Privacy via Scheduling and Connectivity Design in Decentralized Federated Learning

Feng Wang[*]
Google Cloud Platform
fengwaang@google.com

Zixi Wang
Syracuse University
zwang227@syr.edu

M. Cenk Gursoy
Syracuse University
mcgursoy@syr.edu

Senem Velipasalar
Syracuse University
svelipas@syr.edu

Abstract

In gradient-based distributed learning, inference attacks typically encounter significant challenges in reconstructing from combined gradients from a large number of samples that is tens or hundreds of times more than the output size. We notice that such combinations can be achieved by designing specific network topologies with limited number of connections and scheduling clients into sequentially activated subsets in decentralized federated-learning (FL). With this motivation, we propose a novel network topology and connectivity design that optimizes the trade-off between the training performance and privacy protection. We analytically demonstrate the convergence of the proposed decentralized learning, and quantify the privacy leakage via the entropy from the adversary's perspective. Furthermore, we show example topologies that effectively address the trade-off between eliminating privacy leakage and ensuring training convergence. Finally, we validate the performance of our cyclic topology against traditional FL.

1 Introduction

In federated learning (FL), distributed clients transmit gradients instead of their local data to preserve privacy. Among variants of FL, centralized FL (CFL) assumes that one central server orchestrates the training process, while in decentralized FL (DFL) clients exchange model parameters with each other in a peer-to-peer manner. Although data is kept local, inference attacks that reconstruct data from gradients have emerged as a privacy threat to FL. For instance, fully connected DFL (fcDFL) assumes that each client sends the updated model parameters to every other client in every global iteration (see Yuan et al. [2024] and references therein). General topologies require frequent transmissions to neighbors and strong connectivity to maximize the training performance. However, such frequent and dense connectivity makes the clients' gradients become vulnerable to attacks especially if there is an adversarial out-neighbor node. The existing defensive strategies exhibit only sub-optimal trade-offs between maintaining strong training performance and mitigating inference attacks. In this paper, we introduce novel scheduling and network connectivity designs for DFL that achieves the ideal balance between training and privacy protection, and demonstrate the performance against traditional FL.

Our main contributions include the following: 1) We analyze the trade-off in DFL connectivity, and achieve a complete defense against inference attacks while ensuring guaranteed training performance. 2) In contrast to CFL, we do not require any central server and thus avoid the cost of long-distance

[*] Work done during Ph.D. study at Syracuse University.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI and ML for Next-Generation Wireless Communications and Networking (AI4NextG).

transmissions by enabling peer-to-peer exchanges. 3) Compared to conventional DFL, we require significantly less transmission payload as there are less connections. 4) Compared to both CFL and conventional DFL, we have free control on the “batch size". If there is a tight budget on the total training time, requiring many clients to train simultaneously, our topology can explore many gradient directions simultaneously and avoid local minimums.

2 Related Work

2.1 Inference Attacks

One of the major threats to FL clients' privacy is that an adversarial server or client can potentially reconstruct the original data from the gradients. While some prior work focused on reconstructing the labels Melis et al. [2019], Wainakh et al. [2022], more interest has been drawn to reconstructing the inputs. The majority of such attacks assume an “honest-but-curious" central server that coordinates the FL training process and tries to reconstruct inputs from the gradients. The optimization-based attacks, including “deep leakage" Zhu et al. [2019], “gradient inversion" Geiping et al. [2020], and others Wang et al. [2019], Zhao et al. [2020] use random noise as dummy input and label, generate dummy gradients, and iteratively update the dummy input to increase the similarity between the dummy gradients and the true gradients. Other “honest-but-curious" attacks Yin et al. [2021], Hatamizadeh et al. [2022], Li et al. [2022] reconstruct high-resolution input images with pre-trained networks, and assume strong correlation between the distributions of attacker's training data and the clients' data. Furthermore, we notice that all these aforementioned attacks also apply to most DFL schemes including fcDFL, since each potential adversarial client receives individual gradients from a single neighbor.

Several other attacks assume a “malicious" central server that manipulates the neural network structure or parameters, and the majority of these attacks are detectable by suspecting clients. For instance, the studies in Fowl et al. [2021], Boenisch et al. [2021] construct extremely large fully-connected (FC) layers in a convolutional neural network (CNN). The work in Pasquini et al. [2022] and the feature fishing attack in Wen et al. [2022] require the same client to apply different malicious parameters on different samples. The study in Lam et al. [2021] sets the sample index as input (which is unnecessary for FL). Some malicious server attacks limit the manipulation and only modify the parameters in the given neural network structure, such as class fishing Wen et al. [2022] that sets the weights in the last FC layer to all labels other than the target label to zero and eliminate the gradients of most samples, and MKOR Wang et al. [2024] that modifies a small subset of parameters to maximize the information in extracted features and reconstruct every sample. However, the majority of these malicious server attacks deteriorate severely in DFL when the victim parameters deviate from malicious settings. In any DFL topology with at least two in-neighbors for every client, each honest client computes the average of the received parameters before generating new gradients, so the designed malicious parameters are mixed with regular parameters. Therefore, the malicious attacks that are sensitive to parameter noises deteriorate severely due to the difficulty in eliminating the information unnecessary for reconstruction purposes.

2.2 Defense Against Inference Attacks

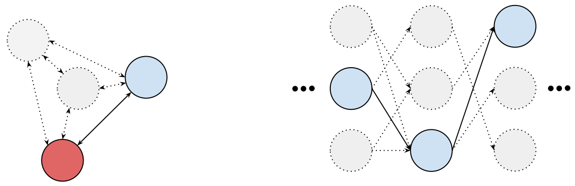
There are three major categories of defense against the aforementioned privacy attacks: encrypting gradients, encoding inputs, and perturbing gradients. Gradient encryption, such as homomorphic encryption Bonawitz et al. [2016] and secure multiparty computation Yao [1982], requires a specific and costly setup to encrypt gradients before sending. While this approach defends against third-party attackers such as eavesdroppers, it does not work if the central server or the out-neighbor is “honest-but-curious". Input encoding, such as MixUp data augmentation Zhang et al. [2017] and InstaHide Huang et al. [2020], proposes data augmentation and mixture with public dataset, respectively, to enhance model robustness and can be applied to FL. However, these methods are only applicable to image classification tasks and introduce significant accuracy loss in training Huang et al. [2021], Carlini et al. [2021]. Gradient perturbation injects additive noise to gradients to confuse the attacker, without assuming a credible third-party. For example, differential privacy Dwork et al. [2006, 2014] advocates adding Gaussian or Laplacian noise to gradients and achieve provable privacy bounds. Gradient pruning Zhu et al. [2019] sparsifies gradients by setting gradients with small magnitudes to zero. Soteria Sun et al. [2021] perturbs the data representation with the highest correlation to the input. In this paper, we consider these gradient perturbation methods as the benchmarks, and evaluate our defense against them via topology design.

3 Decentralized Federated Learning with Optimal Topology

Inference attacks, such as deep leakage and gradient inversion, enable the adversary to reconstruct the individual inputs from the average gradient, if the parameters are given and the number of samples is limited. Therefore, CFL that requires the uploading of the average gradient from a single client may lead to privacy leakage to an adversarial server. Similarly in fcDFL during each global iteration, each client receives individual gradients from every other client and uses the same parameters. For other types of DFL topology, the privacy concerns are not as extensively studied, and there is a lack of analysis on privacy-critical measures and considerations such as the number of in-neighbors and out-neighbors. However, one major constraint of such attacks is that the reconstruction performance drops significantly as the number of samples increases. Thus, to prevent privacy leakage, we propose a novel DFL topology design to “mix-up" the gradients from a large number of clients for any possible adversary.

3.1 Cyclic Topology Design

In this subsection, we introduce the underlying intuition behind our DFL topology design, as demonstrated in Fig. 1. We notice that existing inference attacks require not only the gradients but also the parameters that generated the gradients and the number of samples involved in the final gradient. In this paper, we assume that any client can be a potential attacker, and such an attacker may take a combination of parameters from the in-neighbors and its own parameter from the previous global iteration. To defend against such attacks, we desire a topology where any potentially adversarial client cannot easily separate any other individual gradients. Therefore, there should be limited connectivity in such a topology, so that any gradient information sent to multiple out-neighbors should not arrive at any certain client before being summed up with other gradients multiple times. In this way, the gradient of a victim client cannot be separated from other gradients, and the parameter that generated the gradient remains unknown to the adversary.


Every client shares similar parameter, and the victim's gradient can be easily separated.

Each client uses different parameter, and each individual gradient is highly mixed with others.

Figure 1: Intuition behind our topology design. Traditional FL such as fcDFL on the left side typically do not impose constraints on the connectivity, while our topology on the right side schedules specific connections to mitigate inference attacks.

Therefore, we consider a time-varying and strongly connected directed graph as a DFL cyclic topology with multiple sequentially activated subsets of clients as illustrated in Fig. 2. More specifically, considering a DFL task with U clients in total, we partition the clients into C columns (i.e., subsets), and each column includes R clients (i.e., $U = CR$), which operate simultaneously at a time, and forward their parameters to the next column of clients (which will be activated next). Thus, the connectivity matrix is time-varying. Specifically, we denote the edge weight matrix between two columns as $\boldsymbol{W}_c \in \mathbb{R}^{R \times R}$, where c is the operating column, and each element $\boldsymbol{W}_c[r_1, r_2]$ denotes the weight from (c, r_1) to $(c \,(\text{mod}\, C) + 1, r_2)$ where mod denotes the modulo operation. On the other hand, the adversary may take arbitrary linear combination of parameters received from all in-neighbors, so the maximum number of in-neighbors should be limited.

To ensure that the average model parameter on each column is updated by the average gradient from the previous column consistently, we choose a doubly stochastic matrix $\boldsymbol{W}_c$. Also, strong connectivity leads to sufficient information exchange among the clients and benefits training performance. Furthermore, a desirable topology should also ensure that the averaging happens sufficiently fast.

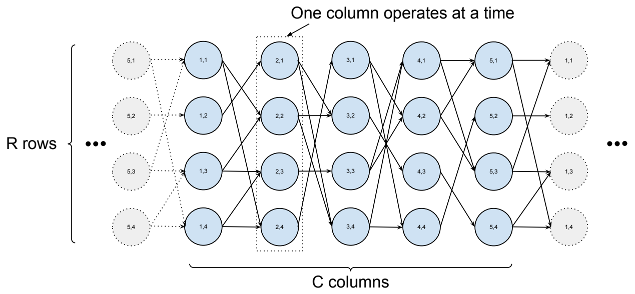


Figure 2: Schematic diagram of the proposed graph, only one column of R clients operate simultaneously at a time.

In the considered cyclic topology design, given the set of U clients, there are two trade-offs. One trade-off is in the choice of the number of columns C and the number of clients/rows R (as depicted in Fig. 2). As C becomes larger, the training performance and defense effectiveness improve. On the other hand, when R is larger, the global processing speed is faster. The other trade-off, as mentioned above, is between the training performance and defense performance. In particular, improved training performance needs more connectivity, while stronger defense performance requires less connectivity, and we need to balance the trade-off between the two performance requirements. In the appendix, we address these two requirements in detail by studying convergence (through upper bounds on the optimality gap and consensus) and privacy leakage (via entropy analysis).

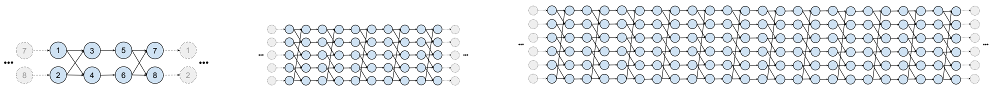
3.2 Designed Graph Topologies

In Appendix A and Appendix B, we have shown that a graph with good training performance should have a small value for the second largest eigenvalue e_2 of the cyclic matrix $\boldsymbol{W}_P = \prod_{c=1}^{C} \boldsymbol{W}_c$, and a graph with inherent protections against inference attacks should have a large value for the squared noise magnitude $\sum_{u' \neq u_v} |\boldsymbol{A}^t_{u_a}[u', t]|^2$, where $\boldsymbol{A}^t_{u_a} = \sum_{i \in \mathcal{I}_{u_a}} \alpha_i \boldsymbol{M}^t_{u_i^{in}}$ is an arbitrary linear combination chosen by the attacker client u_a with a set of in-neighbors $\mathcal{I}_{u_a}$ by setting arbitrary values for $\alpha_i \in \mathbb{R}$ to isolate the gradient from any single victim user u_v. Specifically, we assume that any client can be a potential adversary and any client can be a victim. Thus, we maximize $\min_{u_a, u_v} \sum_{u' \neq u_v} |\boldsymbol{A}^t_{u_a}[u', t]|^2$ to defend against any potential attacker. Overall, we define the following **optimization problem** combining two objectives:

$$\underset{\{\boldsymbol{W}_c | c \in [1, \ldots, C]\}}{\operatorname{argmin}} \left(e_2 - \beta \min_{u_a, u_v} \left(\sum_{u' \neq u_v} |\boldsymbol{A}^t_{u_a}[u', t]|^2 \right) \right) \tag{1}$$

where $\beta > 0$ is a constant. Above, the goal is to optimize the set of $\boldsymbol{W}_c$'s (edge weight matrices between consecutively activated columns of users) in designed topologies against the strongest attacker u_a.

Considering the given guidelines, we provide several examples of designed graph topologies based on the optimization goal. In the examples, we choose the out-neighbors of each client to have equal edge weight.



(a) Best graph by exhaustive search, $C = 4$, $R = 2$.

(b) Designed cyclic topology graph, $C = 10$, $R = 5$.

(c) Designed cyclic topology graph, $C = 24$, $R = 6$.

Figure 3: Graph topology examples.

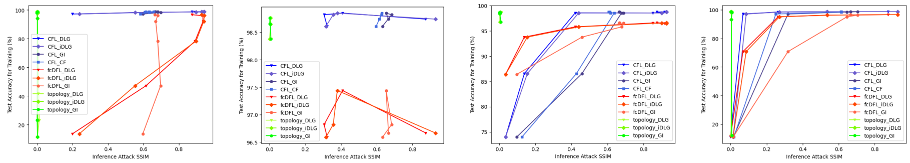
In Fig. 3, we show three example graphs with different numbers of clients. In Fig. 3(a), we demonstrate an optimized graph by exhaustively searching all possible graphs with 4 columns and 2 rows. This graph is the only one with $e_2 = 0$, and each client u_a has the same $\sum_{u' \neq u_v} |\boldsymbol{A}^t_{u_a}[u', t]|^2 = 2.5$, so clients are equally suppressed for privacy protection in case any client acts as an attacker. As an interpretation of this graph, each row behaves as a “bus" that transmits the gradient information of clients in this row, and different “buses" exchange information periodically. The trade-off between training and defense lies on exchanging information once during the activation of two consecutive columns of clients.

Similarly, we design the “rotation" graphs as in Fig. 3(b) and Fig. 3(c), with the intuition of “passing the accumulated gradients to the next bus" to mix the information. We also assume that each client has up to 2 in-neighbors and up to 2 out-neighbors, to prevent any potential attacker from combining many parameters and minimize $\boldsymbol{N} = \sum_{u' \neq u_v}^{U} \boldsymbol{A}^t_{u_a}[u', t]\boldsymbol{G}_{u'}$. In achieving good defensive performance, such a topology does not require the clients to be in physical proximity and have extensive connections. The graph in Fig. 3(b) with $C = 10$, $R = 5$, has $e_2 = 0.3466$ and $\min_{u_a} \sum_{u' \neq u_v} |\boldsymbol{A}^t_{u_a}[u', t]|^2 = 7.5833$, and the graph in Fig. 3(c) with $C = 24$, $R = 6$ has $e_2 = 0.1780$ and $\min_{u_a} \sum_{u' \neq u_v} |\boldsymbol{A}^t_{u_a}[u', t]|^2 = 14.5982$. In both graphs, the strongest potential attackers are located in even-numbered columns. Clearly, the minimum sum, $\min_{u_a} \sum_{u' \neq u_v} |\boldsymbol{A}^t_{u_a}[u', t]|^2$, increases as the number of clients increases, leading to better defensive performance. In the computer vision experiments of the next section, we use even larger graphs, and achieve an excellent trade-off.

4 Experiments

In this section, we evaluate the training-defense trade-off of our cyclic topology designs for DFL on computer vision models.

We consider the MNIST dataset Deng [2012] and Fashion MNIST dataset Xiao et al. [2017] with LeNet5 model LeCun et al. [1998], and the CIFAR-10 dataset Krizhevsky et al. [2009] with ConvNet model Sun et al. [2021]. We perform comparisons with baseline attacks including deep leakage from gradient (DLG) Zhu et al. [2019], improved DLG (iDLG) Zhao et al. [2020], gradient inversion (GI) Geiping et al. [2020], and class fishing (CF) Wen et al. [2022]. In each case, we compare with CFL and fcDFL and apply defensive strategies including differential privacy (DP) noise Dwork et al. [2014], Soteria Sun et al. [2021], quantization, and sparsification Zhu et al. [2019], and show the superior performance with our cyclic topology in all cases.



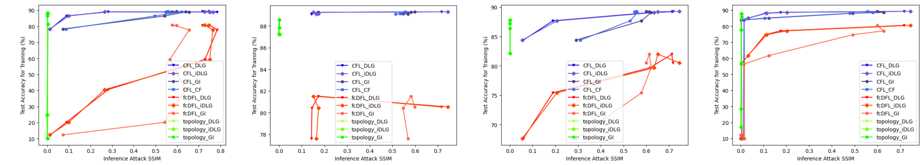
(a) DP noise (b) Soteria (c) Quantization (d) Sparsification

Figure 4: Quantitative comparison of SSIM between different FL architectures, inference attacks and defensive strategies for LeNet5 on MNIST. Optimal trade-off is on top left.

First, we consider training LeNet5 on MNIST, and show the curves of test accuracy for training vs. inference attack performance in terms of structural similarity index measure (SSIM) in Fig. 4. Different points on a curve correspond to different perturbation levels, and they are either DP noise, Soteria prune ratio, quantization bits, or sparsification rate in the respective curves. Higher values of SSIM indicate better reconstruction and worse defense, therefore the best trade-off among FL architectures and defensive strategies lies on the top left of each curve.

In the case without noise, CFL has slightly better training performance than our cyclic topology. However, when perturbation is applied for better defense against inference attack, CFL training performance degrades to the lower left significantly before it reaches a good defense (except Soteria which fails to achieve comparable defense). On the other hand, the reconstruction performance against our topology already reaches zero even without perturbation, as it aggregates gradients from many clients. Therefore, applying perturbations affects only the training performance (indicated by the vertical decrease). Therefore, our topology has better privacy-utility trade-off as it frees the clients from aggressive perturbation applied for privacy concerns.

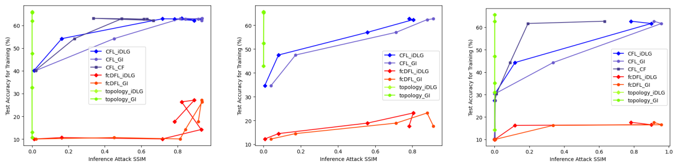
Obviously, DFL with our cyclic topology outperforms traditional FL in all situations. Furthermore, we note that CF fails compared to fcDFL and topology-based DFL, because it cannot preset parameters for all victim clients. Also, the reconstruction performance of Soteria is bounded since it only focuses on one layer, and traditional FL with Soteria does not surpass the defense performance of our topology-based DFL.



(a) DP noise (b) Soteria (c) Quantization (d) Sparsification

Figure 5: Comparison of SSIM between different FL architectures, inference attacks and defensive strategies for LeNet5 on FashionMNIST. Optimal trade-off is on top left.

Similarly, we have the results of LeNet5 on FashionMNIST in SSIM in Fig. 5, . Among different attack methods, our topology stays on the optimal top left point of the right-most curve in each defense strategy.



(a) DP noise (b) Quantization (c) Sparsification

Figure 6: Comparison of SSIM between different FL architectures, inference attacks and defensive strategies for ConvNet on CIFAR-10. Optimal trade-off is on top left.

Finally, we show the performance of ConvNet on CIFAR-10 in SSIM in Fig. 6 . In the previous two sets of experiments, we assume that each client uses its local momentum from previous global iterations. In this experiment, each client transmits both the momentum and the parameter with the same edge weight to reduce the delay of updates on momentum directions. Specifically, we consider the ConvNet without backbone (pre-trained feature extractor) as in Sun et al. [2021], and our fine-tuned test accuracy (63% for CFL and 66% for topology) are much higher than the test accuracy in the original paper (57% for CFL). DFL with our cyclic topology outperforms CFL even without noise since it explores different gradient directions simultaneously, and is more likely to avoid local minimums. Thus, when defense is considered, our designed topology still provides the best trade-off. In Fig. 6, DLG and CF reconstructions fail to converge and are not shown, but GI and iDLG still work. Also, there is no Soteria defense, because it exceeds 11GB RAM memory of our GPU for each client, and may not be practical for edge devices.

In Fig. 7, we compare the original and reconstructed images with different datasets, models, FL architectures, and attack methods without defensive perturbation. While traditional FL completely fails to prevent inference attacks, our cyclic topology achieves perfect defense even without perturbation.

5 Conclusion

We have introduced a novel design of DFL topology with optimized connectivity by measuring the trade-off between training performance and defense against inference attacks. In the appendix, we obtained several analytical results. For training performance, we proved that the upper bound for the regret of loss $\mathcal{L}(\bar{\boldsymbol{\theta}}^{(k)}) - \mathcal{L}(\boldsymbol{\theta}^*)$ diminishes to 0 after sufficient rounds of training. We quantified the privacy leakage using the entropy from the adversary's perspective, and showed that the distortion $\boldsymbol{N} = \sum_{u' \neq u_v}^{U} \boldsymbol{A}_{u_a}^{t}[u', t]\boldsymbol{G}_{u'}$ prevents the attacker from obtaining information from target gradients. By combining these two requirements, we showed that cyclic topology-based DFL outperforms

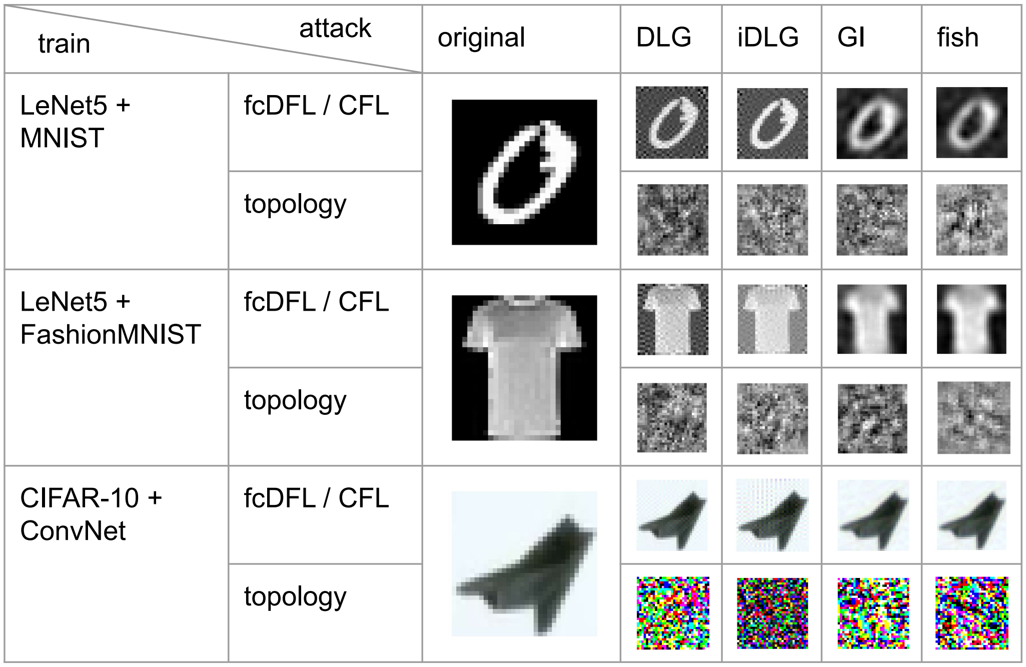


Figure 7: Reconstructed images without defensive perturbation.

traditional FL schemes on multiple models and datasets in the presence of several different inference attacks and defense methods.

References

Franziska Boenisch, Adam Dziedzic, Roei Schuster, Ali Shahin Shamsabadi, Ilia Shumailov, and Nicolas Papernot. When the curious abandon honesty: Federated learning is not private. *arXiv preprint arXiv:2112.02918*, 2021.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*, 2016.

Nicholas Carlini, Samuel Deng, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, Abhradeep Thakurta, and Florian Tramèr. Is private learning possible with instance encoding? In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 410–427. IEEE, 2021.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Liam Fowl, Jonas Geiping, Wojtek Czaja, Micah Goldblum, and Tom Goldstein. Robbing the fed: Directly obtaining private data in federated learning with modified models. *arXiv preprint arXiv:2110.13057*, 2021.

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.

Ali Hatamizadeh, Hongxu Yin, Holger R Roth, Wenqi Li, Jan Kautz, Daguang Xu, and Pavlo Molchanov. Gradvit: Gradient inversion of vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10021–10030, 2022.

Hsiang Hsu, Natalia Martinez, Martin Bertran, Guillermo Sapiro, and Flavio P Calmon. A survey on statistical, information, and estimation—theoretic views on privacy. *IEEE BITS the Information Theory Magazine*, 1(1):45–56, 2021.

Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. Instahide: Instance-hiding schemes for private distributed learning. In *International conference on machine learning*, pages 4507–4518. PMLR, 2020.

Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating gradient inversion attacks and defenses in federated learning. *Advances in Neural Information Processing Systems*, 34:7232–7241, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009.

Maximilian Lam, Gu-Yeon Wei, David Brooks, Vijay Janapa Reddi, and Michael Mitzenmacher. Gradient disaggregation: Breaking privacy in federated learning by reconstructing the user participant matrix. In *International Conference on Machine Learning*, pages 5959–5968. PMLR, 2021.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Zhuohang Li, Jiaxin Zhang, Luyang Liu, and Jian Liu. Auditing privacy defenses in federated learning via generative gradient leakage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10132–10142, 2022.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 691–706. IEEE, 2019.

Dario Pasquini, Danilo Francati, and Giuseppe Ateniese. Eluding secure aggregation in federated learning via model inconsistency. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2429–2443, 2022.

Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9311–9319, 2021.

Aidmar Wainakh, Fabrizio Ventola, Till Müßig, Jens Keim, Carlos Garcia Cordero, Ephraim Zimmer, Tim Grube, Kristian Kersting, and Max Mühlhäuser. User-level label leakage from gradients in federated learning. *Proceedings on Privacy Enhancing Technologies*, 2022(2):227–244, 2022.

Feng Wang, Senem Velipasalar, and M Cenk Gursoy. Maximum knowledge orthogonality reconstruction with gradients in federated learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3884–3893, 2024.

Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2512–2520. IEEE, 2019.

Yuxin Wen, Jonas Geiping, Liam Fowl, Micah Goldblum, and Tom Goldstein. Fishing for user data in large-batch federated learning via gradient magnification. *arXiv preprint arXiv:2202.00580*, 2022.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Andrew C Yao. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 160–164. IEEE, 1982.

Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16337–16346, 2021.

Liangqi Yuan, Ziran Wang, Lichao Sun, Philip S. Yu, and Christopher G. Brinton. Decentralized federated learning: A survey and perspective. *IEEE Internet of Things Journal*, pages 1–1, 2024. doi: 10.1109/JIOT.2024.3407584.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

A Convergence Analysis

In this section, we analytically characterize the training performance by establishing upper bounds for the optimality gap and the consensus difference. We prove that with the proper design of connectivity between neighboring columns, the proposed algorithm would converge to an optimal solution.

A.1 Problem Formulation

We aim at $\boldsymbol{\theta}^* = \text{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{\theta})$ where $\mathcal{L}(\cdot)$ denotes the universal loss function.

In the cyclic topology, we index the number of global cycles by t and the sub-iteration by $k = C(t-1) + c$, where c is the current operating column, and k increments by 1 whenever a column of clients complete their operation. We define $\bar{\boldsymbol{\theta}}^{(k)} = \frac{1}{R} \sum_{r=1}^{R} \boldsymbol{\theta}_r^{(k)}$ as the average model parameter for the corresponding operating column at iteration k. With the doubly stochasticity of the transition matrices $\boldsymbol{W}_c$, we note that $\bar{\boldsymbol{\theta}}^{(k+1)} = \bar{\boldsymbol{\theta}}^{(k)} - \lambda^{(k)} \bar{\boldsymbol{G}}^{(k)}$, where $\lambda^{(k)}$ is the learning rate of the clients at sub-iteration k, and $\bar{\boldsymbol{G}}^{(k)} = \frac{1}{R} \sum_{r=1}^{R} \boldsymbol{G}_{(c,r)}^{(k)}$ is the average gradient, with c denoting the operating column at sub-iteration k.

In the following sections, we adopt the common assumptions that the loss function $\mathcal{L}(\cdot)$ is L-smooth and satisfies the Polyak-Lojasiewicz (PL) condition, and the standard assumptions of bounded gradients and loss.

A.2 Optimality Gap

The optimality gap $\mathcal{L}(\bar{\boldsymbol{\theta}}^{(k)}) - \mathcal{L}(\boldsymbol{\theta}^*)$ characterizes the difference between the loss value achieved by the average model parameter of the operating column of clients and the optimal loss. We show that this gap diminishes to 0, and thus the proposed framework achieves the optimal loss value.

We first establish an upper bound for the term $\mathcal{L}(\bar{\boldsymbol{\theta}}^{(k+1)}) - \mathcal{L}(\boldsymbol{\theta}^*)$, relating to $\mathcal{L}(\bar{\boldsymbol{\theta}}^{(k)}) - \mathcal{L}(\boldsymbol{\theta}^*)$ and characterizing how the optimality gap evolves as k increases.

If the assumptions mentioned above hold, we have

$$\begin{aligned} \mathcal{L}(\bar{\boldsymbol{\theta}}^{(k+1)}) - \mathcal{L}(\boldsymbol{\theta}^*) \leq (1 - \mu(1+\xi_1)\lambda^{(k)}) \times& \\ (\mathcal{L}(\bar{\boldsymbol{\theta}}^{(k)}) - \mathcal{L}(\boldsymbol{\theta}^*)) + {\lambda^{(k)}}^2 \xi \ \ , \forall k \geq k_1 \end{aligned} \tag{2}$$

where $\xi_1 > 0$ is a bounded constant and ξ is a constant. μ and L arise from the PL condition and the L-smooth assumption on $\mathcal{L}(\cdot)$, respectively. k_1 is defined such that $\lambda^{(k)} \leq \frac{\mu}{L^2}$ and $\lambda^{(k)} \leq \frac{1}{\mu}, \quad \forall k \geq k_1$.

We note that (2) depicts the rate at which the optimality gap decreases in terms of the learning rate $\lambda^{(k)}$ and μ (which is some constant as defined by the PL condition).

Theorem A.1. *If $\lambda^{(k)} < \frac{1}{L}$ for $\forall k$, $\sum_{k=1}^{\infty} \lambda^{(k)} = \infty$ and $\sum_{k=1}^{\infty} \left(\lambda^{(k)}\right)^2 < \infty$, then we have*

$$\lim_{k \to \infty} \mathcal{L}(\bar{\boldsymbol{\theta}}^{(k)}) = \mathcal{L}(\boldsymbol{\theta}^*) \tag{3}$$

The requirements we impose above for the framework to converge to the optimal loss essentially indicate that the learning rate should be a sequence decreasing at a moderate speed. For instance, one may set $\lambda^{(k)}$ to decay proportional to $\frac{1}{k}$.

A.3 Consensus Difference

The consensus difference $\sum_{r=1}^{R} ||\boldsymbol{\theta}_r^{(k)} - \bar{\boldsymbol{\theta}}^{(k)}||$ characterizes how different the clients' local model parameters are from the average. We analytically characterize the relationship between the consensus difference and the designed topology.

We first define the cyclic matrix $\boldsymbol{W}_P = \prod_{c=1}^{C} \boldsymbol{W}_c$ to be the product of all the transition matrices within one global iteration (or equivalently C sub-iterations). In the following analysis, we assume $\boldsymbol{W}_P$ is diagonalizable due to the design of $\boldsymbol{W}_c$. The matrix $\boldsymbol{W}_P$ characterizes the transition matrix of one full global iteration starting from the first column.

Proposition 1. The consensus difference of the operating column at sub-iteration K is upper bounded as follows:

$$\sum_{r=1}^{R} ||\boldsymbol{\theta}_r^{(K)} - \bar{\boldsymbol{\theta}}^{(K)}|| \leq 2R^2(R-1) \sum_{k=1}^{K} \lambda^{(k)} ||G^{(k)}|| e_2{}^{\lfloor \frac{K-k}{C} \rfloor} \tag{4}$$

where $\lfloor \cdot \rfloor$ denotes the floor function, $||G^{(k)}||$ is the upper bound of the gradient norm for any stochastic gradient $\boldsymbol{G}_u^{(k)}$ generated in the corresponding sub-iteration, and e_2 is the second largest eigenvalue of the cyclic matrix $\boldsymbol{W}_P$.

We note that, as $\boldsymbol{W}_P$ is doubly stochastic, e_2, the second largest eigenvalue of $\boldsymbol{W}_P$, is upper bounded by 1. It is worth noticing that e_2 relates the consensus difference to the topology of the proposed DFL framework. The smaller e_2 is, the smaller the consensus difference is. Namely, the consensus difference is $O({e_2}^n)$, where $n = \left\lfloor \frac{K-k}{C} \right\rfloor$, the number of full cycles that have been completed.

B Entropy Analysis for Privacy Leakage

In this section, we introduce a privacy metric to evaluate the defensive performance against inference attacks. Specifically, we quantify the privacy leakage from the model parameters using entropy, and show that our topology eliminates privacy leakage when the number of clients is large.

Entropy is a measure of uncertainty of a random variable Cover [1999]. We analytically show the uncertainty level of user data at the inference attacker, and thus utilize entropy to quantify the privacy leakage Hsu et al. [2021] to an attacker. We note that if the conditional entropy of a private feature given the observation is equal to the original entropy then there is no privacy leakage (as concluded in (7) below).

During the training process, client u uses its local data set $\mathcal{D}_u$ to generate the local gradient $\boldsymbol{G}_u$, and hence an inference attacker with access to $\boldsymbol{G}_u$ obtains partial knowledge on $\mathcal{D}_u$.

Definition B.1. In traditional FL such as CFL and fcDFL, an adversary may receive individual gradients from victim u_v, and the adversary's uncertainty on $\mathcal{D}_{u_v}$ is described by the conditional entropy $H\left(\mathcal{D}_{u_v} \mid \boldsymbol{G}_{u_v}\right)$.

In DFL with cyclic topology, we denote the multiplier matrix of client u at global iteration t as $\boldsymbol{M}_u^t \in \mathbb{R}^{U \times t}$. The components of this matrix represent the coefficients of the gradients from every other client u' to client u, and depend on the edge weights in the graph topology. Therefore, the model parameter of user u at global iteration t is essentially the initial parameter $\boldsymbol{\theta}^{(0)}$ plus a weighted sum of all the gradients in the past:

$$\boldsymbol{\theta}_u^t = \boldsymbol{\theta}^{(0)} + \sum_{t'}^{t} \sum_{u'}^{U} \boldsymbol{M}_u^t[t', u'] \boldsymbol{G}_{u'}^{t'}. \tag{5}$$

We note that an attacker desires a mixture of small number of samples, which is equivalent to reconstructing a smaller batch. Therefore, an attacker with multiple in-neighbors can take a linear combination of the model parameters received from different in-neighbors with the goal to isolate the involved clients as much as possible. Thus, any attacker client u_a with a set of in-neighbors $\mathcal{I}_{u_a}$ may take an arbitrary linear combination $\boldsymbol{A}_{u_a}^t = \sum_{i \in \mathcal{I}_{u_a}} \alpha_i \boldsymbol{M}_{u_i^{in}}^t$ by setting arbitrary values for $\alpha_i \in \mathbb{R}$ to isolate the gradient from any single victim user. To distinguish a specific target victim client u_v, the attacker aims at an optimized set of coefficients $\{\alpha_i\}$ to eliminate or suppress the gradients from other clients. Specifically, this can be achieved if $\boldsymbol{A}_{u_a}^t[u_v, t] = 1$, while all other noises $\boldsymbol{A}_{u_a}^t[u', t'] \ \forall (u', t') \neq (u_v, t)$ are minimized.

Definition B.2. In DFL where individual gradients are not available to the adversary, the adversary's uncertainty on $\mathcal{D}_{u_v}$ is described by the conditional entropy $H\left(\mathcal{D}_{u_v} \mid \sum_{u'=1}^{U} \boldsymbol{A}_{u_a}^t[u', t] \boldsymbol{G}_{u'}\right)$.

Compared to $H\left(\mathcal{D}_{u_v} \mid \boldsymbol{G}_{u_v}\right)$, the gradients from other clients is a source of noise/distortion for the inference attack, and will lead to a larger value for $H\left(\mathcal{D}_{u_v} \mid \sum_{u'=1}^{U} \boldsymbol{A}_{u_a}^t[u', t] \boldsymbol{G}_{u'}\right)$, indicating larger uncertainty on the local dataset $\mathcal{D}_{u_v}$ at the adversary in DFL. To further quantify the increase in entropy in DFL, we derive the difference between the two entropy terms as

$$\begin{aligned} &H\left(\mathcal{D}_{u_v} \mid \sum_{u'=1}^{U} \boldsymbol{A}_{u_a}^{t}[u',t]\boldsymbol{G}_{u'}\right) - H\left(\mathcal{D}_{u_v} \mid \boldsymbol{G}_{u_v}\right) \\ =&H\left(\boldsymbol{G}_{u_v} \mid \sum_{u'=1}^{U} \boldsymbol{A}_{u_a}^{t}[u',t]\boldsymbol{G}_{u'}\right) \leq H\left(\boldsymbol{G}_{u_v}\right). \end{aligned} \tag{6}$$

Obviously, since conditioning reduces entropy, the difference above is upper-bounded by $H\left(\boldsymbol{G}_{u_v}\right)$, which is the maximum performance of any possible defense. While $\boldsymbol{A}_{u_a}^{t}[u_v,t] \triangleq 1$, the rest of the weighted gradient terms from other clients $\boldsymbol{N} = \sum_{u' \neq u_v}^{U} \boldsymbol{A}_{u_a}^{t}[u',t]\boldsymbol{G}_{u'}$ acts as noise/distortion in reconstructing $\mathcal{D}_{u_v}$. Therefore, the larger the noise variance is, the better the privacy protection is. For example, if we assume that each element of the gradients is i.i.d. Gaussian distributed $\boldsymbol{G}_u \sim \mathcal{N}(\mu, \sigma^2)$, and $\lim_{U\to\infty} \sum_{u' \neq u_v} |\boldsymbol{A}_{u_a}^{t}[u',t]|^2 = \infty$ (namely, the noise dominates the summation), we will further have the following characterization:

$$\lim_{U\to\infty} H\left(\boldsymbol{G}_{u_v} \mid \sum_{u'=1}^{U} \boldsymbol{A}_{u_a}^{t}[u',t]\boldsymbol{G}_{u'}\right) = H\left(\boldsymbol{G}_{u_v}\right). \tag{7}$$

With this result, we show that it is possible to achieve perfect defense with no privacy leakage via topology design.

With a finite number of clients, we in the aforementioned derivation proved that the “strength" of noise generated from other clients can be measured by the sum squared magnitude $\sum_{u' \neq u_v} |\boldsymbol{A}_{u_a}^{t}[u',t]|^2$, which characterizes the theoretical boundary of defense performance against any inference attack.

Furthermore, we note that it is even harder to attack DFL with a given topology than the result in (6), because the above entropy analysis does not take into account the training momentum and the difference between the training parameters of the attacker and the victims. In a practical DFL scenario, the real entropy difference is greater than that in (6), and additionally depends on the hyperparameters such as the learning rate and momentum factor.