). The instance does not include a GPU. We simulate a decentralized system by deploying 16 parallel processes on this VM, each acting as an individual node in the network. CPU cores and memory are shared among all processes without isolation. Each experiment consists of 8,000 communication iterations. A more detailed overview of the models and parameters used is given in Table II in Appendix A. To ensure reproducibility, a uniform seed was employed for all pseudo-random generators within each node.

For the CIFAR-10 learning task, we conduct two sets of experiments, each using 16-node communication graphs. In both sets, we evaluate the performance of our method alongside baseline approaches across five independent runs with different

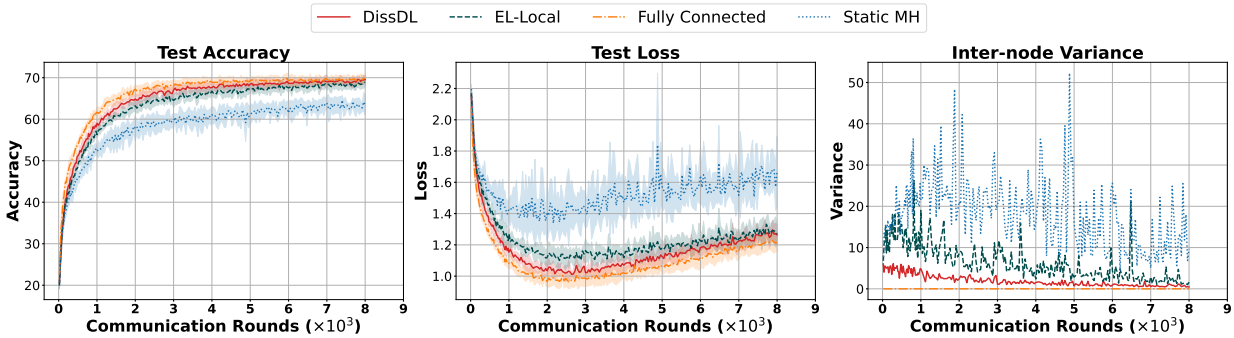

Fig. 1. Performance metrics on CIFAR-10 with 16 nodes in a non-IID setting using degree-7 topologies. The figure presents, from left to right: **(1)** mean top-1 test accuracy over communication rounds (with shaded areas indicating standard deviation across five runs), **(2)** mean test loss, and **(3)** inter-node variance as a measure of convergence stability. **DissDL** denotes our method initialized with a 7-regular topology; **EL-Local** applies the EL-Local algorithm with degree 7; **Fully Connected** operates over a fully connected graph; and **Static MH** uses a static 7-regular topology with Metropolis-Hastings averaging.

random seeds. The first set uses graphs with degree 7, while the second uses a more realistic degree of 3, which is bounded by $\mathcal{O}(\log n)$, where n is the number of nodes.

Our implementation[1] is developed using the `DecentralizePy` framework and executed in a Python 3.11.2 environment. The communication topology is initialized as a random 16-nodes 7-regular or 3-regular graph and is dynamically updated throughout training. Specifically, the topology is re-evaluated every $k = 5$ communication rounds to adapt to changing contribution dynamics, using $\beta = 500$ for the softmax temperature.

C. Baselines

We compare the performance of our algorithm against three DL baselines, specifically variants of D-PSGD.

First, we consider a baseline with a fixed 7-regular or 3-regular random graph, consistent with the initial topology used in our approach. This baseline employs the Metropolis-Hastings (MH) averaging scheme provided by the `DecentralizePy` framework [48] to mitigate topological bias.

The second baseline uses a fully connected topology, which serves as an upper bound on achievable performance [38].

The third baseline is EL-Local [26], which employs a randomly selected k-degree topology at each communication round. We set $k = 7$ or $k = 3$ to match the amount of data sent in our implementation.

D. Evaluation Metrics

We evaluate each method using four primary metrics: mean accuracy, mean test loss, inter-node variance, and total communication cost. All metrics are averaged over five independent runs with different seeds and reported over communication rounds.

Mean accuracy and test loss are computed by evaluating each node's model on a shared test set every 20 rounds, up to 1000 rounds, and every 40 rounds thereafter, then averaging across all 16 nodes. Loss is calculated using cross-entropy. Inter-node variance, used as a stability metric, is computed at the same evaluation rounds by measuring the variance of test accuracies across nodes, and then averaging across five independent runs. Total communication cost reflects bandwidth usage and is measured as the cumulative number of bytes sent per node throughout training.

In addition to plotting trends over communication rounds, we also measure the communication and convergence efficiency of our method by recording how many rounds and how much communication (in bytes) are needed to match the best performance achieved by EL-Local.

E. Degree-7 Topologies

Our first set of experiments is conducted on the CIFAR-10 dataset using topologies of degree 7, except for the fully connected configuration. The results are visualized in Figure 1, while detailed numerical values are provided in Table IV in Appendix A-B.

As expected, the fully connected topology achieves the highest initial performance but incurs more than twice the communication cost of other methods (see Figure 3, Appendix A-B). While it outperforms our approach in the early stages, DissDL catches up in later rounds and ultimately reaches a comparable top-1 accuracy, only 0.28 percentage points lower. This is notable, as the fully connected graph is typically considered an upper bound in decentralized learning scenarios.

Our proposed method converges faster than both EL-Local and Static MH, reaching a target accuracy earlier, an accuracy that EL-Local only approaches later in training. Specifically, DissDL requires $1.34\times$ fewer communication rounds and communication cost to achieve the best top-1 accuracy of EL-Local. Although EL-Local continues to improve over time, it ultimately falls short of the performance reached by our method by 0.69 percentage points. In contrast, the static Metropolis-Hastings-based topology performs the worst in both convergence speed and final accuracy.

In terms of test loss, DissDL closely follows the trend of the fully connected topology, with only slightly higher values throughout training. Toward the later stages, both methods exhibit a mild increase in loss, eventually reaching levels

[1]Code available at: https://gitlab.tudelft.nl/cse3000-2025-robust-decentralized-learning/cse-3000-antreas-ioannou

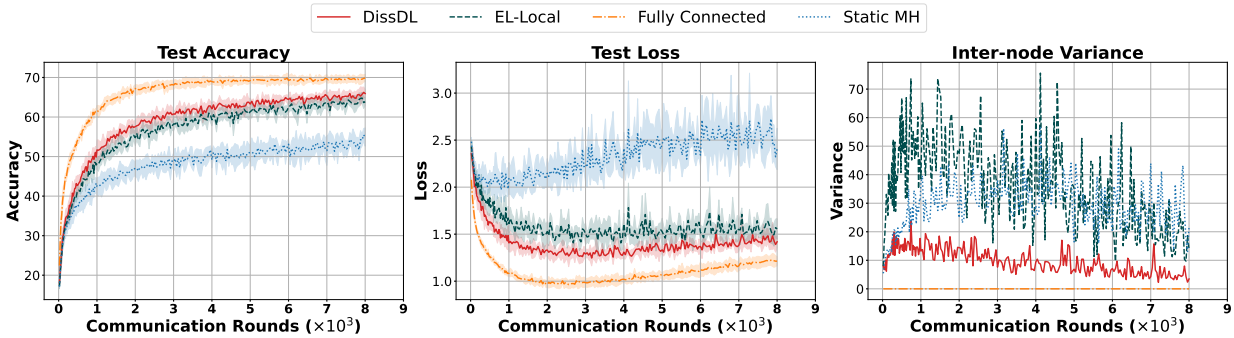

Fig. 2. Performance metrics on CIFAR-10 with 16 nodes in a non-IID setting using 3-regular topologies. The figure shows, from left to right: **(1)** mean top-1 test accuracy over communication rounds (with shaded regions indicating standard deviation), **(2)** mean test loss, **(3)** inter-node variance as a measure of convergence stability. **DissDL** refers to our dissimilarity-driven approach with a 3-regular initial topology; **EL-Local** applies the EL-Local algorithm with degree 3; **Fully Connected** uses a fully connected graph; and **Static MH** uses a static random 3-regular topology with Metropolis-Hastings averaging.

comparable to EL-Local. This behavior is expected, as confident but incorrect predictions in late training can slightly raise cross-entropy loss, even when accuracy remains high.

Also as expected, the fully connected configuration shows zero standard deviation across nodes due to uniform global averaging. Notably, our method achieves lower variance than EL-Local, indicating greater consistency across the network. Finally, the three sparse methods transmit approximately the same amount of data per round, resulting in nearly overlapping communication curves (see Figure 3 in Appenidx A-B).

F. Degree-3 Topologies

In our second set of experiments, we evaluate the same set of algorithms on 3-regular graphs, with the fully connected configuration remaining unchanged. The results are presented in Figure 2, while detailed numerical values are provided in Table V in Appendix A-B.

While the overall trends mirror those observed in the 7-degree setting, accuracy across all sparse methods is lower, an expected outcome due to reduced connectivity and slower information propagation in 3-regular topologies. DissDL and EL-Local exhibit similar convergence dynamics; however, DissDL consistently achieves higher accuracy throughout training, with EL-Local only approaching comparable levels near the end. As in the degree-7 case, DissDL requires $1.34\times$ fewer communication rounds and communication cost to reach the best top-1 accuracy of EL-Local. Additionally, DissDL surpasses EL-Local in best accuracy by 1.44 percentage points.

Once again, the static Metropolis-Hastings baseline lags behind, demonstrating the slowest convergence and the weakest final model performance. In terms of stability, DissDL exhibits significantly lower inter-node variance than the other two sparse methods, indicating more consistent learning across the network.

In terms of test loss, DissDL again tracks closely with EL-Local but maintains lower loss values throughout training, reflecting its stronger predictive performance. The fully connected topology continues to achieve the lowest loss overall, while the static MH baseline shows the highest loss and no improvement. The trends in test loss align well with those observed in accuracy, further reinforcing the relative performance of each approach. The communication cost per round remains similar across the sparse methods, as expected (see Figure 4 in Appenidx A-B).

These results highlight the effectiveness of DissDL in lower-connectivity settings, where dynamic peer selection helps mitigate the limitations imposed by sparse topologies. By adaptively selecting peers based on model dissimilarity, DissDL improves information diversity, enabling faster convergence and more consistent learning across the network, even under non-IID conditions. This demonstrates the broader utility of dissimilarity-driven topology adaptation in decentralized systems.

VI. Conclusion

We introduced DissDL, a fully decentralized learning algorithm that dynamically adapts its communication topology based on local model dissimilarity. By allowing nodes to selectively connect with peers whose models differ meaningfully from their own, DissDL enhances training robustness under non-IID data distributions. Our experimental evaluation on CIFAR-10 and FEMNIST demonstrated that DissDL consistently outperforms static and randomized baselines in terms of accuracy, convergence speed, and inter-node variance, while maintaining comparable communication overhead. Notably, it can approach the performance of fully connected networks while using over $2\times$ less communication. These results highlight model dissimilarity as a promising criterion for adaptive topology optimization in decentralized learning, especially under non-IID data. Future work could explore extending these benefits to even sparser or more heterogeneous environments by refining the peer scoring mechanism or incorporating additional node-level features alongside dissimilarity. One potential limitation of DissDL is that, while the in-degree of each node is fixed, the out-degree is unconstrained, meaning some nodes might be selected too frequently as senders. This could lead to communication imbalance. Future extensions could address this by penalizing over-selected nodes during peer selection, or by enforcing hard limits on the number of outgoing messages per node.

VII. Responsible Research

Throughout this project, careful consideration was given to ensuring the reproducibility and integrity of the research. The study was conducted in accordance with the Netherlands Code of Conduct for Research Integrity [55]. The methodology was implemented entirely in Python using the open-source DecentralizePy framework. All experiments were carried out in a controlled and consistent virtual environment, with fixed random seeds and synchronized versioning of all dependencies. All experiments were conducted on publicly available datasets. Moreover, the full codebase and detailed documentation, as well as results from the experiments, have been published, enabling all results to be replicated under the same conditions.

To uphold research integrity, no data augmentation or cherry-picking techniques were applied to influence experimental outcomes. All baselines and comparisons were implemented faithfully based on existing literature or official repositories. Standard performance metrics such as accuracy and communication cost were selected and reported transparently to reflect both learning effectiveness and system efficiency.

One key reproducibility challenge arose from the inherent stochasticity of decentralized systems. Peer discovery, model updates, and communication patterns introduce non-determinism, particularly under dynamic topologies. To mitigate this, we conducted multiple simulation runs with different seeds and reported variance across runs and/or nodes in performance plots.

However, due to the asynchronous and probabilistic nature of peer discovery in DissDL, even runs with a fixed random seed can yield slight differences in learning dynamics. This stems primarily from the timing of decentralized handshakes, limited by the hardware, which can affect whether a peer participates in a given averaging step. As a result, model updates may differ subtly across runs, even if the communication topology remains unchanged. Therefore, reproducibility in this context refers to consistent performance trends and statistical behaviors over multiple runs, rather than exact repeatability of individual model updates.

Additionally, we openly acknowledge the use of GPT-4o as a coding assistant for tasks such as code refactoring, text shortening, and LaTeX editing. The model was not used for substantive writing, but rather to support through grammar checks, structural feedback, and efficiency improvements. All research design, analysis, and interpretation were conducted independently to maintain academic integrity.

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: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Artificial Intelligence and Statistics. PMLR, Apr. 10, 2017, pp. 1273–1282.

[2] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao. "A Survey on Federated Learning". In: *Knowledge-Based Systems* 216 (Mar. 15, 2021), p. 106775.

[3] M. de Goede, B. Cox, and J. Decouchant. "Training diffusion models with federated learning". In: *arXiv preprint arXiv:2406.12575* (2024).

[4] T. Wittkopp and A. Acker. "Decentralized Federated Learning Preserves Model and Data Privacy". In: *Service-Oriented Computing – ICSOC 2020 Workshops*. Cham: Springer International Publishing, 2021, pp. 176–187.

[5] W. Yu, Q. Li, M. Lopuhaä-Zwakenberg, M. Græsbøll Christensen, and R. Heusdens. "Provable Privacy Advantages of Decentralized Federated Learning via Distributed Optimization". In: *IEEE Transactions on Information Forensics and Security* 20 (2025), pp. 822–838.

[6] Y. Zuo, B. Cox, L. Y. Chen, and J. Decouchant. "Spyker: Asynchronous Multi-Server Federated Learning for Geo-Distributed Clients". In: *Proceedings of the 25th International Middleware Conference*. 2024, pp. 367–378.

[7] B. Cox, A. Mălan, L. Y. Chen, and J. Decouchant. "Asynchronous byzantine federated learning". In: *arXiv preprint arXiv:2406.01438* (2024).

[8] P. Kairouz et al. "Advances and Open Problems in Federated Learning". In: *Foundations and Trends® in Machine Learning* 14.1–2 (June 22, 2021), pp. 1–210.

[9] F. Lai, Y. Dai, S. Singapuram, J. Liu, X. Zhu, H. Madhyastha, and M. Chowdhury. "FedScale: Benchmarking Model and System Performance of Federated Learning at Scale". In: *Proceedings of the 39th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, June 28, 2022, pp. 11814–11827.

[10] 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". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[11] B. Ying, K. Yuan, H. Hu, Y. Chen, and W. Yin. *BlueFog: Make Decentralized Algorithms Practical for Optimization and Deep Learning*. Nov. 8, 2021. arXiv: 2111.04287 [cs]. URL: http://arxiv.org/abs/2111.04287 (visited on 05/05/2025). Pre-published.

[12] X. Ma, J. Zhu, Z. Lin, S. Chen, and Y. Qin. "A State-of-the-Art Survey on Solving Non-IID Data in Federated Learning". In: *Future Generation Computer Systems* 135 (Oct. 1, 2022), pp. 244–258.

[13] J. Xu, C. Hong, J. Huang, L. Y. Chen, and J. Decouchant. "AGIC: Approximate gradient inversion attack on federated learning". In: *2022 41st International Symposium on Reliable Distributed Systems (SRDS)*. IEEE. 2022, pp. 12–22.

[14] A. Shankar, L. Y. Chen, J. Decouchant, D. Gkorou, and R. Hai. "Share Your Secrets for Privacy! Confidential Forecasting with Vertical Federated Learning". In: *arXiv preprint arXiv:2405.20761* (2024).

[15] A. Mălan, J. Decouchant, T. Guzella, and L. Chen. "CCBNet: Confidential Collaborative Bayesian Networks Inference". In: *arXiv preprint arXiv:2405.15055* (2024).

[16] R. Wang, X. Wang, H. Chen, J. Decouchant, S. Picek, N. Laoutaris, and K. Liang. "MUDGUARD: Taming Malicious Majorities in Federated Learning using Privacy-Preserving Byzantine-Robust Clustering". In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 8.3 (2024), pp. 1–41.

[17] R. Ormándi, I. Hegedűs, and M. Jelasity. "Gossip Learning with Linear Models on Fully Distributed Data". In: *Concurrency and Computation: Practice and Experience* 25.4 (2013), pp. 556–571.

[18] I. Hegedűs, G. Danner, and M. Jelasity. "Gossip Learning as a Decentralized Alternative to Federated Learning". In: *Distributed Applications and Interoperable Systems*. Cham: Springer International Publishing, 2019, pp. 74–90.

[19] L. Kong, T. Lin, A. Koloskova, M. Jaggi, and S. Stich. "Consensus Control for Decentralized Deep Learning". In: *Proceedings of the 38th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, July 1, 2021, pp. 5686–5696.

[20] L. Palmieri, C. Boldrini, L. Valerio, A. Passarella, and M. Conti. "Impact of Network Topology on the Performance of Decentralized Federated Learning". In: *Computer Networks* 253 (Nov. 1, 2024), p. 110681.

[21] X. Gao, W. Zhang, T. Chen, J. Yu, H. Q. V. Nguyen, and H. Yin. "Semantic-Aware Node Synthesis for Imbalanced Heterogeneous Information Networks". In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. CIKM '23. New York, NY, USA: Association for Computing Machinery, Oct. 21, 2023, pp. 545–555.

[22] B. L. Bars, A. Bellet, M. Tommasi, E. Lavoie, and A.-M. Kermarrec. "Refined Convergence and Topology Learning for Decentralized SGD with Heterogeneous Data". In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. International Conference on Artificial Intelligence and Statistics. PMLR, Apr. 11, 2023, pp. 1672–1702.

[23] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons. "The Non-IID Data Quagmire of Decentralized Machine Learning". In: *Proceedings of the 37th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, Nov. 21, 2020, pp. 4387–4398.

[24] B. Cox, L. Y. Chen, and J. Decouchant. "Aergia: leveraging heterogeneity in federated learning systems". In: *Proceedings of the 23rd ACM/IFIP International Middleware Conference*. 2022, pp. 107–120.

[25] Y.-C. Lin, J.-J. Kuo, W.-T. Chen, and J.-P. Sheu. "Reinforcement Based Communication Topology Construction for Decentralized Learning with Non-IID Data". In: *2021 IEEE Global Communications Conference (GLOBECOM)*. 2021 IEEE Global Communications Conference (GLOBECOM). Dec. 2021, pp. 1–6.

[26] M. De Vos, S. Farhadkhani, R. Guerraoui, A.-m. Kermarrec, R. Pires, and R. Sharma. "Epidemic Learning: Boosting Decentralized Learning with Randomized Communication". In: *Advances in Neural Information Processing Systems* 36 (Dec. 15, 2023), pp. 36132–36164.

[27] D. Menegatti, A. Giuseppi, C. Poli, and A. Pietrabissa. "Dynamic Topology Optimization for Efficient and Decentralised Federated Learning". In: *2024 IEEE International Conference on Big Data (BigData)*. 2024 IEEE International Conference on Big Data (BigData). Dec. 2024, pp. 7939–7945.

[28] R. Schollmeier. "A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications". In: *Proceedings First International Conference on Peer-to-Peer Computing*. First International Conference on Peer-to-Peer Computing. Aug. 2001, pp. 101–102.

[29] Eng Keong Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes". In: *IEEE Communications Surveys & Tutorials* 7.2 (Sum. 2005), pp. 72–93.

[30] S. B. Mokhtar, J. Decouchant, and V. Quéma. "Acting: Accurate freerider tracking in gossip". In: *2014 IEEE 33rd International Symposium on Reliable Distributed Systems*. IEEE. 2014, pp. 291–300.

[31] J. Decouchant, S. B. Mokhtar, A. Petit, and V. Quéma. "PAG: Private and accountable gossip". In: *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2016, pp. 35–44.

[32] D. Kempe, A. Dobra, and J. Gehrke. "Gossip-Based Computation of Aggregate Information". In: *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* 44th Annual IEEE Symposium on Foundations of Computer Science - FOCS 2003. Cambridge, MA, USA: IEEE Computer. Soc, 2003, pp. 482–491.

[33] L. Wang, Y. Xu, H. Xu, M. Chen, and L. Huang. "Accelerating Decentralized Federated Learning in Heterogeneous Edge Computing". In: *IEEE Transactions on Mobile Computing* 22.9 (Sept. 2023), pp. 5001–5016.

[34] M. Zhou, G. Liu, K. Lu, R. Mao, and H. Liao. "Accelerating the Decentralized Federated Learning via Manipulating Edges". In: *Proceedings of the ACM Web Conference 2024*. WWW '24. New York, NY, USA: Association for Computing Machinery, May 13, 2024, pp. 2945–2954.

[35] N. A. Tuan, A. Rizwan, S. J. S. Moe, A. N. Khan, and D. H. Kim. "DFL Topology Optimization Based on Peer Weighting Mechanism and Graph Neural Network in Digital Twin Platform". In: *Complex & Intelligent Systems* 11.6 (Apr. 22, 2025), p. 257.

[36] M. R. Behera and S. Chakraborty. "pFedGame - Decentralized Federated Learning Using Game Theory in Dynamic Topology". In: *2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. 2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS). Jan. 2024, pp. 651–655.

[37] S. Li, T. Zhou, X. Tian, and D. Tao. "Learning to Collaborate in Decentralized Learning of Personalized Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). June 2022, pp. 9756–9765.

[38] M. Assran, N. Loizou, N. Ballas, and M. Rabbat. "Stochastic Gradient Push for Distributed Deep Learning". In: *Proceedings of the 36th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, May 24, 2019, pp. 344–353.

[39] Z. Song, W. Li, K. Jin, L. Shi, M. Yan, W. Yin, and K. Yuan. "Communication-Efficient Topologies for Decentralized Learning with $O(1)$ Consensus Rate". In: *Advances in Neural Information Processing Systems* 35 (Dec. 6, 2022), pp. 1073–1085.

[40] Y. Dandi, A. Koloskova, M. Jaggi, and S. U. Stich. *Data-Heterogeneity-Aware Mixing for Decentralized Learning*. Apr. 13, 2022. arXiv: 2204.06477 [cs]. URL: http://arxiv.org/abs/2204.06477 (visited on 06/06/2025). Pre-published.

[41] S. A. Aketi, S. Kodge, and K. Roy. *Neighborhood Gradient Clustering: An Efficient Decentralized Learning Method for Non-IID Data Distributions*. Mar. 20, 2023. arXiv: 2209.14390 [cs]. URL: http://arxiv.org/abs/2209.14390 (visited on 04/21/2025). Pre-published.

[42] S. A. Aketi, S. Kodge, and K. Roy. "Neighborhood Gradient Mean: An Efficient Decentralized Learning Method for Non-IID Data". In: *Transactions on Machine Learning Research* (Aug. 29, 2023).

[43] H. Gao, M. Lee, G. Yu, and Z. Zhou. "A Graph Neural Network Based Decentralized Learning Scheme". In: *Sensors* 22.3 (3 Jan. 2022), p. 1030.

[44] Y. Esfandiari, S. Y. Tan, Z. Jiang, A. Balu, E. Herron, C. Hegde, and S. Sarkar. "Cross-Gradient Aggregation for Decentralized Learning from Non-IID Data". In: *Proceedings of the 38th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, July 1, 2021, pp. 3036–3046.

[45] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich. "A Unified Theory of Decentralized SGD with Changing Topology and Local Updates". In: *Proceedings of the 37th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, Nov. 21, 2020, pp. 5381–5393.

[46] B. Ying, K. Yuan, Y. Chen, H. Hu, P. A. N. PAN, and W. Yin. "Exponential Graph Is Provably Efficient for Decentralized

Deep Training". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 13975–13987.

[47] J. Wang, V. Tantia, N. Ballas, and M. Rabbat. "SlowMo: Improving Communication-Efficient Distributed SGD with Slow Momentum". In: International Conference on Learning Representations. Sept. 25, 2019.

[48] A. Dhasade, A.-M. Kermarrec, R. Pires, R. Sharma, and M. Vujasinovic. "Decentralized Learning Made Easy with DecentralizePy". In: *Proceedings of the 3rd Workshop on Machine Learning and Systems*. EuroMLSys '23: 3rd Workshop on Machine Learning and Systems. Rome Italy: ACM, May 8, 2023, pp. 34–41.

[49] E. L. Zec, T. Hagander, E. Ihre-Thomason, and S. Girdzijauskas. "On the Effects of Similarity Metrics in Decentralized Deep Learning under Distribution Shift". In: *Transactions on Machine Learning Research* (June 28, 2024).

[50] E. Schubert. "A Triangle Inequality for Cosine Similarity". In: *Similarity Search and Applications*. International Conference on Similarity Search and Applications. Springer, Cham, 2021, pp. 32–44.

[51] O. Arandjelovic. "Learnt Quasi-Transitive Similarity for Retrieval from Large Collections of Faces". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, June 2016, pp. 4883–4892.

[52] A. Krizhevsky, V. Nair, and G. Hinton. *The CIFAR-10 dataset*. 2009. URL: %5Curl%7Bhttps://www.cs.toronto.edu/~kriz/cifar.html%7D (visited on 05/19/2025).

[53] T.-M. H. Hsu, H. Qi, and M. Brown. *Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification*. Sept. 13, 2019. arXiv: 1909.06335 [cs]. URL: http://arxiv.org/abs/1909.06335 (visited on 05/19/2025). Pre-published.

[54] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar. *LEAF: A Benchmark for Federated Settings*. Dec. 9, 2019. arXiv: 1812.01097 [cs]. URL: http://arxiv.org/abs/1812.01097 (visited on 05/22/2025). Pre-published.

[55] Netherlands Code Committee. *Netherlands Code of Conduct for Research Integrity*. Published by the Association of Universities in the Netherlands (VSNU). 2018.

APPENDIX A
EXPERIMENTAL DETAILS AND FURTHER EVALUATION

TABLE II
OVERVIEW OF DATASETS, MODELS, AND TRAINING CONFIGURATIONS USED IN OUR EXPERIMENTS. FOR EACH DATASET, WE REPORT THE BATCH SIZE (B), THE NUMBER OF LOCAL TRAINING STEPS PER COMMUNICATION ROUND (R), AND THE LIST OF RANDOM SEEDS USED TO ENSURE REPRODUCIBILITY.

Dataset	Model	b	r	Learning Rate	Optimizer	Random Seeds
CIFAR-10	GN-LeNet	5	10	0.05	SGD	90, 180, 270, 360, 450
FEMNIST	CNN	5	10	0.05	SGD	90, 180, 270, 360, 450

A. Experiments on FEMNIST

TABLE III
TOP ACCURACY/LOSS AND EFFORT (IN ROUNDS AND BYTES) TO MATCH EL-LOCAL'S TOP ACCURACY ON FEMNIST USING DEGREE-3 TOPOLOGIES.

Method	Top Acc.	Top Loss	Rounds to Match	Bytes to Match (GB)
DissDL	86.30	0.39	1640	**33.22**
Epidemic	86.06	0.40	1920	38.94
Fully Connected	**86.90**	**0.38**	**1000**	101.41
Static MH	85.94	0.41	–	–

To load the FEMNIST dataset, we use the official LEAF preprocessing script with the settings `-s niid -sf 0.1 -k 0 -t sample`, which samples 10% of the data in a non-IID manner and splits each user's data into training and test sets. In this setup, each client corresponds to a unique writer, resulting in a natural non-IID distribution caused by differences in handwriting style, character frequency, and writing patterns.

This setup contrasts significantly with the form of heterogeneity introduced in CIFAR-10, where we apply Dirichlet-based partitioning with a concentration parameter $\alpha = 0.1$. The Dirichlet distribution induces label (class) heterogeneity, meaning that each client tends to have samples from a limited subset of classes, leading to highly imbalanced class distributions across nodes. In contrast, the LEAF-based partitioning used in FEMNIST leads to feature heterogeneity, where all clients may observe most classes, but the underlying input distributions (e.g., writing styles or character shapes) vary significantly. Feature heterogeneity is generally more subtle and less challenging.

For the FEMNIST experiments, we adopt the same experimental setup as in CIFAR-10, with the only difference being the number of communication rounds, which is reduced to 2000, due to the easier nature of the task.

Figure 5 presents the performance of all methods on FEMNIST, plotted over communication rounds. The fully connected topology performs best overall, achieving slightly higher accuracy and lower test loss than the other methods. However, it requires more than four times the amount of communication data, highlighting its inefficiency despite the performance gain.

DissDL performs slightly better than both EL-Local and Static MH. As shown in Table III, it requires $1.17\times$ fewer communication rounds and communication cost to reach the top-1 accuracy achieved by EL-Local, while surpassing it by 0.24 percentage points in final accuracy. As expected, the static graph performs the worst, though the differences among the methods are relatively small in this setting.

The most noticeable difference among the sparse methods lies in inter-node variance. After some initial communication rounds, as models begin to converge, EL-Local exhibits higher standard deviation compared to the other two methods. While its variance does not increase indefinitely and it still is relatively small, it remains marginally elevated compared to the near-zero variance of the other two methods. This suggests that EL-Local experiences slightly less consistent learning across nodes.

Overall, the narrow performance gap between methods can be attributed to the relative ease of the FEMNIST task and the nature of the heterogeneity. Since each client observes samples from most or all classes, the challenge introduced by data imbalance is significantly reduced, making it easier for all topologies to achieve good performance.

B. Detailed Results and Additional Graphs on CIFAR-10

TABLE IV
TOP ACCURACY/LOSS AND EFFORT (IN ROUNDS AND BYTES) TO MATCH EL-LOCAL'S TOP ACCURACY ON CIFAR-10 USING DEGREE-7 TOPOLOGIES.

Method	Top Acc.	Top Loss	Rounds to Match	Bytes to Match (GB)
DissDL	69.58	1.01	5960	**15.01**
EL-Local	68.89	1.10	8000	20.14
Fully Connected	**69.86**	**0.96**	**3360**	18.12
Static MH	64.56	1.34	–	–

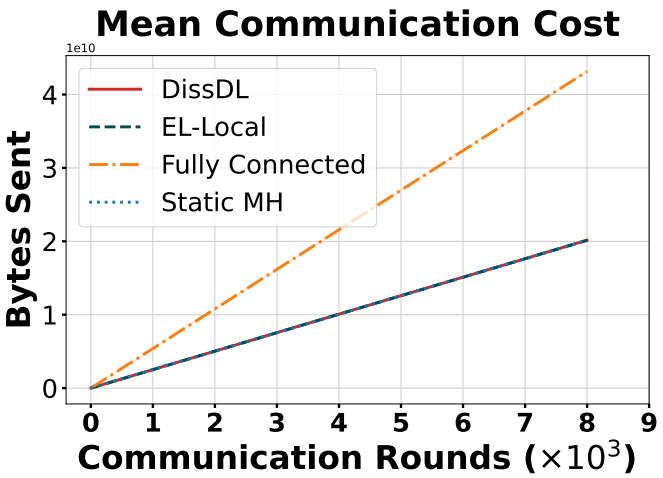

Fig. 3. Total communication cost per node (in bytes) on CIFAR-10 with 16 nodes in a non-IID setting using degree-7 topologies. **DissDL** refers to our dissimilarity-driven approach with a 7-regular initial topology; **EL-Local** applies the EL-Local algorithm with degree 7; **Fully Connected** uses a fully connected graph; and **Static MH** uses a static random 7-regular topology with Metropolis-Hastings averaging.

TABLE V
TOP ACCURACY/LOSS AND EFFORT (IN ROUNDS AND BYTES) TO MATCH EL-LOCAL'S TOP ACCURACY ON CIFAR-10 USING DEGREE-3 TOPOLOGIES.

Method	Top Acc.	Top Loss	Rounds to Match	Bytes to Match (GB)
DissDL	66.24	1.25	5920	**6.39**
EL-Local	64.80	1.41	7920	8.55
Fully Connected	**69.86**	**0.96**	**1520**	8.20
Static MH	55.47	1.97	–	–

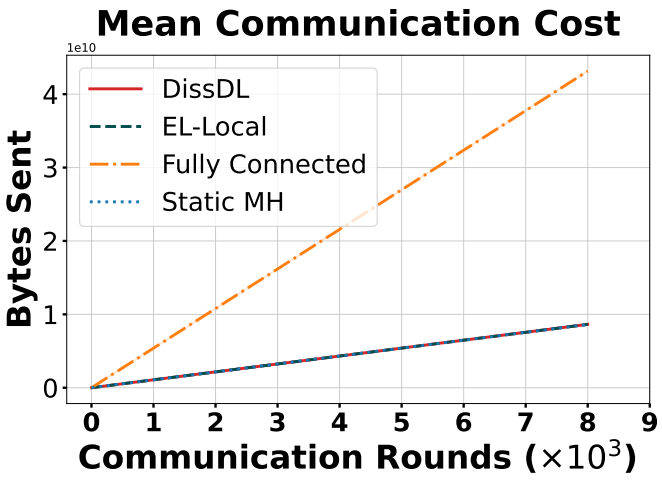

Fig. 4. Total communication cost per node (in bytes) on CIFAR-10 with 16 nodes in a non-IID setting using degree-3 topologies. **DissDL** refers to our dissimilarity-driven approach with a 3-regular initial topology; **EL-Local** applies the EL-Local algorithm with degree 3; **Fully Connected** uses a fully connected graph; and **Static MH** uses a static random 3-regular topology with Metropolis-Hastings averaging.

APPENDIX B
EMPIRICAL EVIDENCE OF IMPLICIT GLOBAL OPTIMIZATION

A. Similarity to Global Model Experiment

Measuring similarity to a global model over time helps quantify how well each node integrates distributed knowledge from the rest of the network. Higher similarity implies more effective parameter consensus and global coordination, which are often desired in decentralized optimization.

We define the global model at a given round as the average of all node models at that round:

$$\theta_t^{\text{global}} = \frac{1}{N} \sum_{i=1}^{N} \theta_t^i \tag{9}$$

where θ_t^i is the model of node i at round t, and N is the total number of nodes.

Each node's similarity to the global model is then computed using the average cosine similarity across all model layers:

$$\text{sim}(\theta_t^i, \theta_t^{\text{global}}) = \frac{1}{L} \sum_{\ell=1}^{L} \cos(\theta_{\ell,t}^i, \theta_{\ell,t}^{\text{global}}) \tag{10}$$

where L denotes the number of layers, and $\cos(\cdot, \cdot)$ represents the cosine similarity between flattened parameter vectors of a given layer.

This layer-wise averaging accounts for differences in layer sizes and ensures that all parts of the model contribute equally to the overall similarity.

To conduct this experiment, we modify the training loop to periodically save each node's model. After each evaluation round (every 20 rounds for the first 1000 iterations and every 40 rounds thereafter), all local models are collected, and the global model is computed as their arithmetic mean. Each node's similarity to this global model is then calculated and stored.

We run this setup under the same configurations as in the main experiments: CIFAR-10 with both 7-regular and 3-regular topologies. Each configuration is executed with five different random seeds to account for initialization variability. We omit FEMNIST from this analysis, as its relative ease and the nature of its feature heterogeneity, being generally more subtle and less challenging, are unlikely to provide additional insight into the effects of similarity-based peer selection.

After training, we extract the similarity scores from each node's log directory, aggregate them by communication round, and compute the mean and standard deviation across all nodes and seeds. This produces a time series showing how the average similarity evolves, with shaded regions representing inter-node variability.

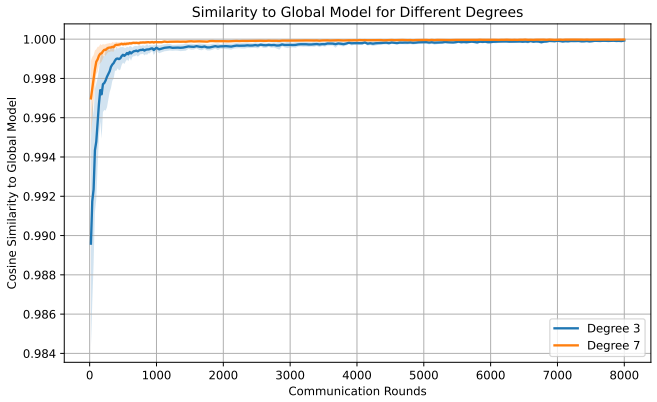

Fig. 6. Cosine similarity to the global model over communication rounds for CIFAR-10 under degree-3 and degree-7 topologies. Curves show the mean similarity across five runs per setup; shaded areas indicate standard deviation across all nodes and runs.

Figure 6 shows the results for CIFAR-10 under both degree-3 and degree-7 topologies. In both cases, the cosine similarity between each node's model and the global model starts high and continues to improve steadily over time. Degree-7 begins with slightly higher alignment and converges more quickly due to its denser connectivity. This similarity-to-global metric provides insights into convergence behavior and network-wide agreement: higher values typically suggest better coordination and alignment with global optimization objectives.

However, in our case, the global model is constructed by aggregating the partial models generated by our own algorithm, rather than referencing an external target. As a result, while the high similarity may reflect effective collaboration, it could

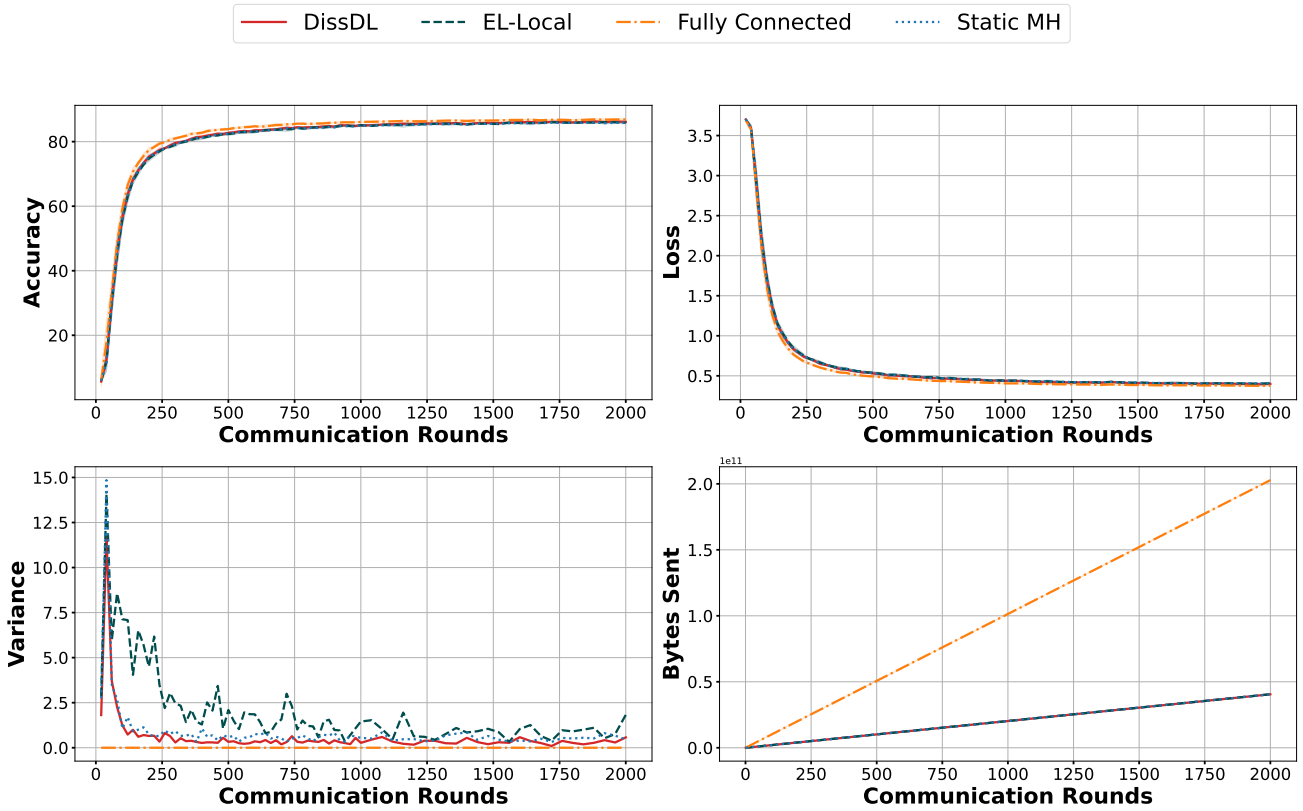

Fig. 5. Performance metrics on FEMNIST with 16 nodes in a non-IID setting using degree-3 topologies. The figure shows, from left to right, top to bottom: **(1)** mean test accuracy over communication rounds (with shaded regions indicating standard deviation), **(2)** mean test loss, **(3)** inter-node variance as a measure of convergence stability, and **(4)** total communication cost per node (in bytes). **DissDL** refers to our dissimilarity-driven approach with a 3-regular initial topology; **EL-Local** applies the EL-Local algorithm with degree 3; **Fully Connected** uses a fully connected graph; and **Static MH** uses a static random 3-regular topology with Metropolis-Hastings averaging.

also represent a self-reinforcing artifact of the system itself. Still, the upward trend and narrowing standard deviation in both configurations suggest that dissimilarity-based peer selection promotes stronger convergence and consensus over time. Nevertheless, this metric is not strong enough on its own to conclude our intuition.

To address this limitation, we also evaluate whether nodes are implicitly exposed to the global distribution by tracking how their indirect exposure to label distributions evolve. This second experiment uses a static global reference and serves to validate whether the topological adaptations promote actual data diversity and global optimization rather than just parameter alignment.

B. Similarity to Global Data Distribution Experiment

To assess whether local peer selection promotes global alignment under data heterogeneity, we estimate how closely each node's cumulative training exposure reflects the global data distribution over time.

Let $\mathbf{g} \in \mathbb{R}^C$ denote the global data distribution across C classes, and let $\mathbf{d}_i \in \mathbb{R}^C$ be the static class distribution of node i. For each round t, we define an effective training distribution for node i, denoted $\tilde{\mathbf{d}}_i^{(t)}$, as the weighted sum:

$$\tilde{\mathbf{d}}_i^{(t)} = t \cdot \mathbf{d}_i + \sum_{j \neq i} \alpha_{ij}^{(t)} \cdot \mathbf{d}_j, \tag{11}$$

where $\alpha_{ij}^{(t)}$ is the cumulative count of how often node i has received model updates from peer j up to round t.

We then compute the cosine similarity between each $\tilde{\mathbf{d}}_i^{(t)}$ and the global distribution $\mathbf{g}$ at every round:

$$\text{sim}_i^{(t)} = \cos\left(\tilde{\mathbf{d}}_i^{(t)}, \mathbf{g}\right). \tag{12}$$

This similarity serves as a proxy for how globally representative node i's training exposure has become by round t.

For this experiment, we focus on the CIFAR-10 dataset with 7- and 3-regular topologies, each run with five different random seeds. We omit FEMNIST, as its relative simplicity and milder heterogeneity make it less informative for evaluating global alignment dynamics.

Table VI quantifies the improvement in alignment with the global data distribution across five runs on CIFAR-10 using a degree 3 graph. All runs start with moderate cosine similarity values and consistently improve by around 0.27–0.31, reaching high final similarities above 0.91.

Figure 7 visualizes this trend across communication rounds.

TABLE VI

PER-RUN SIMILARITY IMPROVEMENTS FOR CIFAR-10 WITH DEGREE-3 TOPOLOGY. WE REPORT THE MEAN COSINE SIMILARITY TO THE GLOBAL DATA DISTRIBUTION AT THE START AND END OF TRAINING, AS WELL AS THE AVERAGE IMPROVEMENT. MIN FINAL AND MAX FINAL REFER TO THE LOWEST AND HIGHEST FINAL SIMILARITY ACROSS ALL NODES WITHIN EACH RUN, INDICATING THE SPREAD OF ALIGNMENT OUTCOMES.

Run ID	Start Sim.	Final Sim.	Avg Impr.	Min Final	Max Final
run_1	0.632	0.926	0.294	0.806	0.993
run_2	0.619	0.927	0.308	0.828	0.994
run_3	0.642	0.914	0.272	0.812	0.988
run_4	0.624	0.914	0.291	0.794	0.989
run_5	0.656	0.931	0.275	0.802	0.994

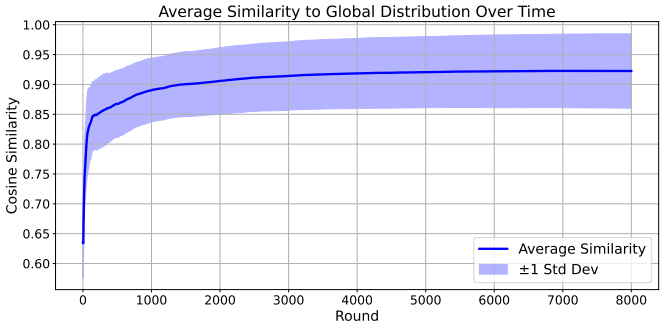

Fig. 7. Average cosine similarity of nodes' aggregated data distributions to the global data distribution over time for CIFAR-10 (degree-3). the shaded area represents standard deviation, capturing both inter-node and inter-run variability across 5 runs.

The curve shows the mean cosine similarity to the global distribution, averaged across all nodes and all runs. The shaded area represents the standard deviation over this set, i.e., how much individual nodes (across five random seeds) differ from the average at each round. This variability captures both inter-node and inter-run divergence, and its relatively narrow range ($\approx \pm\ 0.05$) indicates that the alignment process is robust and consistent across different initializations and data splits. After less than 1500 rounds, nodes reach an average similarity to the global distribution of over 0.9, which is relatively high.

TABLE VII

PER-RUN SIMILARITY IMPROVEMENTS FOR CIFAR-10 WITH DEGREE-7 TOPOLOGY. WE REPORT THE MEAN COSINE SIMILARITY TO THE GLOBAL DATA DISTRIBUTION AT THE START AND END OF TRAINING, AS WELL AS THE AVERAGE IMPROVEMENT. MIN FINAL AND MAX FINAL REFER TO THE LOWEST AND HIGHEST FINAL SIMILARITY ACROSS ALL NODES WITHIN EACH RUN, INDICATING THE SPREAD OF ALIGNMENT OUTCOMES.

Run ID	Start Sim.	Final Sim.	Avg Impr.	Min Final	Max Final
run_1	0.810	0.986	0.176	0.971	0.996
run_2	0.830	0.986	0.156	0.969	0.998
run_3	0.819	0.985	0.166	0.970	0.995
run_4	0.805	0.982	0.177	0.962	0.993
run_5	0.834	0.986	0.152	0.965	0.997

Table VII summarizes the similarity improvements for CIFAR-10 with a degree-7 graph. Compared to the degree-3 case, all runs start with relatively high cosine similarity values

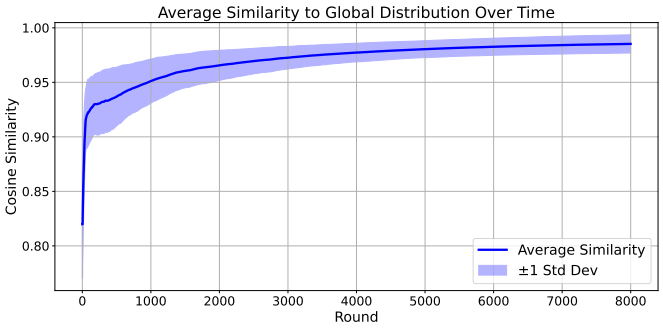

Fig. 8. Average cosine similarity of nodes' aggregated data distributions to the global data distribution over time for CIFAR-10 (degree-7). The shaded area represents standard deviation, capturing both inter-node and inter-run variability across 5 runs.

(≈ 0.81–0.83), reflecting both the increased initial connectivity and the fact that measurements begin at round 20, after each node has already selected a few informative peers. Across all five runs, similarity continues to increase throughout training, reaching very high final values above 0.99 and approaching the perfect 1.0. Notably, the lowest final similarity within any run still exceeds 0.96, indicating that all nodes achieve strong alignment with the global distribution.

Figure 8 illustrates this progression over time. The curve depicts the mean similarity to the global distribution, averaged across all nodes and all runs. The shaded area shows the standard deviation, reflecting how much individual nodes diverge from the mean at each round. Despite the already strong starting alignment, the similarity continues to rise across rounds, showing that nodes benefit from adaptively selecting dissimilar peers. Toward the later rounds, the standard deviation narrows significantly, suggesting stable and consistent convergence behavior across different runs and nodes.

These results support our intuition that the proposed peer selection strategy promotes diverse data exposure and better approximation of the global data distribution. By selectively receiving models from dissimilar peers, nodes are effectively exposed to a more representative subset of the global data. This leads to improved global model quality, approaching the performance one would expect in a fully connected network, where each node has indirect access to the entire data distribution through complete mixing.