Local Graph-Homomorphic Privacy in Distributed Neural Networks

Sergio A. Cáceres-Romero[1 [0009−0006−2294−6296]], Renato Candido[1,2 [0000−0001−8442−0923]], and Magno T. M. Silva[1 [0000−0002−3206−1145]]

[1] Universidade de São Paulo, Escola Politécnica, São Paulo, SP, Brazil
[2] SiDi Intelligence & Innovation Center, Campinas, SP, Brazil
sergiocaceres01@usp.br, renatocan@lps.usp.br, magno.silva@usp.br.

Abstract. The growing concern over the privacy of sensitive data in the training of machine-learning models has driven the development of robust protection techniques. This work investigates *privacy-preserving* methods for *distributed neural networks*, with emphasis on adaptive diffusion techniques. *Random noise*, *global homomorphic*, and *local homomorphic* perturbations are compared in terms of classification utility and empirical information leakage. Experimental results on two real-world datasets show that local homomorphic noise preserves classification accuracy close to that of the non-private model. Moreover, the improved Deep Leakage from Gradients (iDLG) attack, local homomorphic noise substantially degrades the quality of the reconstructed data, as indicated by an increase in reconstruction error and a reduction in the structural similarity index. These results indicate a favorable empirical privacy–utility trade-off for local graph-homomorphic perturbation.

Keywords: Machine learning · Privacy · Distributed training · Sensitive data.

1 Introduction

Distributed learning has become an important paradigm for large-scale machine learning, enabling multiple agents to cooperatively solve optimization problems [17,26,27]. Among the existing approaches, adaptive diffusion networks have attracted attention due to their scalability, robustness, online learning capability, and the absence of a central fusion center [26,27]. More recently, diffusion strategies have also been extended to distributed neural network training [3, 20, 25].

Although diffusion-based learning avoids sharing raw training data, neighboring nodes continuously exchange intermediate model parameters that may reveal sensitive information. Recent studies have shown that gradient inversion attacks, such as the Deep Leakage from Gradients (DLG) and its improved version iDLG, can reconstruct training samples directly from the exchanged gradients [33, 34] [32]. Current evaluations show that the effectiveness of gradient inversion depends strongly on the model architecture, training configuration, batch size, and adversarial assumptions [8, 16]. In decentralized learning, the

communication topology and the adversary's position also determine which participants may be exposed [11]. Figure 1 illustrates this vulnerability using a distributed neural network trained on a single MNIST image [18]. Even without access to the local dataset, iDLG successfully reconstructs the original training sample, demonstrating that privacy is not inherently guaranteed.

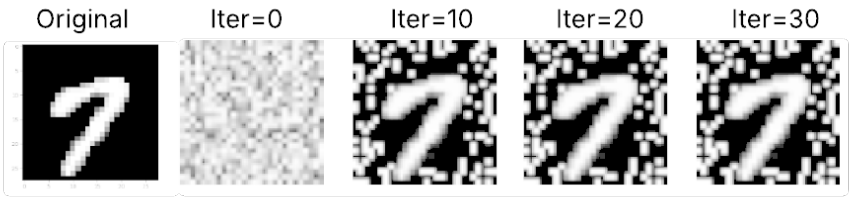


Fig. 1: Model input sample and its reconstruction using iDLG [33]. The simulation setup is described in Section 5.

Differential privacy has become one of the most popular approaches for protecting distributed learning by perturbing the exchanged information with carefully calibrated noise [9, 10]. However, this additional noise often degrades learning performance, motivating the development of more efficient privacy-preserving mechanisms. Recent works have proposed topology-aware perturbation schemes, including graph-homomorphic and local homomorphic noise [24], masked diffusion [13], nonzero-mean protection noise [15], compressive diffusion [31], and other differentially private distributed learning algorithms [14, 22]. These methods significantly improve the privacy–utility trade-off while preserving the performance of diffusion networks.

Despite the growing interest in privacy-preserving distributed and federated learning, an important gap remains in the literature. Graph-homomorphic perturbation has been studied for adaptive filtering and distributed estimation, with its use for distributed neural network training, particularly under diffusion strategies, remaining unexplored. To address this gap, this paper extends graph-homomorphic perturbation mechanisms [24] to distributed neural network training under the adapt-then-combine (ATC) diffusion framework. We investigate three privacy strategies: (i) independent random noise, (ii) global graph-homomorphic noise, and (iii) local graph-homomorphic noise, selected to provide a controlled comparison between conventional independent perturbation and graph-structured perturbation at global and local levels. The impact of these approaches is evaluated on both learning performance and privacy preservation by considering two datasets: MNIST [18] and the Fed-ISIC2019 medical dermoscopic-image dataset, used in a federated-learning setting [4, 5, 28]. A simple three-neuron example illustrates these mechanisms, whose effectiveness is evaluated against the iDLG attack. Furthermore, empirical leakage is esti-

mated using the mutual information neural estimation (MINE) framework [2]. To the best of our knowledge, this is the first study to investigate local graph-homomorphic perturbations specifically for distributed neural network training under an ATC diffusion strategy. The results provide an empirical characterization of the privacy–utility behavior of these mechanisms and establish a basis for their further investigation in distributed learning systems.

The remainder of this paper is organized as follows. Section 2 presents the diffusion framework for distributed neural network training. Section 3 reviews the privacy-preserving mechanisms and evaluates their communication and computational complexity. Section 4 briefly describes the MINE framework and the iDLG attack. Simulation results are presented in Section 5, and concluding remarks are given in Section 6.

2 Problem Formulation

Consider N neural networks with identical architectures and initialization, capable of communicating according to a predefined topology. Each network has access to a local training dataset $\mathcal{D}_k$ of cardinality $|\mathcal{D}_k|$, such that $\mathcal{D}_k \cap \mathcal{D}_\ell = \varnothing$ for all $k, \ell = 1, 2, \cdots, N$ with $k \neq \ell$. The neural networks also share a common test dataset, denoted by $\mathcal{D}_T$. The neighborhood of neural network k consists of all neural networks with which it can communicate directly, including itself, and is denoted by $\mathcal{N}_k$, with cardinality $|\mathcal{N}_k|$ [27].

The objective of distributed neural network training can be formulated as the following optimization problem [27]

$$\min_{\boldsymbol{\mathcal{W}}} J_{\text{global}}(\boldsymbol{\mathcal{W}}) = \min_{\boldsymbol{\mathcal{W}}} \textstyle\sum_{k=1}^{N} J_k(\boldsymbol{\mathcal{W}}, \mathcal{D}_k), \tag{1}$$

where J_k denotes the local cost function and $\boldsymbol{\mathcal{W}}$ represents the set of all neural network parameters. For a multilayer perceptron, for example, $\boldsymbol{\mathcal{W}}$ corresponds to the collection of weight matrices and bias vectors of all layers.

Considering the mean-squared error (MSE) as the cost function and a classification problem with C classes, one has

$$J_k(\boldsymbol{\mathcal{W}}_k, \mathcal{D}_k) = \textstyle\sum_{n_k=1}^{|\mathcal{D}_k|} \sum_{\ell=1}^{C} \left[d_{k,\ell}(n_k) - y_{k,\ell}(n_k, \boldsymbol{\mathcal{W}}_k) \right]^2, \tag{2}$$

where $y_{k,\ell}(n_k, \boldsymbol{\mathcal{W}}_k)$ denotes the ℓ-th output of neural network k for the n_k-th training sample, whose label is $d_{k,\ell}(n_k)$, and $\boldsymbol{\mathcal{W}}_k$ is the current combined estimate of the parameter set of that network. Since each neural network has access only to its own local cost function, cooperative mechanisms are required to minimize J_{global} through local interactions among neighboring networks.

Cooperation can be achieved in different ways, among which the ATC diffusion strategy is one of the most widely adopted [20,26,27]. This strategy consists of two steps. First, each neural network performs a local adaptation by updating its parameters based on the gradient computed from its own training data using, for instance, the backpropagation algorithm [3], i.e.,

$$\boldsymbol{\Psi}_k(m) = \boldsymbol{\mathcal{W}}_k(m-1) - \eta \frac{\partial J_k(\boldsymbol{\mathcal{W}}_k(m-1), \mathcal{D}_k)}{\partial \boldsymbol{\mathcal{W}}_k(m-1)}, \tag{3}$$

where $\boldsymbol{\Psi}_k$, having the same dimensions as $\boldsymbol{\mathcal{W}}_k$, denotes the set of local parameter estimates of neural network k, $m = 1, 2, \cdots, M$ is the iteration index, and η is the learning rate.

After each local update, the intermediate parameters are transmitted to all neighboring neural networks, followed by the combination step,

$$\boldsymbol{\mathcal{W}}_k(m) = \textstyle\sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\Psi}_\ell(m), \tag{4}$$

where $a_{\ell k}$ is the combination weight associated with the link from neural network ℓ to neural network k, i.e., the element located at row ℓ and column k of the $N \times N$ combination matrix $\mathbf{A}$.

Several rules for computing the combination weights have been proposed in the literature [26, 27]. In this work, the Metropolis rule is adopted, according to which the combination weights are given by

$$a_{\ell k} = \begin{cases} 1/\max\{|\mathcal{N}_\ell|, |\mathcal{N}_k|\}, & \ell \in \mathcal{N}_k \text{ and } \ell \neq k \\ 1 - \displaystyle\sum_{j \in \mathcal{N}_k \setminus \{k\}} a_{jk}, & \ell = k, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

where $\mathcal{N}_k \setminus \{k\}$ denotes the neighborhood of node k excluding itself.

The corresponding combination matrix $\mathbf{A}$ is symmetric and doubly stochastic, meaning that the elements in each row and each column sum to one. As will be shown in the next section, this property is essential for the privacy-preserving mechanisms revisited in this paper.

Under the diffusion strategy described above, each neural network uses only its own local training dataset and has no access to the training data of neighboring networks. Nevertheless, the algorithm still faces significant privacy challenges. To mitigate this issue, perturbations must be added to the shared parameters before the combination step, as discussed in the following section.

3 Privacy-Preserving Mechanisms

To ensure the privacy of information exchanged among agents, one may resort to a *privacy-preserving* algorithm. Such mechanisms inject noise into transmitted messages, ensuring that the outputs corresponding to neighboring datasets are statistically indistinguishable and preventing adversaries from inferring sensitive information [10]. In distributed learning, the privacy-preserving mechanism is applied to the messages exchanged among neural networks. Under the ATC diffusion strategy, network k transmits the local parameters $\boldsymbol{\Psi}_k(m)$, which depend on the gradient and may reveal information about the training data. Therefore, privacy can be ensured by perturbing either $\boldsymbol{\Psi}_k(m)$ or $\boldsymbol{\mathcal{W}}_k(m)$. In the following, three strategies for achieving this goal are revisited.

Random Noise Noise $\boldsymbol{\mathcal{G}}_{\ell k}(m)$, having the same dimensions as $\boldsymbol{\Psi}_\ell(m)$, is added to the intermediate estimate $\boldsymbol{\Psi}_\ell(m)$ shared between neural networks ℓ and k. The noise is drawn from a zero-mean Laplace distribution with variance σ_g^2 [21, 23]. After the adaptation step, each neural network k combines the intermediate estimates received from its neighbors according to

$$\boldsymbol{\mathcal{W}}_k(m) = \textstyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\left[\boldsymbol{\Psi}_\ell(m) + \boldsymbol{\mathcal{G}}_{\ell k}(m)\right].$$

As shown in [23] in the context of adaptive diffusion networks, differential privacy is guaranteed if the perturbation $\boldsymbol{\mathcal{G}}_{\ell k}(m)$ is properly calibrated.

Global Homomorphic Noise Although independently perturbing each transmitted message guarantees differential privacy, noise accumulates over the iterations and degrades the convergence of the diffusion algorithm. To alleviate this effect, [29] proposed a structured noise mechanism that exploits the network topology so that the perturbations cancel out during the ATC combination step.

Defining the network centroid as $\overline{\boldsymbol{\mathcal{W}}}(m) \triangleq (1/N)\sum_{k=1}^{N}\boldsymbol{\mathcal{W}}_k(m)$, and substituting (3) into (4), while using the doubly stochastic property of $\mathbf{A}$, yields

$$\begin{aligned}\overline{\boldsymbol{\mathcal{W}}}(m) = \overline{\boldsymbol{\mathcal{W}}}(m-1) &+ \frac{1}{N}\sum_{k=1}^{N}\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\mathcal{G}}_{\ell k}(m)\\ &- \frac{\eta}{N}\sum_{k=1}^{N}\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\frac{\partial J_\ell(\boldsymbol{\mathcal{W}}_\ell(m-1),\mathcal{D}_\ell)}{\partial\boldsymbol{\mathcal{W}}_\ell(m-1)}.\end{aligned} \tag{6}$$

To preserve the centroid dynamics, the structured noise must satisfy

$$\textstyle\sum_{k=1}^{N}\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\mathcal{G}}_{\ell k}(m) = \mathbf{0}. \tag{7}$$

This condition is achieved by generating independent zero-mean Laplace noise $\boldsymbol{\mathcal{G}}_k(m)$, $k = 1, 2, \ldots, N$, with variance σ_g^2 and defining [29]

$$\boldsymbol{\mathcal{G}}_{\ell k}(m) = \begin{cases}\boldsymbol{\mathcal{G}}_\ell(m), & \ell \neq k,\\ -\dfrac{1-a_{kk}}{a_{kk}}\,\boldsymbol{\mathcal{G}}_k(m), & \ell = k.\end{cases} \tag{8}$$

To illustrate this mechanism, consider three neural networks, each consisting of a single linear neuron, with a doubly stochastic combination matrix $\mathbf{A} = [a_{\ell k}]_{\ell,k=1}^{3}$, i.e., $\sum_{\ell=1}^{3} a_{\ell k} = 1, \quad \sum_{k=1}^{3} a_{\ell k} = 1, \quad \forall\, k, \ell \in \{1, \ldots, 3\}$. The corresponding network topology is shown in Fig. 2. Notice that the neighborhood of every node is given by $\mathcal{N}_k = \{1, 2, 3\}$. Let $\boldsymbol{\mathcal{G}}_k(n)$, $k = 1, 2, 3$, denote independent zero-mean noise vectors with variance σ_g^2 generated at each neural network.

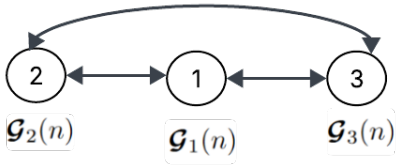


Fig. 2: Example of a fully connected network with $N = 3$ nodes. The vectors $\boldsymbol{\mathcal{G}}_k(n)$ denote the noise generated at each node.

The global noise term is

$$\mathbf{s}(n) = \sum_{k=1}^{3} \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\mathcal{G}}_{\ell k}(n). \tag{9}$$

Using (8), the contribution of each node is

$$\sum_{\ell \in \mathcal{N}_1} a_{\ell 1} \boldsymbol{\mathcal{G}}_{\ell 1}(n) = -(1 - a_{11}) \boldsymbol{\mathcal{G}}_1(n) + a_{21} \boldsymbol{\mathcal{G}}_2(n) + a_{31} \boldsymbol{\mathcal{G}}_3(n), \tag{10}$$

$$\sum_{\ell \in \mathcal{N}_2} a_{\ell 2} \boldsymbol{\mathcal{G}}_{\ell 2}(n) = a_{12} \boldsymbol{\mathcal{G}}_1(n) - (1 - a_{22}) \boldsymbol{\mathcal{G}}_2(n) + a_{32} \boldsymbol{\mathcal{G}}_3(n), \tag{11}$$

$$\sum_{\ell \in \mathcal{N}_3} a_{\ell 3} \boldsymbol{\mathcal{G}}_{\ell 3}(n) = a_{13} \boldsymbol{\mathcal{G}}_1(n) + a_{23} \boldsymbol{\mathcal{G}}_2(n) - (1 - a_{33}) \boldsymbol{\mathcal{G}}_3(n). \tag{12}$$

Adding these terms yields

$$\begin{aligned} \mathbf{s}(n) = {} & [-(1 - a_{11}) + a_{12} + a_{13}] \, \boldsymbol{\mathcal{G}}_1(n) + [a_{21} - (1 - a_{22}) + a_{23}] \, \boldsymbol{\mathcal{G}}_2(n) \\ & + [a_{31} + a_{32} - (1 - a_{33})] \, \boldsymbol{\mathcal{G}}_3(n). \end{aligned} \tag{13}$$

Since $\mathbf{A}$ is doubly stochastic,

$$a_{11} + a_{12} + a_{13} = 1, \quad a_{21} + a_{22} + a_{23} = 1, \quad a_{31} + a_{32} + a_{33} = 1,$$

which implies $\mathbf{s}(n) = \mathbf{0}$ and, therefore, (7) holds.

Local Homomorphic Noise Unlike the previous mechanism, whose perturbations cancel only globally, local noise accumulation may still degrade the performance of individual neural networks, particularly in unbalanced graphs [23, 24]. To address this issue, [24] proposed the *local homomorphic noise* mechanism, in which the perturbations are canceled independently at each receiving network during the combination step. Thus, every transmitted message remains perturbed while the received aggregate is noise-free. The required condition is

$$\sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\mathcal{G}}_{\ell k}(m) = \mathbf{0}. \tag{14}$$

The set $\widetilde{\mathcal{N}}_k = \mathcal{N}_k \setminus \{k\}$ is partitioned into two disjoint subsets, $\widetilde{\mathcal{N}}_{k,+}$ and $\widetilde{\mathcal{N}}_{k,-}$, preferably with equal cardinality. For each pair $\{\ell, p\}$ with $\ell \in \widetilde{\mathcal{N}}_{k,+}$ and $p \in \widetilde{\mathcal{N}}_{k,-}$, the neighboring nodes jointly generate a shared noise $\boldsymbol{\mathcal{G}}_{\{\ell,p\},k}(m)$, which

is injected as $\boldsymbol{\mathcal{G}}_{\ell k}(m) = (1/a_{\ell k})\boldsymbol{\mathcal{G}}_{\{\ell,p\},k}(m)$, $\boldsymbol{\mathcal{G}}_{pk}(m) = -(1/a_{pk})\boldsymbol{\mathcal{G}}_{\{\ell,p\},k}(m)$, while $\boldsymbol{\mathcal{G}}_{kk}(m) = \mathbf{0}$. Consequently, the perturbations cancel at node k, preserving convergence while protecting every transmitted message.

As an illustration, consider again the three-node network in Fig. 2. For node 1, let $\widetilde{\mathcal{N}}_{1,+} = \{2\}$ and $\widetilde{\mathcal{N}}_{1,-} = \{3\}$. The local noise term becomes

$$\mathbf{s}_1(n) = a_{21}\boldsymbol{\mathcal{G}}_{21}(n) + a_{31}\boldsymbol{\mathcal{G}}_{31}(n),$$

since $\boldsymbol{\mathcal{G}}_{11}(n) = \mathbf{0}$. Using

$$\boldsymbol{\mathcal{G}}_{21}(n) = (1/a_{21})\boldsymbol{\mathcal{G}}_{\{2,3\},1}(n), \qquad \boldsymbol{\mathcal{G}}_{31}(n) = -(1/a_{31})\boldsymbol{\mathcal{G}}_{\{2,3\},1}(n),$$

it follows that $\mathbf{s}_1(n) = \boldsymbol{\mathcal{G}}_{\{2,3\},1}(n) - \boldsymbol{\mathcal{G}}_{\{2,3\},1}(n) = \mathbf{0}$, and the same argument applies to the remaining nodes.

An intuitive interpretation is obtained by considering Alice and Bob sending messages to Charlie. Alice adds a secret value $+x$ to her message, while Bob adds $-x$. Charlie only observes the sum, since the perturbations cancel, but neither individual message is revealed.

The shared perturbations must be generated consistently by neighboring nodes without explicitly exchanging them. This can be achieved using the Diffie–Hellman key exchange protocol [7,24]. Consequently, every transmitted message remains protected while the perturbations are canceled locally. The trade-off is an increase in communication overhead due to the coordination required among neighboring networks. Unlike previous mechanisms, the variance of the shared noise no longer affects the learning performance because it is eliminated before the adaptation step.

Communication and Computational Complexity

Let M denote the number of trainable parameters of each neural network, and let $\mathcal{E}$ denote the edge set of the communication graph, with cardinality $|\mathcal{E}|$. In the ATC diffusion strategy, each undirected edge results in two transmissions of an M-dimensional local estimate $\boldsymbol{\Psi}_\ell(m)$. Therefore, the communication complexity per iteration is

$$C_{\text{comm}} = 2|\mathcal{E}|M = \mathcal{O}(|\mathcal{E}|M). \tag{15}$$

Since the perturbations have the same dimension as the transmitted parameters, none of the privacy-preserving mechanisms changes the asymptotic communication cost when the perturbations are generated locally.

The additional computational cost depends on the privacy mechanism. For independent random noise, each transmitted message is perturbed with an M-dimensional noise. Thus, the additional cost is $\mathcal{O}(|\mathcal{E}|M)$ if the perturbations are generated independently for each directed transmission.

For the global graph-homomorphic mechanism, each network k generates a single noise realization $\boldsymbol{\mathcal{G}}_k(m)$ with M entries. Hence, generating the independent noise realizations requires $C_{\text{noise,global}} = \mathcal{O}(NM)$. To construct explicitly

the perturbed messages $\boldsymbol{\mathcal{G}}_{\ell k}(m)$ in (8), the generated noises may need to be associated with the directed edges of the graph. This requires at most $\mathcal{O}(|\mathcal{E}|M)$ additional operations. Consequently, a conservative upper bound for the additional computational cost of the global homomorphic mechanism is

$$C_{\text{global}} = \mathcal{O}\left((N + |\mathcal{E}|)M\right). \tag{16}$$

For the local graph-homomorphic mechanism, each pair formed by the partitions $\widetilde{\mathcal{N}}_{k,+}$ and $\widetilde{\mathcal{N}}_{k,-}$ generates a shared Laplace noise realization with M entries. Therefore, the total number of noise realizations generated per iteration is $P = \sum_{k=1}^{N} |\widetilde{\mathcal{N}}_{k,+}||\widetilde{\mathcal{N}}_{k,-}|$, and the corresponding additional computational cost is

$$C_{\text{local}} = \mathcal{O}(PM). \tag{17}$$

The scaling factors $1/a_{\ell k}$ and $-1/a_{pk}$ required to construct the transmitted perturbations involve only element-wise operations over M-dimensional noises and therefore do not change the asymptotic order. For approximately balanced partitions, the number of pairs associated with node k is approximately $|\widetilde{\mathcal{N}}_{k,+}||\widetilde{\mathcal{N}}_{k,-}| \approx (|\mathcal{N}_k| - 1)^2/4$. Thus, unlike the global mechanism, whose computational cost grows linearly with the number of network edges, the local mechanism can exhibit a stronger dependence on the node degrees, particularly in dense graphs.

If the shared pairwise noise vectors in the local mechanism were explicitly transmitted, the communication complexity would increase to $\mathcal{O}((|\mathcal{E}| + P)M)$. Alternatively, shared values can be established through a key-agreement protocol such as Diffie–Hellman [7, 24], requiring only a fixed-size exchange during initialization. In this case, the communication complexity during training remains $\mathcal{O}(|\mathcal{E}|M)$, although the cryptographic operations introduce an additional initialization cost.

4 Evaluating Privacy

To evaluate the privacy-preserving mechanisms, this paper uses MINE to estimate mutual information [2] and iDLG as a reconstruction attack [33]. MINE provides a neural-network-based approach to estimate the mutual information between high-dimensional continuous variables. It relies on a variational lower bound of the mutual information that can be optimized through gradient descent, enabling the assessment of statistical dependence between variables. In this work, MINE is employed to quantify the amount of information about the training labels contained in the exchanged messages.

To evaluate privacy leakage, we consider the iDLG attack [33], which improves upon DLG [34] by exploiting the gradient structure of the last network layer to directly infer the true label before input reconstruction. For classification tasks with cross-entropy loss and one-hot encoded labels, the gradient component associated with the correct class typically has the most negative value, providing a distinctive signature for label inference. After estimating the label, iDLG optimizes only a dummy input whose gradients are iteratively matched to those

observed by the adversary, allowing the reconstruction of an approximation of the original training sample.

The privacy evaluation conducted in this work is empirical. MINE provides an estimated lower bound on the dependence between private labels and intercepted messages, while iDLG evaluates reconstruction quality under a specific attack configuration. Consequently, lower MINE estimates or poorer iDLG reconstructions indicate reduced leakage under the evaluated settings, but do not constitute a differential-privacy guarantee or resistance to attacks outside the stated threat model.

5 Simulation Results

In this section, we compare the three privacy-preserving mechanisms described in Section 3: (i) random noise, (ii) global homomorphic noise, and (iii) local homomorphic noise. In addition, the noise-free baseline is also considered. For methods (i) and (ii), the noise intensity is controlled by the standard deviation σ_g, which was varied over the interval $[10^{-5}, 5 \times 10^{-2}]$. The communication topology is generated according to the Erdős–Rényi model, and the combination matrix $\mathbf{A}$ is computed using the Metropolis rule [26, 27].

We consider a passive external eavesdropper (*honest but curious* [19]) that intercepts one directed ATC message transmitted from a victim node v to a neighboring node r at each iteration. The adversary observes only the intercepted message and does not modify, delay, or inject messages. We conservatively assume that the adversary knows the network architecture, loss function, learning rate η, communication topology, combination weights, and the victim's pre-adaptation parameters $\boldsymbol{\mathcal{W}}_v(m-1)$. From the intercepted message $\widetilde{\boldsymbol{\Psi}}_{v\to r}(m)$, the adversary estimates the victim's gradient as

$$\frac{\boldsymbol{\mathcal{W}}_v(m-1) - \widetilde{\boldsymbol{\Psi}}_{v\to r}(m)}{\eta}.$$

The estimated gradient is then supplied to the iDLG attack [33] to reconstruct the victim's training data. The adversary does not observe both messages involved in the same cancellation pair.

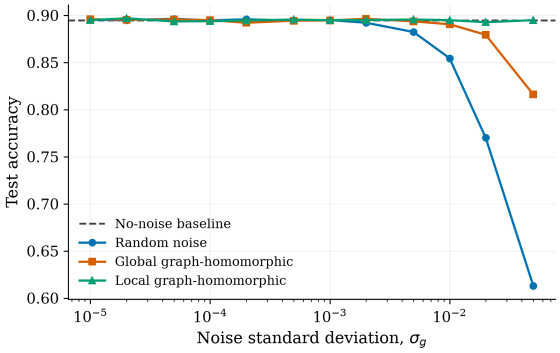
First, we revisit the simulation shown in Fig. 1, which considers the distributed training of multilayer perceptron (MLP) neural networks [3] using the MNIST dataset [18], normalized to have zero mean and unit variance. The 6×10^4 training samples are randomly shuffled and partitioned among the $N = 6$ neural networks. Each MLP consists of two hidden layers and one output layer containing 64, 128, and 10 neurons, respectively. Rectified linear unit (ReLU) activation functions are employed in the hidden layers, while a *softmax* activation function is used in the output layer. Cross-entropy loss and Xavier initialization are adopted [12]. The networks are trained using the conventional *backpropagation* algorithm with a learning rate of 0.01 [3].

Fig. 3(a) shows the average MNIST test accuracy as a function of σ_g over 30 independent repetitions. The noise-free configuration achieves an accuracy of

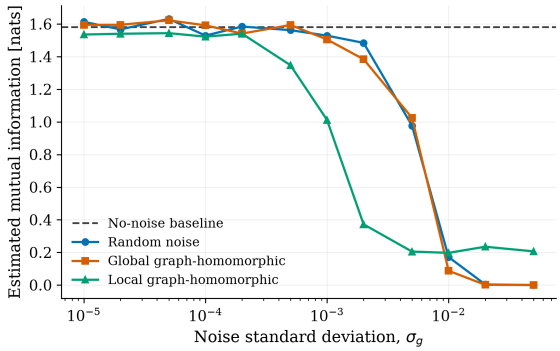
0.8946 ± 0.0100. Local graph-homomorphic noise maintains a comparable performance throughout the evaluated range, attaining 0.8956 ± 0.0105, 0.8927 ± 0.0131, and 0.8949 ± 0.0107 for $\sigma_g = 0.005$, 0.02, and 0.05, respectively. These variations are within the standard deviation across repetitions and therefore indicate preservation of the noise-free learning performance. In contrast, independent random noise reduces the accuracy to 0.8824 ± 0.0129 at $\sigma_g = 0.005$, 0.7703 ± 0.0249 at $\sigma_g = 0.02$, and 0.6130 ± 0.0518 at $\sigma_g = 0.05$. Global graph-homomorphic noise produces a less pronounced degradation, retaining accuracies of 0.8937 ± 0.0110, 0.8794 ± 0.0120, and 0.8160 ± 0.0345 at the same noise levels. Thus, local cancellation preserves classification utility, whereas random and global perturbations progressively affect the learning dynamics as the noise intensity increases.

Fig. 3(b) reports the mutual-information estimates obtained with MINE as a function of σ_g, providing an empirical measure of the information leaked through the exchanged messages. Since the gradients are high-dimensional, a principal component analysis (PCA) transformation [3] is fitted independently in each experimental repetition and used to project all privacy mechanisms being compared onto a 20-dimensional space. Mutual information between the private labels and the projected intercepted gradient is then estimated. As expected, increasing the noise intensity reduces the statistical dependence between the intercepted gradients and the private labels for all privacy-preserving mechanisms. The random-noise and global graph-homomorphic approaches exhibit the lowest MINE values for large noise levels, indicating stronger privacy protection, but this improvement comes at the expense of a substantial loss in classification accuracy, as observed in Fig. 3(a). In contrast, the local graph-homomorphic mechanism also achieves a significant reduction in mutual information while preserving virtually the same classification performance as the noise-free algorithm. These results highlight its more favorable privacy–utility trade-off, demonstrating that it effectively limits information leakage without compromising the learning process.

All methods were subjected to the iDLG attack. The final input image reconstructed by the iDLG algorithm is shown in Fig. 4 for $\sigma_g = 0.05$. In the absence of noise, iDLG is able to recover a visually accurate approximation of the original image. Conversely, when noise is added to the transmitted messages, the reconstruction quality deteriorates significantly. The classifier performance is evaluated in terms of the overall classification accuracy of the distributed training process. The quality of the reconstructed input image is assessed using the mean-squared error (MSE) and the structural similarity index (SSIM) [30]. The SSIM ranges from -1 to 1, attaining a value of 1 when the two images are identical. The corresponding MSE and SSIM values, obtained by comparing the reconstructed images with the original one, are reported in Table 1. These results quantitatively confirm the visual impressions of Fig. 4. In the absence of noise, iDLG reconstructs the input image with relatively high fidelity, yielding the lowest MSE (0.0088) and the highest SSIM (0.530). In contrast, all three privacy-preserving mechanisms substantially degrade the reconstruction quality, producing MSE values around 0.15–0.18 and SSIM values close to zero. Although the random-noise mechanism achieves the lowest SSIM, the differences among



(a)



(b)

Fig. 3: Utility and empirical privacy results for the MNIST experiment: (a) average test accuracy and (b) mutual-information estimate.

the three protection strategies are relatively small, indicating that they are all effective in preventing accurate image reconstruction. Combined with the classification results in Fig. 3 (a), these findings further highlight the advantage of the local graph-homomorphic mechanism, which provides a level of privacy protection comparable to the other approaches while preserving the learning performance.

We also evaluate the mechanisms on the federated dermoscopic-image dataset (Fed-ISIC2019) [28], which contains 23,247 dermoscopic images collected from six centers, resulting in naturally imbalanced and non-identically distributed local datasets. We consider binary classification between melanocytic nevus (NV) and melanoma (MEL), since both classes are represented at all six centers. The resulting subset contains 15,511 images, comprising 11,326 NV and 4,185 MEL samples. The original center assignments and fixed train-test partitions are preserved, yielding 12,416 training images and 3,095 test images distributed across six nodes. Each image is resized to 28×28 pixels and standardized indepen-

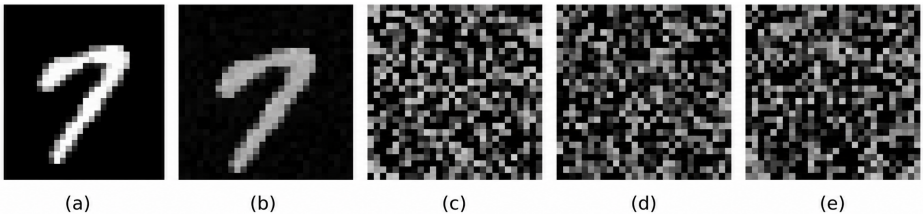


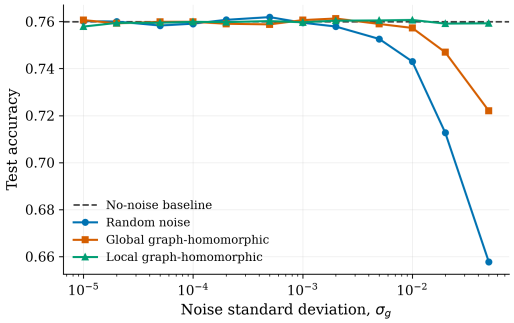
Fig. 4: Reconstructions obtained by iDLG: (a) original image, (b) without noise, (c) random noise, (d) global homomorphic noise, and (e) local homomorphic noise.

Table 1: Reconstruction metrics for the image labeled 7.

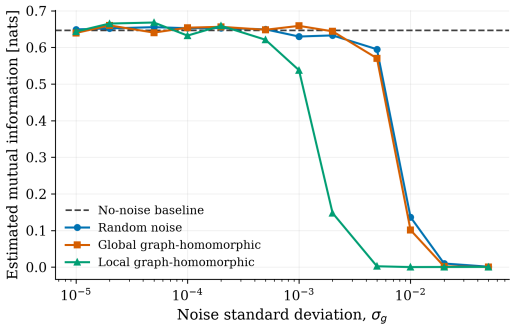
Case	**Label**	**MSE**	**SSIM**
Original image	7	0.0000	1.000
Without noise	7	0.0088	0.530
Random noise	7	0.1841	0.013
Global homomorphic noise	7	0.1537	0.041
Local homomorphic noise	7	0.1755	0.019

dently for each color channel. The CNN comprises a convolutional layer with 32 3×3 filters, followed by ReLU activation and 2×2 max pooling, and a second convolutional layer with 64 3×3 filters, followed by ReLU activation and adaptive average pooling to 4×4. The resulting feature maps are flattened and passed through a fully connected layer with 128 neurons and ReLU activation, followed by a two-neuron output layer. MEL is treated as the positive class. Cross-entropy loss is used. Convolutional kernels are initialized using Kaiming initialization, whereas the fully connected layers use Xavier initialization. The networks are trained using the conventional backpropagation algorithm (learning rate of 0.01) [3].

Fig. 5(a) shows the average test accuracy as a function of σ_g, averaged over 30 independent repetitions. The noise-free baseline achieves a test accuracy of 0.7599 ± 0.0047. Independent random noise progressively degrades classification performance, reducing the accuracy to 0.7127 ± 0.0171 at $\sigma_g = 0.02$ and 0.6577 ± 0.0648 at $\sigma_g = 0.05$. Global graph-homomorphic noise provides better utility under the same noise intensities, attaining 0.7469 ± 0.0103 and 0.7220 ± 0.0154, respectively. In contrast, local graph-homomorphic noise maintains an accuracy close to the noise-free baseline throughout the evaluated range, reaching 0.7591 ± 0.0058 at $\sigma_g = 0.02$ and 0.7593 ± 0.0052 at $\sigma_g = 0.05$. Fig. 5(b) presents the estimated mutual information between the private labels and the gradient reconstructed from the intercepted message. As occurs in the MNIST case, increasing σ_g reduces the dependence measured by MINE, indicating that the perturbed messages disclose less information about the private labels.



(a) Test accuracy as a function of σ_g.



(b) MINE estimate as a function of σ_g.

Fig. 5: Utility and empirical privacy results for the Fed-ISIC2019 experiment: (a) average test accuracy and (b) mutual-information estimate.

6 Conclusions

Diffusion-based distributed neural networks may disclose sensitive information because the exchanged intermediate parameters depend directly on local gradients. The iDLG experiments demonstrated that an honest-but-curious adversary can reconstruct training images and infer their labels from intercepted messages, confirming that decentralized training alone does not guarantee privacy. Experiments with MNIST and Fed-ISIC2019 showed that independent random perturbations and global graph-homomorphic noise improve privacy at the expense of classification performance, whereas the local graph-homomorphic mechanism achieved a more favorable privacy–utility trade-off. It preserved test accuracy close to the noise-free baseline while reducing the mutual-information estimates obtained with MINE across both datasets. While graph-homomorphic perturbation mechanisms have previously been investigated in the context of adaptive diffusion networks, their application to privacy-preserving distributed neural network training has remained unexplored. This work bridges that gap by demonstrating their effectiveness in this setting. Although MINE and iDLG pro-

vide empirical privacy assessments rather than formal differential privacy guarantees, the results demonstrate the effectiveness of the proposed approach for privacy-preserving distributed neural network training. Evaluating other adversary models and deriving guarantees comparable to those established in recent formal analyses of decentralized privacy [1,6] remain directions for future work.

Acknowledgments. This work was supported by CAPES (Finance Code 001 and Grant 88887.001105/2024-00), FAPESP (Grant 2026/01471-7), and CNPq (Grants 301743/2026-6 and 404081/2023-1).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Allouah, Y. *et al.*: The privacy power of correlated noise in decentralized learning. In: Proc. 41st Int. Conf. Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 1115–1143. (2024).
2. Belghazi, M.I. *et al.*: MINE: Mutual information neural estimation. arXiv preprint arXiv:1801.04062 (2018).
3. Bishop, C.M.: Deep Learning: Foundations and Concepts. Springer (2024).
4. Codella, N.C.F. *et al.*: Skin lesion analysis toward melanoma detection: A challenge at Int. Symp. Biomed. Imaging (ISBI). arXiv preprint arXiv:1710.05006 (2017).
5. Combalia, M. *et al.*: BCN20000: Dermoscopic lesions in the wild. arXiv preprint arXiv:1908.02288 (2019).
6. Cyffers, E., Bellet, A., Upadhyay, J.: Differentially private decentralized learning with random walks. In: Proc. 41st Int. Conf. Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 9762–9783. (2024).
7. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theory **22**(6), 644–654 (1976).
8. Du, J. *et al.*: SoK: On gradient leakage in federated learning. In *Proc. 34th USENIX Security Symposium*, pp. 3045–3064. (2025).
9. Dwork, C.: Differential privacy. In: Proc. ICALP. pp. 1–12 (2006).
10. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds) Theory of Cryptography. Lecture Notes in Computer Science, vol. 3876, 265–284. Springer (2006).
11. El Mrini, A., Cyffers, E., Bellet, A.: Privacy attacks in decentralized learning. In: Proc. 41st Int. Conf. Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 36419–36433. (2024).
12. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In Proc. AISTATS. pp. 249–256 (2010).
13. Han, H., Zhang, S., Chen, H., Sayed, A.H.: Masked diffusion strategy for privacy-preserving distributed learning. IEEE Trans. Inf. Forensics Security **20**, 8402–8416 (2025).
14. Han, H., Zhang, S., So, H.C.: Privacy-preserving distributed adaptive filtering via input perturbation and amplitude-shifted data exchange over networks. IEEE Trans. Signal Process. **74**, 639–653 (2026).

15. Han, H., Zhang, S., Wen, G.: Privacy-preserving diffusion adaptive learning with nonzero-mean protection noise. IEEE Trans. Syst., Man, Cybern. **55**(6), 4140–4150 (2025).
16. Hatamizadeh, A. *et al.*: Do gradient inversion attacks make federated learning unsafe? *IEEE Transactions on Medical Imaging*, **42**(7), 2044–2056 (2023).
17. Kairouz, P. *et al.*: Advances and Open Problems in Federated Learning. arXiv preprint arXiv:1912.04977 (2021).
18. LeCun, Y., Cortes, C., Burges, C.J.C.: The MNIST database of handwritten digits. Netw. [Online], available at `https://doi.org/10.24432/C53K8Q`, (1998).
19. Lindell, Y., Pinkas, B.: Secure multiparty computation for privacy-preserving data mining. J. Privacy and Confidentiality **1**(1), 59–98 (2009).
20. Liu, B., Ding, Z., Lv, C.: Distributed training for multi-layer neural networks by consensus. IEEE Trans. Neural Netw. Learn. Syst. **31**(5), 1771–1778 (2020).
21. Muthukrishnan, G., Kalyani, S.: Differential privacy with higher utility by exploiting coordinate-wise disparity: Laplace mechanism can beat gaussian in high dimensions. IEEE Trans. Inf. Forensics Security **20**, 2836-2851 (2025).
22. Papachristou, M., Rahimian, M.A.: Differentially private distributed estimation and learning. IISE Transactions **57**(7), 756–772 (2025).
23. Rizk, E., Vlaski, S., Sayed, A.H.: Enforcing privacy in distributed learning with performance guarantees. IEEE Trans. Signal Process. **71**, 3385–3398 (2023).
24. Rizk, E., Vlaski, S., Sayed, A.H.: Local graph-homomorphic processing for privatized distributed systems. In: Proc. ICASSP. pp. 1–5 (2023).
25. Sattler, F., Wiedemann, S., Müller, K.R., Samek, W.: Robust and communication-efficient federated learning from non-i.i.d. data. IEEE Trans. Neural Netw. Learn. Syst. **31**(9), 3400–3413 (2020).
26. Sayed, A.H.: Adaptation, learning, and optimization over networks. Foundations and Trends in Machine Learning **7**(4–5), 311–801 (2014).
27. Tiglea, D.G., Candido, R., Silva, M.T.M.: Adaptive diffusion networks: An overview. Signal Process. **223**, 109570 (2024).
28. Tschandl, P., Rosendahl, C., Kittler, H.: The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific Data **5**, 180161 (2018).
29. Vlaski, S., Sayed, A.H.: Graph-homomorphic perturbations for private decentralized learning. In: Proc. ICASSP. pp. 5240–5244 (2021).
30. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004).
31. Xu, T. *et al.*: Privacy-preserving distributed estimation via compressive diffusion lms for secure IoT. IEEE Internet Things J. (2026).
32. Wu, R., Chen, X., Guo, C., Weinberger, K.Q.: Learning to invert: Simple adaptive attacks for gradient inversion in federated learning. In: Proc. 39th Conf. Uncertainty in Artificial Intelligence. Proceedings of Machine Learning Research, vol. 216, pp. 2293–2303. (2023).
33. Zhao, B., Mopuri, K.R., Bilen, H.: iDLG: Improved deep leakage from gradients. arXiv preprint arXiv:2001.02610 (2020).
34. Zhu, L., Liu, Z., Han, S.: Deep leakage from gradients. arXiv preprint arXiv:1906.08935 (2019).