Dropout is NOT All You Need to Prevent Gradient Leakage

Daniel Scheliga[1], **Patrick Mäder**[1,2], **Marco Seeland**[1]

[1] Technische Universität Ilmenau, Germany
[2] Friedrich Schiller Universität Jena, Germany
daniel.scheliga@tu-ilmenau.de, patrick.maeder@tu-ilmenau.de, marco.seeland@tu-ilmenau.de

Abstract

Gradient inversion attacks on federated learning systems reconstruct client training data from exchanged gradient information. To defend against such attacks, a variety of defense mechanisms were proposed. However, they usually lead to an unacceptable trade-off between privacy and model utility. Recent observations suggest that dropout could mitigate gradient leakage and improve model utility if added to neural networks. Unfortunately, this phenomenon has not been systematically researched yet. In this work, we thoroughly analyze the effect of dropout on iterative gradient inversion attacks. We find that state of the art attacks are not able to reconstruct the client data due to the stochasticity induced by dropout during model training. Nonetheless, we argue that dropout does not offer reliable protection if the dropout induced stochasticity is adequately modeled during attack optimization. Consequently, we propose a novel *Dropout Inversion Attack (DIA)* that jointly optimizes for client data and dropout masks to approximate the stochastic client model. We conduct an extensive systematic evaluation of our attack on four seminal model architectures and three image classification datasets of increasing complexity. We find that our proposed attack bypasses the protection seemingly induced by dropout and reconstructs client data with high fidelity. Our work demonstrates that privacy inducing changes to model architectures alone cannot be assumed to reliably protect from gradient leakage and therefore should be combined with complementary defense mechanisms.

1 Introduction

Federated Learning strategies were designed to leverage the collaborative use of distributed data to learn a common machine learning model. Since training data is not shared between participating clients, systemic privacy risks can be mitigated (Kairouz et al. 2021). Recent work, however, shows that the privacy of participating clients can be compromised by reconstructing sensitive data from gradients or model states that are exchanged during the federated training. The most versatile reconstruction techniques are realized as iterative gradient inversion attacks (Zhu and Han 2020; Zhao, Mopuri, and Bilen 2020; Wei et al. 2020; Geiping et al. 2020; Yin et al. 2021; Lu et al. 2022; Hatamizadeh et al. 2022). These attacks optimize randomly initialized dummy images so that their resulting dummy gradients match the targeted client gradient.

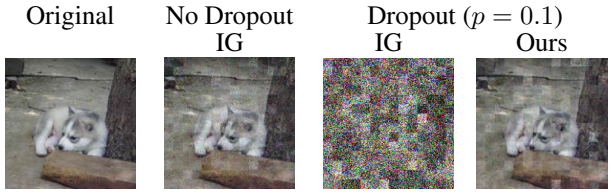


Figure 1: Reconstructing data from gradients without and with dropout. (a) Original image. (b) State of the art IG attack (Geiping et al. 2020) without dropout. (c) State of the art IG attack (Geiping et al. 2020) with dropout. (d) Our proposed Dropout Inversion Attack with dropout.

Defense strategies to protect against such attacks are based on: (1) adjustments to the training process, e.g. increasing the number of local training iterations or the batch-size (Wei et al. 2020), (2) changes to the input data, e.g. perturbation or input encryption (Huang et al. 2020a,b), (3) perturbation of exchanged gradient information, e.g. through the addition of noise, compression or pruning (Bonawitz et al. 2017; Jayaraman and Evans 2019; Zhu and Han 2020; Sattler et al. 2020; Lyu 2021; Wei and Liu 2021), or (4) application of specifically designed architectural features or modules (Scheliga, Mäder, and Seeland 2022b,a; Sun et al. 2021). The use of most defense mechanisms, however, results in a trade-off between privacy and model utility (Dwork and Roth 2013; Jayaraman and Evans 2019; Zhu and Han 2020; Wei et al. 2020; Huang et al. 2021; Scheliga, Mäder, and Seeland 2022b,a).

Dropout is a regularization technique that aims to reduce overfitting in deep neural networks (Hanson 1990; Hinton et al. 2012). While the use of dropout can boost the performance of neural networks (Srivastava et al. 2014), recent publications suggest that it could also protect shared gradients from gradient leakage (Wei et al. 2020; Zheng 2021). Inspired by these observations, we show that the stochasticity introduced by dropout indeed protects shared gradients from gradient leakage through iterative gradient inversion attacks. However, we claim that this protection is only ap-

parent, because the attacker has no access to the specific realization of the stochastic client model used during training. Moreover, we argue that an attacker can sufficiently approximate this specific realization of the client model using the shared gradient information. To reveal the vulnerability of dropout protected models, we formulate a novel **Dropout Inversion Attack** (DIA) that jointly optimizes for client data and the dropout masks applied during local training.

Our contributions can be summarized as follows:

- We systematically show that the application of dropout during neural network training seems to prevent gradient leakage by iterative gradient inversion attacks.
- We formulate a novel attack that, contrary to previous attacks, successfully reconstructs client training data from dropout protected shared gradients. Note that the components of our proposed attack can be universally used to extend any other iterative gradient inversion attack.
- We perform an extensive systematic evaluation of our attack on two dense connection based (Multi Layer Perceptron, Vision Transformer) and two CNN based (LeNet, ResNet) model architectures as well as three image classification datasets of increasing complexity (MNIST, CIFAR-10, ImageNet).

2 Related Work

2.1 Gradient Inversion Attacks

Consistent with related work (Geiping et al. 2020; Enthoven and Al-Ars 2020; Yin et al. 2021; Kaissis et al. 2021; Jin et al. 2021; Scheliga, Mäder, and Seeland 2022b,a; Zhang et al. 2022; Gupta et al. 2022), we assume a *honest-but-curious* server threat model. In this scenario the attacker has insight into the training process, *i.e*. knowledge of the model F, the loss function $\mathcal{L}$ used to optimize the model parameters θ and the client gradient $\nabla\mathcal{L}_\theta(F(x), y)$ which is exchanged during federated training. Given this knowledge, the attacker aims to reconstruct training data (x, y) of clients that participate in federated training.

To achieve this, the attacker iteratively minimizes the distance D between the client gradient $\nabla\mathcal{L}_\theta(F(x), y)$ and a dummy gradient $\nabla\mathcal{L}_\theta(F(x'), y')$. The dummy gradient is obtained by forward propagation of randomly initialized dummy data (x', y') through the model F. A gradient based optimizer, *e.g*. Adam (Kingma and Ba 2014), adjusts the dummy data (x', y') until convergence.

Attack optimization can be formally expressed as:

$$\underset{(x',y')}{\arg\min}\, D(\nabla\mathcal{L}_\theta(F(x), y), \nabla\mathcal{L}_\theta(F(x'), y')) + \lambda\Omega. \quad (1)$$

Depending on the specific attack, the regularization term $\lambda\Omega$ can take different forms. Generally $\lambda\Omega$ aims to stabilize optimization and to improve reconstruction quality.

The first iterative gradient inversion attack was introduced by Zhu *et al*. (Zhu and Han 2020). They use Euclidean distance for D and no regularization. The authors of (Zhao, Mopuri, and Bilen 2020) and (Yin et al. 2021) proposed methods to analytically reconstruct the ground-truth labels y in advance. As long as the training batch contains disjoint classes, an attacker can reliably reconstruct label information. This eliminates optimization for y' in Eq. 1 and accelerates the overall attack.

Geiping *et al*. further improve the reconstruction process with their *Inverting Gradients* (IG) attack (Geiping et al. 2020). They minimize the cosine distance between client and dummy gradients instead of Euclidean distance to disentangle gradient direction and magnitude. Furthermore, they add a total variation (Rudin, Osher, and Fatemi 1992) prior of the dummy image x' as regularization term to increase the fidelity of their reconstructions.

Lu *et al*. (Lu et al. 2022) specifically target transformer based architectures. They find that the trainable position embedding in transformers can be greatly abused for reconstruction. Their iterative *Attention PRIvacy Leakage* (APRIL) attack uses Euclidean distance for D and adds the cosine distance between client and dummy gradients of the positional embedding as regularization term.

Other related work in the area of iterative gradient inversion attacks mainly focuses to improve the reconstruction quality through the choice of the 1) gradient distance function D, 2) regularization term Ω, 3) initialization of the dummies (x', y') and 4) label reconstruction method (Wei et al. 2020; Wang et al. 2020; Yin et al. 2021; Jin et al. 2021; Jeon et al. 2021). A detailed overview of recent attack combinations can be found in (Li et al. 2022) and (Zhang et al. 2022).

2.2 Dropout

Dropout (Hanson 1990; Hinton et al. 2012) is a commonly used regularization method that randomly masks the output of neurons with a chosen probability p. Hence, each forward pass realizes a different version of the neural network. This makes dropout an efficient technique for model averaging and in turn prevents models from overfitting to training data (Srivastava et al. 2014).

Formally, we consider a neural network $F : X \rightarrow Y$, $F(x) = y$ to be a deterministic function that calculates an output $y \in Y$ from an input $x \in X$. Given the output $z^{(i)}$ of the ith layer $L^{(i)}$ in F, a succeeding dropout layer $L_D^{(i)}$ multiplies $z^{(i)}$ element-wise with a random dropout mask $\psi^{(i)}$ and scales the remaining outputs according to the dropout rate p to preserve the output magnitude:

$$L_D^{(i)}(z^{(i)}) = \frac{1}{1-p} \cdot z^{(i)} \circ \psi^{(i)} \quad (2)$$

For every dropout layer $L_D^{(i)}$ $i \in \{1, \ldots, l\}$, $\psi^{(i)}$ is a vector of independent Bernoulli variables, *i.e*. $\psi^{(i)} \sim \text{Bernoulli}(p)$. We define $\tilde{\Psi}_p = \{\psi^{(1)}, \ldots, \psi^{(l)}\}$ as the set of l random dropout masks for a neural network with l dropout layers.

The use of dropout turns a deterministic neural network into a stochastic one. Hence, the set of all functions F that depend on the dropout masks $\tilde{\Psi}_p$ is $\mathcal{F}_p = \{F_\Psi | \Psi \sim \tilde{\Psi}_p\}$. We denote Ψ as one arbitrary but fixed sample from $\tilde{\Psi}_p$. At each training step a new Ψ is sampled. Consequently, this realizes a different version $F_\Psi \in \mathcal{F}_p$ of the neural network that is used for forward propagation and gradient calculation.

As dropout introduces noise into the training process, a decrease in reconstruction quality of iterative gradient

inversion attacks is observed in recent work (Wei et al. 2020; Zheng 2021). Contrary to these findings, Enthoven *et al*. (Enthoven and Al-Ars 2020) find that the use of dropout after the first fully connected layer of a neural network increases the success to analytically reconstruct client data from larger batches. Such analytical attacks, however, can be easily mitigated by removing bias weights from the model (Scheliga, Mäder, and Seeland 2022a).

3 Dropout vs Gradient Leakage

Although no systematic studies have yet been conducted, recent observations suggest that dropout can decrease the success of iterative gradient inversion attacks (Wei et al. 2020; Zheng 2021). To confirm these observations, we first conduct a series of experiments that evaluate the effect of increased dropout rates on reconstruction quality and model utility. Next, we argue that an attacker would be able to successfully reconstruct client training data if given knowledge about the specific realization of the stochastic client model. Therefore, we conduct proof of concept experiments that consider an attacker who knows the dropout masks applied during client model training, *i.e*. a *well-informed attacker*.

3.1 Attacking Dropout Protected Models

To confirm the impact of dropout on iterative gradient inversion attacks, we first attack a Multi Layer Perceptron (MLP) (Rumelhart, Hinton, and Williams 1985) and a Vision Transformer (ViT) (Dosovitskiy et al. 2020) trained on the MNIST (Deng 2012) and CIFAR-10 (Krizhevsky, Hinton et al. 2009) datasets. We chose these architectures as they typically use dropout as regularization technique. We use the publicly available PyTorch implementation of IG[1] provided by (Geiping et al. 2020) as gradient inversion attack.

To observe the effect of dropout on model utility we follow the federated scenario and hyperparameters used in (Scheliga, Mäder, and Seeland 2022a). We report the test accuracy of the global model state after convergence. More details on the experimental setup can be found in Section 5.

When dropout is used, the attacker has two options to generate the dummy gradients required for attack optimization. Analogous to client training, the first option uses the model in training mode, *i.e*. the stochastic model. In this case, the attacker applies randomly sampled dropout masks Ψ_A in each forward propagation so that a different realization $F_{\Psi_A} \in \mathcal{F}_p$ is used in each iteration during attack optimization. Consequently, the dummy gradients $\nabla\mathcal{L}_\theta(F_{\Psi_A}(x'), y'))$ differ greatly for each attack iteration and are "elusive and unable to converge" (Wei et al. 2020) to match the client gradient $\nabla\mathcal{L}_\theta(F_{\Psi_C}(x), y))$.

The second option uses the model in inference mode, *i.e*. dropout is not applied. Note that in this case all dropout masks $\psi_A^{(i)} = I \; \forall i = 1, \ldots, l$. Hence, the same realization $F_{\Psi_A} \in \mathcal{F}_{p=0} = \{F\}$ is used in each iteration during attack optimization. The attacker's dummy gradients are more stable compared to when the stochastic model is used. However, since the client used dropout during training, $\Psi_A \neq \Psi_C$ causes the dummy gradients to differ from the client gradients despite attack optimization.

[1]https://github.com/JonasGeiping/invertinggradients

	Model	p	Accuracy [%] ↑	IG SSIM ↑	WIIG SSIM ↑
MNIST	MLP	0.00	98.53	**1.00**	-
		0.25	98.28	0.79	**1.00**
		0.50	97.48	0.59	**1.00**
		0.75	93.50	0.30	*0.82*
	ViT	0.00	98.76	0.98	-
		0.25	**98.98**	0.04	0.99
		0.50	98.67	*0.02*	0.99
		0.75	*87.36*	*0.02*	**1.00**
CIFAR-10	MLP	0.00	54.72	**1.00**	-
		0.25	52.52	0.68	**0.98**
		0.50	38.89	0.51	0.84
		0.75	*27.09*	0.38	*0.78*
	ViT	0.00	64.47	0.87	-
		0.25	**70.83**	0.01	0.93
		0.50	67.01	0.01	0.96
		0.75	45.08	*0.00*	0.95

Figure 2: Model accuracy after federated training of a MLP and ViT on MNIST and CIFAR-10 as well as SSIM computed from gradients attacked with IG. WIIG indicates that the attacker has knowledge of the victim dropout masks Ψ_V. Arrows indicate direction of improvement. Bold and italic formatting highlight best and worst results respectively.

Fig. 2 shows the global model accuracy after federated training of the MLP and ViT on MNIST and CIFAR-10, as well as the privacy as measured by SSIM. Dropout rates were selected as $p \in \{0, 0.25, 0.50, 0.75\}$. With increasing p the SSIM steadily decreases for the MLP; hence, privacy increases. However, we also observe a negative impact of dropout on MLP model utility. Findings in (Hofmann and Mäder 2021) confirm this effect. Furthermore, Piotrowski *et al*. (Piotrowski, Napiorkowski, and Piotrowska 2020) argue that MLPs with a low width require very low dropout rates to achieve improvements in model utility.

The effect of dropout is even more pronounced for the ViT architecture. A moderate dropout rate $p = 0.25$ causes the SSIM to immediately drop from $0.98/0.87$ to $0.04/0.01$ for MNIST/CIFAR-10, respectively. No visually recognizable information can be reconstructed (cf. Fig. 4 and 5). Furthermore, the accuracy of the ViT benefits from dropout with an absolute increase of $0.22\%/6.36\%$ for MNIST/CIFAR-10 at $p = 0.25$. Note that we have also used APRIL (Lu et al. 2022) to attack the ViT but found IG to perform better when dropout is applied. More detailed results on the comparison of IG and APRIL, as well as more reconstruction quality metrics can be found in the technical appendix. To ensure a consistent experimental setup, we stick with IG as baseline attack for the remaining experiments.

Fig. 3 illustrates the behavior of the reconstruction loss during attack optimization. Without dropout, *i.e*. $p = 0$ and hence $\Psi_A = \Psi_C$ (blue lines in Fig. 3), the dummy gradients quickly converge towards the client gradients. The optimization becomes unstable as soon as dropout is used, *e.g*.

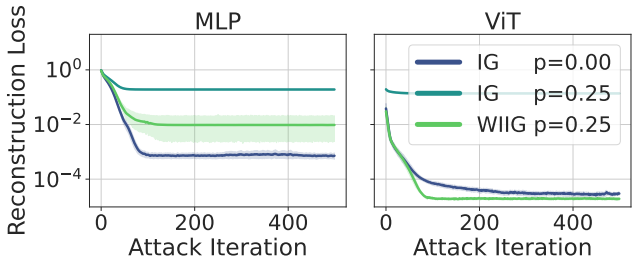


Figure 3: Exemplary reconstruction loss for a MLP and ViT on CIFAR-10. WIIG indicates that the attacker has knowledge of the client dropout masks Ψ_C.

with a dropout rate of $p = 0.25$. The attacker is forced to base the attack optimization on a model realization F_{Ψ_A} that is different from the realization F_{Ψ_C} used during training. This causes a mismatch between dummy and client gradients. Corresponding visual examples are displayed in Fig. 4.

3.2 The Well-Informed Attacker

The previous experiments show that the attack optimization cannot converge because the attacker and the client calculate their gradients based on different realizations F_{Ψ_A} and F_{Ψ_C}. We argue that the attacker would be able to reconstruct the client's training data if she is either informed about F_{Ψ_C} or finds a suitable approximation thereof. As a proof of concept, we conduct a series of experiments where the attacker applies the same dropout masks that were applied by the client during training, *i.e.* we use a *well-informed attacker*. Consequently, the attack optimization is based on the same realization $F_{\Psi_A} = F_{\Psi_C}$, and the gradient matching loss can be effectively minimized as in a model without dropout.

To empirically validate this argument we give the attacker knowledge of Ψ_C. During the iterative attack optimization, the attacker uses Ψ_C in the forward propagation of the dummy images to compute the dummy gradients. We denote this as *well-informed inverting gradients* attack (WIIG).

Fig. 2 displays the reconstruction quality measured in SSIM for the well-informed attacker. The MLP still shows a slight decrease in SSIM for high dropout rates p. However, even with the highest considered dropout rate $p = 0.75$ the SSIM is increased by $0.52/0.40$ compared to the baseline IG attack for MNIST/CIFAR-10, respectively.

The increase in reconstruction quality for the ViT is even more remarkable. For dropout rates $p > 0$, the IG based reconstructions yield a SSIM ≈ 0. The well-informed attacker WIIG achieves almost perfect reconstructions, *i.e.* SSIM ≈ 1, for both datasets. Interestingly, the SSIM increases compared to IG with $p = 0$. This indicates that dropout could, in principle, allow even better reconstructions. We attribute this effect to the attacker's additional knowledge about Ψ_C. Because the ground truth masks Ψ_C are applied during forward propagation, dropout related zero values in the client and dummy gradients match by default. The additional information makes the problem easier as it reduces the total number of gradient values that need to be matched to find an optimal solution.

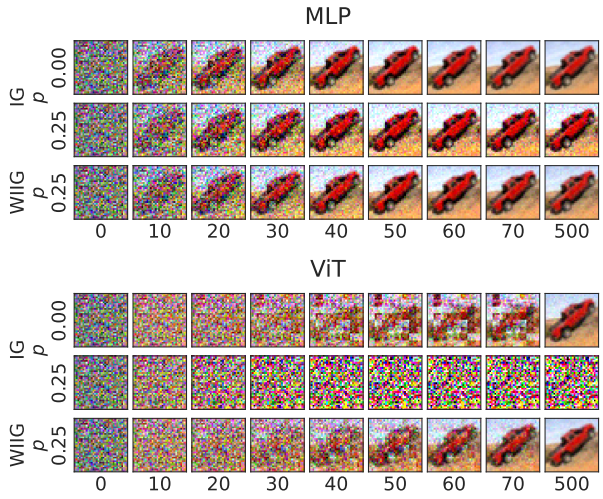


Figure 4: Exemplary reconstruction progress for a MLP and ViT on CIFAR-10. WIIG indicates that the attacker has knowledge of the client dropout masks Ψ_C. Numbers on the ordinate indicate the attack iteration.

4 DIA – Dropout Inversion Attack

In a realistic scenario the attacker does not have information on the client's dropout masks Ψ_C used during training. However, we argue that if the attacker finds a close enough approximation $F_{\Psi_A} \approx F_{\Psi_C}$, she still bypasses the privacy inducing effect of dropout.

Assuming an honest-but-curious threat model, the attacker has knowledge of the model architecture and the positions of dropout layers in the model. To find a realization $F_{\Psi_A} \in \mathcal{F}_p$ that approximates F_{Ψ_C}, the attacker has to find dropout masks $\psi_A^{(1)}, \ldots, \psi_A^{(l)}$ such that $\psi_A^{(i)} \approx \psi_C^{(i)}$ $\forall i = 1, ..., l$, where $\psi_C^{(1)}, \ldots, \psi_C^{(l)}$ are the dropout masks that were applied during the forward propagation of a local client training step.

To find a realization $F_{\Psi_A} \approx F_{\Psi_C}$, we propose to optimize the dropout masks Ψ_A used for the forward propagation of dummy data during the gradient inversion attack. For each dropout layer the corresponding mask $\psi_A^{(i)}$ is initialized randomly from a Bernoulli distribution[2] with probability p. Instead of optimizing solely for the dummy data (x', y'), the attacker optimizes the dropout masks Ψ_A and the dummy data jointly. We rewrite the optimization problem as follows:

$$\underset{(x', y', \Psi_A)}{\arg\min}\ D(\nabla\mathcal{L}_\theta(F_{\Psi_C}(x), y), \nabla\mathcal{L}_\theta(F_{\Psi_A}(x'), y')) + \lambda\Omega. \quad (3)$$

The pseudo code for our proposed Dropout Inversion Attack is given as Algorithm 1.

To calculate the dummy gradient $\nabla\mathcal{L}_\theta(F_{\Psi_A}(x'), y'))$ the attacker forwards the dummy image x' through the model realization F_{Ψ_A}. The reconstruction loss between the shared client gradient and dummy gradient is computed and backpropagated. The gradients for the dummy data (x', y') and

[2]Other initializations are discussed in the technical appendix. It can be found at: https://arxiv.org/abs/2208.06163.

Algorithm 1: Dropout Inversion Attack

Input: F: neural network; $\mathcal{L}$: training loss function; D: gradient distance function;
$\nabla_C = \nabla\mathcal{L}_\theta(F_{\Psi_C}(x), y)$: shared client gradient; p: dropout rate; η: learning rate

Output: (x', y'): training data reconstructions; $\Psi_A = \{\psi_A^{(1)}, \dots \psi_A^{(l)}\}$: learned dropout masks

$x', y' \leftarrow \mathcal{N}(0,1)$; $\psi_A^{(1)}, \dots, \psi_A^{(l)} \leftarrow$ Bernoulli(p); ▷ initialize dummy data and dropout masks
while not converged **do** ▷ reiterate until some optimization criterion is reached
  $\nabla_A \leftarrow \nabla\mathcal{L}_\theta(F_{\Psi_A}(x'), y')$; ▷ calculate dummy gradient
  $\mathcal{L}_A \leftarrow D(\nabla_C, \nabla_A)$; ▷ calculate gradient distance
  $x' \leftarrow x' - \eta\frac{\delta\mathcal{L}_A}{\delta x'}$; $y' \leftarrow y' - \eta\frac{\delta\mathcal{L}_A}{\delta y'}$; $\psi_A^{(i)} \leftarrow \psi_A^{(i)} - \eta\frac{\delta\mathcal{L}_A}{\delta\psi_A^{(i)}}\ \forall i \in 1, ..., l$; ▷ update dummy data and dropout masks
end while
return (x', y'), Ψ_A

the masks $\psi_A^{(i)}$ are calculated and used for optimization. Note that elements of the client dropout masks $\psi_C^{(i)} \in \{0, 1\}$ are binary, whereas the optimized masks $\psi_A^{(i)} \in [0, 1]$ are *fuzzy*, since they are adjusted iteratively. We found that discretization of the masks destabilizes the attack optimization. To avoid scaling effects, we clip the masks between 0 and 1. We provide a PyTorch implementation of DIA[3].

5 Experiments

We use MNIST (Deng 2012) and CIFAR-10 (Krizhevsky, Hinton et al. 2009) datasets that are separated into train and test splits according to the benchmark protocols. For the attacks, we randomly sample a victim client dataset of 128 images from the training data of one federated client as used in the training. For experiments on ImageNet (Russakovsky et al. 2015), we randomly sample 128 images from different classes from the training dataset. Client gradients are computed by performing one training step on victim client data.

Initial experiments are carried out on a Multi Layer Perceptron (MLP) (Rumelhart, Hinton, and Williams 1985) and a small version of a Vision Transformer (ViT) (Dosovitskiy et al. 2020). For experiments conducted on CNN based architectures we modify the LeNet implementation from (Zhao, Mopuri, and Bilen 2020) and a ResNet-18 (He et al. 2016) by adding a dropout layer right before the final fully connected classification layer. We use IG (Geiping et al. 2020) as baseline attack. More details on the model architectures, attack configuration and hyperparameter selection can be found in the technical appendix.

To measure reconstruction quality we calculate the *Structural Similarity* (SSIM) (Wang et al. 2004) between the original and reconstructed images. Higher SSIM indicates higher reconstruction quality. Additional metrics, *i.e.* MSE, PSNR and LPIPS, are reported in the technical appendix.

To measure the similarity between the approximated model F_{Ψ_A} and the client model F_{Ψ_C}, we compute the *Mean Mask Distance* (MMD) between the optimized dropout masks Ψ_A and the client's dropout masks Ψ_C:

$$\text{MMD}(\Psi_A, \Psi_C) = \frac{1}{l}\sum_{i=1}^{l} ||\psi_A^{(i)} - \psi_C^{(i)}||^2. \quad (4)$$

[3]https://github.com/dAI-SY-Group/DropoutInversionAttack

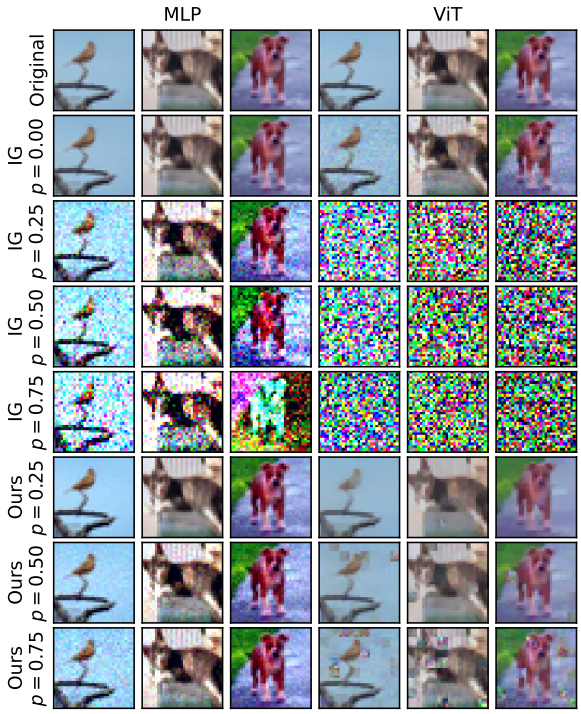


Figure 5: Example reconstructions for batchsize $\mathcal{B} = 1$ for MLP and ViT on CIFAR-10.

Hence, MMD $= 0$ indicates $F_{\Psi_A} = F_{\Psi_C}$, i.e. the attacker model equals the client model. For each metric we report the average across the 128 samples of each victim client dataset.

5.1 Dropout Inversion Attack

In the first set of experiments the MLP and ViT with batchsizes $\mathcal{B} \in \{1, 4, 8, 16\}$ are attacked. Although model utility did not benefit from dropout rates $p > 0.25$ (cf. Fig. 2), we choose $p \in \{0.25, 0.50, 0.75\}$ to assess the efficacy of DIA at increased difficulty. Example reconstructions are visualized in Fig. 5. Numeric results are reported in Fig. 6.

We find that, in contrast to IG (cf. Fig. 2), DIA is able to successfully reconstruct client data from shared gradients even if dropout was used during model training. However,

SSIM decreases with increasing dropout rates and batchsizes. For the MLP with dropout rate $p = 0.75$ and batchsize $\mathcal{B} = 16$, DIA based reconstructions achieve a SSIM of 0.8/0.63 on MNIST/CIFAR-10. For the ViT, increased p and $\mathcal{B}$ affect the reconstruction quality more notably. SSIM drops below a critical value of 0.6 if $p \geq 0.5$ and $\mathcal{B} \geq 4$. However, dropout rates $p \geq 0.25$ also have negative impact on model utility (cf. Fig. 2) and should be avoided for ViTs.

We observe that the joint optimization of dummy data and dropout masks in DIA finds a suitable approximation $F_{\Psi_A} \approx F_{\Psi_C}$ that allows to reconstruct the client data. For the ViT, DIA based reconstructions achieve smaller SSIM compared to WIIG based reconstructions (cf. Fig. 2), *i.e.* if the attacker is informed about Ψ_C. In fact, we observe an inverse correlation between SSIM and MMD (cf. Fig. 6(b)), *i.e.* high reconstruction quality (high SSIM) correlates with small mask distance (low MMD), which is a measure for the similarity between F_{Ψ_A} and F_{Ψ_C}. Since dropout masks have to be approximated per sample, increased batchsizes $\mathcal{B}$ increase the number of attack parameters. In addition, different samples in a batch cause overlapping neuron activations (Pan et al. 2020) and lead to joint gradients. This increases the difficulty of the attack, as can be observed by decreased SSIM and increased MMD in Fig. 6.

5.2 Improving Dropout Mask Approximations

We observe that masks optimized by DIA deviate from client masks with increasing dropout rate and batchsize. To mitigate this effect, we propose to regularize the optimized masks Ψ_A by $\Omega(\Psi_A)$ to match the client's dropout rates:

$$\Omega(\Psi_A) = \sum_{i=1}^{l} \left| p - \left(1 - \frac{||\psi_A^{(i)}||}{n_i}\right) \right|, \tag{5}$$

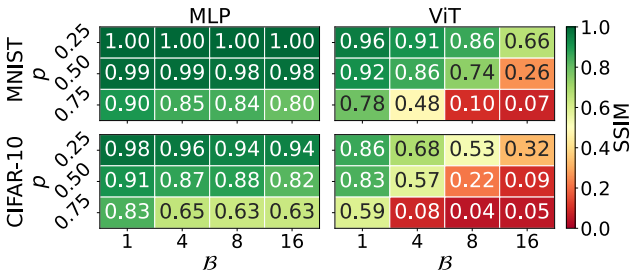
where n_i is the size of dropout mask $\psi_A^{(i)}$. The client's dropout rate p is part of the model architecture and hence known by the attacker by default (cf. Sec. 2.1).

We evaluate the efficacy of $\Omega(\Psi_A)$ for a fixed dropout rate of $p = 0.25$ since higher rates did not improve model utility (cf. Fig. 2). In addition, we tune the impact of $\Omega(\Psi_A)$ by weighting with $\lambda_{\text{mask}} \in \{10^{-4}, 10^{-3}, 10^{-2}\}$.

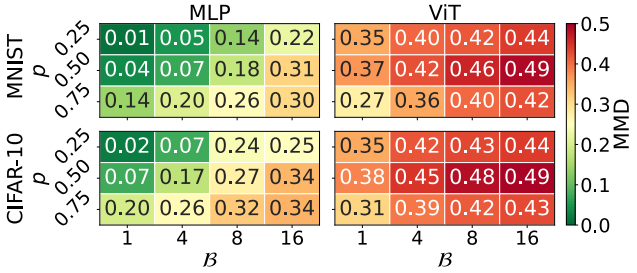
The results of this mask regularization are displayed in Fig. 6 (c). As the SSIM for the MLP is already close to 1, only marginal improvement is observed upon addition of $\Omega(\Psi_A)$. For the ViT the added mask regularization shows a notable increase in SSIM and hence improved reconstruction quality, especially for $\mathcal{B} > 1$. Since we find our proposed mask regularization to improve reconstruction quality, we utilize it with $\lambda_{\text{mask}} = 10^{-4}$ for all further experiments.

5.3 Attacking Dropout at Higher Scales

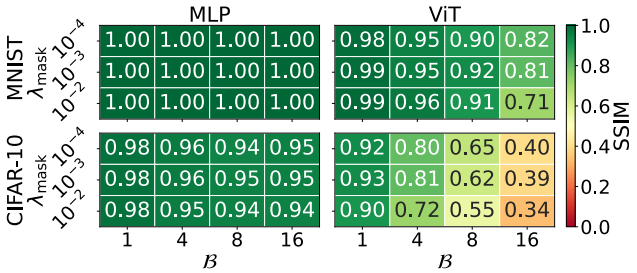
Since DIA jointly optimizes for dummy data and dropout masks, the number of optimized parameters increases with (1) the number of dropout layers l in the model and (2) the input batchsize. ViTs also use an image patch embedding; hence, both input dimensions and batchsize further influence the number of parameters. We therefore want to investigate the applicability of our proposed attack on a state of the art



(a) SSIM for different dropout rates p and batchsizes $\mathcal{B}$.



(b) MMD for different dropout rates p and batchsizes $\mathcal{B}$.



(c) SSIM for different regularization parameter selections λ_{mask} and batchsizes $\mathcal{B}$ with fixed dropout rate $p = 0.25$.

Figure 6: DIA reconstruction results for MLP (left) and ViT (right) on MNIST and CIFAR-10.

sized ViT-B/16 and a practical image classification dataset, *i.e.* ImageNet. Following the recommendations of the original ViT paper, we apply a dropout rate of $p = 0.1$ for the ViT-B/16 (Dosovitskiy et al. 2020).

Fig. 7 shows that even for such a low dropout rate p, IG is not able to reconstruct the data. In comparison, DIA based reconstructions achieve a SSIM of 0.72. As observed before, the reconstruction quality for DIA with dropout is higher compared to IG without dropout. Reconstruction examples are visualized in Fig. 8.

5.4 Attacking Dropout in CNNs

Recent work commonly evaluates gradient inversion attacks for CNN based architectures like LeNet and ResNet (Zhao, Mopuri, and Bilen 2020; Geiping et al. 2020; Wei et al. 2020; Yin et al. 2021). Furthermore, a drop in reconstruction quality was reported when dropout is used before the output layer of a LeNet (Zheng 2021). We therefore investigate the

	Model	p	IG SSIM ↑	Ours SSIM ↑
IN	ViT-B/16	0.00	**0.56**	-
		0.10	*0.01*	***0.72***
MNIST	LeNet	0.00	**0.95**	-
		0.25	0.57	0.94
		0.50	0.40	**0.95**
		0.75	0.23	0.94
	ResNet	0.00	0.88	-
		0.25	0.37	0.94
		0.50	0.18	*0.93*
		0.75	*0.09*	*0.93*
CIFAR-10	LeNet	0.00	**0.89**	-
		0.25	0.48	**0.89**
		0.50	0.32	0.88
		0.75	0.21	0.88
	ResNet	0.00	0.64	-
		0.25	0.28	0.71
		0.50	0.15	*0.70*
		0.75	*0.08*	0.71

Figure 7: SSIM computed from gradients with $\mathcal{B} = 1$ attacked with IG and DIA (Ours) for ViT-B/16 on ImageNet (IN) as well as LeNet and ResNet on MNIST and CIFAR-10. Arrows indicate direction of improvement. Bold and italic formatting highlight best and worst results respectively.

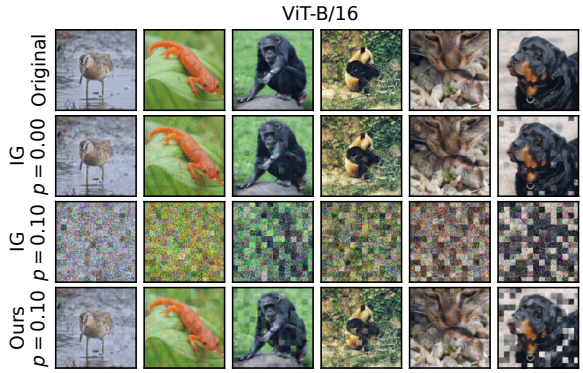


Figure 8: Example reconstructions for batchsize $\mathcal{B} = 1$ for ViT-B/16 on ImageNet.

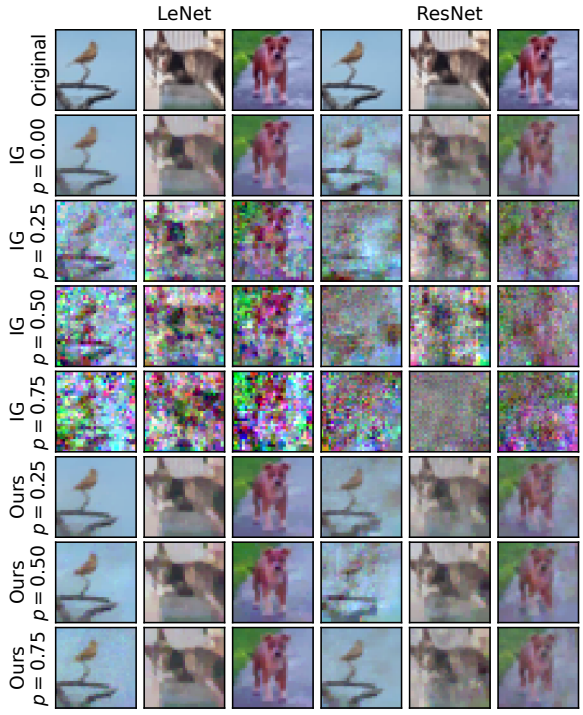


Figure 9: Example reconstructions for batchsize $\mathcal{B} = 1$ for LeNet and ResNet on CIFAR-10.

efficacy of our proposed attack on these CNN based classifiers if dropout is applied before the output layer. The results in Fig. 7 confirm that for the baseline IG attack reconstruction quality decreases for increased dropout rates for both model architectures. In contrast, when DIA is used as attack, client data is successfully reconstructed regardless of enabled dropout. Moreover, compared to the MLP and ViT architectures, SSIM remains at the same level even with increased dropout rates. We argue that since the CNN based architectures utilize only one dropout layer, the gradients of the other layers retain sufficient information for reconstruction. Reconstruction examples are visualized in Fig. 9.

6 Conclusion

Recent work suggests that dropout in neural networks improves data privacy during federated learning, because it seems to prevent gradient inversion attacks. We formalize the impact of dropout on such inversion attacks based on specific realizations of a stochastic model. Dropout causes an inherent mismatch between the model realizations of the attacker and client, which in turn prevents reconstruction of client data. However, this offers a premature sense of security, because an attacker can still reconstruct client data either by being informed about the client's dropout masks or by approximating them. To showcase the vulnerability of dropout protected neural networks, we formulate a novel Dropout Inversion Attack (DIA) that jointly optimizes for client data and dropout masks to approximate the client's model realization. We conduct an extensive systematic empirical study to investigate the impact of dropout on four seminal model architectures and three image classification datasets of increasing complexity. We show that our proposed attack successfully bypasses the seemingly induced protection of dropout and allows to reconstruct data with high fidelity. Although we evaluate our proposed attack solely in an image classification setting, we expect DIA to be universally applicable since the underlying mechanism can be trivially integrated into other iterative inversion attacks. We confirm that the strategic use of architectural features, such as dropout, cannot be assumed to sufficiently protect client privacy in federated learning scenarios. We conclude that a combination of complementary defense mechanisms should be applied in order to protect privacy and maintain model utility.

Acknowledgments

We are funded by the Thuringian Ministry for Economic Affairs, Science and Digital Society (Grant: 5575/10-3).

References

Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H. B.; Patel, S.; Ramage, D.; Segal, A.; and Seth, K. 2017. Practical secure aggregation for privacy-preserving machine learning. In *Practical secure aggregation for privacy-preserving machine learning*, 1175–1191. ISBN 9781450349468.

Deng, L. 2012. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29: 141–142.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929*.

Dwork, C.; and Roth, A. 2013. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9: 211–487.

Enthoven, D.; and Al-Ars, Z. 2020. Fidel: Reconstructing Private Training Samples from Weight Updates in Federated Learning. *arXiv preprint arXiv:2101.00159*.

Geiping, J.; Bauermeister, H.; Dröge, H.; and Moeller, M. 2020. Inverting Gradients-How easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33: 16937–16947.

Gupta, S.; Huang, Y.; Zhong, Z.; Gao, T.; Li, K.; and Chen, D. 2022. Recovering private text in federated learning of language models. *arXiv preprint arXiv:2205.08514*, 1–20.

Hanson, S. J. 1990. A stochastic version of the delta rule. *Physica D: Nonlinear Phenomena*, 42: 265–272.

Hatamizadeh, A.; Yin, H.; Roth, H. R.; Li, W.; Kautz, J.; Xu, D.; and Molchanov, P. 2022. Gradvit: Gradient inversion of vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10021–10030.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem: 770–778.

Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. R. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hofmann, M.; and Mäder, P. 2021. Synaptic Scaling–An Artificial Neural Network Regularization Inspired by Nature. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15.

Huang, Y.; Gupta, S.; Song, Z.; Li, K.; and Arora, S. 2021. Evaluating Gradient Inversion Attacks and Defenses in Federated Learning. *Advances in Neural Information Processing Systems*, 34: 7232–7241.

Huang, Y.; Song, Z.; Chen, D.; Li, K.; and Arora, S. 2020a. TextHide: Tackling Data Privacy in Language Understanding Tasks. *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, 1368–1382.

Huang, Y.; Song, Z.; Li, K.; and Arora, S. 2020b. Instahide: Instance-hiding schemes for private distributed learning. In *International conference on machine learning*, 4507–4518. PMLR.

Jayaraman, B.; and Evans, D. 2019. Evaluating differentially private machine learning in practice. *Proceedings of the 28th USENIX Security Symposium*, 1895–1912.

Jeon, J.; Lee, K.; Oh, S.; Ok, J.; et al. 2021. Gradient Inversion with Generative Image Prior. *Advances in Neural Information Processing Systems*, 34: 29898–29908.

Jin, X.; Chen, P.-Y.; Hsu, C.-Y.; Yu, C.-M.; and Chen, T. 2021. Catastrophic Data Leakage in Vertical Federated Learning. *Advances in Neural Information Processing Systems*, 34.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; D'Oliveira, R. G.; Eichner, H.; Rouayheb, S. E.; Evans, D.; Gardner, J.; Garrett, Z.; Gascón, A.; Ghazi, B.; Gibbons, P. B.; Gruteser, M.; Harchaoui, Z.; He, C.; He, L.; Huo, Z.; Hutchinson, B.; Hsu, J.; Jaggi, M.; Javidi, T.; Joshi, G.; Khodak, M.; Konecní, J.; Korolova, A.; Koushanfar, F.; Koyejo, S.; Lepoint, T.; Liu, Y.; Mittal, P.; Mohri, M.; Nock, R.; Özgür, A.; Pagh, R.; Qi, H.; Ramage, D.; Raskar, R.; Raykova, M.; Song, D.; Song, W.; Stich, S. U.; Sun, Z.; Suresh, A. T.; Tramèr, F.; Vepakomma, P.; Wang, J.; Xiong, L.; Xu, Z.; Yang, Q.; Yu, F. X.; Yu, H.; and Zhao, S. 2021. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14: 1–210.

Kaissis, G.; Ziller, A.; Passerat-Palmbach, J.; Ryffel, T.; Usynin, D.; Trask, A.; Lima, I.; Mancuso, J.; Jungmann, F.; Steinborn, M. M.; Saleh, A.; Makowski, M.; Rueckert, D.; and Braren, R. 2021. End-to-end privacy preserving deep learning on multi-institutional medical imaging. *Nature Machine Intelligence 2021 3:6*, 3: 473–484.

Kingma, D. P.; and Ba, J. L. 2014. Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning Multiple Layers of Features from Tiny Images. 1–60.

Li, Z.; Wang, L.; Chen, G.; Shafq, M.; and Gu, Z. 2022. A Survey of Image Gradient Inversion Against Federated Learning. *TechRxiv*, 1–13.

Lu, J.; Zhang, X. S.; Zhao, T.; He, X.; and Cheng, J. 2022. APRIL: Finding the Achilles' Heel on Privacy for Vision Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10051–10060. IEEE.

Lyu, L. 2021. DP-SignSGD: When efficiency meets privacy and robustness. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2021-June: 3070–3074.

Pan, X.; Zhang, M.; Yan, Y.; Zhu, J.; and Yang, M. 2020. Exploring the Security Boundary of Data Reconstruction via Neuron Exclusivity Analysis. *Proceedings of the 31st USENIX Security Symposium, Security 2022*, 3989–4006.

Piotrowski, A. P.; Napiorkowski, J. J.; and Piotrowska, A. E. 2020. Impact of deep learning-based dropout on shallow neural networks applied to stream temperature modelling. *Earth-Science Reviews*, 201: 103076.

Rudin, L. I.; Osher, S.; and Fatemi, E. 1992. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60: 259–268.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1985. Learning Internal Representations by Error Propagation. 1–49.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115: 211–252.

Sattler, F.; Wiedemann, S.; Muller, K. R.; and Samek, W. 2020. Robust and Communication-Efficient Federated Learning from Non-i.i.d. Data. *IEEE Transactions on Neural Networks and Learning Systems*, 31: 3400–3413.

Scheliga, D.; Mäder, P.; and Seeland, M. 2022a. Combining Variational Modeling with Partial Gradient Perturbation to Prevent Deep Gradient Leakage. *arXiv preprint arXiv:2208.04767*, 1–21.

Scheliga, D.; Mäder, P.; and Seeland, M. 2022b. PRECODE - A Generic Model Extension to Prevent Deep Gradient Leakage. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 3605–3614.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15: 1929–1958.

Sun, J.; Li, A.; Wang, B.; Yang, H.; Li, H.; and Chen, Y. 2021. 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*, 9311–9319.

Wang, Y.; Deng, J.; Guo, D.; Wang, C.; Meng, X.; Liu, H.; Ding, C.; and Rajasekaran, S. 2020. SAPAG: A Self-Adaptive Privacy Attack From Gradients. *arXiv preprint arXiv:2009.06228*, 1–8.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13: 600–612.

Wei, W.; and Liu, L. 2021. Gradient Leakage Attack Resilient Deep Learning. *IEEE Transactions on Information Forensics and Security*.

Wei, W.; Liu, L.; Loper, M.; Chow, K.-H.; Gursoy, M. E.; Truex, S.; and Wu, Y. 2020. A Framework for Evaluating Gradient Leakage Attacks in Federated Learning. *arXiv preprint arXiv:2004.10397*, 1–25.

Yin, H.; Mallya, A.; Vahdat, A.; Alvarez, J. M.; Kautz, J.; and Molchanov, P. 2021. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16337–16346.

Zhang, R.; Guo, S.; Wang, J.; Xie, X.; and Tao, D. 2022. A Survey on Gradient Inversion: Attacks, Defenses and Future Directions. *arXiv preprint arXiv:2206.07284*.

Zhao, B.; Mopuri, K. R.; and Bilen, H. 2020. iDLG: Improved Deep Leakage from Gradients. *arXiv preprint arXiv:2001.02610*, 1–5.

Zheng, Y. 2021. Dropout against Deep Leakage from Gradients. *arXiv preprint arXiv:2108.11106*, 1–6.

Zhu, L.; and Han, S. 2020. Deep Leakage from Gradients. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12500 LNCS: 17–31.