IEEE INFOCOM 2025 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 979-8-3315-4370-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/INFOCOMWKSHPS65812.2025.11152750

Connectivity Enrichment for Decentralized Federated Learning Networks with Teleportation

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

***Abstract*—The six generation (6G) networks demand intelligent and decentralized solutions to meet dynamic service requirements and high quality-of-service expectations. Federated Learning (FL) emerges as a promising framework for collaborative machine learning in 6G, ensuring data privacy while supporting diverse artificial intelligence (AI)-driven services. Yet, extending FL to decentralized architectures, as necessitated by 6G heterogeneous and distributed environments, faces the challenge of data heterogeneity, resulting in catastrophic forgetting. Addressing this challenge is essential for realizing pervasive network intelligence in 6G. To address this problem, we analyze the impact of data distribution on the stability and efficiency of decentralized federated learning by analyzing the propagation of bias among nodes and examining the frequency of incorrect identification for each digit. In addition, we investigate how varying local model learning rates influence stability and efficiency. To enhance the convergence speed while maintaining stability, we propose to add a small number of teleportation links to reduce the average pairwise distance, thereby enhancing connectivity and accelerating knowledge dissemination. The experimental results demonstrate the effectiveness of the proposed method.**

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

I. Introduction

Artificial intelligence (AI) and machine learning (ML) are essential for achieving pervasive network intelligence in six generation (6G) networks. These technologies provide advanced, data-informed capabilities that enhance the management and utilization of network resources, enabling uninterrupted connectivity and sophisticated services. Moreover, AI and ML analyze large datasets to predict trends, adapt to network changes, and meet quality of service demands like latency, reliability, and accuracy.

The emerging trend for distributed intelligence and efficient resource utilization has driven the development of innovative approaches, where federated learning (FL) stands out, enabling collaborative ML across decentralized systems. FL enables multiple devices to train a shared model while keeping their data localized to ensure privacy and efficient use of resources [1]–[4]. This method significantly enhances data privacy, as sensitive information remains on local devices and is not transmitted to a central server [5], [6]. Despite these advantages, the performance of FL faces significant challenges, including scalability issues and communication bottlenecks at the central server [7]. These bottlenecks can hinder efficient model updates as the number of participating devices increases. In addition, relying on a central server introduces a single point of failure, posing a risk to system reliability and robustness. To address these limitations, decentralized FL (DFL) extends the FL approach by distributing the aggregation process across peers without relying on a server [8], [9]. In DFL, each peer node trains a local model on its own data and periodically exchanges model updates with its neighbours rather than a central server. These updates are then aggregated at each peer to refine the local model. This iterative exchanging and aggregation of model updates continue until the models converge. By decentralizing the learning process, DFL enhances scalability, bypasses communication bottlenecks, and improves fault tolerance while maintaining data privacy.

However, DFL faces new challenges, primarily arising from the non-independent and identically distributed (non-i.i.d.) nature of data across different nodes. This heterogeneity in data distribution complicates model convergence and consistency, posing significant obstacles to achieving a robust and reliable global model. One of the most pressing issues arising from these challenges is catastrophic forgetting [10]. It involves a model losing a significant amount of previously acquired knowledge, causing unstable convergence in a decentralized learning system with significant implications across various critical sectors. For example, in healthcare, an unstable FL model can result in inconsistent and unreliable diagnoses with potential to harmful treatments. In autonomous driving, it poses safety risks by causing erratic vehicle behaviors. For financial institutions, it undermines fraud detection mechanisms, potentially leading to financial losses. In smart cities, it can cause inefficiencies in traffic management and resource allocation. In industrial settings, it can result in operational inaccuracies and safety hazards with collaborative robots.

The server-less nature of DFL and the non-i.i.d. data distribution are the main reasons for catastrophic forgetting. At the beginning of the training process, each node has access only to its own training data and that of its closest neighbours. This initial phase leads to a high degree of model specialization to

The work of Xu Wang and Yuanzhu Chen is financially sponsored by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant, RGPIN-2017-05201. The work of Octavia Dobre is supported by NSERC, through its Discovery program.

Xu Wang and Yuanzhu Chen are with the School of Computing, Queen's University, Kingston, ON K7L3N6 Canada.

Qiang (John) Ye is with the Department of Electrical Software Engineering, University of Calgary, Calgary, AB, T2N1N4 Canada.

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

local data, making the model susceptible to forgetting when it encounters new and varied data. As training continues, nodes start to share insights and aggregate models from more distant nodes, which sets the stages for catastrophic forgetting. The incorporation of new information during these updates can impair the performance of the model, which specializes for the data in the neighbourhood. The continuous model updates and information exchanges, fundamental to FL, exacerbate this issue as they can displace or overwrite previously learned information, resulting in decreased performance on prior tasks.

Various studies have proposed alternative approaches to overcome these limitations. SkewScout optimizes DFL by adjusting communication thresholds, selectively transmitting important parameters to save the overall model accuracy [11]. Yet, overly restrictive thresholds may slow convergence, degrade accuracy, and require careful tuning for optimal performance. P2PK-SMOTE rebalances non-IID training data by generating synthetic minority samples and selectively sharing them to enhance anomaly detection performance [12]. Nevertheless, it faces challenges in real-world deployment, as sharing synthetic training samples may still introduce privacy concerns and practical data-sharing constraints.

To address this problem, it is crucial to consider the influence of network topology in DFL [13]. Network topology significantly affects the efficiency of information flow and the convergence of the global model in several ways. First, the structure of the network determines the paths through which data travels among nodes. In a highly connected network, information can disseminate quickly and efficiently, reducing the time it takes for updates to reach all nodes. Second, the presence of highly connected nodes (hubs) can facilitate faster communication and enhance the overall speed of information propagation. These hubs act as central points that can rapidly distribute information to multiple nodes simultaneously. In addition, the average pair-wise distance within the network affects how quickly information can spread from one node to another. The clustering coefficient, a metric that measures the presence of closed triangles, also plays a significant role by indicating the likelihood of which neighbours of a given node are next to each other.

To accelerate the dissemination of information within DFL networks, we strategically introduce a limited number of teleportation links to alter the network topology. These teleportation links operate at the transport layer and serve as logical shortcuts, connecting nodes that would otherwise be forced to go through other parts of the network. In this work, we use a genetic algorithm to add teleportation links to the DFL network. The contributions of this work are summarized as follows: (1) We analyze the impact of data distribution on the stability and efficiency of DFL by examining the propagation of bias across nodes; (2) We explore the effect of the local model learning rate on stability and efficiency; (3) We propose to add a small number of teleportation links to increase network connectivity, thereby accelerating convergence while maintaining stability. Unlike [14], which primarily emphasizes accelerating DFL, this work provides a broader analysis by investigating the impact of data distribution on stability and efficiency, examining bias propagation among nodes, and analyzing the frequency of incorrect identification for each digit.

The remainder of the paper is organized as follows. Section II details the methodology for connectivity enrichment. Section III evaluates the performance of our proposed method. Section IV presents concluding remarks, and future research directions.

II. Connectivity Enrichment

In this section, we begin with catastrophic forgetting in DFL. We then explore the role of network connectivity in DFL. Finally, we address how network connectivity can be improved by introducing a small number of teleportation links.

A. Catastrophic forgetting in DFL

Catastrophic forgetting occurs when a ML model, upon being trained on new data, significantly loses its ability to perform well on previously learned tasks or data [15]. This phenomenon is particularly problematic in DFL, when training data is non-i.i.d. Each node initially has access only to its local data and the data of its immediate neighbours. During the initial training phase,the training model becomes highly specialized to the local data of the node, increasing the risk of forgetting when exposed to new data. As training progresses, nodes acquire insights from distant nodes through model aggregation, further exacerbating the issue. The delay in information exchange, even if there is a small number of hops, implies that updates can be outdated by the time they are received. Consequently, these updates can lead to catastrophic forgetting, as the model parameters are adjusted to incorporate new information.

B. Connectivity in DFL network topology

Nodes in DFL represent individual participants, such as devices or servers, each holding local data and performing computations. These nodes train local models and share model updates with other nodes, typically involving model parameters to construct a global model while preserving privacy. Edges in the network represent communication links, determining direct information exchange paths. These links define how nodes interact and share information directly with each other. In addition, the edges influence the network topology and the overall efficiency of information spread. Several key metrics can be employed to assess network connectivity in DFL networks, each offering distinct advantages and disadvantages. These metrics include network density, diameter, average pair-wise distance, flow capacity, and clustering coefficient [16]. Each of these metrics provides unique insights on different aspects of network structure and performance [17]. Network density represents the ratio of existing connections to the total possible connections, offering a quick overview of overall connectivity, and does not distinguish the importance of individual connections. Diameter determines the maximum distance between any two nodes, revealing the network spatial extent, and is less stable due to its sensitivity to changes. Average pair-wise distance measures the mean distance between all node pairs, indicating the efficiency of information flow within the network. The average distance is more robust compared to the network diameter as it is pooled from all node pairs. Flow capacity determines the maximum volume that can be transported through a network. Information throughput is limited by the network bottlenecks, where the connectivity is weaker than other richly connected regions. The clustering coefficient quantifies the extent to which nodes form triangles, indicating cohesion in a neighbourhood.

In our assessment of DFL network connectivity, we choose average pair-wise distance and the clustering coefficient. Average pair-wise distance provides a comprehensive, global measure of the network efficiency in information flow. Formally, given a connected undirected graph $G = (V, E)$ of n vertices in V, denoted as $v_1, v_2, \ldots$, we use $d_{i,j}$ to denote the distance between nodes v_i and v_j, i.e., the length of the shortest path joining these two nodes. Hence, the average pair-wise distance, $\langle d \rangle$, can be expressed as

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

A small average pair-wise distance enables nodes to efficiently acquire knowledge from distant nodes, promoting broader dissemination of information across the network. Conversely, a higher clustering coefficient facilitates fast and efficient local information exchange by ensuring that neighboring nodes are richly connected. With high connectivity achieved either way, nodes continuously refine their models with diverse knowledge from others, retaining previously learned information while integrating new insights. The clustering coefficient $\langle C \rangle$ indicates the tendency of nodes to form triangles, i.e., the mean of the per-node clustering coefficients C_i over all nodes, given by

$$\langle C \rangle = \frac{1}{n} \sum_{i=1}^{n} C_i \tag{2}$$

where the per-node clustering coefficient of node v_i with degree k_i is

$$C_i = \frac{2L_i}{k_i(k_i - 1)}. \tag{3}$$

Here, L_i denotes the number of links between the k_i neighbors of node v_i. Both C_i and $\langle C \rangle$ range from 0 to 1.

C. Teleportation links

To speed up the convergence and improve the stability of DFL systems, we strategically enhance network connectivity through addition of teleportation links. These overlay links are logical connections, at the transport layer, and function as shortcuts, enabling direct communication between nodes that would otherwise be indirectly connected. This reduces the average pair-wise distance across the network. By integrating teleportation links, DFL networks achieve faster convergence, improved generalization, and more stable and efficient learning processes.

We employ a genetic algorithm to add teleportation links with the aim of increasing network connectivity [18], [19]. We encode an individual as all $\frac{n(n-1)}{2}$ possible node pairs using a binary representation, where 0 indicates the absence of an edge and 1 indicates the presence of an edge. The fitness function is defined as the average pair-wise distance within the corresponding network, with lower values indicating more optimal solutions. There are two primary operations applied to these individuals. For the mutation process, node pairs connected by edges in the original graph have a mutation rate of 0, while those not connected in the original graph have a mutation rate of 0.2. Here, we enforce a constant number of added edges. Crossover combines segments from two parent individuals to generate two new offsprings. We maintain the population of 50 individuals in the evolution. The selection retains the fittest individuals based on their fitness scores. We iteratively apply these operations for 100 generations to reduce the network connectivity and terminate the process afterwards.

III. Experimental Evaluation

In this section, we assess the effectiveness of the proposed method on the MNIST dataset [20]. We use MNIST for two key reasons. First, its simplicity ensures that evaluation focuses on the method rather than dataset complexity. Large datasets introduce additional challenges that can obscure method assessment, whereas MNIST allows for a clearer evaluation. Second, MNIST is a widely recognized benchmark, extensively studied and commonly used for method comparison. Poor performance on MNIST often signals potential issues on more complex datasets.

Initially, we examine the stability and efficiency of DFL by exploring bias propagation and comparing the impact of different learning rate of the local model on stability and efficiency in DFL. Next, we demonstrate the addition of teleportation links across various topologies and evaluate their performance.

A. Experiment settings

1) Topologies: We evaluate the proposed method on three typical network topologies.

a) Cycle: In a cycle, each node connects to exactly two other nodes, forming a closed loop. This configuration ensures that there is a single continuous path for data to travel, looping back to the starting point [21], [22].

b) Ring of cliques: In a ring of cliques, each clique connects to exactly two other cliques, forming a ring. For a setup with 100 nodes, there are 25 cliques, each containing 4 nodes [23].

c) Sphere: In a sphere, nodes are distributed evenly on the surface of a sphere. We use the Fibonacci sequence to place nodes. The nodes are spaced to minimize overlap and maximize coverage [24].

2) Data distribution: We evaluate the method on 100 nodes, following a non-i.i.d. data distribution. To achieve this, we initially categorize the data by digit label and then divide it into 200 segments, with each segment containing 300 samples.

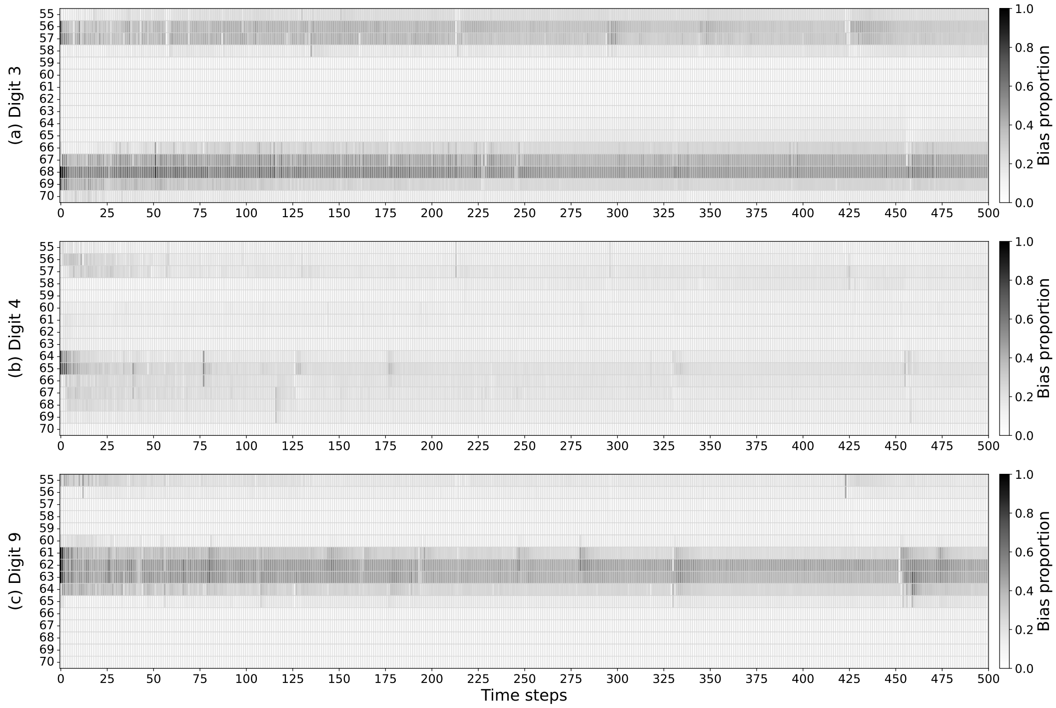


Fig. 1: Bias proportion for different digits over time across nodes.

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. 2: Data distribution for nodes 54 to 70.

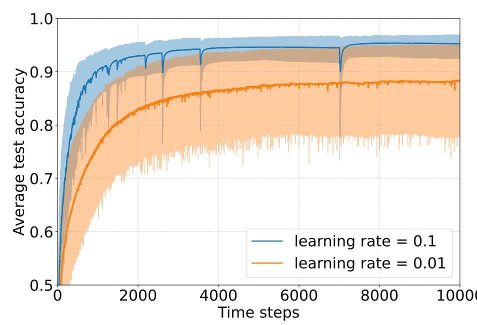


Fig. 3: Average test accuracy over time.

Each node is assigned two segments. This leads to a highly non-i.i.d. distribution where the majority of nodes possess samples from only two different digits.

3) Federated learning parameters: We train a multilayer perceptron with a total of 199,210 parameters. The network includes two hidden layers, each with 200 units using ReLU activations. We employ stochastic gradient descent for optimization, with learning rates of 0.1 and 0.01, and use a local training batch size of 10.

4) Hardware configuration: The proposed algorithm is implemented with PyTorch. It is tested on a computing node equipped with an Intel Xeon Gold 6530 CPU. The system has 512GB of RAM and an Nvidia RTX 6000 GPU.

B. Stability and efficiency in DFL

Stability and efficiency in DFL are significantly influenced by data distribution and the learning rate of the local model. In the first experiment, we illustrate the impact of data distribution on the stability and efficiency of DFL by analyzing bias propagation across nodes in a cycle topology of 100 nodes. Here, the learning rate is 0.1. We are interested in observing the frequency at which each digit is incorrectly identified. That is, the number of false positives for each digit divided by the total number of misclassifications. Fig. 1 shows the bias proportion for three example digits 3, 4, and 9 over time (x-axis) across nodes 55 through 70 (y-axis). The patterns in this Fig. 1 are caused by the specific data distributions summarized in Fig. 2.

In panel (a) for digit 3, nodes 56 and 57 exhibit a tendency to misclassify other digits as digit 3. Over time, the bias of these two nodes towards digit 3 decreases, transitioning from a darker gray to a lighter gray. As shown in Fig. 2, both nodes 56 and 57 have digit 3 in their training set, so they have a tendency towards digit 3 from the very beginning. Such a tendency tapers due to the knowledge acquired from other nodes over time. In this case, the biased nodes are next to each other.

In contrast, the biased nodes can be spaced out and nodes between them can still be affected. Fig. 1(a) shows that nodes 67, 68, and 69 show a similar propensity to misclassify other digits as 3. Notably, node 68, which are trained with digits 1 and 4 but not digit 3, exhibits an even stronger bias towards digit 3 than nodes 67 and 69. In particular, node 67 exclusively contains digit 3, whereas node 69 includes both digits 3 and 8, as depicted in Fig. 2. The observation above can be explained as follows. 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.

Not only can bias towards a given digit propagate to nearby nodes, but also biases towards different digits at the same node compete with each other. 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 does not contain digit 9 but includes digits 2 and 6. Node 64 contains digits 4 and 9, and node 63 develops a bias towards digit 9 due to the influence from nodes 62 and 64. Node 65 contains digits 4 and 7. Initially, nodes 64 and 65 exhibit a bias towards digit 4 as their knowledge of digit 4 mutually reinforces one another. Over time, the bias of these nodes towards digit 4 or 9 diminishes due to the knowledge of other digits acquired from more distant neighbours.

There are three key takeaways from the observations above. First, nodes initially exhibit specific biases towards certain digits based on the digits represented in their training set. Second, nodes can influence each other's biases. Third, over time, the initial biases of nodes towards specific digits decrease. This reduction in bias is attributed to the acquisition of knowledge from other nodes, including distant neighbours.

Next, we will illustrate the impact of the learning rate of the local model on stability and efficiency in DFL. While data distribution plays a crucial role in bias propagation, the learning rate also significantly affects the system performance in a different dimension.

Fig. 3 presents a performance comparison of the average test accuracy between DFL with learning rates of 0.1 and 0.01 over a cycle topology of 100 nodes. The shaded area shows the 90% confidence interval. There is a trade-off between stable and fast convergence regulated by the choice of learning rate. As shown in Fig. 3, a smaller learning rate results in stable but slower convergence due to the gradual exchange of information. In contrast, a larger learning rate can accelerate convergence but often leads to instability. Specifically, at a learning rate of 0.1, the average test accuracy reaches 0.9 within 1,000 time steps. Meanwhile, at a learning rate of 0.01, the average test accuracy is still below 0.9 after 10,000 time steps. If our target learning accuracy is 0.8, learning rate of 0.1 gives us a convergence rate around four times faster than 0.01.

TABLE I: 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%)

It is observed that the DFL system with a smaller learning rate exhibits a larger confidence interval compared to a higher learning rate. That is, at a low learning rate, a considerable number of nodes are trapped at local optima while many may reach high learning accuracies, leading to a large performance variance. Furthermore, these nodes exhibit considerable stability regardless of their learning accuracy level. In contrast, at the higher learning rate of 0.1, the DFL system successfully converges to a higher level, but the system also demonstrates a greater instability. For example, at approximately 7,000 time steps, the accuracy noticeably drops from 94% to 75%.

C. Performance evaluation of adding teleportation links

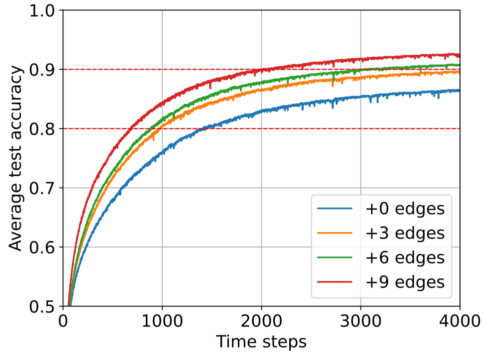
Fig. 4 presents the average test accuracy for different topologies with varying numbers of teleportation links. As illustrated in the figure, strategically adding a small number of teleportation links reduces the average pair-wise distance and accelerates convergence speed while maintaining stability in convergence.

To further quantify the benefits of teleportation links, Table I shows time steps required for different topologies with varying numbers of teleportation links to achieve 0.8 and 0.9 test accuracies. It demonstrates that adding teleportation links significantly reduces the time needed to reach the target accuracies across all topologies. For instance, adding 9 edges to the ring of cliques reduces the time to reach 0.8 by 42% and the time to reach 0.9 by 47%.

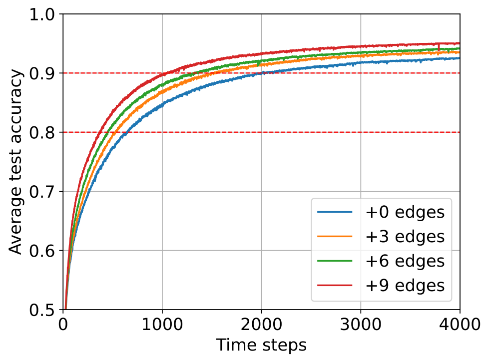
IV. Conclusion and Future Work

In this paper, we strategically add a small number of teleportation links to decentralized federated learning networks, enhancing connectivity and accelerating knowledge dissemination. The cost of a communication network is usually measured by the number of communication links in it. This work achieves fast and stable convergence simultaneously with little network cost increase. The experimental results demonstrate that these teleportation links speed up model convergence while preventing catastrophic forgetting.

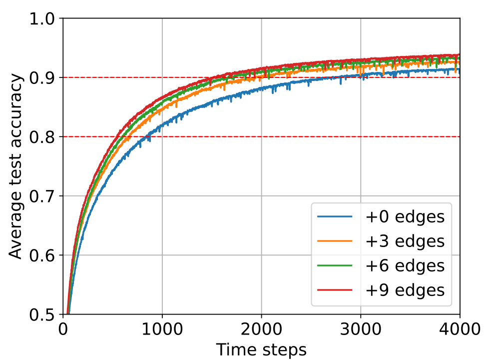
For future work, we will investigate possible performance degradation at specific nodes to trace anomalies during model training and quantitatively assess the instability of DFL under varying learning rates. We will evaluate the impact of adding



(a) Cycle.



(b) Ring of cliques.



(c) Sphere.

Fig. 4: Average test accuracy for different topologies with different numbers of teleportation links.

a varying number of edges on network connectivity across different topologies. We will also examine the scalability and efficiency of the proposed method in a larger DFL network. In addition, the proposed method can be applied to other different 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.

References

[1] X. Yuan, J. Chen, N. Zhang, Q. J. Ye, C. Li, C. Zhu, and X. Sherman Shen, "Low-Cost Federated Broad Learning for Privacy-Preserved Knowledge Sharing in the RIS-Aided Internet of Vehicles," *Engineering*, vol. 33, pp. 178–189, Feb. 2024.

[2] X. Wang, Y. Liang, Y. Chen, and O. A. Dobre, "Robust Federated Learning for Energy Storage Systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Dubai, UAE, Jul. 2024, pp. 01–06.

[3] D. Yang, W. Zhang, Q. Ye, C. Zhang, N. Zhang, C. Huang, H. Zhang, and X. Shen, "DetFed: Dynamic Resource Scheduling for Deterministic Federated Learning Over Time-Sensitive Networks," *IEEE Trans. Mob. Comput.*, vol. 23, no. 5, pp. 5162–5178, Aug. 2024.

[4] P. Zheng, Y. Zhu, Y. Hu, Z. Zhang, and A. Schmeink, "Federated Learning in Heterogeneous Networks With Unreliable Communication," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 4, pp. 3823–3838, Apr. 2024.

[5] 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. Computer Commun. Workshops (INFOCOM WKSHPS)*, May 2022, pp. 1–6.

[6] X. Wang, Y. Chen, and O. A. Dobre, "Federated Learning for Anomaly Detection: A Case of Real-World Energy Storage Deployment," in *Proc. Int. Conf. Commun. (ICC)*, Seoul, Korea, May 2022, pp. 4312–4317.

[7] Z. Yang, M. Chen, K.-K. Wong, H. V. Poor, and S. Cui, "Federated Learning for 6G: Applications, Challenges, and Opportunities," *Engineering*, vol. 8, pp. 33–41, Jan. 2022.

[8] E. T. Martínez Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized Federated Learning: Fundamentals, State of the Art, Frameworks, Trends, and Challenges," *IEEE Commun. Surv. Tutor.*, vol. 25, no. 4, pp. 2983–3013, Sept. 2023.

[9] J. Zhang, L. Chen, X. Chen, and G. Wei, "A Novel Hierarchically Decentralized Federated Learning Framework in 6G Wireless Networks," in *Proc. IEEE Int. Conf. Computer Commun. (INFOCOM)*, Hoboken, NJ, USA, Aug. 2023, pp. 1–6.

[10] M. McCloskey and N. J. Cohen, *Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem*. Academic Press, 1989, vol. 24.

[11] K. Hsieh, A. Phanishayee, O. Mutlu, and P. B. Gibbons, "The Non-IID Data Quagmire of Decentralized Machine Learning," in *Proc. 37th Int. Conf. Machine Learning. (ICML)*, Jul. 2020.

[12] H. Wang, L. Muñoz González, D. Eklund, and S. Raza, "Non-IID Data Re-balancing at IoT Edge with Peer-to-Peer Federated Learning for Anomaly Detection," in *Proc. ACM Conf. Security and Privacy in Wirel. and Mob. Networks*, New York, NY, USA, Jun. 2021, p. 153–163.

[13] 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. and Commun. (ISCC)*, Athens, Greece, Dec. 2021, pp. 1–6.

[14] M. Zhou, G. Liu, K. Lu, R. Mao, and H. Liao, "Accelerating the Decentralized Federated Learning via Manipulating Edges," in *Pro. ACM Web Conf.*, New York, NY, USA, May 2024, p. 2945–2954.

[15] Y. Kong, L. Liu, H. Chen, J. Kacprzyk, and D. Tao, "Overcoming Catastrophic Forgetting in Continual Learning by Exploring Eigenvalues of Hessian Matrix," *IEEE Trans. Neural Netw. Learn.*, pp. 1–15, Jul. 2023.

[16] Y. Li, Y. Shang, and Y. Yang, "Clustering Coefficients of Large Networks," *Information Sciences*, vol. 382-383, pp. 350–358, Mar. 2017.

[17] A.-L. Barabasi and M. Posfai, *Network Science*. Cambridge, England: Cambridge University Press, 2016.

[18] A. Awad, A. Hawash, and B. Abdalhaq, "A Genetic Algorithm (GA) and Swarm-Based Binary Decision Diagram (BDD) Reordering Optimizer Reinforced With Recent Operators," *IEEE Trans. Evol. Comput.*, vol. 27, no. 3, pp. 535–549, Jun. 2023.

[19] Y. Zhao, Y. Zhu, and S. Wang, "User Scheduling in Wireless Networks for Deterministic Service: An Efficient Genetic Algorithm Method," *IEEE Netw. Lett.*, vol. 6, no. 1, pp. 1–5, Dec. 2024.

[20] L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Oct. 2012.

[21] D. West, *Introduction to Graph Theory, Second Edition*. Prentice Hall Upper Saddle River, 2001, vol. 2.

[22] J.-w. Lee, J. Oh, S. Lim, S.-Y. Yun, and J.-G. Lee, "Tornadoaggregate: Accurate and Scalable Federated Learning via the Ring-based Architecture," *arXiv preprint arXiv:2012.03214*, 2020.

[23] A. Bellet, A.-M. Kermarrec, and E. Lavoie, "D-cliques: Compensating Noniidness in Decentralized Federated Learning with Topology," *arXiv preprint arXiv:2104.07365*, vol. 1, no. 2, p. 5, 2021.

[24] B. Keinert, M. Innmann, M. Sänger, and M. Stamminger, "Spherical Fibonacci Mapping," *ACM Trans. Graph.*, vol. 34, no. 6, Nov. 2015.