Teleportation Links: Mitigating Catastrophic Forgetting in Decentralized Federated Learning

Xu Wang, *Student Member, IEEE*, Yuanzhu Chen, *Senior Member, IEEE*, Qiang Ye, *Senior Member, IEEE*, and Octavia A. Dobre, *Fellow, IEEE*

***Abstract*—Decentralized approaches are inspired by the self-organizing principles observed in natural and social systems. These methods offer a scalable and resilient framework for collaborative learning. Decentralized federated learning (DFL) uses these principles to avoid the dependency on a central controller. However, a key challenge arises from data heterogeneity. This often leads to catastrophic forgetting, where the model significantly loses its ability to remember and use previously learned information. This loss of knowledge can reduce the reliability of DFL in critical applications, such as autonomous systems, financial services, energy management, and transportation networks. To tackle these challenges, in this paper, we investigate how data distribution affects DFL performance, with a focus on how bias propagates across nodes. We also explore how local model learning rates affect the trade-off between learning stability and convergence speed. At the same time, we evaluate the performance drops at individual nodes. Furthermore, to improve connectivity and speed up knowledge sharing, we propose adding a limited number of teleportation links, which aims to reduce the average distance between pairs of nodes. Extensive experimental results demonstrate the effectiveness of this strategy, showing reduced catastrophic forgetting, faster convergence, and improved resilience across various scenarios.**

***Index Terms*—Decentralized federated learning, network topology, connectivity, teleportation links.**

I. Introduction

COMPLEX natural systems, such as flocks of birds [1], ant colonies [2], neural networks in the brain [3], and ecosystems [4], operate efficiently without centralized control. These systems are decentralized, adaptive, and capable of achieving global objectives through simple local interactions [5], [6]. For example, in a flock of birds, there is no single leader; instead, each bird follows a set of simple rules based on its neighbors: collision avoidance to steer away from crowding local flockmates, velocity matching to attempt to match speed and direction with nearby flockmates, and flock centering to attempt to stay close to nearby flockmates. This local coordination allows the flock to move as a unified entity, achieving a global objective without centralized control. This directly parallels the way decentralized learning models operate. Just as natural systems achieve global coordination through local interactions, decentralized learning offers a compelling paradigm for modern machine learning, especially for scaling algorithms to manage large datasets distributed across many devices [7], [8]. In these systems, local model updates and peer-to-peer data sharing among devices enable a global objective to be achieved without a central server [9]. Furthermore, this paradigm enhances data privacy in sensitive domains, such as healthcare and finance, while improving model robustness in dynamic environments [10], [11], [12].

The demand for decentralized collaboration is continually growing in domains such as smart cities [13], [14], autonomous systems [15], [16], and personalized healthcare [17], [18], [19]. These applications require solutions emphasizing privacy, scalability, and resilience, thereby enabling adaptation to complex and dynamic situations. Federated learning (FL) has emerged as a key solution for privacy-preserving and collaborative learning in data-driven systems [20], [21]. FL enables multiple parties to collaboratively train models without sharing sensitive data, thereby facilitating secure and efficient learning across decentralized networks [22]. However, its reliance on a central server introduces several limitations, including potential performance bottlenecks, single points of failure, and scalability challenges [23]. Decentralized federated learning (DFL) fundamentally addresses this core architectural limitation by eliminating the central server altogether. Through peer-to-peer collaboration, DFL bypasses the need for a central orchestrator, thereby enhancing both scalability and resilience [24], [25].

Despite its promise, DFL faces a major challenge: data heterogeneity. In real-world decentralized systems, the data across nodes is typically non-independent and identically distributed (non-i.i.d.) [26], [27]. This non-i.i.d. nature complicates the training process and directly leads to a critical issue known as catastrophic forgetting. In this phenomenon, a model trained on data from one node catastrophically overwrites previously learned knowledge when subsequently exposed to disparate data from another node [28]. This phenomenon makes it difficult for the system to synthesize knowledge across all participating nodes, posing a significant challenge to achieving coherent and effective decentralized learning.

Received 8 June 2025; revised 27 August 2025; accepted 14 September 2025. Date of publication 17 September 2025; date of current version 11 December 2025. The work of Xu Wang and Yuanzhu Chen was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Program under Grant RGPIN-2017-05201. The work of Octavia A. Dobre was supported by NSERC Discovery Program under Grant RGPIN-2019-04123. An earlier version of this paper was presented in part at the 2025 IEEE Conference on Computer Communications Workshops [DOI: 10.1109/INFOCOMWKSHPS65812.2025]. Recommended for acceptance by Dr. Yuan Wu. *(Corresponding author: Yuanzhu Chen.)*

Xu Wang and Yuanzhu Chen are with the School of Computing, Queen's University, Kingston ON K7L3N6, Canada (e-mail: yuanzhu.chen@queensu.ca).

Qiang Ye is with the Department of Electrical and Software Engineering, University of Calgary, Calgary, AB T2N1N4, Canada (e-mail: qiang.ye@ucalgary.ca).

Octavia A. Dobre is with the Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's NL A1C5S7, Canada.

Digital Object Identifier 10.1109/TNSE.2025.3611109

To address the challenge of catastrophic forgetting and slow information propagation within decentralized learning environments, we propose the strategic implementation of teleportation links. These links emulate the efficient long-range connections observed in natural systems. By design, they shorten communication paths, accelerate information flow, and enhance knowledge retention across nodes. These teleportation links draw inspiration from the efficiency of small-world networks. In these networks, a small number of long-range links or "shortcuts" dramatically reduce the average path length between any two nodes, thereby promoting efficient information flow [29], [30], [31]. For instance, social networks rapidly disseminate information through a few key connections that bridge distant communities [32], [33]. Similarly, metabolic networks optimize chemical reactions by using short pathways to connect otherwise distant metabolites [34], [35]. Pollinator networks improve resource distribution in ecosystems by connecting otherwise isolated plant species [36], [37]. By emulating these natural connectivity patterns, our proposed teleportation links provide logical shortcuts that directly connect distant nodes, bypassing multiple intermediate network segments. This mechanism significantly enhances overall network efficiency and learning effectiveness in a manner analogous to these real-world systems. Operating at the transport layer, teleportation links provide logical shortcuts that directly connect distant nodes, thereby bypassing intermediate network segments. In this work, we employ a genetic algorithm (GA) to strategically place teleportation links that minimize communication paths and enhance information flow across the network [38].

Our main contributions are as follows: (1) To address the challenge of data heterogeneity and its role in catastrophic forgetting, we investigate how varying data distributions affect the stability and efficiency of DFL, focusing on the propagation of biases across interconnected nodes. (2) To mitigate the negative impact of heterogeneous data and enhance convergence, we explore how local learning rates influence overall system performance, focusing on their effect on stability and efficiency, especially the performance drops observed at certain nodes. (3) To overcome the limitations of poor network connectivity and slow information flow, we introduce a method for incorporating a minimal number of teleportation links to enhance network connectivity, demonstrating how this strategic addition accelerates convergence without compromising system reliability.

The rest of the paper is organized as follows. Section II reviews the related work. Section III details the methodology for mitigating DFL catastrophic forgetting via optimized teleportation links. Section IV evaluates the performance of our proposed method. In Section V, we present concluding remarks, limitations and future research.

II. Related Work

This section reviews key literature relevant to our research concerning teleportation links in DFL. We begin by examining the persistent challenge of catastrophic forgetting, which refers to the tendency of local models to adapt to new knowledge while forgetting previously learned information. We then discuss the critical role of network topology in DFL, reviewing various network architectures and optimization methods. We conclude by identifying a gap between these two areas and motivating the need for topological solutions that can simultaneously mitigate catastrophic forgetting and promote fast, stable convergence.

A. Catastrophic Forgetting in Learning Systems

The phenomenon of catastrophic forgetting was first identified in 1989 [39]. It refers to a critical challenge in machine learning models, particularly neural networks. Such models exhibit a significant decline in performance concerning previously learned tasks after training on new tasks [40]. This degradation occurs primarily because new training updates the model weights that were optimized for earlier tasks [41], [42].

In centralized learning, several strategies address catastrophic forgetting. Regularization-based methods, such as elastic weight consolidation [43] and synaptic intelligence [44], add penalty terms to protect important weights. Replay-based methods, including experience replay [45] and generative replay [46], mix stored or generated past data with new training data. Architectural methods, like progressive neural networks [47] and dynamic expandable networks [48], adapt model structure to retain previously learned knowledge.

While these strategies are effective in centralized settings, federated learning presents unique characteristics. Its decentralized data distribution and communication constraints introduce complexities that necessitate specialized approaches. GradMA addresses this by using gradient memory on both server and client sides to stabilize updates [49]. FedReg introduces pseudo-data to regularize local training and preserve past performance [50]. The federated global twin generator trains privacy-preserving generative models on the server to produce synthetic data covering all class labels, helping clients retain old knowledge while learning new tasks [51].

In DFL, the absence of a central server and the reliance on peer-to-peer communication make catastrophic forgetting more difficult to address. One approach uses mutual knowledge transfer among nodes, which avoids simple model averaging and improves knowledge retention [52]. Despite recent advances, catastrophic forgetting remains a core challenge in DFL, largely due to limited and uneven knowledge exchange across nodes. This issue is closely tied to the network topology, which determines how information propagates, how quickly models converge, and how stable the learning process is. Therefore, understanding and optimizing network topology is essential for developing DFL systems that achieve fast, stable convergence and effectively mitigate catastrophic forgetting.

B. Network Topology Design in Decentralized Federated Learning

In DFL, network topology plays a crucial role by shaping the dynamics of information exchange between nodes. Such dynamics subsequently impact convergence speed, training stability, and the magnitude of catastrophic forgetting. Given this significant impact, researchers have investigated various network architectures suitable for DFL.

Several foundational graph structures have been explored as potential network topologies. In a fully connected topology, each node communicates with all others, enabling maximum information exchange and redundancy [10]. However, its communication cost grows quadratically with network size, limiting

scalability. In contrast, a ring topology connects each node to two neighbors, reducing per-node communication while supporting continuous information flow [53].

Specialized topologies like the hypercube offer a balance between connectivity and node degree. Each node represents a corner of an n-dimensional hypercube and connects to its immediate neighbors. When combined with data-similarity-based neighbor selection, this structure helps mitigate data heterogeneity and speeds up training [54]. Tree and hierarchical topologies also play a key role. While often used in hierarchical federated learning with intermediate servers, decentralized versions can support layered communication in DFL [55].

Several foundational graph structures, such as Erdős–R'enyi, small-world, and scale-free have been evaluated for decentralized training. The study in [56] found that clustered networks are generally more effective, with small-world graphs converging faster in small-scale settings and scale-free networks performing better at large scale. While scale-free structures benefit from hierarchy, they also introduce node load imbalance and vulnerability to failures. Expander graphs have shown strong potential, offering sparse yet highly connected topologies. DFL systems using d-regular expanders achieve faster convergence and greater resilience than rings or Erdős–R'enyi graphs, with lower communication costs [57].

In dynamic DFL environments, static topologies often result in slow convergence due to inefficient peer-to-peer connections. To address this issue, prior work [58] proposes an efficient algorithm to accelerate DFL by strategically introducing a limited number of additional edges. This work establishes a crucial link between the convergence rate and the second smallest eigenvalue of the Laplacian matrix of the graph, thereby providing a theoretical basis for such augmentations. Energy efficiency is another critical concern; thus, an energy-aware topology optimization algorithm proposed in [59] aims to maximize algebraic connectivity while adhering to energy and communication constraints.

Furthermore, the dynamic nature of network bandwidth and heterogeneous device capacities necessitates adaptive topologies. Greedy algorithms in [60], [61] reconstruct network structures based on node resources, enabling responsiveness to changing conditions. To further improve efficiency, dynamic link management adjusts connections based on latency, and [62] proposes pruning redundant links to form sparser, low-latency topologies.

C. Motivation for Teleportation Links in DFL

Mitigating catastrophic forgetting and ensuring stable convergence remain key challenges in DFL. Existing approaches to address forgetting mainly focus on algorithmic changes or data-level interventions. Although these methods provide partial solutions, they often overlook the influence of network topology in facilitating knowledge retention and transfer. In contrast, recent work on topology optimization, such as strategic edge additions [58], has shown improvements in convergence speed. However, such techniques typically do not target catastrophic forgetting. This reveals a gap and motivates the need for topological enhancements that can support both faster convergence and long-term knowledge preservation through teleportation links in DFL.

Our work on teleportation links addresses this identified gap. We propose a method that strategically introduces long-range connections to facilitate efficient information flow across the network. These links are primarily designed to mitigate catastrophic forgetting by enhancing the propagation and reinforcement of learned knowledge. By promoting a more consistent global understanding across nodes, teleportation links also contribute to faster and more stable convergence. A summary of existing methods, along with their key characteristics and limitations, is provided in Table I.

III. Mitigating Catastrophic Forgetting in DFL Through Teleportation Links

Catastrophic forgetting presents significant challenges in DFL, as it undermines the accumulation of global knowledge and reduces overall effectiveness. The fundamental problem stems from poor connectivity and inefficient communication across the network. These limitations, particularly when combined with non-i.i.d. data and decentralized updates, lead to delays and inconsistencies in information exchange. Consequently, learning becomes isolated, models struggle to generalize effectively, and the integration of knowledge across nodes is impaired.

A. Impact of Network Connectivity on Catastrophic Forgetting

Poor connectivity and inefficient communication contribute to catastrophic forgetting through several mechanisms. First, local specialization can occur when nodes train exclusively on their local datasets, a scenario especially common in non-i.i.d. settings. Such specialization can cause models to overfit to local data, preventing them from generalizing well across diverse data distributions. Second, delayed updates occur when distant nodes are unable to exchange information efficiently, which can lead to inconsistencies and knowledge gaps throughout the network. Third, conflicting updates, especially those originating from non-i.i.d. datasets, can further destabilize the global model by introducing incompatible parameter changes. Collectively, these disruptions degrade the performance of the model, destabilize parameters, and slow global convergence. This situation forces the network into frequent and inefficient relearning cycles, which in turn increase training time and reduce learning efficiency.

B. Connectivity Metrics for DFL Networks

Strong network connectivity enables timely and consistent knowledge exchange, thereby lowering the risk that outdated updates will lead to catastrophic forgetting. When nodes are well connected, the global model can integrate information more efficiently, which reduces inconsistencies and enhances stability. Several graph metrics can be employed to assess and enhance network connectivity. For instance, the clustering coefficient indicates the extent to which nodes form local clusters, while the k-core number reveals resilient subnetworks composed of highly interconnected nodes [29], [63], [64]. The average pairwise distance, on the other hand, measures the efficiency of global information flow across the network [65], [66]. While each of these metrics reflects a different aspect of the network structure, some are more suited to specific types of analysis. In this work,

TABLE I
SUMMARY OF METHODS ADDRESSING CATASTROPHIC FORGETTING AND TOPOLOGY DESIGN IN DFL

Method	Characteristics	Limitations
Elastic Weight Consolidation [43]	Regularization to preserve important weights.	Requires task identity; limited for DFL.
Synaptic Intelligence [44]	Tracks weight importance during training.	Centralized setting; limited DFL applicability.
Experience Replay [45]	Replays stored past data with new tasks.	Conflicts with FL privacy.
Generative Replay [46]	Generates pseudo-data to mimic old tasks.	High overhead; generator training complexity.
Progressive Neural Networks [47]	Adds subnetworks for new tasks.	Poor scalability.
Dynamic Expandable Networks [48]	Adds task-specific components.	Increases model size and complexity.
GradMA [49]	Gradient memory on server and clients.	Requires coordination and storage.
FedReg [50]	Pseudo-data for regularization.	Depends on data quality.
Twin Generator [51]	Server-generated synthetic data.	Training cost; privacy risk.
Mutual Knowledge Transfer [52]	Peer-to-peer knowledge transfer.	Sensitive to network structure.
Fully Connected Topology [10]	Full communication among nodes.	Poor scalability.
Ring Topology [53]	Neighbor-based communication.	Slow convergence.
Hypercube [54]	Connects similar nodes in high dimensions.	Grows complex with dimension.
Tree/Hierarchical [55]	Layered node hierarchy.	Fragile under failure.
Clustered Networks [56]	Small-world/scale-free graphs.	Hub overload risk.
Expander Graphs [57]	Sparse but well-connected.	Hard to design.
Edge Augmentation [58]	Adds links to boost convergence.	Ignores catastrophic forgetting.
Energy-Aware Topology [59]	Optimizes algebraic connectivity.	May neglect learning goals.
Greedy/Dynamic Topology [60], [61]	Adapts to device resources.	Requires real-time resource information.
Teleportation Links (Ours)	Adds long-range links for knowledge reinforcement.	Requires careful design balance.

we focus on the average pairwise distance as the primary metric for evaluating and optimizing global communication efficiency. This metric is chosen because it directly quantifies how information propagates across the entire network. The average pairwise distance is computed as the mean of the shortest path lengths among all pairs of nodes in a connected, undirected graph $G = (V, E)$ with n nodes. Let $d_{i,j}$ denote the shortest distance between nodes v_i and v_j. Then, the average pairwise distance, d, is calculated as:

$$\langle d \rangle = \frac{2}{n(n-1)} \sum_{1 \le i < j \le n} d_{i,j}. \tag{1}$$

A smaller value of $\langle d \rangle$ signifies potentially faster information propagation throughout the network, thus indicating better global communication efficiency.

C. Enhancing Connectivity With Teleportation Links

Global efficiency and rapid knowledge dissemination are key objectives in DFL. Minimizing the average pairwise distance across the network is crucial for achieving these objectives, as shorter distances facilitate faster integration of updates and accelerate convergence. To significantly reduce this average pairwise distance and thereby enhance network connectivity, we propose the strategic addition of teleportation links. These links act as logical shortcuts, creating direct, near-instant communication paths between selected pairs of nodes, regardless of their separation in the underlying physical network. Thus, this approach effectively alleviates bottlenecks in connectivity.

Identifying the optimal set of teleportation links from all possible combinations presents a combinatorially complex search problem. The quantity of potential new links in a network of N nodes can be very large, as it is determined by the $N(N-1)/2$ total possible pairs minus the number of links already present. Exhaustively evaluating all subsets of these potential links is computationally infeasible for networks of non-trivial size.

D. Optimizing Teleportation Link Placement Using a Genetic Algorithm

To address this challenge, we employ a genetic algorithm [67], [68], [69]. The genetic algorithm was chosen for its advantages over heuristic and deterministic approaches. Heuristics, such as linking nodes with the highest degree or betweenness, are fast but limited to local improvements and perform poorly on complex, non-linear problems. Deterministic methods, like integer linear programming or convex optimization, can yield exact solutions but require strong assumptions and become intractable for large networks. In contrast, GA searches the global solution space, avoids local optima, and offers interpretability by showing how link placements evolve over generations. Because of these strengths, GA is a robust and suitable method for optimizing teleportation link placement. The algorithm evaluates node pairs to identify configurations that enhance connectivity by minimizing the average pairwise distance across the network in DFL. The GA process for optimizing the placement of teleportation links involves several key elements:

1) Chromosome Representation: Each potential solution, representing a specific set of teleportation links, is encoded as an individual, also termed a chromosome. Potential sets of teleportation links (pairs of nodes) are represented as binary configurations within the GA. We represent this using a binary string of length L, where L denotes the total number of potential teleportation links in the network (e.g., pairs of nodes that are not directly connected).

- *Chromosome (C):* A chromosome C is a binary vector $C = [g_1, g_2, \ldots, g_L]$.
- *Gene (g_i):* Each element g_i in the chromosome is a gene, where $g_i \in \{0, 1\}$.
- *Active Link:* If $g_i = 1$, the i-th potential link is selected as an active teleportation link. To enforce a limit on the number of added links, we restrict the total count of "1"s in each individual to the total number of teleportation links.
- *Inactive Link:* If $g_i = 0$, the i-th potential link is not selected.

For example, if there are $L = 500$ potential locations where teleportation links could be added, each chromosome would be a binary string of length 500.

2) Population Initialization: The genetic algorithm starts by generating an initial population, denoted as P_0, consisting of M chromosomes (individuals). This population is typically created randomly to ensure a diverse set of starting solutions:

$$P_0 = \{C_1, C_2, \ldots, C_M\}. \tag{2}$$

3) Fitness Evaluation: To minimize the average pairwise distance $\langle d \rangle$, the fitness function $f(C)$ evaluates each chromosome C_j, which represents a configuration of added teleportation links. Since the GA is designed to minimize the objective function $f(C)$, this fitness value is set directly as the calculated $\langle d \rangle$ for the network configuration represented by the chromosome. The search process, therefore, favors configurations that result in lower $\langle d \rangle$ values.

The evaluation process for a given chromosome C_j involves the following steps:

1) The set of active teleportation links encoded by chromosome C_j is temporarily added to the base network graph G, thereby creating an augmented graph G'.
2) The average pairwise distance, denoted as $\langle d \rangle_{G'}$, is then calculated for the augmented graph G'.
3) The fitness $f(C_j)$ is defined as the computed average pairwise distance. Since the GA aims to minimize this value, the fitness directly reflects the optimization objective:

$$f(C_j) = \langle d \rangle_{G'}. \tag{3}$$

Consequently, the selection mechanism of the GA favors chromosomes that yield lower fitness values $f(C_j)$. In other words, it prioritizes configurations C_j that result in smaller average pairwise distances $\langle d \rangle_{G'}$, thereby guiding the evolutionary search toward solutions that effectively enhance network connectivity by minimizing $\langle d \rangle$.

4) Genetic Operators: Following the selection process, the genetic algorithm applies evolutionary operations to generate the next generation of solutions. Through iterative operations such as crossover and mutation, the algorithm refines the population of potential solutions.

- *Mutation:* This operator introduces variation by disabling one existing teleportation link and enabling a new one when mutation occurs. Each individual has a small mutation probability p_m. Specifically, when mutation is triggered, we randomly select a bit set to 1 and change it to 0 to deactivate that link; then we randomly select a bit set to 0 and change it to 1 to activate a different link. This process maintains genetic diversity, explores new regions of the solution space, and prevents premature convergence to suboptimal solutions.
- *Crossover:* This operator combines genetic material from two parent chromosomes to generate new offspring. We implement a single-point crossover strategy, where a random point is selected along the chromosome, and segments beyond this point are exchanged between the two parents. After generating each offspring, we verify that the total number of teleportation links (bits set to 1) still matches the required constraint; if the count does not match, the offspring is discarded and we move on. This check ensures that all offspring remain valid and that the algorithm continues exploiting high-quality solutions without violating the link-count constraint.

5) Selection: Parents for the subsequent generation are selected based on their fitness values. Individuals that achieve lower average pairwise distances are assigned higher selection probabilities and are more likely to contribute to the next generation. Roulette wheel selection is employed for this selection process.

In roulette wheel selection, the probability p_j of selecting a particular chromosome C_j is proportional to its fitness score. This probability is typically calculated as:

$$p_j = \frac{f_{\text{sel}}(C_j)}{\sum_{k=1}^{M} f_{\text{sel}}(C_k)}. \tag{4}$$

Here, M is the total number of individuals in the population and $f_{\text{sel}}(C_j)$ is the fitness value of chromosome j, transformed specifically for the selection process.

When the GA aims to minimize an objective function, such as $f(C_j) = \langle d \rangle_{G'}$, the standard roulette wheel selection requires a transformation of $f(C_j)$. The selection fitness $f_{\text{sel}}(C_j)$ used in the formula must be positive and defined such that higher values indicate better solutions, which correspond to smaller values of $\langle d \rangle_{G'}$. For instance, if $f(C_j) > 0$, the selection fitness $f_{\text{sel}}(C_j)$ can be computed as $f_{\text{sel}}(C_j) = 1/f(C_j)$. This transformation ensures that individuals with lower original $f(C_j)$ values, which represent better solutions in a minimization problem, are more likely to be selected.

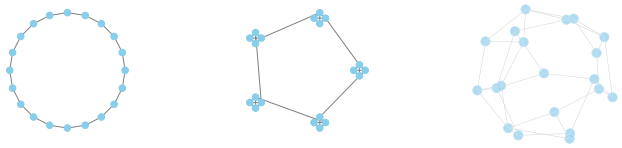
6) Creating the Next Generation and Elitism: The offspring generated through crossover and mutation form the basis of the subsequent generation. We also employ an elitism strategy, where a small number of the best-fitting individuals (e.g., the top 5) from the current generation are carried over directly to the next generation. This practice ensures that high-quality solutions are preserved and not lost during the evolutionary process.

7) Termination Criteria: The evolutionary process includes evaluation, selection, crossover, and mutation. It terminates when a predefined condition is met, such as reaching a maximum number of generations or observing no significant improvement in fitness over several consecutive generations. The chromosome that exhibits the best fitness score upon termination is considered the near-optimal set of teleportation links identified by the GA.

Ultimately, the evolutionary search identifies strategic placements for teleportation links that enhance global network connectivity and accelerate information exchange in DFL systems.

IV. Experimental Evaluation

In this section, we assess the effectiveness of the proposed method on the MNIST dataset [70], a widely used benchmark for multi-class image classification. We selected this dataset for two key reasons. First, the relative simplicity allows our evaluation to focus clearly on the impact of the method. It avoids confounding factors often present in more complex datasets. Second, the widespread use of MNIST as a standard benchmark facilitates comparison with existing work and provides



(a) Cycle graph (b) Ring of cliques (c) Sphere graph

Fig. 1. Three representative network structures with 20 nodes illustrating different topologies.

a fundamental performance baseline. We begin by examining the stability and efficiency of DFL in the absence of teleportation links, focusing on how bias propagates through the network and comparing the trade-off between "learning fast" and "learning well." This baseline evaluation serves as a reference for assessing the impact of teleportation in later experiments. Next, we demonstrate how teleportation links are added across various topologies to minimize average pair-wise distance. Following on this, we evaluate how the addition of teleportation links affects performance, focusing on improvements in average test accuracy and convergence speed. We then further assess the impact of this link placement strategy within larger DFL networks.

A. Experimental Settings

1) Network Structure: We conduct experiments on networks of two different sizes: 100 nodes and 1,000 nodes. For each size, we evaluate three representative topologies:

- *Cycle:* Fig. 1 panel (a) shows a topology where each node connects to exactly two neighbors and forms a closed loop [71], [72].
- *Ring of Cliques:* Fig. 1 panel (b) shows small fully connected groups called cliques linked together in a ring. Each clique connects to its two adjacent cliques [73]:
  - For 100 nodes: 25 cliques of 4 nodes each.
  - For 1000 nodes: 250 cliques of 4 nodes each.
- *Sphere:* Fig. 1 panel (c) shows nodes placed as uniformly as possible on the surface of a sphere using the Fibonacci lattice method [74].

2) Data Setup: We use the MNIST dataset, distributed in a strongly non-i.i.d. manner across nodes in both network sizes [75].

- *Partitioning:* Data is sorted by digit label and split into 200 segments, each containing 300 samples.
- *100-node setup:* Each node is randomly assigned two distinct segments, leading most nodes to contain samples from only two digit classes.
- *1,000-node setup:* Each node receives data from at most two digit classes, maintaining non-i.i.d. conditions.

3) Model and Training Parameters:

- *Model:* Multilayer perceptron.
- *Total Parameters:* 199,210.
- *Architecture:* Two hidden layers with 200 units each; ReLU activation [76].
- *Optimizer:* Stochastic gradient descent.
- *Learning Rates:* 0.1 for initial exploration and 0.01 for stable convergence.
- *Local Batch Size:* 10.

4) Genetic Algorithm Parameters: We use a genetic algorithm to optimize teleportation link placement, with the following configuration:

- *Population Size:* 50 individuals.
- *Generations:* 100 iterations.
- *Mutation Rate:* 0.2.

5) Computational Environment:

- *Software:* PyTorch.
- *Hardware:* Intel Xeon Gold 6530 CPU, 512 GB RAM, Nvidia RTX 6000 GPU.

B. Stability and Efficiency in DFL

The overall stability and efficiency of DFL systems are strongly affected by both the distribution of data across nodes and the learning rate used for local model training. Our first experiment demonstrates how data distribution impacts DFL stability and efficiency. This is achieved by analyzing the propagation of classification bias among nodes arranged in a 100-node cycle topology. A learning rate of 0.1 is used in this experiment. Specifically, our analysis focuses on how frequently each digit is misclassified. We measure this using the bias proportion: for a given digit, this is the count of its false positives (times it was predicted incorrectly) divided by the total number of misclassifications overall. Fig. 2 illustrates this bias proportion for three sample digits (3, 4, and 9). The visualization tracks these proportions over time (x-axis) for a specific range of nodes (nodes 55-70, y-axis). The patterns in this Fig. 2 are caused by the specific data distributions summarized in Fig. 3.

1) Bias Propagation: Referring to panel (a) for digit 3 in Fig. 2, nodes 56 and 57 initially show a tendency to misclassify other digits as 3. Over time, this bias towards digit 3 at these two nodes decreases, as indicated by the color changing from darker to lighter gray in the figure. As shown in Fig. 3, nodes 56 and 57 both possess digit 3 samples in their local training set. Consequently, they exhibit an initial bias towards digit 3. This tendency diminishes over time due to knowledge gained from other nodes through the DFL process. In this specific example, the two biased nodes (56 and 57) are adjacent in the network topology.

In contrast, biased nodes do not need to be adjacent; intermediate nodes situated between separated biased nodes can also be affected. Fig. 2(a) also shows that nodes 67, 68, and 69 exhibit a similar tendency to misclassify other digits as 3. Notably, node 68, which is trained only with digits 1 and 4 (and not 3), exhibits an even stronger bias towards digit 3 than nodes 67 and 69 do. Referring again to Fig. 3, node 67 only contains digit 3 samples, whereas node 69 contains samples from both digits 3 and 8. Node 68 demonstrates a strong propensity towards digit 3, jointly boosted by the tendencies of nodes 67 and 69. Specifically, node 67's inclination towards digit 3 is diluted by its knowledge of digits 1 and 8 (from node 66) and of digits 1 and 4 (from node 68). Likewise, node 69's tendency towards digit 3 is moderated by its understanding of digits 1, 4, 6, and 7, acquired from nodes 68 and 70.

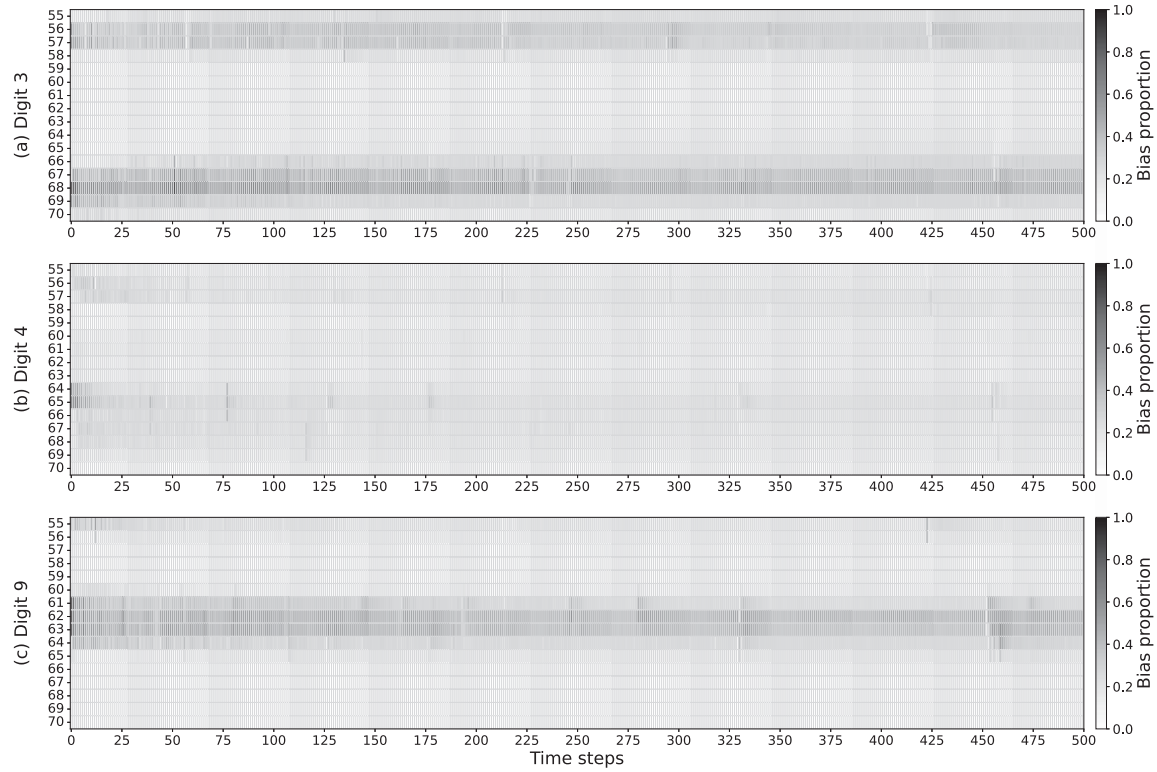


Fig. 2. Bias proportion for different digits over time across nodes.

54	0				6				
55		1							9
56				3				8	
57	0			3					
58		1	2						
59		1					7		
60					5				
61				4					9
62								8	9
63			2			6			
64				4					9
65				4			7		
66		1						8	
67				3					
68		1			4				
69				3				8	
70						6	7		

Fig. 3. Local data distribution of nodes 54–70 in the MNIST dataset, where each node holds samples of up to two digit classes. For example, node 55 holds digits 1 and 9.

In addition to bias propagating between nodes, competing biases towards different digits can exist within a single node. For example, at the beginning, nodes 61, 62, and 63 exhibit a bias towards digit 9. Specifically, node 61 contains digits 4 and 9; node 62 contains digits 8 and 9; and node 63 contains digits 2 and 6 (but no 9). Node 64 contains digits ‘4’ and ‘9’. Consequently, node 63 develops a bias towards digit ‘9’, likely influenced by nodes 62 and 64. Node 65 contains digits 4 and 7. Node 64, holding both 4 and 9 locally, serves as an example where competing biases can arise. Initially, node 64 (along with node 65) exhibits a bias towards digit 4, possibly due to mutual reinforcement as both hold digit 4 data. However, node 64 also possesses data for digit 9, creating an internal potential for bias towards 9 as well. Later, the bias towards ‘9’ becomes more prominent Over time, these specific initial biases within nodes like 64 and 65 diminish as they gain broader knowledge about other digits from more distant neighbors in the network.

Three key takeaways emerge from these observations: First, nodes initially develop biases reflecting the digits present in their local training set. Second, nodes mutually influence each other’s biases through communication. Third, initial node biases towards specific digits tend to decrease over time. This bias reduction occurs as nodes acquire broader knowledge from other nodes across the network, including from distant neighbors.

2) ”Learn Fast” Vs. “Learn Well”: Next, we will examine the impact of the learning rate of the local model on stability and efficiency in DFL. While data distribution primarily drives bias propagation, the learning rate also significantly affects system convergence speed and stability.

Fig. 4 compares the average test accuracy of DFL systems using learning rates of 0.1 and 0.01, both evaluated on a 100-node cycle topology. The shaded regions in the figure represent the 90% confidence intervals. The choice of learning rate clearly involves a trade-off between convergence stability and speed. As shown in Fig. 4, the smaller learning rate (0.01) results in stable but slower convergence, reflecting a more gradual exchange of information among nodes. In contrast, the larger learning rate (0.1) can accelerate convergence but also lead to instability.

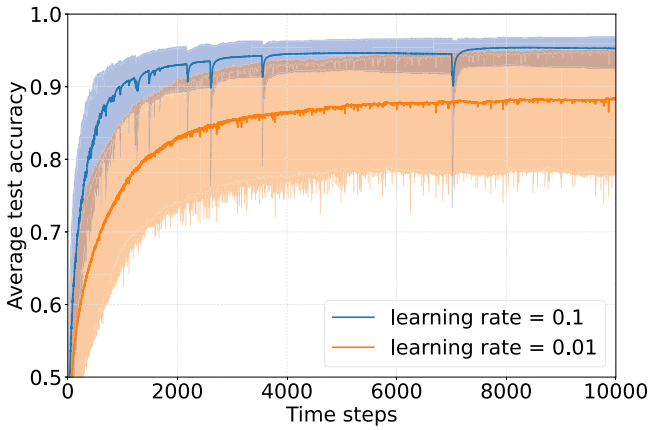


Fig. 4. Average test accuracy over time.

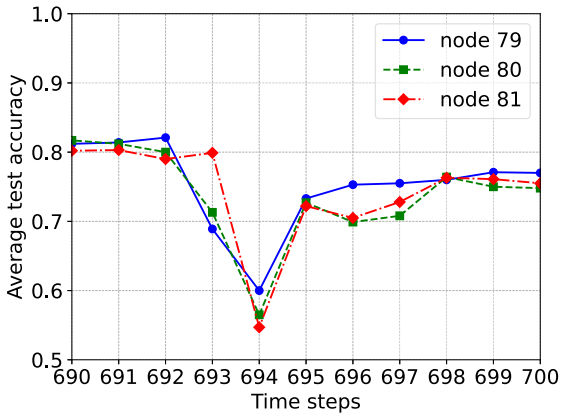


Fig. 5. Test accuracy for nodes 79 to 81.

Specifically, with a learning rate of 0.1, the average test accuracy reaches 90% within approximately 1,000 time steps. Meanwhile, with a learning rate of 0.01, the average test accuracy remains below 90% even after 10,000 time steps. If achieving a target accuracy of 80% is the goal, the learning rate of 0.1 provides a convergence speed roughly four times faster than that of 0.01.

Observing the confidence intervals, the DFL system with the smaller learning rate (0.01) exhibits a wider interval compared to the system using the higher learning rate (0.1). This indicates that at the low learning rate, a considerable number of nodes may become trapped in local optima, even while many others reach high accuracies, leading to large performance variance across the network. Furthermore, these nodes using the low learning rate exhibit considerable stability regardless of their learning accuracy level. While the higher learning rate of 0.1 enables convergence to a higher average accuracy, it also introduces greater instability. This instability is evident in the larger performance fluctuations observed over time. For example, with learning rate of 0.1, around the 7,000 time step mark, the average accuracy noticeably drops from approximately 94% to 75%.

Next, we look into the loss of performance at nodes 79 to 81 to trace how this happened. Fig. 5 zooms into time steps 690 to 700 given the data distribution in Fig. 6. In addition, Fig. 7 illustrates the evolution of the confusion matrices for nodes 79 to 81. As shown in Fig. 7, at $t = 691$, node 80 misclassifies numerous

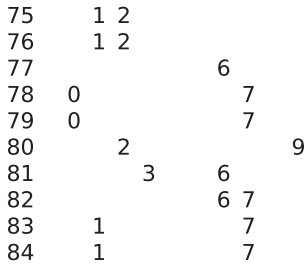


Fig. 6. Data distribution for nodes 75 to 84.

instances of digit 4 as digits 7 or 9 due to the absence of digit 4 in its own dataset and the data of nearby nodes, making digit 4 prone to misclassification. Node 80 contains digits 2 and 9. At $t = 692$, in comparison to $t = 691$, more instances of digit 4 are misclassified as digit 9. At $t = 693$, many instances of digits 4 and 9 are misclassified as digit 7. At $t = 694$, many instances of digit 2 are misclassified as digits 7 or 9, while digits 4, 5, 7, and 8 are all misclassified as digit 9. At $t = 695$, digit 4 is misclassified as digit 9, digit 5 as digits 3 or 9, and digit 8 as digits 2, 6, or 9. The bias towards digit 7 at $t = 693$ is transferred to node 80 from nodes 78 and 79. At $t = 694$, node 80 transmits its bias towards digit 9 to nodes 79 and 81, resulting in node 81 inheriting this bias and being unable to recover in the short term. Next, we will discuss strategies to achieve stable and fast DFL, aiming to optimize both convergence speed and system reliability, to mitigate the potential negative impacts.

Quantitatively, we measure the instability of DFL with learning rates of 0.1 and 0.01 with a kind of high-pass filter. To do that, we first segment the average test accuracy into intervals of length $\tau = 20$ and compute the mean for each interval. We then calculate the squared differences between successive means to evaluate instability. As shown in Fig. 8 higher learning rates are associated with greater instability, whereas lower learning rates enhance stability.

C. Increasing Connectivity by Adding Teleportation Links

The experiments above show that stable convergence can only be achieved with a low learning rate. Yet, it results in a slow convergence due to the gradual exchange of knowledge across the network. To accelerate convergence while maintaining stability, we strategically add a small number of teleportation links to the network. This improves network connectivity and accelerates the dissemination of knowledge by reducing the average pair-wise distance. Fig. 9 shows examples of increasing connectivity by adding varying number of edges across different topologies.

Table II shows the average pair-wise distance for them with varying numbers of added teleportation links. For the cycle topology, the average pair-wise distance decreases significantly from 25.25 hops with no extra edges to 7.92 with 9 extra edges, showing a reduction of 68.64%. The ring of cliques also shows a considerable decrease from 13.12 to 5.41, resulting in a 58.77% reduction. Similarly, the sphere's average pair-wise distance decreases from 15.42 to 6.54, with a 57.59% reduction.

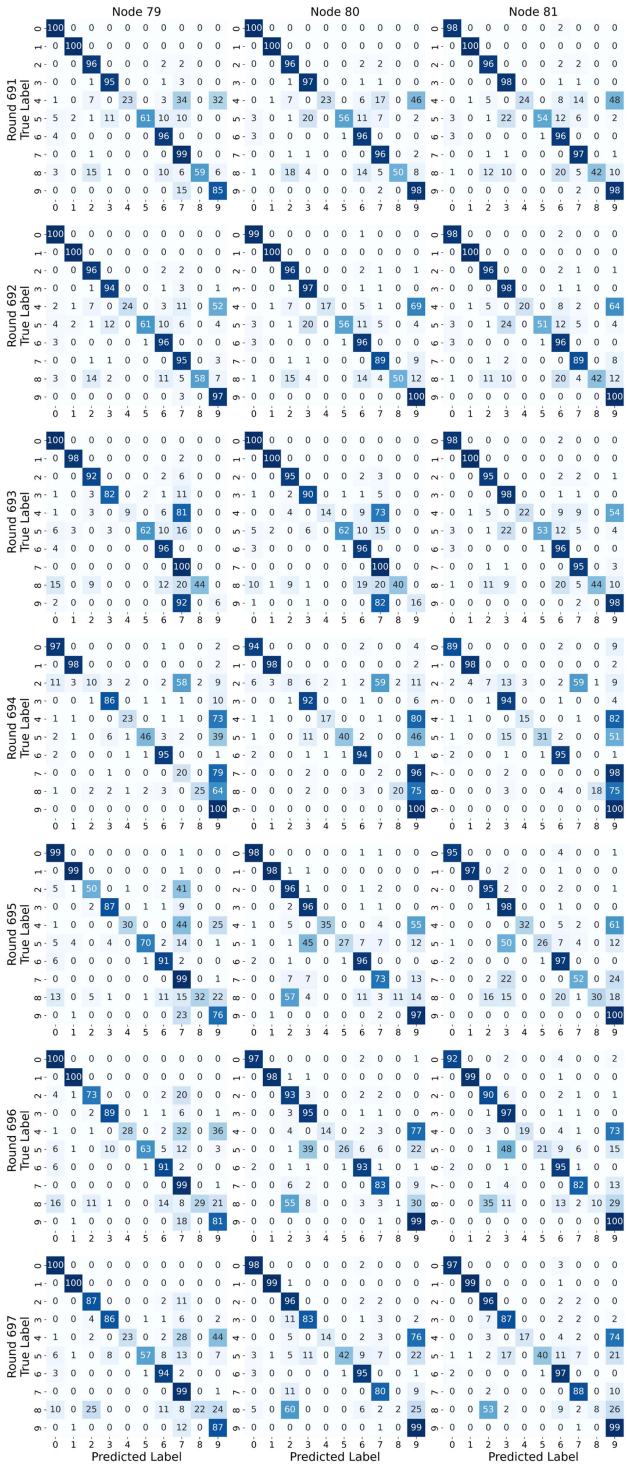


Fig. 7. Evolution of confusion matrix for nodes 79 to 81.

TABLE II
AVERAGE PAIR-WISE DISTANCE FOR DIFFERENT TOPOLOGIES WITH VARYING NUMBER OF TELEPORTATION LINKS

Topology	+0 edges	+3 edges	+6 edges	+9 edges
Cycle	25.25	12.34 (↓51.14%)	9.54 (↓62.22%)	7.92 (↓68.64%)
Ring of Cliques	13.12	7.39 (↓43.66%)	6.17 (↓52.96%)	5.41 (↓58.77%)
Sphere	15.42	8.12 (↓47.34%)	7.17 (↓53.52%)	6.54 (↓57.59%)

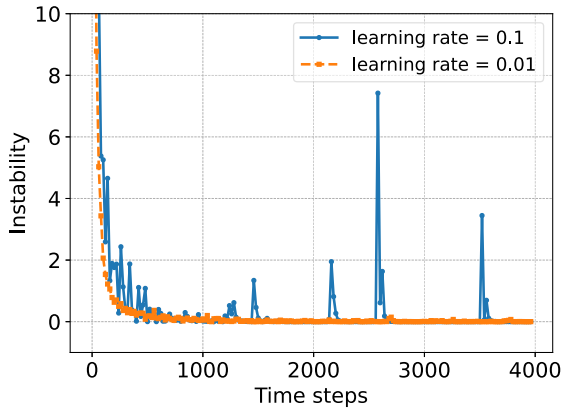


Fig. 8. Instability comparison for DFL with different learning rate.

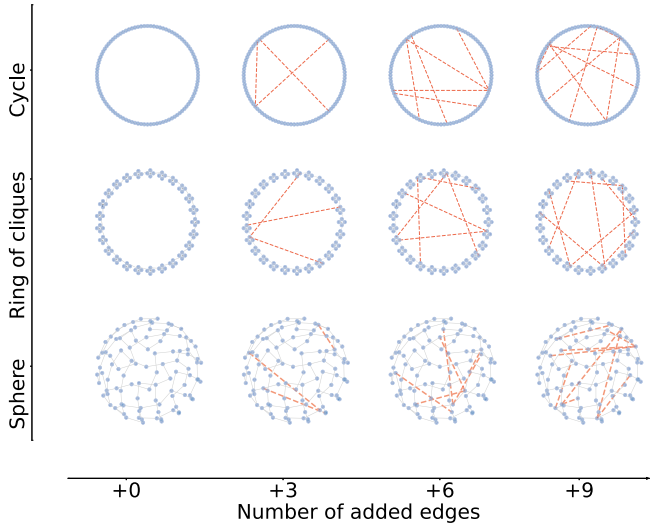


Fig. 9. Examples of implementing varying numbers of teleportation links across different topologies.

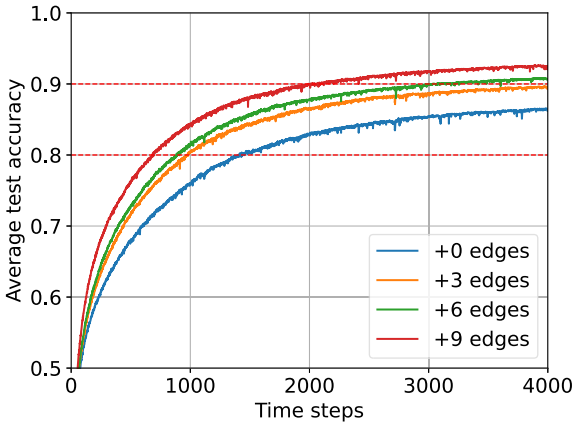
TABLE III
CLUSTERING COEFFICIENT FOR DIFFERENT TOPOLOGIES WITH VARYING NUMBER OF TELEPORTATION LINKS

Topology	+0 edges	+3 edges	+6 edges	+9 edges
Cycle	0	0	0	0
Ring of Cliques	0.75	0.75	0.75	0.75
Sphere	0	0	0	0

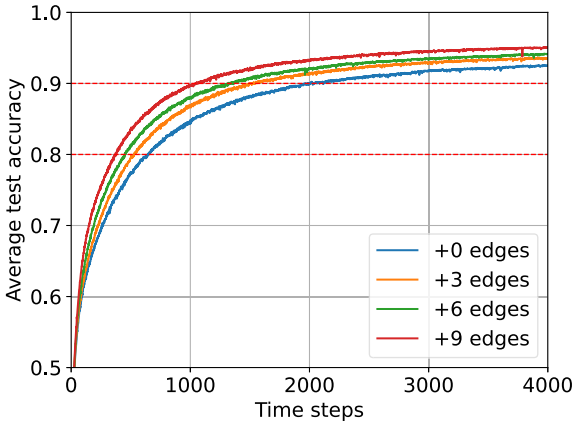
Table III shows the clustering coefficients for these network topologies with different numbers of teleportation links. The cycle and sphere topologies have a clustering coefficient of 0, regardless of the number of teleportation links added. In contrast, the ring of cliques consistently shows a high clustering coefficient of 0.75, unaffected by the addition of teleportation links. This indicates that the ring of cliques inherently supports higher clustering compared to the cycle and sphere, which accounts for its better convergence compared to cycle and sphere.

D. Performance Evaluation of Adding Teleportation Links

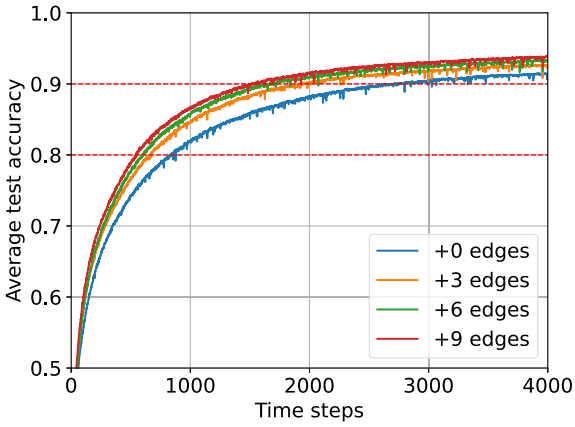
Using the examples above, we evaluate the practical impact of teleportation links on network performance. Fig. 10 shows



(a) Cycle.



(b) Ring of cliques.



(c) Sphere.

Fig. 10. Average test accuracy for different topologies with different numbers of teleportation links.

the average test accuracy achieved on different topologies when varying numbers of teleportation links are introduced. As illustrated in the figure, strategically adding even a small number of teleportation links improves network structure by reducing the average pair-wise distance. This structural improvement then leads to accelerated convergence speed without sacrificing convergence stability.

To further quantify the benefits of teleportation links, Table IV lists the number of time steps required for various

TABLE IV
TIME STEPS REQUIRED FOR DIFFERENT TOPOLOGIES WITH VARYING NUMBERS OF TELEPORTATION LINKS TO ACHIEVE 0.8 AND 0.9 TEST ACCURACY

Topology	Target accuracy	+0 edges	+3 edges	+6 edges	+9 edges
Cycle	0.9	–	–	3034	1992
	0.8	1412	967 (↓31%)	884 (↓37%)	679 (↓52%)
Ring of cliques	0.9	1965	1493 (↓24%)	1318 (↓33%)	1038 (↓47%)
	0.8	643	528 (↓18%)	450 (↓30%)	373 (↓42%)
Sphere	0.9	2655	1929 (↓27%)	1664 (↓37%)	1509 (↓43%)
	0.8	837	659 (↓21%)	595 (↓29%)	546 (↓35%)

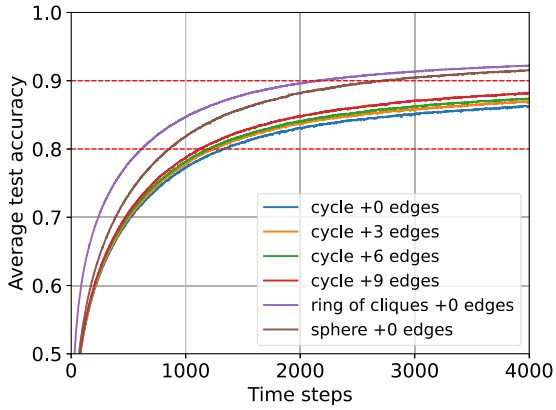


Fig. 11. Average test accuracy for 1,000-node network topologies.

TABLE V
AVERAGE PAIR-WISE DISTANCE FOR THE CYCLE TOPOLOGY WITH VARYING NUMBER OF TELEPORTATION LINKS

Topology	+0 edges	+3 edges	+6 edges	+9 edges
Cycle	250.25	115.11 (↓54.00%)	79.42 (↓68.26%)	60.83 (↓75.70%)

topologies, configured with varying numbers of teleportation links, to achieve target test accuracies of 0.8 and 0.9. The table illustrates that adding teleportation links significantly reduces the convergence time (measured in time steps) needed to reach the target accuracies for all tested topologies. For instance, in the ring of cliques topology, adding just 9 teleportation links reduces the time required to reach 0.8 accuracy by 42% and the time to reach 0.9 accuracy by 47%.

E. Teleportation Links in Larger DFL Networks

Fig. 11 illustrates the average test accuracies across 1,000-node network topologies. It indicates that introducing a small number of edges to the cycle topology can enhance the convergence speed while maintaining stability. This occurs because the addition of these edges dramatically decreases the average pairwise distance, as demonstrated in Table V.

Additionally, in scenarios without teleportation links, the ring of cliques consistently outperforms the sphere topology, which in turn surpasses the performance of the cycle topology. A detailed examination of the topological properties, as reported in Table VI, shows that the ring of cliques has a larger average

TABLE VI
AVERAGE DEGREE, AVERAGE PAIR-WISE DISTANCE, AND CLUSTERING COEFFICIENT FOR DIFFERENT TOPOLOGIES WITH 100 AND 1,000 NODES

Topology	Nodes	Average degree	Average pair-wise distance	Clustering coefficient
Cycle	100	2.00	25.25	0.00
	1000	2.00	250.25	0.00
Ring of cliques	100	3.50	13.12	0.75
	1000	3.50	**125.63**	0.75
Sphere	100	2.16	15.42	0.00
	1000	2.06	**47.56**	0.00

pairwise distance compared to the sphere. This suggests that, on average, information takes longer to propagate globally across the ring of cliques. Nevertheless, it still achieves superior learning performance, making this result particularly notable.

The strong performance of the ring of cliques can be primarily attributed to two key structural advantages: a significantly higher average node degree and an exceptionally high clustering coefficient of 0.75. In contrast, the sphere topology has a lower average degree and a clustering coefficient of zero. The high clustering coefficient reflects the presence of densely connected local communities, or cliques, where nodes form tightly knit groups. These strong local connections, supported by a larger number of neighbors, enable each node to interact with a more diverse set of peer models. For example, on the MNIST dataset, where each node holds data from at most two digit classes, the ring of cliques structure enables rich information exchange within each local community. Each clique in the ring consists of four fully connected nodes, enabling every node to interact with peers that collectively represent a broader range of digit classes. This dense local interaction promotes the development of more robust and well-informed local models. In contrast, nodes in the sphere topology have a lower average degree and no clustering, which limits their opportunities to communicate with diverse neighbors. As a result, they are less effective at aggregating varied local information, leading to slower convergence and reduced performance in the early stages of training.

By enabling rich local communication and effective information sharing within communities, the ring of cliques accelerates model refinement and leads to faster convergence toward a high-quality global model. These advantages demonstrate that strong local structure and high local connectivity can outweigh the limitations introduced by longer global communication paths. Fig. 12 compares the test accuracy between the ring of cliques and sphere topologies. In this analysis, we compute the average test accuracy within each clique for the ring of cliques, and compare it with the node-level test accuracy in the sphere topology to evaluate local consensus. As shown in Fig. 12, even after just 5 time steps, the 25th, 50th, and 75th percentiles of the ring of cliques already outperform those of the sphere. This early advantage reflects faster local consensus facilitated by the community structure. As training progresses, the performance gap between the two topologies gradually narrows due to increased information exchange across the entire network. By round 4000, the accuracies become nearly indistinguishable.

Besides, we quantitatively assess the instability across 1,000-node topologies as defined in Section IV-B. As illustrated in Fig. 13, the introduction of a small number of edges maintains

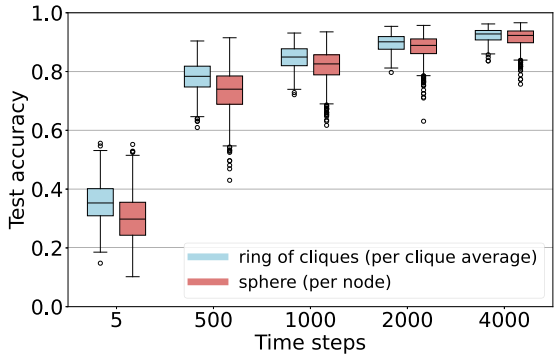


Fig. 12. Boxplot of test accuracy under ring of cliques and sphere topologies.

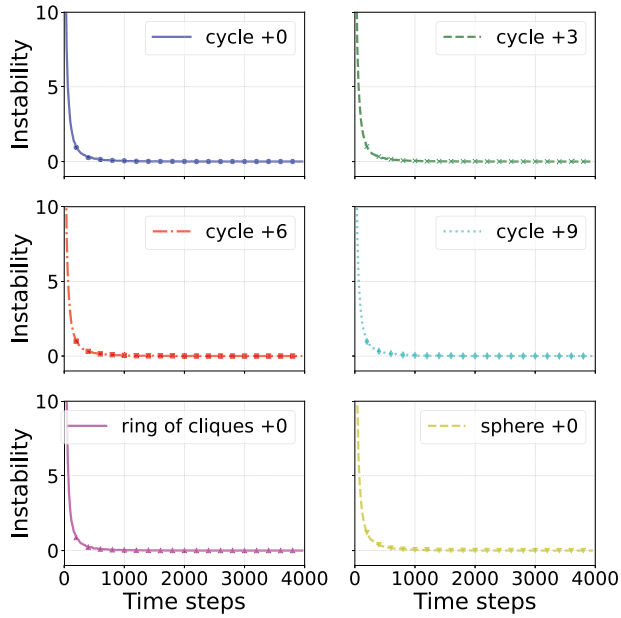


Fig. 13. Instability comparison for 1,000-node network tolologies.

stable convergence. Moreover, the three different topologies exhibit a high stability.

V. CONCLUDING REMARKS

We address the significant challenge of catastrophic forgetting in decentralized federated learning, which is primarily caused by non-i,i,d. data. We investigate how data heterogeneity hinders knowledge retention, and how local learning rates impact overall model stability. To enhance information flow between learning nodes, we introduce teleportation links inspired by small-world networks. These long range links are strategically positioned using a genetic algorithm, which aims to shorten communication paths and thereby enhance overall network communication. The cost of a communication network is typically measured by its total number of links. Therefore, we focus on adding only a few carefully selected links to improve performance, ensuring we do not significantly increase overall network complexity. The experimental results demonstrate that these teleportation links speed up model convergence while preventing catastrophic forgetting.

This work has several limitations. It does not account for highly dynamic environments where network topology and data distribution change frequently. The ability to adapt to such changes is not explored. In addition, the performance of the proposed method is not evaluated in heterogeneous networks. Such heterogeneous networks have nodes with varying capabilities and resources, where message asynchrony can be very common. Thus, more challenges would need to be addressed, including inconsistent model updates, delayed convergence, and reduced model accuracy due to nodes processing outdated information. Asynchronous message passing can also result in increased computational and communication overhead for the need of managing and re-incorporating out-of-order messages.

This work can be extended in a few interesting ways as future research. First, we will explore dynamically activating and deactivating a small number of teleportation links to adapt to environments where network topology and data distribution change frequently. Second, hardware heterogeneity and the resulting straggler effect are critical real-world factors. Our current work assumes homogeneous hardware for clarity, but this is a limitation. Investigating the impact of heterogeneity and developing strategies to manage stragglers are important directions for future research, as the benefits of teleportation links may otherwise be diminished. Third, we will measure and manage the age of information to effectively handle out-of-order messages, ensuring timely and agile model updates. Fourth, we will test the proposed method in more general synthetic and real-world topologies. In addition, the proposed method can be applied to other application domains. For instance, it can optimize communication in distributed sensor networks, enhance data flow in peer-to-peer systems, and improve efficiency in transportation networks. Finally, catastrophic forgetting may manifest differently in more complex tasks, and evaluating the method on real-world data, such as medical images or time-series sensor data, is considered a crucial next step.

REFERENCES

[1] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proc. 14th Annu. Conf. Comput. Graph. Interactive Techn.*, New York, NY, USA, Aug. 1987, pp. 25–34.

[2] D. M. Gordon, "The organization of work in social insect colonies," *Nature*, vol. 380, no. 6570, pp. 121–124, Mar. 1996.

[3] D. S. Bassett and O. Sporns, "Network neuroscience," *Nature Neurosci.*, vol. 20, no. 3, pp. 353–364, Feb. 2017.

[4] S. A. Levin, "Ecosystems and the biosphere as complex adaptive systems," *Ecosystems*, vol. 1, pp. 431–436, Sep. 1998.

[5] P. Csermely, A. London, L.-Y. Wu, and B. Uzzi, "Structure and dynamics of core/periphery networks," *J. Complex Netw.*, vol. 1, no. 2, pp. 93–123, Oct. 2013.

[6] J. Tang, G. Liu, and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends," *IEEE J. Automat. Sinica*, vol. 8, no. 10, pp. 1627–1643, Oct. 2021.

[7] A. Lalitha, O. C. Kilinc, T. Javidi, and F. Koushanfar, "Peer-to-peer federated learning on graphs," 2019, *arXiv:1901.11173*.

[8] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized SGD with changing topology and local updates," in *Proc. 37th Int. Conf. Mach. Learn.*, Vancouver, BC, Canada, Jul. 2020, pp. 5381–5393.

[9] P. Kairouz et al., "Advances and open problems in federated learning," *Foundations Trends Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, Jun. 2021.

[10] L. Yuan, Z. Wang, L. Sun, P. S. Yu, and C. G. Brinton, "Decentralized federated learning: A survey and perspective," *IEEE Internet Things J.*, vol. 11, no. 21, pp. 34617–34638, Nov. 2024.

[11] N. Rieke et al., "The future of digital health with federated learning," *NPJ Digit. Med.*, vol. 3, no. 1, Sep. 2020, Art. no. 119.

[12] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, Jan. 2019.

[13] R. Al-Huthaifi, T. Li, W. Huang, J. Gu, and C. Li, "Federated learning in smart cities: Privacy and security survey," *Inf. Sci.*, vol. 632, pp. 833–857, Jun. 2023.

[14] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[15] J. C. S. D. Anjos et al., "A survey on collaborative learning for intelligent autonomous systems," *ACM Comput. Surveys*, vol. 56, no. 4, pp. 1–37, Nov. 2023.

[16] Z. Yang, Y. Shi, Y. Zhou, Z. Wang, and K. Yang, "Trustworthy federated learning via blockchain," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 92–109, Jan. 2022.

[17] C. Shiranthika, P. Saeedi, and I. V. Bajić, "Decentralized learning in healthcare: A review of emerging techniques," *IEEE Access*, vol. 11, pp. 54188–54209, Jun. 2023.

[18] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The Internet of Things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, Jun. 2015.

[19] A. P. Kalapaaking, I. Khalil, and X. Yi, "Blockchain-based federated learning with SMPC model verification against poisoning attack for healthcare systems," *IEEE Trans. Emerg. Topics Comput.*, vol. 12, no. 1, pp. 269–280, Jan./Mar. 2023.

[20] X. Wang, Y. Chen, and O. A. Dobre, "Federated learning for anomaly detection: A case of real-world energy storage deployment," in *Proc. IEEE Int. Conf. Commun.,* Seoul, Korea, May 2022, pp. 4312–4317.

[21] D. Yang et al., "DetFed: Dynamic resource scheduling for deterministic federated learning over time-sensitive networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5162–5178, May 2023.

[22] X. Yuan, J. Chen, N. Zhang, C. Zhu, Q. Ye, and X. S. Shen, "FedTSE: Low-cost federated learning for privacy-preserved traffic state estimation in IoV," in *Proc. IEEE Int. Conf. Comput. Commun. Workshops*, New York, NY, USA, Jun. 2022, pp. 1–6.

[23] E. T. M. Beltrán et al., "Fedstellar: A platform for decentralized federated learning," *Expert Syst. Appl.*, vol. 242, May 2024, Art. no. 122861.

[24] E. T. M. Beltrán et al., "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Commun. Surv. Tuts.*, vol. 25, no. 4, pp. 2983–3013, Fourthquarter 2023.

[25] Q. Li, W. Yu, Y. Xia, and J. Pang, "From centralized to decentralized federated learning: Theoretical insights, privacy preservation, and robustness challenges," Mar. 2025, *arXiv:2503.07505*.

[26] Z. Lu, H. Pan, Y. Dai, X. Si, and Y. Zhang, "Federated learning with non-IID data: A survey," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19188–19209, Jun. 2024.

[27] M. Chen, Y. Xu, H. Xu, and L. Huang, "Enhancing decentralized federated learning for non-IID data on heterogeneous devices," in *Proc. IEEE Int. Conf. Data Eng.*, Anaheim, CA, USA, Apr. 2023, pp. 2289–2302.

[28] X. Yang, H. Yu, X. Gao, H. Wang, J. Zhang, and T. Li, "Federated continual learning via knowledge fusion: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 8, pp. 3832–3850, Aug. 2024.

[29] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.

[30] X.-F. Wang and G. Chen, "Complex networks: Small-world, scale-free and beyond," *IEEE Circuits Syst. Mag.*, vol. 3, no. 1, pp. 6–20, Dec. 2003.

[31] C. L. N. Oliveira, P. A. Morais, A. A. Moreira, and J. S. Andrade, "Enhanced flow in small-world networks," *Phys. Rev. Lett.*, vol. 112, no. 14, Apr. 2014, Art. no. 148701.

[32] R. Albert, H. Jeong, and A.-L. Barabási, "Diameter of the World Wide Web," *Nature*, vol. 401, no. 6749, pp. 130–131, Sep. 1999.

[33] S. Pei, L. Muchnik, J. S. Andrade Jr., Z. Zheng, and H. A. Makse, "Searching for superspreaders of information in real-world social media," *Sci. Rep.*, vol. 4, no. 1, Jul. 2014, Art. no. 5547.

[34] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, no. 6804, pp. 651–654, Oct. 2000.

[35] Y. Cheng et al., "Machine learning for metabolic pathway optimization: A review," *Comput. Struct. Biotechnol. J.*, vol. 21, pp. 2381–2393, 2023.

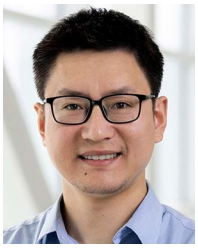
[36] J. Bascompte, P. Jordano, C. J. Melián, and J. M. Olesen, "The nested assembly of plant–animal mutualistic networks," *Proc. Nat. Acad. Sci.*, vol. 100, no. 16, pp. 9383–9387, Jul. 2003.

[37] P. Ren, R. K. Didham, M. V. Murphy, D. Zeng, X. Si, and P. Ding, "Forest edges increase pollinator network robustness to extinction with declining area," *Nature Ecol. Evol.*, vol. 7, no. 3, pp. 393–404, Jan. 2023.

[38] M. A. Nayeem, M. K. Rahman, and M. S. Rahman, "Transit network design by genetic algorithm with elitism," *Transp. Res. Part C: Emerg. Technol.*, vol. 46, pp. 30–45, Sep. 2014.
[39] M. McCloskey and N. J. Cohen, *Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem*. vol. 24, Cambridge, CA, USA: Academic Press, 1989.
[40] G. M. van de Ven, N. Soures, and D. Kudithipudi, "Continual learning and catastrophic forgetting," 2024, *arXiv:2403.05175*.
[41] Z. Wang, E. Yang, L. Shen, and H. Huang, "A comprehensive survey of forgetting in deep learning beyond continual learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 3, pp. 1464–1483, Mar. 2024.
[42] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5362–5383, Aug. 2024.
[43] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.
[44] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. 34th Int. Conf. Mach. Learn.,* Sydney, Australia, Aug. 2017, pp. 3987–3995.
[45] A. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Sci.*, vol. 7, no. 2, pp. 123–146, Jul. 1995.
[46] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, vol. 30, pp. 2994–3003.
[47] A. A. Rusu et al., "Progressive neural networks," 2016, *arXiv:1606.04671*.
[48] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," in *Proc. Int. Conf. Learn. Represent.*, Vancouver, Canada, Apr. 2018, pp. 1788–1799.
[49] K. Luo, X. Li, Y. Lan, and M. Gao, "GradMA: A gradient-memory-based accelerated federated learning with alleviated catastrophic forgetting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, Jun. 2023, pp. 3708–3717.
[50] C. Xu, Z. Hong, M. Huang, and T. Jiang, "Acceleration of federated learning with alleviated forgetting in local training," in *Proc. Int. Conf. Learn. Representations*, 2022.
[51] T. Nguyen, K. D. Doan, B. T. Nguyen, D. Le-Phuoc, and K.-S. Wong, "Overcoming Catastrophic Forgetting in Federated Class-Incremental Learning Via Federated Global Twin Generator," 2024, *arXiv:2407.11078*.
[52] C. Li, G. Li, and P. K. Varshney, "Decentralized federated learning via mutual knowledge transfer," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1136–1147, May 2021.
[53] J. Wu, F. Dong, H. Leung, Z. Zhu, J. Zhou, and S. Drew, "Topology-aware federated learning in edge computing: A comprehensive survey," *ACM Comput. Surv.*, vol. 56, no. 10, pp. 1–41, Jun. 2024.
[54] Y. Duan, X. Li, and J. Wu, "Topology design and graph embedding for decentralized federated learning," *Intell. Converged Netw.*, vol. 5, no. 2, pp. 100–115, Jun. 2024.
[55] Y. Xu, Y. Zhu, Z. Wang, H. Xu, and Y. Liao, "Enhancing federated learning through layer-wise aggregation over non-IID data," *IEEE Trans. Serv. Comput.*, vol. 18, no. 2, pp. 798–811, Mar./Apr. 2025.
[56] H. Kavalionak, E. Carlini, P. Dazzi, L. Ferrucci, M. Mordacchini, and M. Coppola, "Impact of network topology on the convergence of decentralized federated learning systems," in *Proc. IEEE Symp. Comput. Commun.*, Athens, Greece, Sep. 2021, pp. 1–6.
[57] Y. Hua, K. Miller, A. L. Bertozzi, C. Qian, and B. Wang, "Efficient and reliable overlay networks for decentralized federated learning," *SIAM J. Appl. Math.*, vol. 82, no. 4, pp. 1558–1586, 2022.
[58] M. Zhou, G. Liu, K. Lu, R. Mao, and H. Liao, "Accelerating the decentralized federated learning via manipulating edges," in *Proc. ACM Web Conf.,* Singapore, May 2024, pp. 2945–2954.
[59] D. Menegatti, A. Giuseppi, C. Poli, and A. Pietrabissa, "Dynamic topology optimization for efficient and decentralised federated learning," in *Proc. IEEE Int. Conf. Big Data (BigData)*, Washington, DC, USA, Dec. 2024, pp. 7939–7945.
[60] S. Chen, Y. Xu, H. Xu, Z. Ma, and Z. Wang, "Enhancing decentralized and personalized federated learning with topology construction," *IEEE Trans. Mobile Comput.*, vol. 23, no. 10, pp. 9692–9707, Oct. 2024.
[61] Y. Liao, Y. Xu, H. Xu, L. Wang, and C. Qian, "Adaptive configuration for heterogeneous participants in decentralized federated learning," in *Proc. IEEE Int. Conf. Comput. Commun.*, New York City, NY, USA, May 2023, pp. 1–10.
[62] J. Wang, B. Liang, Z. Zhu, E. T. Fapi, and H. Dalal, "Communication-efficient network topology in decentralized learning: A joint design of consensus matrix and resource allocation," *IEEE/ACM Trans. Netw.*, vol. 33, no. 2, pp. 761–776, Apr. 2024.
[63] I. Malvestio, A. Cardillo, and N. Masuda, "Interplay between K-core and community structure in complex networks," *Sci. Rep.*, vol. 10, no. 1, Sep. 2020, Art. no. 14702.
[64] K. Burleson-Lesser, F. Morone, M. S. Tomassone, and H. A. Makse, "K-core robustness in ecological and financial networks," *Sci. Rep.*, vol. 10, no. 1, Feb. 2020, Art. no. 3357.
[65] C. M. Valensise, A. Serra, A. Galeazzi, G. Etta, M. Cinelli, and W. Quattrociocchi, "Entropy and complexity unveil the landscape of memes evolution," *Sci. Rep.*, vol. 11, no. 1, Oct. 2021, Art. no. 20022.
[66] G. Bertagnolli, R. Gallotti, and M. D. Domenico, "Quantifying efficient information exchange in real network flows," *Commun. Phys.*, vol. 4, no. 1, Jun. 2021, Art. no. 125.
[67] M. S.K. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *IEEE Comput.*, vol. 27, no. 6, pp. 17–26, Jun. 1994.
[68] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm: A literature review," in *Proc. Int. Conf. Mach. Learning, Big Data, Cloud Parallel Comput.*, Faridabad, India, Feb. 2019, pp. 380–384.
[69] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools Appl.*, vol. 80, pp. 8091–8126, Feb. 2021.
[70] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
[71] D. B. West, *Introduction to Graph Theory*, 2nd ed. New Jersey, NJ, USA: Prentice Hall, 2001.
[72] J. woo Lee, J. Oh, S. Lim, S.-Y. Yun, and J.-G. Lee, "TornadoAggregate: Accurate and scalable federated learning via the ring-based architecture," Dec. 2020, *arXiv:2012.03214*.
[73] A. Bellet, A.-M. Kermarrec, and E. Lavoie, "D-cliques: Compensating for data heterogeneity with topology in decentralized federated learning," in *Proc. Int. Symp. Reliable Distrib. Syst.*, Vienna, Austria, Sep. 2022, pp. 1–11.
[74] B. Keinert, M. Innmann, M. Sänger, and M. Stamminger, "Spherical fibonacci mapping," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–7, Nov. 2015.
[75] X. Wang, Y. Chen, and O. A. Dobre, "Designing robust 6G networks with bimodal distribution for decentralized federated learning," in *Proc. IEEE Int. Conf. Comput. Commun. Workshops*, Vancouver, BC, Canada, May 2024, pp. 1–6.
[76] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, Fort Lauderdale, USA, Apr. 2017, pp. 1273–1282.

Xu Wang (Student Member, IEEE) is currently working toward the Ph.D. degree with Queens University, Kingston, ON, Canada. His research interests include computer networking, complex networks, cybersecurity, and federated learning.

Yuanzhu Chen (Senior Member, IEEE) received the B.Sc. degree from Peking University, Beijing, China, in 1999, and the Ph.D. degree from Simon Fraser University, Burnaby, BC, Canada, in 2004. He has been a Professor of computing science since 2005. From 2004 to 2005, he was a Postdoctoral Researcher with Simon Fraser University. In 2005, he joined Memorial University as an Assistant Professor. He was the Deputy Head for undergraduate studies and for graduate studies with Memorial from 2012 to 2015 and from 2016 to 2019, respectively, and Department Head from 2019 to 2021. He then joined Queen's School of Computing in 2021. He is currently with the School of Computing, Queen's University, Kingston, ON, Canada. His research interests include complex networks, computer networking, online social networks, mobile computing, graph theory, and evolutionary computation. He received funding from national agencies and various university programs and awards. He was the recipient of the President's Award for Distinguished Teaching in 2018, and is an IEEE Senior Member.



Qiang (John) Ye (Senior Member, IEEE) received the Ph D degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2016. He was an Assistant Professor with the Memorial University of Newfoundland, St. Johnrz's, NL Canada from 2021 to 2023, and with Minnesota State University, Mankato, MN, USA from 2019 to 2021. Since 2023, he has been an Assistant Professor with the Department of Electrical and Software Engineering, Schulich School of Engineering, University of Calgary, Calgary, AB Canada. He was with the Department of Electrical and Computer Engineering, University of Waterloo, as a Postdoctoral Fellow and then a Research Associate from 2016 to 2019. He has authored or coauthored more than 90 research papers in top ranked journals and conference proceedings. He is/was a General, Publication, Publicity, TPC, or Symposium Co-Chair for different reputable international conferences and workshops (such as, IEEE INFOCOM, GLOBECOM, VTC, ICCC, ICCT, WISEE, and SWC). He was also the IEEE Vehicular Technology Society Region 7 Chapter Coordinator in 2024, IEEE Communications Society (ComSoc) Southern Alberta Chapter Vice Chair from 2024, and the VTS Regions 1-7 Chapters Coordinator from 2022 to 2023. He is the SIG leading Co-Chair in the IEEE ComSoc IoT AHSN Technical Committee. He is an Associate Editor for the prestigious IEEE journals, such as the *IEEE Internet of Things Journal*, *IEEE Transactions on Vehicular Technology*, *IEEE Transactions on Cognitive Communications and Networking*, *IEEE Open Journal of the Communications Society*, and PPNA. He was the recipient of the Best Paper Award in the IEEE ICCC in 2024 and the IEEE TCCN Exemplary Editor Award in 2023 Early Career Research Excellence Award, Schulich School of Engineering, University of Calgary, in 2024. He has been selected as an IEEE ComSoc Distinguished Lecturer for the class of 2025–2026.

Octavia A. Dobre (Fellow, IEEE) is currently a Professor and Tier-1 Canada Research Chair with Memorial University, St. John's, NL, Canada. She has authored or coauthored more than 600 publications in her research interests which include wireless communications and networking technologies, and optical and underwater communications. She was the recipient of Ten Best Paper Awards, including the IEEE Communications Society Heinrich Hertz Award.

Dr. Dobre is with the VP Publications of the IEEE Communications Society. She was the founding Editor-in-Chief of the *IEEE Open Journal of the Communications Society* and the Editor-in-Chief of the *IEEE Communications Letters*. He is a Clarivate Highly Cited Researcher, fellow of the Royal Society of Canada, Canadian Academy of Engineering, and the Engineering Institute of Canada. She is also an Elected Member of the European Academy of Sciences and Arts and the Academia Europaea.