Supplementary Material: Reducing Training Time in Cross-Silo Federated Learning using Multigraph Topology

Tuong Do[1†], Binh X. Nguyen[1†], Vuong Pham[1], Toan Tran[2], Erman Tjiputra[1], Quang D.Tran[1], Anh Nguyen[3]
[1]AIOZ, Singapore
[2]VinAI Research, Vietnam
[3]University of Liverpool, UK
{tuong.khanh-long.do, binh.xuan.nguyen} @aioz.io

This appendix contains additional materials for the paper "*Multigraph Topology Design for Cross-Silo Federated Learning*". We include training procedures, experimental setup details, additional results, and further discussions.

1. Training Procedure

The decentralized periodic averaging stochastic gradient descent (DPASGD) [7] is a popular algorithm for training federated learning setup since it allows local-update in each silo during the learning process. This section shows how the DPASGD is used for training cross-silo federated learning co-operated with our proposed multigraph. Note that the convergence analysis of DPASGD can be found in [7].

For convenience, we re-write the updater in each training round as follows:

$$\mathbf{w}_i\left(k+1\right) = \begin{cases} \sum_{j\in\mathcal{N}_i^{++}\cup\{i\}} \mathbf{A}_{i,j}\mathbf{w}_j\left(k-h\right), \\ \qquad \text{if k} \equiv 0 \,(\text{mod}\; u+1)\;\&\; \left|\mathcal{N}_i^{++}\right| > 1, \\ \mathbf{w}_i\left(k\right) - \alpha_k \frac{1}{b}\sum_{h=1}^{b} \nabla L_i\left(\mathbf{w}_i\left(k\right), \xi_i^{(h)}\left(k\right)\right), \\ \qquad \text{otherwise.} \end{cases} \tag{1}$$

where $(k-h)$ is the index of the considered weights; h is initialized to 0 and: $h = h+1$, if $e_{k-h}(i,j) = \mathbb{0}$. The training procedure of our multigraph using DPASGD is described in Algorithm 1. We first select the appropriate state from the multigraph that is synchronized with the communication round to identify the connection status between silos. Then, the DPASGD updater in Eq. 1 is used to compute the accumulation between learned weights of local silos that have corresponding strong-connected edges.

Algorithm 1: Multigraph Training Procedure.

Input: List of multigraph states $\mathcal{S}$;
Initial weight $\mathbf{w}_i(0)$ for each silo i;
Maximum training round K.

```
c ← Create states counting variable which is initialized by zero.
for round k = 0 to K − 1 do
    G^s_{m_c} ← Establish the state the need to use by selecting th c-th graph G^s_m in the input list of multigraph S
    c ← Update the counting variable by +1 when there is a state has been used.
    if S[c] ∉ S then
        Reset c to 0
    for i = 0 to N do
        N_i^{++} ← strongly-connected edges list of i using G^s_{m_i}.
    // The loop below is parallel
    foreach silo i ∈ N do
        for batch ♭ = 0 to u do
            m_♭ ← Sampling from local dataset of silo i
            w_i(k + 1) ← Update model using Eq. 1.
```

2. Algorithm Complexity

It is trivial to see that the complexity of our proposed Algorithm 1, Algorithm 2 in the main paper, and our training procedure is $\mathcal{O}(n^2)$. In practice, since the cross-silo federated learning setting has only a few hundred silos ($n < 500$) [2], the time to execute our algorithms is just a tiny fraction of training time. Therefore, our proposed topology still can significantly reduce the overall wall-clock training time.

3. Network Setup

Table 1 shows the statistic of five distributed networks in our experiments: Exodus, Ebone, Géant, Amazon [6], and Gaia [1]. The Exodus, Ebone, and Géant are from the Internet Topology Zoo [3]. The Amazon and Gaia network are

Network	#Silos	#Maximum Links
Gaia [1]	11	55
Amazon [6]	22	231
Géant [3]	40	61
Exodus [3]	79	147
Ebone [3]	87	161

Table 1. The network setups in our experiments.

synthetic networks that are constructed using the geographical locations of the data centers.

4. Additional Results

4.1. Results on different networks and datasets

Figure 1, Figure 2, and Figure 3 show more experiment results of our multigraph and other methods on different datasets and network setups. These figures show that our multigraph achieves state-of-the-art accuracy compared to other methods (top row) while significantly reducing the total wall-clock time (bottom row).

4.2. Access Link Capacities Analysis

Following [5], we analyse the effect of access link capacity on our multigraph topology. Access link capacity is related to the bandwidth when packages are transmitted between silos. Figure 4 shows the results under Exodus network and FEMNIST dataset in two scenarios: all access links have the same 1 Gbps capacity and one orchestra node has a fixed 10 Gbps access link capacity. From Figure 4, we can see that our multigraph topology slightly outperforms RING when the link capacity is low. However, when the capacity between silos is high, then our method clearly improves over RING. In all setups, our method archives the best cycle time and training time.

5. Further Discussion

Limitations. Since our multigraph is designed based on RING [5], our method inherits both the strengths and weaknesses of RING. We can see that the "lower bound" of our multigraph is the overlay of RING when there are no isolated nodes. In this case, all states in our multigraph are the input overlay. Hence, there is no improvement. Furthermore, compared to RING, our multigraph is more sensitive to the low bandwidth capacity setup (Figure 4).

Privacy Concerns. The privacy of federated learning has been discussed in many papers [4, 2]. Generally, any federated learning topologies (such as ours) can be accompanied by any privacy preservation methods [5]. Since our method focuses on optimizing topology, studying the privacy concern is not our main focus. In practice, our proposed topology potentially can enhance privacy as it reduces the data exchanges between nodes.

Broader Impacts. Decentralized inference mechanisms are a workable alternative that balances utility with crucial normative values like privacy, transparency, and accountability. Given the ease of access to sensitive data, large-scale centralized data sources are not feasible to uphold the best interests and privacy of the users. We hope that using our proposed topology, we can reduce the training time of the whole network, and consequently would bring direct benefits in real-world applications such as saving energy when training a model, reducing waiting time, and potentially reducing the privacy breach when exchanging data between silos.

References

[1] Kevin Hsieh, Aaron Harlap, Nandita Vijaykumar, Dimitris Konomis, Gregory R Ganger, Phillip B Gibbons, and Onur Mutlu. Gaia: Geo-distributed machine learning approaching lan speeds. In *USENIX Symposium on Networked Systems Design and Implementation*, 2017. 1, 2

[2] Peter Kairouz, H Brendan McMahan, et al. Advances and open problems in federated learning. *arXiv*, 2019. 1, 2

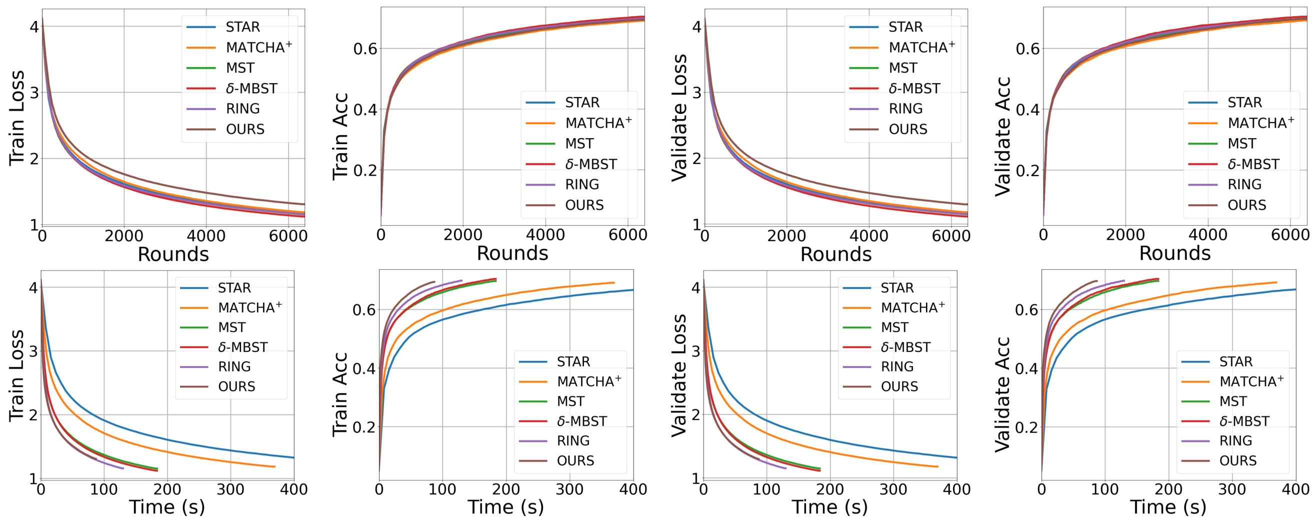
[3] Simon Knight, Hung X Nguyen, Nickolas Falkner, Rhys Bowden, and Matthew Roughan. The internet topology zoo. *IEEE Journal on Selected Areas in Communications*, 2011. 1, 2

[4] Chengxi Li, Gang Li, and Pramod K Varshney. Decentralized federated learning via mutual knowledge transfer. *IEEE Internet of Things Journal*, 2021. 2

[5] Othmane Marfoq, Chuan Xu, Giovanni Neglia, and Richard Vidal. Throughput-optimal topology design for cross-silo federated learning. In *NIPS*, 2020. 2

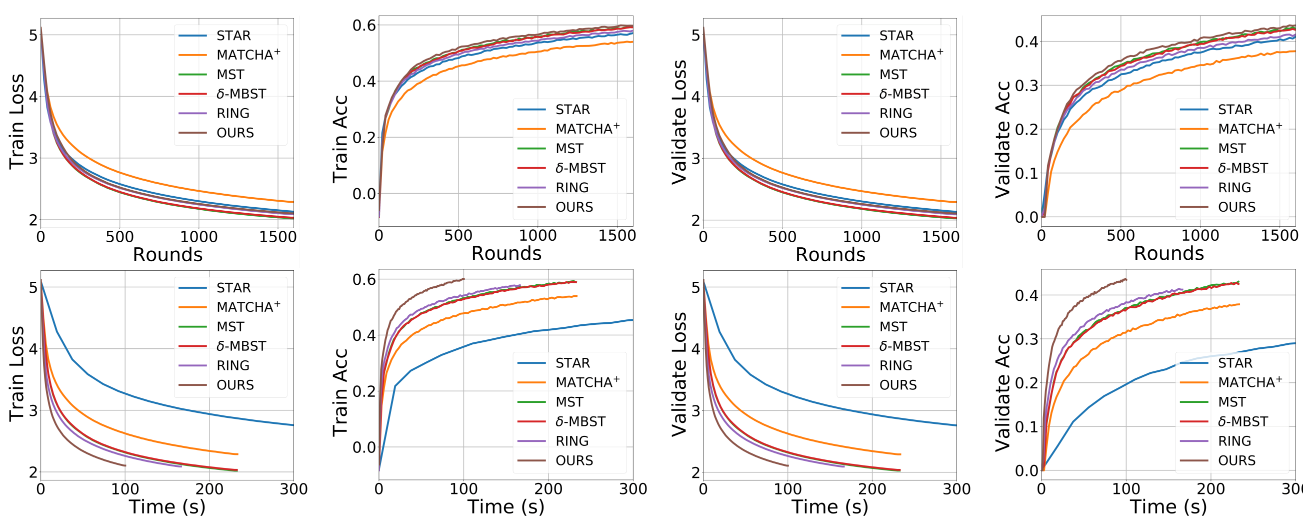
[6] Frederic P. Miller, Agnes F. Vandome, and John McBrewster. Amazon web services. *Retrieved November*, 2011. 1, 2

[7] Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. In *ICLRW*, 2018. 1



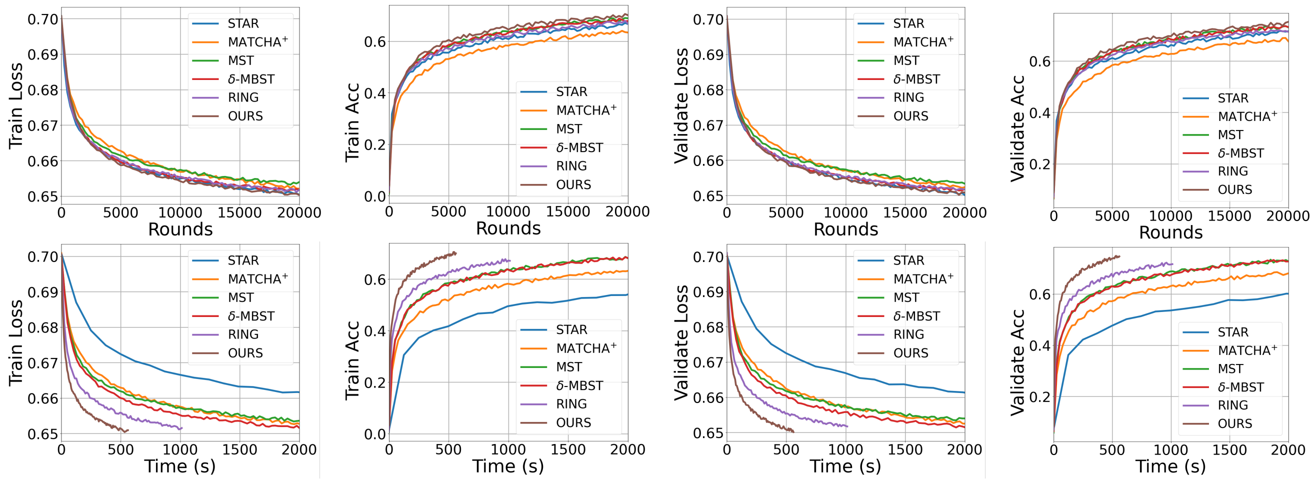
(a) Train Loss (b) Train Acc (c) Validate Loss (d) Validate Acc

Figure 1. The comparison of our multigraph and other methods on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row). Amazon network and FEMNIST dataset are used. The wall-clock time is counted until the training process of all setups reaches $6,000$ communication rounds. Best viewed in color.



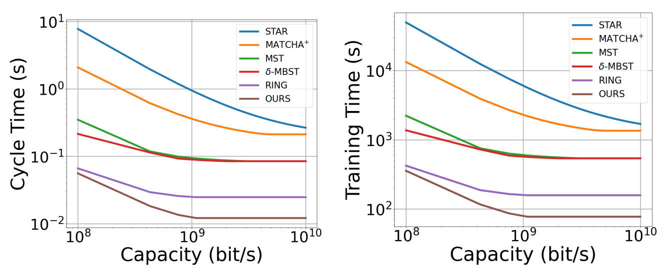
(a) Train Loss (b) Train Acc (c) Validate Loss (d) Validate Acc

Figure 2. The comparison of our multigraph and other methods on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row). Exodus network and iNaturalist dataset are used. The wall-clock time is counted until the training process of all setups reaches $1,500$ communication rounds. Best viewed in color.

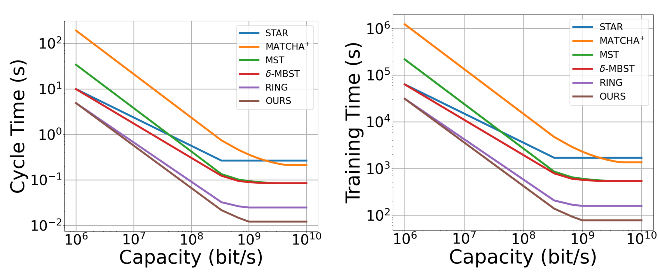


(a) Train Loss (b) Train Acc (c) Validate Loss (d) Validate Acc

Figure 3. The comparison of our multigraph and other methods on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row). Exodus network and Sentiment140 dataset are used. The wall-clock time is counted until the training process of all setups reaches $20,000$ communication rounds. Best viewed in color.



(a) Homogeneous access link capacity



(b) Orchestra node with 10 Gbps access capacity

Figure 4. The effect of access link capacity on cycle time and training time of different approaches. (a) All access links have the same 1 Gbps capacity. (b) One orchestra node has a fixed 10 Gbps access link capacity. The training time is counted until the training process of all setups reaches $6,400$ communication rounds.