

A EXPERIMENTAL DETAILS

- *CIFAR10* (Krizhevsky et al., 2009) consists of 60,000 32x32 color images across 10 classes (6,000 images per class), with 50,000 training and 10,000 test images. It represents a standard benchmark for image classification.
- *DomainNet* (Peng et al., 2019) contains objects across six domains (*sketch, real, quickdraw, painting, infographic, clipart*). We select ten classes (*suitcase, teapot, pillow, streetlight, table, bathtub, wine glass, vase, umbrella, bench*) based on sample availability. Unless stated otherwise, experiments use *real, sketch* and *clipart* domains.
- *Camelyon17* (Bandi et al., 2018) comprises histopathological images from multiple hospitals, with each hospital treated as a distinct domain. We create domain-specific test sets for binary tumor tissue classification.
- *Digit-Five* (Li et al., 2021b) combines MNIST (LeCun et al., 1998), SVHN (Netzer et al., 2011), USPS (Hull, 1994), SynthDigits and MNIST-M (Ganin & Lempitsky, 2015), with each dataset as a separate domain. Single-channel samples are extended to three channels by duplication.

Model Settings We use the **ResNet-10** model architecture for all our experiments. It consists of 4 residual blocks containing 2 consecutive convolutions, batch norm, and ReLU sub-blocks with residual connections. Based on our experiments, it displays similar behavior as the ResNet-18 (He et al., 2016) architecture while allowing us to run more users on the same hardware.

B SYSTEM DESIGN DETAILS

Listing 3. Universal API function calls for training

```

setup() # initializes the node
local_train() # one round of local training
receive() # pulls model from neighbors
aggregate() # averages param weights
    
```

B.1 Supported Network Topologies

Supported Collaboration Topologies

Ring: Nodes form a cycle, each connected to two neighbors.

Star: A central node connects to all others.

Grid: Nodes form a 2D lattice for structured communication.

Torus: A grid with wraparound connections, improving symmetry.

Fully Connected: Every node connects to all others, maximizing bandwidth.

Circle Ladder: A ring with additional cross-links forming a ladder-like structure.

Erdős-Rényi Graph: Randomized edges based on probability, modeling real-world networks.

Watts-Strogatz Network: A small-world model with local and long-range connections.

Tree: A hierarchical structure ideal for aggregation and spanning communication.

C INTER-DOMAIN COLLABORATION AND COLLABORATOR COLLAPSE

We view the problem of collaborator selection as designing and evolving the collaboration graph of users in a decentralized manner. However, the interplay between collaborator selection and the collaboration process (weight aggregation) makes the problem challenging. The edges in the collaboration network (the selected neighbors) alter the state of the network nodes (models). Consequently, changes in the models lead to changes in the collaboration network edges, creating interesting dynamics. Adaptive complex systems (Thurner et al., 2019) exhibiting such dynamics and have been studied extensively in their respective domains such as infectious diseases (Keeling & Eames, 2005), opinion dynamics (Peralta et al., 2022), and spin glass systems (Amit et al., 1985). In our case, local training of models (a non-linear function with a non-convex optimization) makes the problem difficult to analyze. Therefore, almost all the works have relied on heuristics. For instance, several works in centralized FL (Huang et al., 2019; Sattler et al., 2020; Duan et al., 2021) as well as decentralized FL (Fan et al., 2023; Onoszko et al., 2021; Sui et al., 2022; Li et al., 2022b) have proposed the idea of selecting users based on similarity between users.

Emergence of Collaborator Collapse: If collaborators are selected purely based on similarity, a feedback loop gets initiated where the same collaborator may be repeatedly selected. This results in a lack of diversity in collaborator selection, causing users to miss out on knowledge relevant to their task and ultimately limiting their performance. These

Table 3. Full performance overview (AUC) of all topologies with different numbers of collaborators. Reported as mean (std) over 3 runs.

Topology	Train			Test		
	1	2	3	1	2	3
Isolated	208.0 _(0.3)	208.0 _(0.3)	208.0 _(0.3)	44.5 _(6.9)	44.5 _(6.9)	44.5 _(6.9)
Central*	208.5 _(0.0)	208.5 _(0.0)	208.5 _(0.0)	33.97 _(14.2)	33.97 _(14.2)	33.97 _(14.2)
Random	205.4 _(1.0)	205.5 _(0.9)	205.9 _(0.8)	54.9 _(5.3)	56.0 _(5.8)	56.2 _(5.6)
Ring	198.8 _(3.1)	198.7 _(3.3)	198.7 _(3.4)	47.8 _(7.3)	46.9 _(6.4)	47.6 _(7.0)
Grid	202.6 _(1.5)	203.9 _(1.4)	204.5 _(1.3)	49.3 _(6.0)	48.8 _(6.0)	48.1 _(6.1)
Torus	202.0 _(1.2)	203.2 _(1.2)	204.0 _(1.3)	50.2 _(6.2)	50.7 _(6.6)	50.3 _(6.2)
Similarity based (top-k)	206.4 _(1.6)	197.6 _(7.3)	200.4 _(4.4)	47.3 _(8.3)	48.4 _(8.5)	52.8 _(7.2)
Swarm	183.2 _(3.5)	183.1 _(3.6)	183.2 _(3.6)	52.2 _(8.7)	52.3 _(8.7)	52.4 _(8.6)
L2C	167.0 _(25.4)	158.8 _(30.6)	152.8 _(35.2)	37.6 _(7.4)	36.6 _(7.4)	35.8 _(7.7)

Table 4. Performance overview(AUC) of various topologies with different number of collaborators.

Num Users	DomainNet		Camelyon17		Digit-Five	
	Within Domain	Random	Within Domain	Random	Within Domain	Random
12	56.6267	50.1772	178.3622	145.6398	57.1168	68.9724
18	59.5647	54.2480	179.5941	145.9916	66.8201	69.8341
24	61.8006	54.3855	178.5976	149.2037	71.6536	72.5333
30	66.5896	58.4835	179.1761	153.0658	74.4239	72.6996
39	68.3743	59.6090	179.1404	149.8618	163.8116	163.9892
45	68.1240	59.7852	180.0231	147.4649	77.0248	73.0634

Table 5. Performance comparison across datasets as system scales. Random collaboration effectiveness diminishes with increasing user count. Within Domain approach would require novel topology design algorithms (current approaches do not outperform random), as users lack ground truth domain information in practical settings. This result gives empirical evidence that better topologies can be designed.

dynamics are inherent to any similarity metric used between two models.

We perform several experiments and observe the collapse of collaborator diversity when a pair of users initially find each other similar and start collaborating exclusively, enhancing the likelihood of selecting each other repeatedly. This phenomenon happens either within or across domains. In the former scenario, the collapsed users lack diversity and are prone to overfitting their combined local data, hindering their ability to generalize to their actual domain distribution. In the latter case, arguably leads to worse performance as the collapsed users lack diversity and relevance. Similarly to the first scenario, they are likely to overfit; however, their local data is even less representative of their respective underlying domain distribution. Both scenarios are observed in Fig. 6 where users collapse into smaller groups instead of learning a single model per domain, missing relevant knowledge. Users collapsing between domains (points in the center) find themselves situated far from their domain optimal model.

C.0.1 Validation of Collaborator Collapse using Top-K:

To examine these dynamics, we introduce *Top K*, a greedy collaborator selection scheme that follows the protocol of similarity based selection. This scheme enables us to regulate the level of diversity in client selection through the

parameter K . In each round, every client evaluates and ranks all other users according to their estimated (potentially asymmetric) similarity. From the top K most similar users, $C(=1$ here) users are then randomly sampled with uniform probability.

In Fig. 9, we study the importance of the choice of K ¹. To accomplish this, we define an additional baseline: *Top K with domain knowledge* it assumes users know in which domain they are part of, and sample collaborator from a fixed set of K users within their domain when K is smaller than the domain size. *Similarity averaging* enhances the robustness of the similarity metric by computing its running average over the last $R(=10$ here) rounds.

When the value of K is much smaller than the domain size, users tend to *collapse* into small groups, consequently missing out on a significant portion of the knowledge available within their domain as seen in Fig. 6. On the other hand, as K increases, collaboration extends beyond the confines of the domain, ultimately hurting the performance. However, the optimal value for K during collaboration may differ from its ground truth value, which is the number of other

¹When domains have the same number of clients, it makes sense for all users to share the same value of K . However, in real-world situations, users are unlikely to be evenly distributed across domains. Consequently, each client would need to determine its optimal diversity threshold independently. This variability in K values contributes to increased heterogeneity within the system.

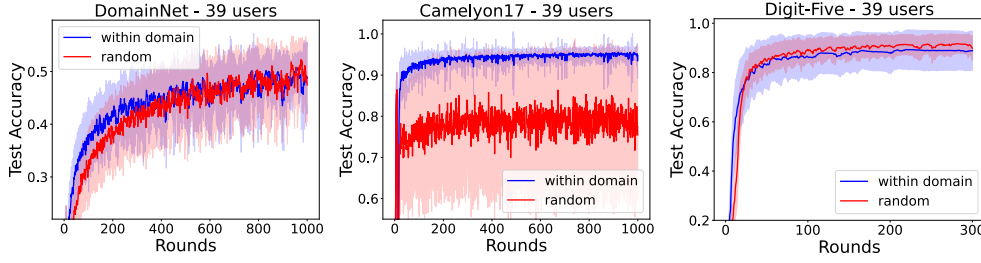


Figure 8. **Evolution of within-domain collaboration with increasing number of rounds.** We compare the convergence rate of the test accuracy for random and within-domain collaboration for a large number of rounds.

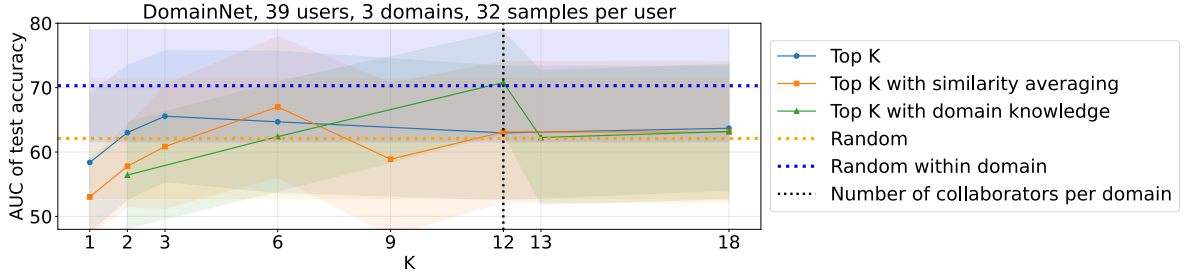


Figure 9. **Sensitivity of K in Top K .** The performance of top K is highly sensitive to the noise in the similarity metric. Even if we assume the domain sizes are known, finding the optimal K is not trivial. If K is too small, it can result in collaborator collapse, with performance dipping below that of random collaboration. Conversely, exceeding the optimal K threshold leads to inter-domain collaboration, hindering any potential benefits of collaborator selection.

users within the domain. The empirical value of K is closely intertwined with the noisiness of the similarity metric. In our case, representations, models, and similarity serve as sources of noise. As the similarity metric becomes noisier, the optimal K diminishes as the inherent noise already contributes to increased diversity. This trend is discernible in Fig Fig. 9: as similarity becomes more robust and accurate in finding same-domain users, the best K increases from 3 to 12, which corresponds to the number of other users within the same domain.

Any similarity score comparing the knowledge present in any two users should be reduced after a collaboration event takes place between the two users. Thus, in the next selection round, the closest users of the previous round are even closer and, thus, more likely to be selected again.

The observation we have made so far is that the similarity metrics can be imperfect. This imperfectness gets amplified as the users rely on using this similarity metric to select other users and update their model weights with selected users' parameters. This leads to a systemic collapse that either occurs among a small number of collaborators within a domain or users across different domains. Both collapses reduce the performance of the overall system. To the best of our knowledge, this phenomenon has not been reported before in the literature.

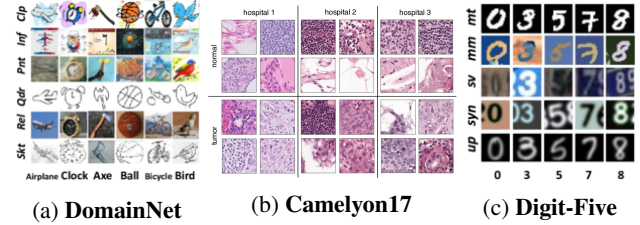


Figure 10. **Dataset Samples for Various Domains.** DomainNet contains objects across multiple representations, image [source](#). Camelyon17 histopathological images are split across hospital of capture, image [source](#). Digit-Five is a collection of digit images datasets, image [source](#).

D EXTENDED DISCUSSION: ROBUSTNESS AND PRIVACY MODULES

D.1 Robustness to Adversarial Behavior

In decentralized learning, robustness to malicious attacks is paramount due to the absence of centralized validation and control. Model updates aggregated from multiple users or devices are vulnerable to adversarial behavior such as *data poisoning*, *model poisoning*, and *backdoor attacks*. Fan et al. (Fan et al., 2023) propose consensus-based defenses for peer-to-peer (P2P) settings, but their effectiveness remains strongly topology-dependent. SONAR implements these attack families as modular components that can be toggled

or combined with defenses such as median and trimmed-mean aggregation, allowing controlled benchmarking of system-level resilience.

D.2 Gradient Inversion and Privacy Leakage

A critical privacy threat in both federated and decentralized learning is *gradient inversion*, where intermediate parameter or gradient exchanges can be exploited to reconstruct private training data (Zhu et al., 2019; Geiping et al., 2020; Zhao et al., 2020; Chen & Campbell, 2021; Zhu & Blaschko, 2021). The reconstruction is formulated as an optimization problem:

$$\arg \min_{\hat{X}, \hat{y}} \|\Delta \hat{W} - \Delta W\|^2 + \text{Reg}(\hat{X}, \hat{y}), \quad (1)$$

where $\Delta \hat{W}$ is the gradient computed on the dummy input $(\hat{X}, \hat{y})$ and ΔW is the intercepted update, with Reg denoting image or feature regularization. For algorithms that transmit parameters W_t , the gradient update can be approximated as $\Delta W_t = \frac{1}{\eta}(W_t - W_{t-1})$ given learning rate η . We primarily adopt the formulation from (Geiping et al., 2020) due to its extensibility across architectures, while also considering variants involving trap weights (Fowl et al., 2022), trap layers (Boenisch et al., 2023), and graph-based gradient inversion (Sinha et al., 2025). To evaluate decentralized reconstruction, we use the distributed reconstruction matrix method of (Mrini et al., 2024) to invert gradients from nodes several hops away.

D.3 Membership Inference

In addition to GIA, SONAR implements loss-based membership inference attacks (Hu et al., 2022), where nodes compare prediction entropy between in-training and held-out data to estimate exposure risk. These experiments quantify how topological sparsity, data heterogeneity, and communication frequency jointly modulate privacy leakage in decentralized settings.

Figures 12 and 7 illustrate how different topologies respond to the presence of malicious attackers in the collaborative learning system, as well as the volume of network connection the system logged in total.

E EXTENDED SECURITY MODULE: ROBUSTNESS AND PRIVACY

E.1 Robustness to Adversarial Behavior

Decentralized learning lacks centralized validation, making it vulnerable to malicious updates that can compromise global performance. SONAR integrates established attack and defense mechanisms for decentralized learning (Sharma & Marchang, 2024; Nguyen et al., 2023) through a modular interface that allows researchers to inject, combine,

and evaluate robustness strategies under diverse network conditions.

Data Poisoning. Implements label flipping, random label assignment, and input corruptions (e.g., CIFAR-10 (Hendrycks & Dietterich, 2019)) to simulate compromised local datasets.

Model Poisoning. Includes gradient- and backdoor-based attacks that alter optimization trajectories or implant hidden inference triggers.

Weight Manipulation. Supports aggregation-targeting attacks such as *free-riding* (submitting null updates) and *sign-flip* (gradient reversal) to test defense reliability.

Defenses. Provides robust aggregation mechanisms—median and trimmed-mean—to mitigate the influence of outlier updates. All defenses can be toggled or extended through the same interface, ensuring reproducibility and fair comparison across algorithms.

E.2 Privacy and Information Leakage

Built on the same instrumentation backbone, SONAR implements standardized modules for evaluating privacy risks in decentralized training, focusing on *gradient inversion* and *membership inference*.

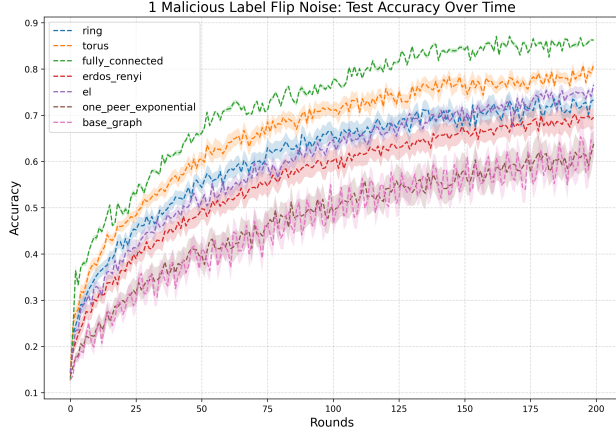
Gradient Inversion. Following (Geiping et al., 2020), the attacker reconstructs input data from exchanged gradients by solving:

$$\arg \min_x \left(1 - \frac{\langle \Delta \hat{W}, \Delta W \rangle}{\|\Delta \hat{W}\| \|\Delta W\|} \right) + \alpha TV(x), \quad (2)$$

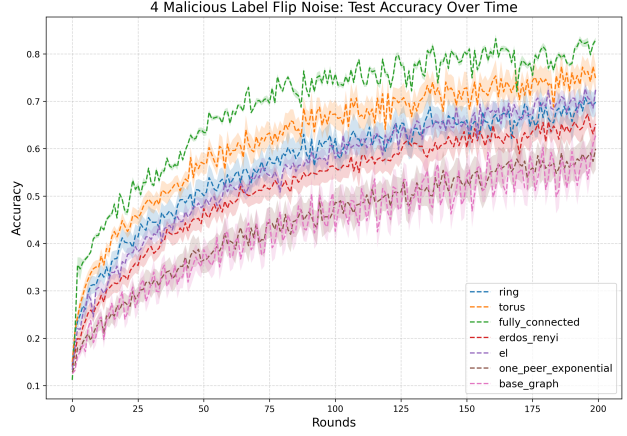
where cosine similarity measures gradient alignment and $TV(x)$ imposes a natural-image prior. To generalize to parameter-exchange settings, we approximate gradients as $\Delta W_t = \frac{1}{\eta}(W_t - W_{t-1})$ with learning rate η . We adopt the formulation from (Geiping et al., 2020) and include extensions such as trap weights (Fowl et al., 2022), trap layers (Boenisch et al., 2023), and distributed reconstruction matrices (Mrini et al., 2024) for multi-hop gradient inversion.

Membership Inference. Implements loss-based membership inference (Hu et al., 2022), where each node compares confidence and entropy metrics between training (IN) and held-out (OUT) data before aggregation. These measurements quantify per-node exposure risk and enable correlation with topology and data heterogeneity.

Operational Design. Both privacy and robustness components operate locally, require no modification to the learning



Test accuracy under noisy label attack with 1 malicious node



Test accuracy under noisy label attack with 4 malicious nodes

Figure 11. Test accuracy of noisy label attack with 1 vs 4 attackers

pipeline, and log all results through the telemetry stack for reproducible analysis. Together, they provide a unified and extensible framework for evaluating security trade-offs in decentralized learning.

E.3 Gradient Inversion Attack Results

We evaluate the vulnerability of different network topologies to gradient inversion attacks (GIA), where an adversary attempts to reconstruct training data from shared gradients. Our analysis reveals that network topology significantly influences reconstruction quality, with certain graph statistics showing strong correlations with attack success.

E.4 Decentralized GIA result at various network connectivity

16 showcases the reconstruction result at different levels of network connectivity on a small world graph.

Benchmarking topology and collaboration in Decentralized Machine Learning

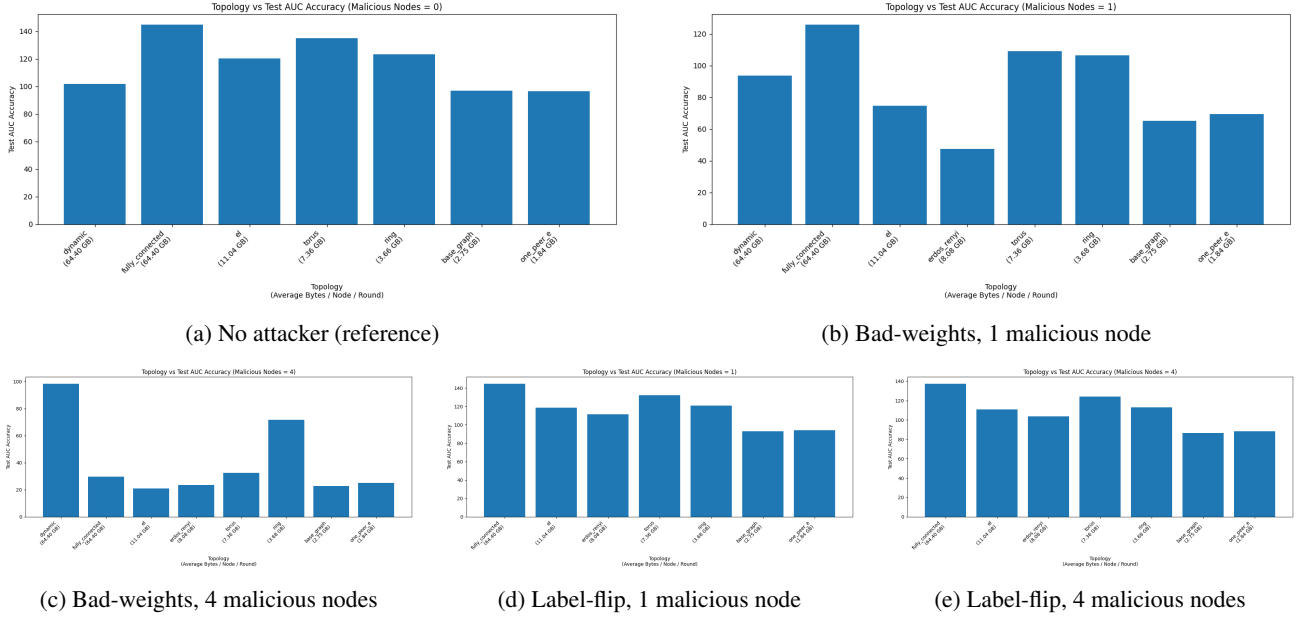


Figure 12. **Aggregate AUC under failures.** End-of-run AUC across topologies for clean reference and two attack families (bad-weights, label-flip) at 1 and 4 malicious nodes. Dense graphs incur the largest AUC loss; structured topologies are more resilient. (Curves in Fig. 7)



Figure 13. Original Dataset



Figure 14. Reconstruction

Figure 15. Gradient inversion attack. Reconstruction qualitative results

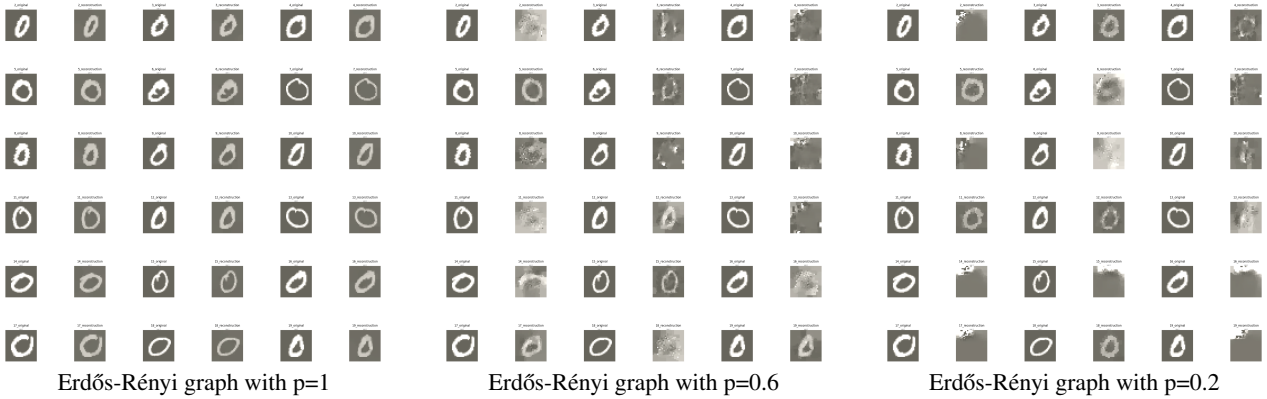


Figure 16. Decentralized gradient inversion attack reconstruction results on Erdős-Rényi graphs with 18 nodes, each holding a single MNIST image, using the technique from (Mrini et al., 2024). The fully connected graph ($p=1.0$, left) successfully reconstructs gradient images from all nodes, while sparser connectivity ($p=0.6$, center and $p=0.2$, right) increasingly obscures gradient signals, making it more difficult to separate and reconstruct the original input images.

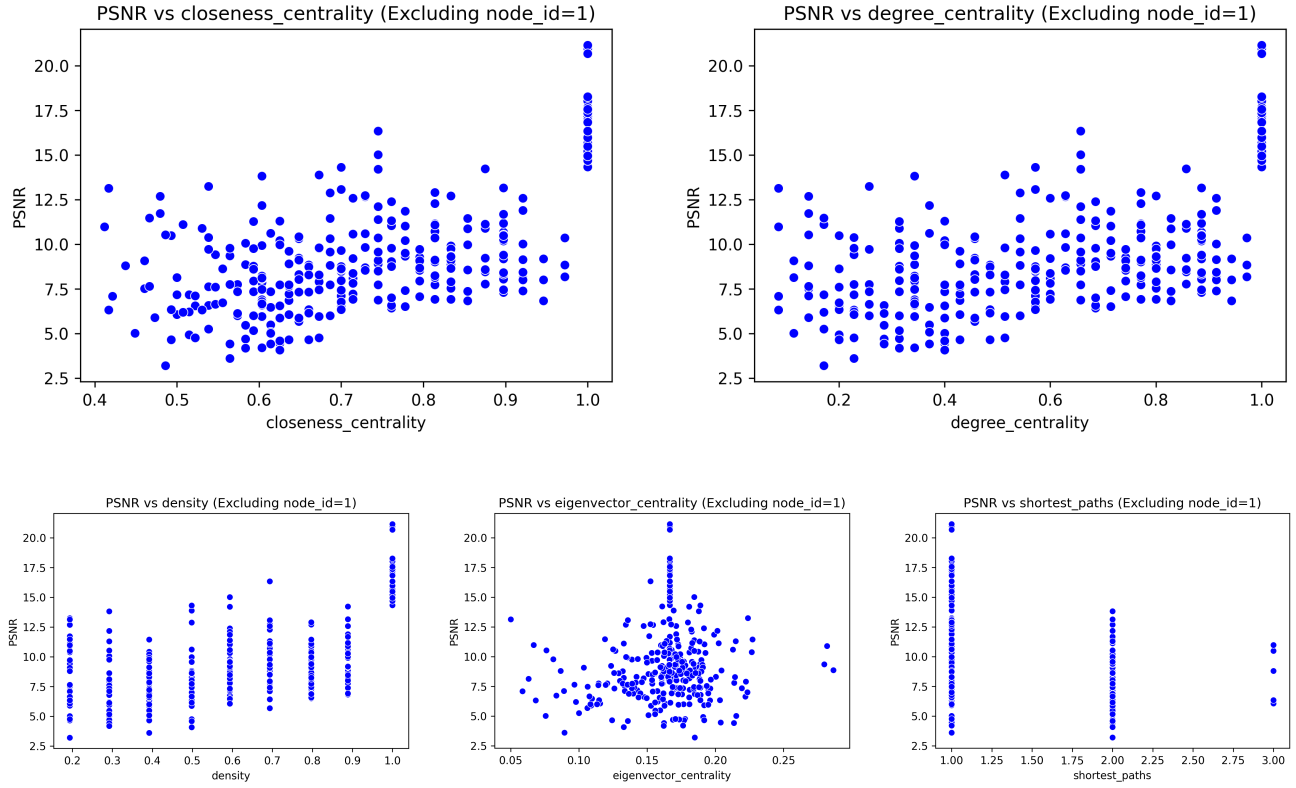


Figure 17. Comparison of Attacker's PSNR of reconstructed images against the communication graph statistic.

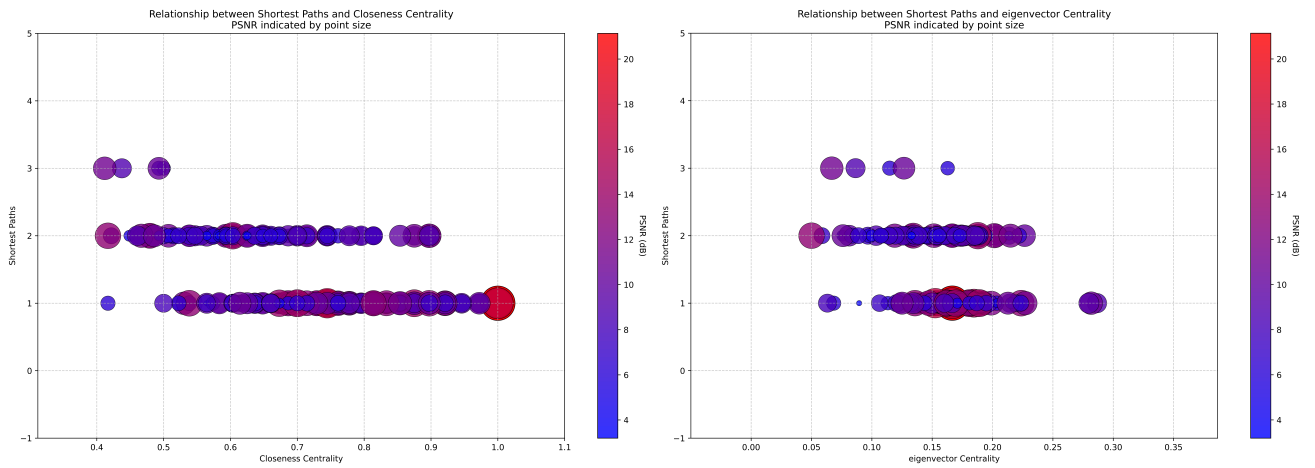


Figure 18. Comparison between graph statistics and its correlation with PSNR.

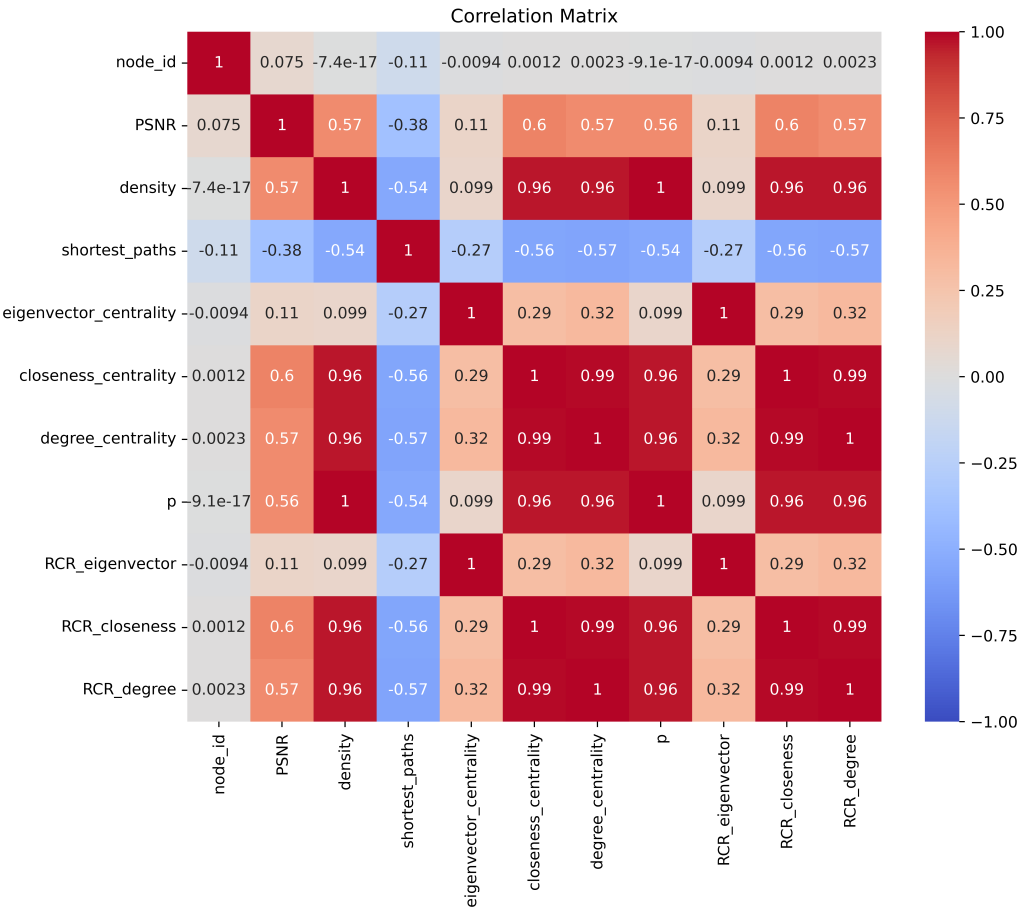


Figure 19. Correlation Matrix between GIA PSNR values and communication graph statistics.

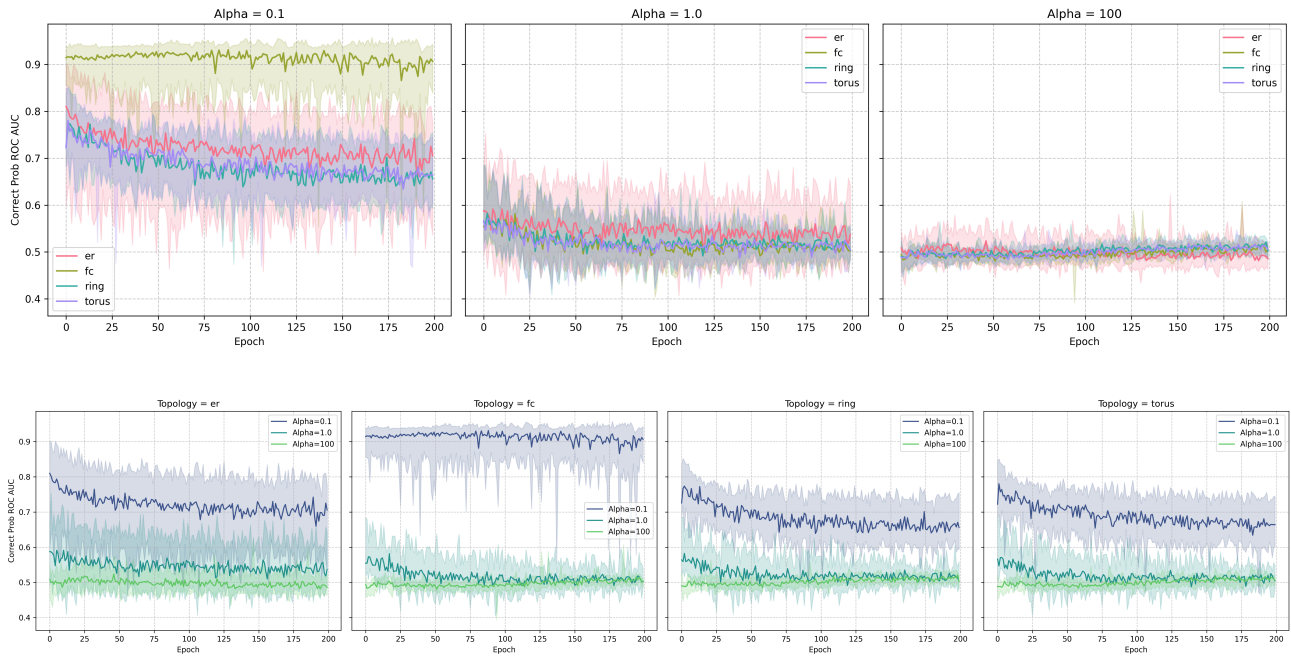


Figure 20. Membership Inference Attack evaluation. Top graph compares topologies when distribution similarities are fixed. Bottom graph compares distribution similarity when topology is fixed: As distributions between each participant becomes more and more non-IID, attack quality decreases for all topologies.