Exploring the Security Boundary of Data Reconstruction via Neuron Exclusivity Analysis

Xudong Pan, Mi Zhang✉, Yifan Yan, Jiaming Zhu, Min Yang✉
Fudan University, China
{xdpan18, mi_zhang, yanyf20, 19210240146, m_yang}@fudan.edu.cn

Abstract

Among existing privacy attacks on the gradient of neural networks, *data reconstruction attack*, which reverse engineers the training batch from the gradient, poses a severe threat on the private training data. Despite its empirical success on large architectures and small training batches, unstable reconstruction accuracy is also observed when a smaller architecture or a larger batch is under attack. Due to the weak interpretability of existing learning-based attacks, there is little known on why, when and how data reconstruction attack is feasible.

In our work, we perform the first analytic study on the security boundary of data reconstruction from gradient via a microcosmic view on neural networks with rectified linear units (ReLUs), the most popular activation function in practice. For the first time, we characterize the insecure/secure boundary of data reconstruction attack in terms of the *neuron exclusivity state* of a training batch, indexed by the number of ***Ex**clusively **A**ctivated **N**eurons* (ExANs, i.e., a ReLU activated by only one sample in a batch). Intuitively, we show a training batch with more ExANs are more vulnerable to data reconstruction attack and vice versa. On the one hand, we construct a novel deterministic attack algorithm which substantially outperforms previous attacks for reconstructing training batches lying in the insecure boundary of a neural network. Meanwhile, for training batches lying in the secure boundary, we prove the impossibility of unique reconstruction, based on which an exclusivity reduction strategy is devised to enlarge the secure boundary for mitigation purposes.

1 Introduction

From G. Hinton's Turing-award-winning work on *backpropagation* in 1986 [39] to modern optimizers standardized in popular deep learning libraries like Google's Tensorflow [1] and Facebook's PyTorch [36], the *gradient* plays a ubiquitous role in the learning process of most deep learning models. Intuitively, taking the task of image classification for example, the gradient provides the image classifier with a good direction to adapt its parameters for narrowing the errors (i.e., the *loss function*) between the predictions and the ground-truth class labels. As the model iteratively updates its parameters along the opposite direction of the gradient on different training samples, the loss function gradually decreases and the prediction of the learning model becomes more accurate.

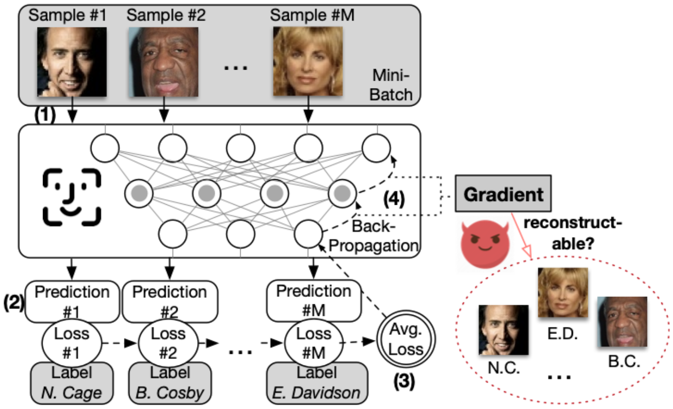


Figure 1: The information flow of producing the average gradient of a training batch in a face recognition model and the scenario of data reconstruction attack.

However, accompanied with the fundamental role of gradient in deep learning is its tell-tale heart. As Fig. 1 shows, in a typical face recognition system, a batch of training images are first input to the neural network classifier. The classifier then predicts the labels, computes the average loss function, and uses back-propagation to calculate the gradient as the parameter derivative of the average loss. As the gradient is explicitly derived from the data inputs and the labels, it is reasonable for an attacker to expect the gradient would leak sensitive information about the original training data. With the booming of novel distributed learning paradigms [5, 51], several research works start to explore the feasibility of inferring the data property [29], the membership [29, 31], the class representatives [16, 49], or the data inputs [12, 52, 53] from the gradient potentially leaked to a man-in-the-middle attacker or an honest-but-curious server [19, 28].

Despite the feasibility of privacy attacks via the gradient, most previous attacks notice a common yet unclear performance bottleneck on the privacy leakage of their proposed approaches. For example, Melis et al. [29] report the precision of sensitive word inference from the gradient decreases by over 30% when the batch size increases by 8×. Nasr et al. [31] report a shallower neural network model is observed to leak less membership information. Zhu et al. [53] report the iterations required to reverse engineer a training batch from its average gradient increase by 10× when the batch size increases from 1 to 8, while the proposed attack is more likely to fail when the neural network is shallow. To summarize, the information leakage from the gradient seemingly decreases for a larger training batch and a shallower neural network model. However, whether this phenomenon has a common root cause interwoven with the underlying mechanism of deep learning? To the best of our knowledge, existing literature provides almost no clue to this fundamental question.

Our Work. We investigate the above question by dissecting the mechanism of *data reconstruction attack* [12, 52, 53], an emerging privacy threat which exploits the leaked average gradient of a deep learning model to reverse engineer the corresponding training batch. As shown in the right part of Fig. 1, data reconstruction attack targets at reconstructing the training samples from the corresponding gradient, which poses severe threats on the confidentiality of private training data. As one of the earliest data reconstruction attacks, Zhu et al. [53] propose a learning-based approach to restore the training batch, which views the unknown training batch as learnable variables (i.e., *dummy data*). By minimizing the L2 distance between the gradient calculated on the dummy data and the ground-truth gradient (i.e., *gradient matching*), they surprisingly observe the reconstruction is possible when the batch size is no larger than 8 on CIFAR-100 [22], while the reconstruction quality can be unstable for different trials and relatively small victim models. Follow-up works [12, 52] present technical adjustments to the learning-based framework in [53], with similar bottlenecks observed on data reconstruction. However, due to their weak interpretability, none of the previous works have successfully characterized *why, when and how data reconstruction from gradient is feasible*, which, from our perspective, can be a key entrance to understand and strengthen the privacy properties of the gradient.

To explore the security boundary of data reconstruction from gradient, we present the first analytic study of data reconstruction attacks on the family of fully-connected neural networks (FCNs) with rectified linear units (ReLUs [13]), a quintessential neural network architecture which has been commonly used for demonstrating novel attack and defense insights [17, 38, 46]. As probably the most popular activation function in deep learning practices [13], a ReLU lets non-negative inputs pass through without modification and blocks the negative inputs. This special gate-like behavior of ReLU allows each input sample to hold its own set of *activation paths* as its *activation pattern* [25, 30]. We construct deterministic algorithms which decode the hidden information in the average gradient to determine the activation patterns of every single sample, a critical step to reduce the otherwise highly nonlinear gradient-matching problem to a linear equation system regarding the inputs to ease the further analytical studies. Investigating the conditions under which the activation patterns can be reconstructed from the gradient, we mainly make the following key contributions:

(1) Neuron Exclusivity State Analysis. For the first time, we point out *neuron exclusivity state*, indexed by the number of ***Ex**clusively **A**ctivated **N**eurons* (ExANs, i.e., a ReLU activated by only one sample in a batch during a forward pass), is critical to the feasibility of data reconstruction attack. Specifically, we characterize the following boundary conditions for the neuron exclusivity state of a training batch under attack.

(2) Boundary of Insecure Exclusivity States. We discover the condition of *sufficient exclusivity*, i.e., when each sample in a batch has at least 2 ExANs at the last ReLU layer and 1 at the other layers, as a strong indicator to insecure neuron exclusivity states (Section 5). Specifically, we show a deterministic attack algorithm with guaranteed reconstruction accuracy (Theorem 1) can be constructed for any training batch satisfying the sufficient exclusivity condition. Evaluation on 5 real-world scenarios covering medical, face recognition and visual datasets and a diverse set of FCNs of varied depth and width shows, our attack consistently outperforms previous attacks by a large margin in terms of reconstruction recognizability and reaches 100% label inference accuracy (Section 7). Besides, we also extend our attack algorithm to classifiers based on convolutional neural networks (CNNs) by combining analytical and optimization-based techniques.

(3) Boundary of Secure Exclusivity States. By dissecting the remaining exclusivity state space, we further determine the *lack of exclusivity* condition, i.e., when each sample has 0 ExAN at the first ReLU layer, as an indicator to the impossibility of unique reconstruction (Section 6). For these states, we prove there always exist infinitely many artifact batches which yield exactly the same gradient as the victim's ground-truth batch, and derive the lower bound for the largest distance between an artifact batch and the ground-truth batch (Theorem 2). This observation inspires us to devise an exclusivity reduction strategy, which replaces the first ReLU layer as a linear layer, to enhance the privacy of an arbitrary batch of training samples when its size is larger than the number of neurons in the first layer, with almost no degradation on the model performance. For the completeness of our study, we also present preliminary experimental results in Section 8 to empirically analyze the performance of data reconstruction on the remaining states.

2 Related Work

Data Reconstruction Attack. Different from inferring class representatives [9, 16], data reconstruction attack primarily aims at recovering each single training sample behind the intermediate computational results accessed by the attacker. Although [41] first refers to such an attack class as data reconstruction attack, their work mainly study reconstructing a batch of training samples from the changes of their outputs from an updated neural network, which is merely a realistic threat model in most distributed learning paradigms. Parallel to this work, [49] improves [16] with a multi-task GAN to generate individual samples by refining the recovered class representatives, which however requires strong inner-class similarity of the datasets. These limitations make these two attacks not directly applicable to our threat model.

Recently, starting from [53], a branch of research [12, 52, 53] begins to explore a brute-force yet general approach towards data reconstruction attacks with meaningful empirical results. Solving the gradient matching problem via optimization, these works mainly differ in the choice of the distance function to minimize (L2 distance in [52, 53] and cosine distance in [12]). Although [52] uses the property of neural networks to recover the label of a single sample in prior before the learning-based attack, the trick only works for the gradient of a single sample, which makes their method identical to [53] when applied to the average gradient. Nevertheless, existing attacks mainly stay at an empirical level and aim at showing the feasibility of data reconstruction attacks from the average gradient. Yet, almost no existing works attempt to explain the feasibility and the underlying mechanisms of data reconstruction attack.

Privacy Attacks on Training Data and Beyond. As gradients can be more easily accessed in open-network distributed learning systems, a number of recent works begin to study various types of information leakage from gradients [16, 29, 31]. For example, [29] demonstrates the possibility of inferring from the gradient whether the training samples share certain properties (e.g., whether the faces are wtih eye-glasses) and [16] leverages a generative adversarial learning paradigm to infer the class representatives, while [31] exploits the gradient for membership inference. Different from these existing studies, we are more curious about the feasibility and the theoretical limit of data reconstruction attack, considering its severe threats posed on the private training data [53]. Besides exploiting the gradient for breaking the training data privacy, researchers also explore, e.g., using the model parameters to infer the properties of training data [4, 11], using the intermediate data representations to infer the sensitive attribute values of data samples [9, 10, 34], or using model explanations to reconstruct significant parts of the training set [44]. Aside from training data privacy, previous studies also cover many other aspects of machine learning privacy, including the privacy risks of the data membership [26, 42, 43], the parameters [46], the hyper-parameters [48], the model architecture [8] or its functionality [17, 33].

3 Preliminary

Gradient in Deep Learning. Gradient plays an indispensable and ubiquitous role in modern deep learning systems, especially during the model training phase. In the following, we focus on the K-class classification task which covers many real-world use cases of deep learning. We denote a learning model as $f(\cdot;W)$, where W denotes its learnable parameters, and a training sample (X,Y), where X is called the data input and Y is the ground-truth label, ranging in $\{1,\ldots,K\}$. By convention, the learning model takes in the data input X and outputs a vector $f(X;W) \in \mathbb{R}^K$ (abbrev. f), where the c-th element of this vector after a softmax operation predicts the probability of X in class c, i.e., $p_c := [\text{softmax}(f(X;W))]_c = \exp f_c / \sum_{c=1}^K \exp f_c$, where the operator $[\cdot]_c$ takes the c-th entry/row of a vector/matrix, or the c-th row of a matrix.

With this prediction, the loss function $\ell(f(X;W),Y)$ (abbrev. ℓ) is usually calculated as the cross-entropy loss between the predicted probabilities and the ground-truth label, i.e., $\ell(f(X;W),Y) := -\log p_Y = -f_Y + \log \sum_{c=1}^K \exp f_c$. With the aid of modern optimization algorithms (e.g., SGD [37] and Adam [20]), the model parameters are updated along the opposite direction of the gradient, i.e., $\overline{G}(X,Y;W) := \nabla_W \ell(f(X;W),Y)$, with a prescribed step size, which guarantees the loss function to decrease iteratively, indicating that the learning model would make more accurate predictions.

In practice, deep learning systems mainly use the average gradient calculated on multiple training samples (i.e., a batch) for parameter updating, which is usually more suitable for modern parallel computation devices and results in much faster convergence rate [3]. Formally, given a batch of M training samples $\{(X_m,Y_m)\}_{m=1}^M$, the average gradient is calculated as the coordinate-wise arithmetic average of the gradients for each single sample, which formally writes $\overline{G}(\{(X_m,Y_m)\}_{m=1}^M;W) := \frac{1}{M}\sum_{m=1}^M \nabla_W \ell(f(X_i;W),Y_i)$.

From Gradient Matching to Gradient Equation. Existing data reconstruction attacks suppose the attacker captures the average gradient of an unknown batch and has a white-box knowledge about the victim's learning model (i.e., the parameters and the architecture). In practice, such an attacker may be a man-in-the-middle attacker or an honest-but-curious server in distributed learning systems deployed in open networks (e.g., federated learning [21]/collaborative training [5]). Given the leaked average gradient $\overline{G}$, previous attacks commonly adopt a learning-based approach to solve the following *gradient matching* problem,

$$\min_{\{X_m,Y_m\}_{m=1}^M} D\left(\frac{1}{M}\sum_{m=1}^M \frac{\partial \ell(f(X_m;W),Y_m)}{\partial W}, \overline{G}\right) \tag{1}$$

where $\{X_m, Y_m\}_{m=1}^{M}$ are the learnable variables (i.e., *dummy inputs/labels*) in the gradient matching problem, and a predefined function D measures the distance between the gradient produced by the variables under optimization with the ground-truth average gradient. For example, [52, 53] implement D as the layerwise L2 distance between the ground-truth gradient and the gradient calculated from the dummy inputs and dummy labels, while [12] proposes to use the layerwise cosine distance alternatively. Using standard optimizers like L-BFGS [27] or Adam [20] to minimize the learning objective in (1) w.r.t. the dummy inputs and labels, one is expected to find a batch of $\{X_m, Y_m\}_{m=1}^{M}$ which yield an average gradient close to the ground-truth gradient. According to the results in [12, 52, 53], the authors find the learned dummy inputs are perceptually close to the ground-truth inputs. However, the effectiveness of previous learning-based reconstruction attacks are also observed to rapidly deteriorate when the batch size M increases and the size of the learning model decreases. Yet, there is still little known about the mechanisms which determine this commonly observed yet unclear phenomenon.

In our viewpoint, to optimize the gradient matching problem in (1) is equivalent to solve the *gradient equation*:

$$\sum_{m=1}^{M} \frac{\partial \ell(f(X_m;W), Y_m)}{\partial W} = M\overline{G} \tag{2}$$

where $\{(X_m, Y_m)\}_{m=1}^{M}$ are the variables. In other words, the solvability and the uniqueness of the solutions to the gradient equation would largely determine the feasibility of data reconstruction attacks, which is however scarcely explored.

Fully-Connected Neural Networks with ReLU. Considering the generality of this open problem, our first analytical study mainly focus on fully connected neural networks (FCNs) with rectified linear units (ReLUs). On the one hand, FCN is a quintessential neural network architecture [13] which is commonly used for demonstrating novel attack and defense insights [17, 38, 46], and a popular choice for classification tasks on data samples in vector form or feature vectors extracted from upstream feature extraction models [18]. On the other hand, due to its numeric stability [13], ReLU is commonly implemented in a very broad class of popular neural network architectures including both FCNs and deep convolutional neural networks (CNN). Intuitively, a ReLU $\sigma(\cdot)$ can be viewed as a gate structure which allows non-negative values to pass through without any change and meanwhile blocks negative values by outputting 0 instead, which is formally written as $\sigma(x) = x$ if $x \geq 0$; $\sigma(x) = 0$ if $x < 0$.

For simplicity, we refer to an FCN with ReLU as an FCN. Formally, an $(H+2)$-layer FCN has the following formulation $f(X; W_0, W_1, \ldots, W_H, b_0, b_1, \ldots, b_H) = W_H\sigma(W_{H-1}\ldots(W_1\sigma(W_0X + b_0) + b_1)\ldots + b_{H-1}) + b_H$, where $W_i \in \mathbb{R}^{d_{i+1}\times d_i}$ is the weight matrix at the i-th layer, $b_i \in \mathbb{R}^{d_{i+1}}$ is the bias vector, the data input $X \in \mathbb{R}^{d_0}$, $d_{H+1} = K$, i.e., the class number, and σ is the ReLU activation function. For example, when $H = 1$, the model $W_1\sigma(W_0X + b_0) + b_1$ is called a three-layer FCN. We denote an FCN architecture in the form of $(d_0\text{-}d_1\text{-}\ldots\text{-}d_{H+1})$. Moreover, without loss of generality, we would omit the bias terms in our analysis for the simplicity of notations.

Activation Patterns. Considering the gate-like behavior of ReLU, when a representation is input to a neural network with ReLU, each coordinate of the representation selectively passes through a part of neurons at the current layer and meanwhile is blocked by the remaining neurons due to the negativity or a vanishing weight of the neural connection. As Fig. 2 shows, after forwarding through the whole neural network layer by layer, each sample has a set of computation paths in the neural network, which forms its *activation pattern*. Below, we develop the idea of activation pattern in a formal way.

ReLU is applied to a vector in a coordinate-wise way. For example, the i-th output of the first layer, i.e., $\sigma(W_0X + b_0)$, is reformulated as $\sigma(W_0X + b_0) := D_1(X; W_0, b_0)(W_0X + b_0)$ [25], where $D_1(X; W_0, b_0) = \text{diag}(\mathbf{1}\{\sigma(W_0X + b_0) \succ 0\})$, i.e., a diagonal matrix whose j-th diagonal entry is 1 when the j-th output of the first layer is positive and otherwise 0. For simplicity, we denote the last term as $D_1(X)(W_0X + b_0)$. We call such a matrix $D_1(X)$ the *activation matrix* of X at the first layer. Similarly, we can reformulate the whole ReLU FCN as $f(X) = W_HD_H(W_{H-1}\ldots(W_1D_1(W_0X + b_0) + b_1)\ldots + b_{H-1}) + b_H$ where the sequence of activation matrices $(D_1, \ldots, D_H)$ describes the *activation pattern* for the data input X.

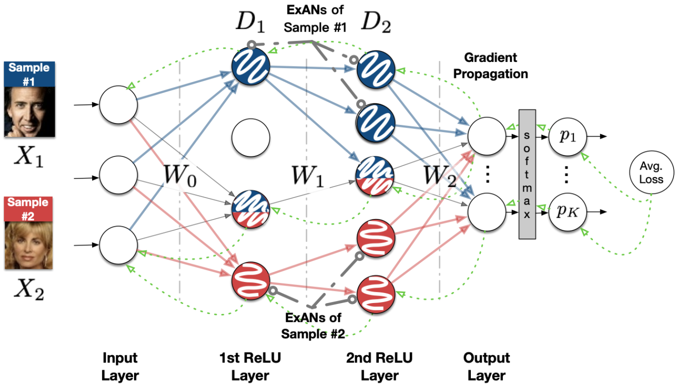


Figure 2: The forward and backward phase of a training batch in a 4-layer FCN (better viewed in color).

Finally, we would like to mention a useful property of the activation pattern during the gradient back-propagation, that is, the activation matrix commutes with the derivative operation, i.e., $\nabla_{W_0}D_i(X)W_{i-1}\ldots W_0X = D_i\nabla_{W_0}W_{i-1}\ldots W_0X$. In other words, the gradient backpropagates along the same activated path as in the forwarding phase. Fig. 2 illustrates the role of the activation pattern in the forward and the backward phases of an FCN, where each data sample is forwarded through a set of computation paths which composes its *activation pattern* (D_1, D_2). For example, as the blue directed lines show, Sample #1 passes through the 1st and the 3rd neuron at the first ReLU layer, which means its activation matrix at the first layer D_1 is $\text{diag}(1,0,1,0)$. Similarly, at the second ReLU layer, its activation matrix D_2 is $\text{diag}(1,1,1,0,0)$. Moreover, we call a neuron which is only activated by one sample in an input batch as the *exclusively activated* neuron (i.e., ExAN) of the corresponding sample (marked in the same color of the sample). For simplicity, the green dashed lines plot parts of the back-propagation paths: the gradient signal is non-vanishing only along the same activation pattern in the forward phase.

4 Overview of Analytic Framework

Threat Model. As summarized in Table 1, we follow almost the same threat model as in existing data reconstruction attacks [12, 53], where the attacker has the knowledge of:

1. The ground-truth average gradient $\overline{G}$ calculated on a batch of M training samples.
2. The architecture and the parameters of an FCN with respect to which the gradient is calculated.

Unlike previous attacks, we do not require the knowledge of the batch size M. As Section 5 will show, the attacker can determine the batch size from the gradient alone for certain exclusivity states. In our analysis on the boundary conditions, we additionally assume the model has random weights to ensure an attacker cannot exploit the otherwise trained parameters for better attacks. Nevertheless, we later show this assumption has no influence on the effectiveness of our proposed attack.

Summary of Key Results. As one of our major contributions, we for the first time unveil and prove the strong relation between the feasibility of data reconstruction attacks on FCNs and the *exclusivity* of activation patterns of samples in a batch (i.e., *neuron exclusivity state*), indexed by the number of ***Ex**clusively **A**ctivated **N**eurons* (*ExAN*s) in each ReLU layer. First, we formally define what is an ExAN.

Definition 1 (ExAN). *Given a batch* $\{(X_m, Y_m)\}_{m=1}^{M}$*, we call the* j*-th neuron at the* i*-th layer is an ExAN if* $\sum_{m=1}^{M}[D_i(X_m)]_j = 1$*, where* $[D_i(X_m)]_j$ *denotes the* j*-th diagonal entry of the activation pattern of* X_m *at the* i*-th layer.*

Literally, an ExAN is a ReLU activated by only one sample in a batch during the forward pass. For intuition, Fig. 2 illustrates two data samples and their corresponding ExANs during their computation in a four-layer FCN. We further denote the number of ExANs for the m-th sample at the i-th layer as N_i^m, which is calculated as $n(\{j : [D_i(X_m)]_j = 1 \bigwedge \forall m' \neq m, [D_i(X_m)]_j = 0\})$, where $n(\cdot)$ denotes the cardinality of a set. Based on the definition, we present the following boundary conditions which provide sufficient conditions for both the insecure and the secure neuron exclusivity states respectively.

- **Insecure Boundary Condition.** *(Sufficient Exclusivity)*: $N_H^m \geq 2$ and $\forall i = 1, \ldots, H-1$, $N_i^m \geq 1$. Intuitively, the condition of *sufficient exclusivity* characterizes that each sample in a batch has at least 2 ExANs at the last ReLU layer and has at least 1 ExAN at the other ReLU layers. We call such a batch as an *insecure* batch. In this case, we present in Section 5 the construction of a deterministic attack algorithm which has guaranteed reconstruction accuracy and stably outperforms previous attacks in evaluation (Sections 7.2).
- **Secure Boundary Condition.** *(Lack of Exclusivity)*: $N_1^m = 0$ and $M > d_1$. As a contrast, the condition of *lack of exclusivity* covers the situations when each sample in a batch activates the same set of neurons in the first layer. In this case, we prove the impossibility of unique reconstruction based on the gradient only, and correspondingly derive a simple yet effective privacy enhancing strategy based on a slight modification on the FCN architecture (Section 6).

5 Reconstruction under Sufficient Exclusivity

In this section, we present a novel deterministic algorithm for reconstructing an unknown insecure batch $\{(X_m, Y_m)\}_{m=1}^{M}$ from the average gradient $(\overline{G}_0, \ldots, \overline{G}_H)$ with guaranteed accuracy.

Gradient Equation of an FCN. As mentioned in the first part of Section 2, the loss function $\ell_m := \ell(f(X_m), Y_m)$ is usually implemented as the cross-entropy between the ground-truth label Y_m and the "softmax-ed" $f(X_m)$. With simple calculations, the gradient of the entropy loss on the c-th output of $f(X_m)$, i.e., f_c^m, has the following closed form:

$$\partial \ell_m / \partial f_c^m = \overline{g}_c^m = -1 + p_c^m \text{ if } c = Y_m \text{ else } p_c^m \quad , \qquad (3)$$

where $p_c^m := p_c(X_m)$ is the predicted probability for the sample X_m in class c. For convenience, we use the *loss vector* $\overline{g}^m$ to denote $(\overline{g}_1^m, \ldots, \overline{g}_K^m)$.

Based on the chain rule, the gradient of W_i (i.e., the weight of the $(i+1)$-th layer) contributed by the m-th sample is $\nabla_{W_i} \ell_m = \sum_{c=1}^{K} \overline{g}_c^m \nabla_{W_i} f_c^m$. By summing over m and replacing the left side as the captured gradient at the i-th layer, i.e., $\overline{G}_i$, we have the following gradient equation for W_i, $M\overline{G}_i = \sum_{m=1}^{M} \sum_{c=1}^{K} \overline{g}_c^m \nabla_{W_i} f_c^m$, which provides a highly complicated nonlinear equation system for the attacker to solve, where the nonlinearity lies in $\overline{g}_c^m$ and the activation patterns $D_i(X_m)$ (or, concisely, D_i^m) contained in f_c^m.

Simplification to Linear Equation System. Under the condition of sufficient exclusivity, we show both $\{(\overline{g}_c^m)_{c=1}^{K}\}_{m=1}^{M}$ and $\{(D_i^m)_{i=1}^{H}\}_{m=1}^{M}$ can be uniquely determined to reduce the nonlinear gradient equation above to a linear equation system.

(1) Inference of Loss Vectors: First, to infer $\overline{g}_c^m$, we con-

Table 1: Summary of threat models of different data reconstruction attacks.

	DLG [53]	**iDLG** [52]	**Inverting** [12]	**Ours**
Target Architecture	Unspecified	Unspecified	Unspecified	FCN/Extensible to CNN
Attack Technique	Optimization	Optimization	Optimization	Analytic/Hybrid
Type of Leaked Gradient *(Average/Single-Sample)*	Both	Single-Sample	Both	Both
Batch Size is Required?	Required	N/A	Required	Not Required

sider the gradient equation for W_H, i.e.,

$$M[\overline{G}_H]_c = \sum_{m=1}^{M} \overline{g}_c^m f_{H-1}^m, \tag{4}$$

where $f_{H-1}^m := D_H^m W_{H-1} \ldots D_1^m W_0 X_m$. We discover the following sufficient condition for recovering $\overline{g}_c^m$.

Proposition 1. *A sufficient condition for determining the ratio of $\overline{g}_c^m$ over $\overline{g}_1^m$ is, each data sample has at least two ExANs at the last but one layer.*

As a proof, we construct the following algorithm to determine the ratios $\{(\overline{g}_c^m/\overline{g}_1^m)_{c=2}^K\}_{m=1}^M$. For better intuition, we consider the case in Fig. 2 where each sample X_m in a batch of size 2 has two ExANs at the last layer and one commonly activated neuron (i.e., X_1 takes up the 1st and the 2nd neurons, and X_2 the 4th and 5th). In other words, both samples activate two different neurons at the last ReLU layer of the neural network. According to the gradient equation above, by forming the ratio vector $[\overline{G}_H]_4/[\overline{G}_H]_3$, we notice that for each ExAN of the 1st sample (i.e., the 1st & the 2nd neuron), the element $[[\overline{G}_H]_4/[\overline{G}_H]_3]_1 = [[\overline{G}_H]_4/[\overline{G}_H]_3]_2 = \overline{g}_4^1/\overline{g}_3^1$. We also provide a schematic proof of this property in Fig. 3. Based on this property, we can practically detect the repetitive values in $[\overline{G}_H]_c/[\overline{G}_H]_1$ to determine the ExANs for the m-th sample and then collect the value at the corresponding index of the ratio vector $[\overline{G}_H]_c/[\overline{G}_H]_1$ as the corresponding ratio $\overline{g}_c^m/\overline{g}_1^m$. Similarly, by enumerating the class index c, we can again reduce the $M \times K$ variables in $\{(\overline{g}_c^m)_{c=1}^K\}_{m=1}^M$ to K variables. Below, we present two noteworthy remarks on inferring the label and determining the concrete values of $\overline{g}_c^m$ based on the ratio equations. For more implementation details, please refer to Algorithm B.1.

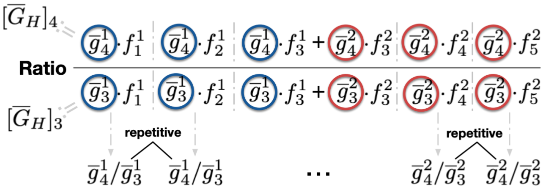

$[\overline{G}_H]_4$
$\overline{g}_4^1 \cdot f_1^1$ | $\overline{g}_4^1 \cdot f_2^1$ | $\overline{g}_4^1 \cdot f_3^1 + \overline{g}_4^2 \cdot f_3^2$ | $\overline{g}_4^2 \cdot f_4^2$ | $\overline{g}_4^2 \cdot f_5^2$
Ratio
$\overline{g}_3^1 \cdot f_1^1$ | $\overline{g}_3^1 \cdot f_2^1$ | $\overline{g}_3^1 \cdot f_3^1 + \overline{g}_3^2 \cdot f_3^2$ | $\overline{g}_3^2 \cdot f_4^2$ | $\overline{g}_3^2 \cdot f_5^2$
$[\overline{G}_H]_3$
repetitive
repetitive
$\overline{g}_4^1/\overline{g}_3^1$ $\overline{g}_4^1/\overline{g}_3^1$
...
$\overline{g}_4^2/\overline{g}_3^2$ $\overline{g}_4^2/\overline{g}_3^2$

Figure 3: A schematic proof on the observation that ExANs at the last ReLU layer help solve the ratios among $\{g_c^m\}_{c=1}^K$ for each m.

Remark 1 (Exact Label Inference)**.** *From (3), only if c hits the ground-truth label Y, then g_c is negative while the others are positive. This observation is also noticed by [52] independently. As a result, by checking the signs of the recovered ratios, the attacker can easily determine the ground-truth label Y of the data X. For details, please see Algorithm B.2.*

Remark 2 (Feasible Range of $\overline{g}_1$)**.** *Moreover, with the constraint that $\sum_{c \neq Y} \overline{g}_c = \sum_{c \neq Y} p_c \leq 1$, we can determine the feasible range $[0, \delta]$ of $\overline{g}_1$, where δ is a rather small constant in practice, which allows the attacker to use a random value in the range or run binary search to get satisfying results. Below, it is reasonable to assume $\overline{g_1}$ is known.*

(2) Inference of Activation Patterns: Based on the knowledge of the ExANs at the last ReLU layer, we present the following exclusivity condition under which the attacker can uniquely determine the activation pattern $(D_i^m)_{i=1}^H$ for each data sample.

Proposition 2. *Given the knowledge on the ExANs at the last ReLU layer, the attacker can determine $\{(D_i^m)_{i=1}^H\}_{m=1}^M$ with uniqueness, if each data sample X_m has at least one ExAN in D_i^m, $i \in \{1, \ldots, H-1\}$.*

Below, we provide a brief algorithmic proof. In general, the procedure of determining the activation patterns is recursively done from the last to the first ReLU layer. Initially, we have already recovered at least two ExANs in D_H^m for each input X_m. Therefore, if we consider the j-th neuron as the ExAN for X_m, then the j-th column of $\overline{G}_{H-1}$ only consists of the gradient w.r.t. X_m. Hence, by checking the non-zero positions of the j-th column, we immediately get the diagonal terms of D_{H-1}^m. Similarly, with the $(H-1)$-th layer solved, the procedure can be done for the $(H-2)$-th layer, and so on, until the first layer. Readers may refer to Fig. 2 for better intuition. Meanwhile, the attacker can further determine the whole D_H^m for each m-th sample by solving the gradient equation w.r.t. the last bias vector b_{H-1} via dynamic programming. Details on the above algorithm can be found in Algorithm B.3.

An Upper Bound on Reconstruction Errors. After the loss vectors and the activation patterns are determined, the nonlinear gradient equation collapses to a system of linear scalar equations, which can be solved with off-the-shelf linear equation solvers (e.g., LSMR [6]). When the gradient equation is reduced to a linear form, the reconstruction error is influenced by the number of scalar linear equations available to the attacker and the number of samples the attacker wants to solve. Specifically, for an attacker who solves the least-square-error solution of the linear gradient equation as an approximation to the victim's ground-truth data inputs, we derive the following error upper bound of data reconstruction.

Theorem 1 (Reconstruction Error Bound)**.** *Under the insecure boundary condition, when the sparsity of the gradient at each i-th layer satisfies* $1-\beta(\overline{G}_i) < \varepsilon_i \frac{\sqrt{d_i d_{i+1}}}{M \dim \mathcal{X}}$ *(Note:* $\beta(\overline{G}_i)$ *denotes the ratio of non-zero elements in the full gradient), then the attacker can reconstruct the labels exactly and recover the ground-truth data inputs* $\{X_m^*\}_{m=1}^M$ *within the following mean square error bound:*

$$\frac{1}{M}\sum_{m=1}^{M}\|X_m - X_m^*\|_2 < O(\sum_{i=0}^{H}\varepsilon_i(1-\beta(\overline{G}_i)))(\sum_{m=1}^{M}\|X_m^*\|_2)) \quad (5)$$

Omitted technical proofs are all provided at the following link[1]. Intuitively, Theorem 1 details the quantitative relation between the upper bound of the average reconstruction error and several key characteristics about the victim. For example, when the gradient information provided to the adversary is sparser, the batch size or the dimension of the problem space is larger, then the ε_i increases according to the inequality in the premise, which in turn makes the error bound at the RHS of (5) larger and hence causes the reconstruction quality less stable. On the contrary, when the layer width d_i, d_{i+1} are enlarged and the gradient information stays at a similar level, the ε_i decreases and therefore the attacker can expect a smaller reconstruction error bound.

Extension to Convolutional Neural Networks. When attempting to extend the above analytical results to convolutional neural networks (CNNs), we notice *the weight parameters are shared among each input dimension for a convolutional layer but not for a linear layer* would inhibit a direct extension. Although a convolutional layer is mathematically equivalent to a sparse fully-connected layer in the forward phase, the former has a rather different behavior from a sparse linear layer during the backward phase, as the gradient signals conceptually propagated to each dimension of the weight of the equivalent sparse fully-connected layer are actually accumulated to the same weight parameter in the convolutional filter. In this situation, we could neither check the non-zero/zero elements in the gradients to determine the activation state of each neuron in the feature map, nor to determine the ExANs for each data sample, which inhibits the reduction of the otherwise nonlinear gradient equation to a solvable linear equation system. Moreover, even if the reduction were possible, the number of scalar gradient equations provided by convolution filters can be highly insufficient to form a determined equation system with a satisfying solution.

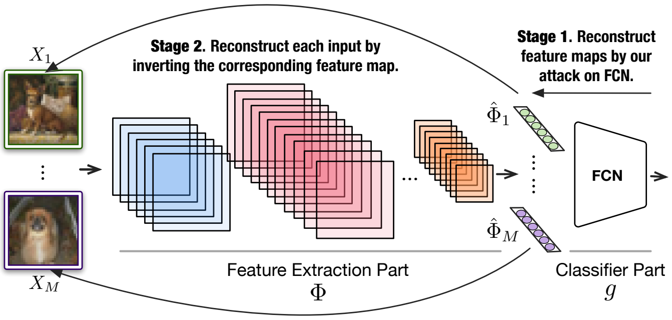

X_1
Stage 2. Reconstruct each input by inverting the corresponding feature map.
Stage 1. Reconstruct feature maps by our attack on FCN.
$\hat{\Phi}_1$
...
FCN
$\hat{\Phi}_M$
X_M
Feature Extraction Part
Φ
Classifier Part
g

Figure 4: Overview of our hybrid attack on CNN-based classification models.

In this work, we alternatively extend our proposed attack algorithm on FCN as a two-stage hybrid approach towards data reconstruction attacks on CNN-based classification models. As a mild assumption, we assume the target CNN-based classification model can be decomposed into explicitly as $f = \Phi \circ g$, where Φ is a feature extraction model mainly composed of convolutional and pooling operations, and g is an FCN for classification. This characterizes a common practice of CNN models in the real world [18, 45]. As Fig. 4 shows, our extended attack pipeline contains the following stages:

- **Stage 1.** At the first stage, we reconstruct the inputs (i.e., the feature maps) to the FCN g. Based on our obtained results on FCNs, the feature maps of each sample can be reconstructed with guaranteed reconstruction accuracy, under the condition of sufficient exclusivity, which we denote as $\{\hat{\Phi}_1, \ldots, \hat{\Phi}_M\}$.
- **Stage 2.** At the second stage, with the reconstructed feature maps, we aim to solve the input-output constraint $h(X_m) = \hat{\Phi}_m$ for each $m = 1, \ldots, M$ with gradient-based optimization algorithms. This is equivalent to an optimization problem $\arg\min_{X_m} \|\Phi(X_m) - \hat{\Phi}_m\|^2$. We name the optimization problem as the *feature matching problem* to distinguish it from the *gradient-matching problem* in (1) solved in learning-based data reconstruction attacks. In our implementation, we utilize the technique proposed in an interpretability-related work [47], which mainly models the variable X_m as the output of a trainable neural network $h(\cdot;\psi) : \mathbb{R}^d \to \mathcal{X}$ on a fixed random noise z, corresponding to the following optimization objective of our hybrid attack:

$$\arg\min_{\psi} \|\Phi(h(z_m;\psi)) - \hat{\Phi}_m\|^2, \quad (6)$$

where the optimization is conducted on the parameters of the model h.

As a final remark, we highlight the tight relation of our proposed hybrid attack on CNN with our analytic and attack techniques on FCN. On the one hand, our hybrid attack still exploits the key condition of sufficient exclusivity to separate out and reconstruct the feature map of each individual sample.

[1] https://tinyurl.com/2p8pvyra

From our perspective, how to separate the information of each single data sample which is otherwise mixed in the average gradient is critical to the feasibility of data reconstruction attacks. On the other hand, without our attack algorithm on FCNs to reconstruct the feature map for each sample to a tolerably small error, one cannot bootstrap the otherwise challenging task of data reconstruction from the average gradient to the feature matching problem.

6 Privacy Enhancement via Exclusivity Reduction

Impossibility Results under Lack of Exclusivity. First, we show the lack of exclusivity leads to the impossibility of unique reconstruction, i.e., *given a ground-truth batch of data samples $\{X_i\}_{i=1}^M$, there always exist an infinite number of artifact batches which have exactly the same gradients as the ground-truth one.*

Theorem 2 (Impossibility of Reconstruction)**.** *For an FCN $f(X) = W_H\sigma(W_{H-1}\ldots(W_1\sigma(W_0X+b_0)+b_1)\ldots+b_{H-1})+b_H$ s.t. $d_1 < d_0$ and a batch of samples $\{X_i\}_{i=1}^M$, if $N_1^m = 0$ and $M > d_1$, then there always exists a linear space $Q \subseteq \mathbb{R}^{d_0\times M}$, which satisfies: $\forall\Delta \in Q$ and $\forall i = 1,\ldots,H$,*

$$\overline{G}(\{(X_m,Y_m)\}_{m=1}^M;W_i) = \overline{G}(\{(X_m+\Delta_m,Y_m)\}_{m=1}^M;W_i) \quad (7)$$

$$\overline{G}(\{(X_m,Y_m)\}_{m=1}^M;b_i) = \overline{G}(\{(X_m+\Delta_m,Y_m)\}_{m=1}^M;b_i) \quad (8)$$

Moreover, when the input space has the interval constraints $X+\Delta \in [-1,1]^d$ (common for the image domain), the L2 norm of the largest perturbation Δ has the following lower bound,

$$\|\Delta\|_2^2 \geq \sum_{i=1}^M \|\eta_i\|_2^2 - Tr(A^\dagger AY^TY), \quad (9)$$

where $A = [\alpha_1^T,\ldots,\alpha_M^T]$, $\vec{\alpha}_m = \sum_{c=1}^K \overline{g}_c^m([W_H]_c^T D_H^m \ldots W_1 D_1^m)$, $Y = [\eta_1^T,\ldots,\eta_1^T]$, and $\eta_i = |TP_0X_i|$ (where $|X|$ takes the absolute values of entries in X), with $P_0 = (I - W_0^\dagger W_0)$ and the columns of T are the left singular vectors of W_0.

In other words, Theorem 2 indicates, without additional information, each artifact batch is indistinguishable from the ground-truth batch for the adversary in our threat model. At the left of Fig. 5, we report the empirical values of the lower bound of the largest perturbation norm, where the largest perturbation that can be added to one pixel without changing the gradient is as large as 0.5 when the width of the first layer is 10, which forms a 25% relative deviation compared with the $[-1,1]$ range of a pixel's value. At the right of Fig. 5, we further visualize a batch of size 8 when the largest perturbation is added to the ground-truth data samples while preserving the average gradient calculated on a (d_0-7-512-K) FCN. As is shown, almost each single input can be obfuscated to an unrecognizable level while the average gradient of the obfuscated batch differs from the ground-truth one by an 10^{-8} numeric error. Combining the results above, we expect the existence of the perturbation subspace with a considerable size under the lack of exclusivity will have a positive effect on inhibiting the attacker from reconstructing useful information from the average gradient only.

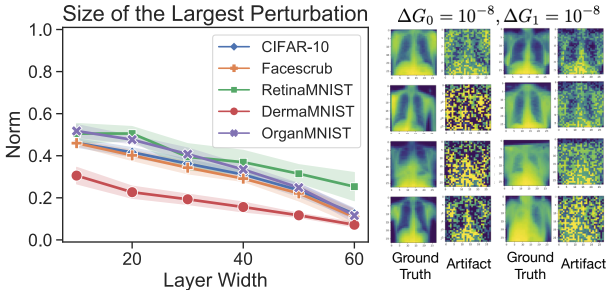


Figure 5: **Left**: The empirical values of the largest perturbation (per dim.) available in the perturbation subspace when the layer width varies, where the range of the y-axis marks the largest possible modification to an input dimension while making it stay in $[-1,1]$. **Right**: The artifact batches which share the same gradient (to a 10^{-8} numeric error) with the ground-truth batches on OrganMNIST.

Enhancing Gradient Privacy by Exclusivity Reduction. Although the above impossibility result under the lack of exclusivity poses a natural defense against data reconstruction attacks, we however notice with experiments that the situation of a batch of samples sharing the same activation pattern at the first hidden layer rarely happens. To utilize the above observation, we propose the exclusivity reduction strategy below to modify the conventional FCN architecture for ensuring the lack of exclusivity and thus the impossibility of unique reconstruction.

Corollary 1 (Exclusivity Reduction)**.** *When we remove the first ReLU layer in a conventional FCN, i.e.,*

$$W_H\sigma(W_{H-1}\ldots(W_1\hat{\sigma}(W_0X+b_0)+b_1)\ldots+b_{H-1})+b_H \quad (10)$$

where $\hat{}$ denotes the omission of the term, then, for a batch of samples $\{X_i\}_{i=1}^M$ s.t. $M > d_1$, there always exists a linear space $Q \subseteq \mathbb{R}^{M\times d_0}$ such that for each $\Delta \in Q$ and $i = 1,\ldots,H$, Theorem 2 holds.

The motivation behind is straightforward: after the first ReLU layer is removed, every sample in a batch activates all the neurons in the first layer, which naturally guarantees the lack of exclusivity. Consequently, according to Theorem 2, we can construct infinitely many artifact batches which are considerably different from the ground-truth batch in perception yet indistinguishable in terms of the gradients (Fig. 5). Further, we show in Fig. C.3 that such a modification would cause almost no performance degradation for the practical usage of FCNs.

As a final remark, exclusivity reduction is essentially different from collapsing the first two layers (e.g., $W_0 \in \mathbb{R}^{d_1 \times d_0}, W_1 \in \mathbb{R}^{d_2 \times d_1}$) into a single layer (i.e., $\tilde{W}_1 = W_2, \tilde{W}_0 = W_1 W_0 \in \mathbb{R}^{d_2 \times d_0}$), because, in the backward phase, the gradient information accessible to the attacker becomes $\nabla_{W_1 W_0} \ell(X,Y)$ after exclusivity reduction, which provides at most $d_0 \times d_2$ scalar equations to solve, instead of $\nabla_{W_0} \ell(X,Y)$, $\nabla_{W_1} \ell(X,Y)$, which brings at most $(d_0 + d_2) \times d_1$ equations to solve.

7 Evaluation Results

7.1 Overview of Evaluation

Datasets. We provide an overview on the 5 real-world datasets and the corresponding learning tasks in Table A.1. Based on considerations of research ethics, we choose public datasets to construct the data-sensitive scenarios for evaluations. As our attack requires almost no prior knowledge about the datasets, we do think the reported results would faithfully reflect the potential threats to the confidentiality of private training data in the real world. For more details on each scenario, please refer to Appendix A.

Evaluation Protocols. Following [12, 53], we first leverage the Hungarian algorithm [24] to find the best-matching pairs of reconstructed and ground-truth data inputs according to the pairwise mean square error (MSE). Then we compute the average of the following set of performance metrics over the best-matching pairs. We denote each reconstructed (ground-truth) data input as $\hat{X}_m$ (X_m).

• **Mean Square Error (MSE)** measures the L2 difference between the reconstructed input and the ground-truth input, averaged over coordinates. Formally, the MSE metric writes $\mathrm{MSE}(\hat{X}_m, X_m) = \|\hat{X}_m - X_m\|_2 / \dim \mathcal{X}$, where $\dim \mathcal{X}$ is the dimension of the input space. The MSE is the lower the better.

• **Peak Signal-to-Noise Ratio (PSNR)** measures the ratio of the effective information and noises in the reconstructed images, which is also used in [12]. It formally computes as $\mathrm{PSNR}(\hat{X}_m, X_m) = -10 \times \log_{10}(\mathrm{MSE}(\hat{X}_m, X_m))$. It is worth to notice, although PSNR is a derived metric from MSE, it behaves slightly different when being averaged and provides a better perspective on comparing the recognizability of the reconstructed input, especially for the visual scenarios.

Besides, we report the label recovery accuracy, i.e., *LAcc*, which computes the ratio between the number of the labels present in both the ground-truth and the reconstructed label sets with the ground-truth batch size. Moreover, we also visualize the reconstructed results and incorporate human evaluation to better reflect the perceptual reconstruction quality.

7.2 Attacks inside Insecure Boundary

Comparison of Reconstruction Accuracy. We compare the performance of our proposed data reconstruction attack with two previous attacks, i.e., DLG [53] and Inverting [12], on each scenario in Table A.1, where the target FCN architecture is (d-512-K) and the batch size $M = 8$. We do not involve iDLG [52] as it is only applicable to gradient calculated on a single sample (Table 1). The **FCN** rows in Table 2 compare the performance of our proposed attack with the baselines.

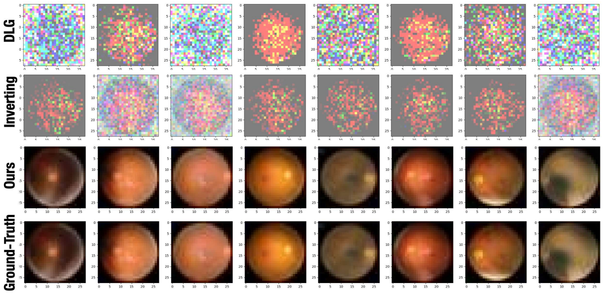


Figure 6: Sampled reconstruction results on RetinaMNIST.

As the **LAcc** columns of Table 2 show, our attack algorithm reaches 100% accuracy when reconstructing the labels of each single sample in the batch, which conforms to the theoretical guarantee in Theorem 1. In terms of the MSE and PSNR metrics, our attack algorithm substantially outperforms all the baselines in most test cases. For example, the average PSNRs of our reconstruction results are observed to be larger than 35 in most cases, which corresponds to highly recognizable reconstruction results for human observers (Fig. 6). As a comparison, previous attacks tend to produce less recognizable reconstruction results. In the following, we provide more ablation studies to validate the robustness of our proposed attack once the batch has sufficient exclusivity. Due to the space limit on the main text, we omit the full results on all the datasets only if they do not violate the observations we make. The omitted results are all presented in Appendix C.

Attacks on Partially/Fully Trained Models. We provide experiments to show the effectiveness of our attack algorithm is not limited to attacking a randomly initialized neural network, but it can also successfully attack partially/fully trained neural networks. Specifically, we train a three-layer fully connected neural network (d_0-512-K) for 100 epochs, during which the model checkpoints are stored for every 10 epochs. We conduct our attack and the best baseline *Inverting* on 10 randomly sampled insecure batches. Fig. 7(a) reports the PSNR metrics on CIFAR-10 when the training epoch proceeds from 0 (i.e., *initial stage*) to 100 (i.e., *convergence*) with a stride of 10, where the shaded region reports the 95% confidence interval. As Fig. 7(a) shows, the performance of our attack remains stable throughout the whole training process. On CIFAR-10, the MSE of the reconstruction results remain at the 10^{-4} error

Table 2: Comparisons of reconstruction attacks on different scenarios. All statistics are averaged on 10 controlled repetitive tests, with the best in **bold**.

		DLG			Inverting			Ours		
		MSE	**PSNR**	**LAcc**	**MSE**	**PSNR**	**LAcc**	**MSE**	**PSNR**	**LAcc**
FCN	**CIFAR-10**	0.503	8.75	0.475	0.296	12.50	0.775	**0.001**	**48.12**	**1.000**
	RetinaMNIST	1.102	4.48	0.500	0.993	4.97	0.513	**0.030**	**19.88**	**1.000**
	DermaMNIST	0.15	10.87	0.450	0.095	17.11	0.775	**0.005**	**41.42**	**1.000**
	OrganMNIST	0.565	7.77	0.375	0.263	12.95	0.775	**0.012**	**43.56**	**1.000**
	Facescrub	0.604	6.94	0.475	0.360	11.59	0.588	**0.002**	**35.48**	**1.000**
LeNet-5	**ImageNet**	0.496	13.46	0.375	0.213	13.26	1.000	**0.046**	**19.52**	**1.000**
	ISIC	0.438	9.68	0.375	0.086	17.31	1.000	**0.071**	**24.93**	**1.000**
	Facescrub	0.699	7.77	0.500	0.245	12.73	0.625	**0.007**	**28.88**	**1.000**
AlexNet	**ImageNet**	0.513	9.06	0.375	0.370	10.57	0.875	**0.229**	**12.79**	**1.000**
	ISIC	0.247	12.51	0.500	0.093	17.32	0.875	**0.018**	**24.90**	**1.000**
	Facescrub	0.677	7.86	0.625	0.298	11.59	0.875	**0.037**	**20.48**	**1.000**
VGG-13	**ImageNet**	0.404	10.11	0.375	0.292	11.89	1.000	**0.087**	**17.55**	**1.000**
	ISIC	0.173	14.20	0.625	0.114	16.17	1.000	**0.006**	**28.14**	**1.000**
	Facescrub	0.255	12.01	0.125	0.212	13.25	0.875	**0.007**	**29.53**	**1.000**

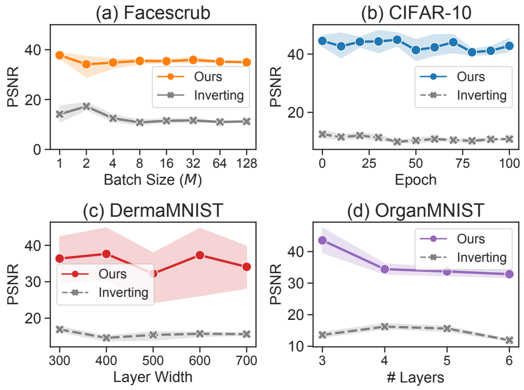


Figure 7: The PSNR curve of reconstruction attacks when **(a)** the batch size, **(b)** the training epoch, **(c)** the layer width and **(d)** the number of layers vary.

level and the PSNR remains over 40. Conforming to Theorem 1, these phenomenons further validate that our attack algorithm works independent from the attack epoch.

Scalability for Realistic Batch Sizes. To validate the scalability of our proposed attack, we alternatively leverage an auxiliary algorithm in [35, Section 3.3] (referred to as *SOW*) to arbitrarily manipulate the activation pattern of a given input by adding a slight perturbation to the input. We specify the expected activation pattern of each sample in a randomly sampled batch of realistic batch sizes to satisfy the sufficient exclusivity condition. Then, we invoke SOW to generate the perturbations, and conduct our proposed attack on the average gradient of the perturbed batches of size varying from 1 to 128 by a multiplier of 2. Fig. 7(b) reports the PSNR metrics of our proposed attack and *Inverting* on Facescrub. We repeat the experiments on 10 randomly sampled batches, where the shaded part reports the 95% confidence interval of the results. As Fig. 7(b) shows, the performance of our attack remains strong when the batch size increases from 1 to realistic batch sizes like 64 and 128 (Visualization results can be accessed following Appendix C). For example, the average PSNR of our attack is 37.8 and 35.1 when the batch size is 1 and 128 respectively on Facescrub, while the PSNR of Inverting is only 14.1 and 11.2. Besides, according to the MSE and PSNR curves, the performance of our proposed attack is almost not correlated with the size of the batch to reconstruct only if the batch stays within the insecure boundary.

Attacks on Different Architectures. To test our attack on different FCN architectures, we vary the width d_1 of the ReLU layer of a 3-layer FCN (d_0-d_1-K) from 300 to 700 with a stride of 100. The corresponding PSNR for $M = 8$ on DermaMNIST is plotted in Fig. 7(c). Fixing the layer width as 512, we also increase the depth of the target FCN (d-512-K) by inserting additional ReLU layers of the same width incrementally to obtain FCNs of 3-6 layers. we report the corresponding PSNR curves on OrganMNIST in Fig. 7(d).

From Fig. 7(c)-(d), we observe when the layer width and the number of layers increase, the performance of the learning-based reconstruction attack does not show a clear upward trend, mainly because the gradient-descent-based optimizer is likely to get stuck at a local optimum [7] when the learning process converges, which is however distant from the ground-truth results. Consequently, the corresponding PSNR metrics only loosely reflect the intrinsic relation between the model size and the attack effectiveness. Meanwhile, as the PSNR of our attack remains over 20 in most cases, the improvement of attack performance is also not clear. Nevertheless, a deeper, wider FCN architecture does facilitate data reconstruction attacks according to our analysis: On the one hand, it increases the possibility of a batch to be insecure (Section 7.4). On the other hand, it provides the adversary more scalar equations to determine the data input, which, according to Theorem 1, lowers down the upper bound on the reconstruction error (Appendix B). To alleviate the threats of data reconstruction, one may consider reduce the size of the neural networks especially when the utility requirement is already met.

Hybrid Attacks on CNN-based Classification Models. we conduct our hybrid attack on a classical shallow CNN model, i.e., LeNet-5 [22], and two state-of-the-art deep CNN models, i.e., AlexNet [23] and VGG-13 [45], with three real-world datasets, namely, ImageNet [40], ISIC skin cancer dataset [14] and Facescrub [32] (upsampled to 224 × 224). The corresponding rows of Table 2 report the quantitative performance of our proposed attack and the baseline methods when the batch size is 8. For better intuition, we also visualize the reconstructed results for VGG-13 on batches from ISIC skin cancer dataset in Fig. 8. For accessing the omitted visualization on other datasets, please refer to Appendix C.

As we can see from Table 2, our newly proposed hybrid attack on CNN-based classification models outperforms previous attacks, namely, DLG and Inverting, by a non-trivial

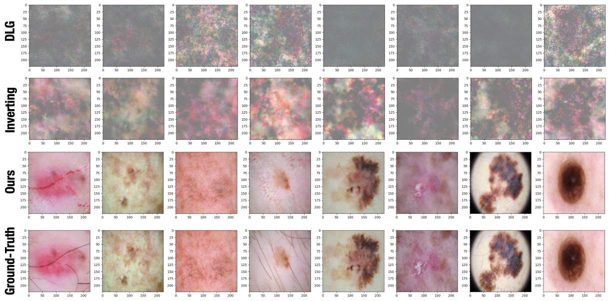


Figure 8: Sampled results on the ISIC skin cancer dataset, reconstructed from the average gradient of VGG-13.

margin. For example, when reconstructing a batch from ISIC and Facescrub, our proposed attack achieves a PSNR over 20.0 consistently on all the three representative CNN architectures (with the highest PSNR very close to 30.0), which conforms to highly recognizable reconstruction results in Fig. 8. Moreover, by leveraging our proposed attack algorithm on the FCN classifier, we reach 100% accuracy in inferring the labels of each sample in the target batch.

Human Evaluation. Finally, we measure the reconstruction quality from the perspective of human perception. Specifically, we collect one group of reconstruction results of DLG, Inverting and our attack on the same batch in 8 test cases when the batch size is 8. Then we prepare a survey composed of 24 questions, each of which shows 4 images (3 reconstruction results for the same ground-truth image and the corresponding ground-truth image in a random order) and asks the participant to rank the 4 images in a decreasing order of recognizability. The study is conducted with 71 volunteer graduate students. This whole study has been approved by our institution's IRB. The approval process is similar to the exempt review in the US, as this study is considered as "minimal risk" by IRB staffs. After collecting the completed surveys, we evaluate the performance of our attack and the baselines in terms of the average discounted cumulative gain (DCG) of the corresponding reconstruction results in each ranking results. Table 3 reports the DCG score of our attack and the baselines on different models averaged over all the participants and the datasets, alongwith the 95% confidence interval. Appendix C presents more details, with a sample question in Fig. C.1.

Table 3: Comparison of different attacks in terms of perceptual reconstruction quality in terms of discounted cumulative gain (DCG).

	DLG	Inverting	Ours	Ground-Truth
FCN	0.432 ± 0.002	0.503 ± 0.003	**0.83 ± 0.01**	0.80 ± 0.01
LeNet-5	0.448 ± 0.004	0.487 ± 0.003	0.72 ± 0.02	**0.91 ± 0.01**
AlexNet	0.436 ± 0.004	0.502 ± 0.005	0.630 ± 0.006	**0.993 ± 0.005**
VGG-13	0.442 ± 0.006	0.496 ± 0.003	0.70 ± 0.01	**0.93 ± 0.01**

As Table 3 shows, the human evaluation results are strongly consistent with the performance evaluated with the automatic metrics. For example, on CNNs, our attack always has the second largest DCG score, which is only lower than the ground-truth, for all the target architectures, which conforms to the reported performance in Table 2 and indicates the effectiveness of our proposed hybrid extension. More strikingly, on FCNs, our attack even has a higher DCG score under human evaluation compared with the ground-truth, indicating that the reconstructed results from our algorithm are more frequently ranked as the most recognizable than the ground-truth, and conforms to the over 30 PSNR of our attack on FCNs.

7.3 Protection Effect inside Secure Boundary

In this part, we provide preliminary experimental results on how our proposed exclusivity reduction strategy weakens the privacy leakage from gradients. We include differentially-private SGD (DPSGD) [2] as a potential defense based on gradient obfuscation, orthogonal to our exclusivity reduction strategy which is based on architecture modification. Besides, we further consider a hybrid defense which combines exclusivity reduction with DPSGD. Specifically, we implement the gradient obfuscation procedure of DPSGD as in [2], where the gradient clipping constant is set as 1.0 and the standard deviation σ of the Gaussian noise as 0.1, 0.5 and 1.0. In the experiments, we simulate an attacker who leverages the best baseline data reconstruction attack *Inverting* on the average gradient (w/ or w/o obfuscation) of the same batch calculated on the following comparison groups.

- **Group A.** The base FCN (d_0-512-K), i.e., *Base*;
- **Group B.** An FCN of the same architecture as in Group A except that a ReLU layer of width 7 is inserted at the first layer (d_0-7-512-K), i.e., *Compression*;
- **Group C.** An FCN which shares the same parameters with the model in Group B but has the ReLUs in the first layer removed, i.e., *Compression+w/o ExAN*;
- **Group D.** An FCN of the same architecture as in Group B and the gradient is obfuscated with DPSGD, i.e., *Compression+DPSGD* ($\sigma = 0.1, 0.5, 1.0$);
- **Group E.** An FCN of the same architecture as in Group C and the gradient is obfuscated with DPSGD, i.e., *Compression+DPSGD+w/o ExAN* ($\sigma = 0.1, 0.5, 1.0$),

where we choose the width of the non-ReLU layer as 7 because this setting is expected to enhance the privacy of a batch with its size $M \geq 8$ $(= 7 + 1)$ according to Corollary 1, which is also a common setting on the maximal size of a batch under attack in previous attacks. For all the five comparison groups, we repetitively conduct the attack on 100 randomly sampled batches, and collect the average MSE and PSNR as indicators of the reconstruction quality. Fig. 9 presents the box-plots of the performance metrics on RetinaMNIST. The omitted results on other datasets are in Appendix C.

First, comparing the PSNR on Group A & B in Fig. 9, we

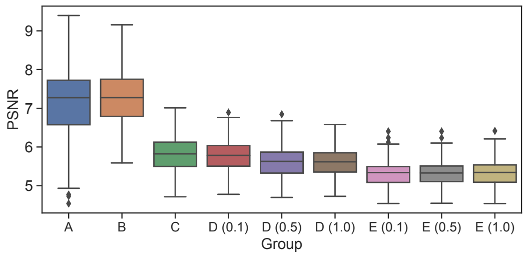


Figure 9: The reconstruction quality of *Inverting* on RetinaMNIST when applied on 5 comparison groups with different architectures or implemented with different defense strategies.

observe the Inverting attack on RetinaMNIST has almost the same performance whether a 7-unit ReLU layer is inserted into the original model, which indicates the model compression only has a very slight effect in weakening the reconstruction quality. As a comparison, our proposed exclusivity reduction strategy substantially decreases the PSNR of the reconstruction: The PSNR for Group C is 20% lower than the PSNR of Group A & B. The results imply that exclusivity reduction does play a non-trivial role in weakening the effectiveness of data reconstruction when the compression effect of the shallow layer of width 7 is left out.

Next, comparing the attack performance on Group C and D, we observe that the attack effectiveness of Inverting is weakened on both groups, which supports that the mitigation strategies via architecture modification or via gradient obfuscation can both alleviate the information leakage from the gradient. Meanwhile, by comparing the decrease in PSNR, we observe that the DPSGD provides as a slightly more effective defense than exclusivity reduction, for which we infer the reason is DPSGD works by directly obfuscating the gradient, the immediate information source exploited by data reconstruction, while our strategy works by reducing the neuron exclusivity, a more in-depth factor which guarantees the non-uniqueness of reconstruction. The orthogonality of these two approaches further inspires us to evaluate a more effective defense which combines our strategy for eliminating the insecure exclusivity state and DPSGD for gradient obfuscation. As the reported performance on Group E shows, this new combination exhibits a larger decrease on the reconstruction quality, while, with regression tests, we observe almost no further trade-off on the normal utility.

7.4 Impact Factors on Exclusivity States

Finally, we empirically study how the layer width, the network depth, the training epoch and the label composition in a batch would influence the statistics of batches which satisfy the sufficient exclusivity condition (i.e., *insecure batch*). Generally, we set the base FCN architecture as a three-layer FCN (d_0-512-K), vary the architecture as specified by the experimental purpose, test the validity of the sufficient exclusivity condition for 1000 randomly sampled batches of size 8, and report the proportion of the insecure batches in Fig. 10. Specifically, the model configurations in Fig. 10(a)-(c) are the same as the ones in Fig. 7, while, in Fig. 10(d), we report the proportion of valid batches which consist of 8 samples from the same class, which are averaged over all the classes during the training process, to measure the impact of label composition in a training batch on its exclusivity state.

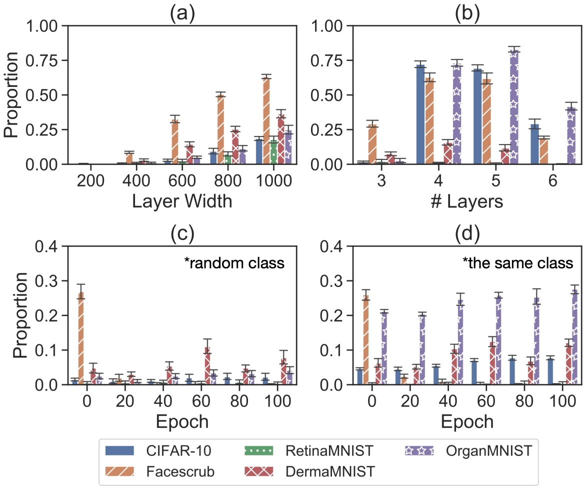


Figure 10: The proportion of insecure batches in 1000 randomly sampled batches under different configurations.

As Fig. 10(a) shows, on all the five datasets, the width plays a strong impact factor on the proportion of insecure batches. For example, when the layer width is 1000, the proportion of insecure batches is over 60% on RetinaMNIST, which, in other words, indicates that over 60% batches of size 8 can be reconstructed with high recognizability only if the average gradient is leaked in this case. From Fig. 10(b), we observe that the influence of the network depth on the neuron exclusivity state is complicated. In most cases, the proportion of the insecure batches reaches the maximal when the network depth is 4 and 5 but radically decreases when the depth is further enlarged. This may serve as an explanation on our previously reported results in Fig. 7(d), where the baseline attacks do not show a clear upward trend when the depth increases. In Fig. 10(c), we do not observe a common principle which characterizes the influence of the training epoch on the neuron exclusivity. For example, on Facescrub, the proportion of insecure batches decreases when the training epoch accumulates, while, on DermaMNIST, the proportion first increases and then remains stable. From Fig. 10(d), we observe that, on some datasets, the proportion of insecure batches is even higher compared to the case when the batches contain randomly sampled inputs, which conforms to the observed complexity of activation patterns even for samples from the same class [15], considering the exponentially many

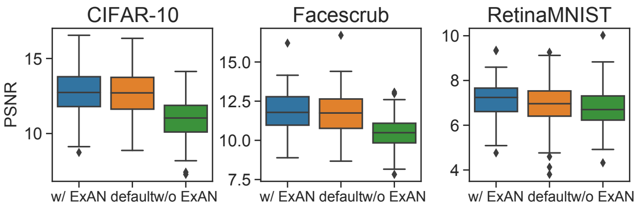


Figure 11: Impact of *sufficient exclusivity* (i.e., *w/ ExAN*) and *lack of exclusivity* (i.e., *w/o ExAN*) conditions on the quality of data reconstruction, where the *default* columns collect the results on a randomly sampled batches with no control on its exclusivity state.

possibilities (e.g., 2^{512}).

8 Discussions

On the Remaining Exclusivity States. We further explore how the neuron exclusivity state of a batch would influence the attack effectiveness in general. Specifically, with the aid of the SOW algorithm [35], we prepare the following three comparison groups of batches to attack: **A.** 100 randomly sampled batch of size 8 from the original dataset (i.e., *default*); **B.** slightly perturbed versions of batches in Group A such that each batch satisfies the condition of sufficient exclusivity (i.e., *w/ ExAN*); **C.** slightly perturbed versions of batches in Group A such that all the samples in the same batch have no ExAN with one another. Then we conduct the *Inverting* attack on all the 100 batches from the three comparison groups respectively. Fig. 11 presents the box plots of the PSNR of *Inverting* on three datasets, with the omitted results in Appendix C. As Fig. 11 shows, in most cases, we observe that the attack performance decreases in the following order of the comparison groups: *w/ ExAN* $>$ *default* $>$ *w/o ExAN*. For example, on Facescrub, the average PSNR is respectively 11.83, 11.55, and 10.47 for the *w/ ExAN*, the *default* and the *w/o ExAN* comparison group. Although the performance margin of *Inverting* between the default group and the *w/ ExAN* group is not as substantial as that between the default group and the w/o ExAN group, we should notice *Inverting* does not exhibit the optimal attack effectiveness on insecure batches. In fact, our constructed attack algorithm indicates, the attacker can achieve a much higher reconstruction quality (e.g., with PSNR over 30) on the insecure batches. Combining these results, we expect ExAN as a promising metric for understanding and measuring the data leakage from the gradient. Yet, there lacks rigorous statements whether an effective data reconstruction attack can be constructed for the remaining cases as in Section 5, or whether the impossibility of unique reconstruction can be proved as in Section 6, which is left as an open question for future research.

Data Reconstruction vs. Model Extraction. As pointed out in [17], model extraction becomes less feasible when the model is expansive (i.e., the model contains a layer with a higher output dimension than the input dimension), while, under the same condition, data reconstruction attack in turn becomes stronger, according to our analysis. It is mainly because, the information exposed to model extraction attacks is a number of data inputs (i.e., queries) and their predictions, from which he/she wants to recover the model parameters. Therefore, model extraction on an expansive model has to recover more unknown variables than either the input dimension or the prediction dimension, which becomes an issue. In contrast, the information exposed to data reconstruction attacks is the gradient, the information of which grows when the model becomes larger. When the model is expansive, the gradient information accessible to the adversary is sufficiently more than the dimension of the unknown inputs the adversary wants to solve, which further facilitates data reconstruction.

Limitation & Future Directions. To further improve the reconstruction accuracy of our analytic attack, future works may consider design a solution refinement procedure based on nonconvex optimization techniques [17], e.g., by using the solved solution from the linear equation solver as the initial guess and then refining the solution iteratively by gradient descents on the gradient matching objective. Besides, our current work mainly characterizes the defensive effectiveness of exclusivity reduction with the theory of linear equation systems. As a promising future work, one may consider extend the analytic results to the language of differential privacy. Finally, future works may also study the role of neuron exclusivity states in other gradient-based privacy attack classes. For example, our proposed exclusivity reduction may also weaken the effectiveness of gradient-based property inference attacks [29], because the gradient information after exclusivity reduction can also correspond to many other mini-batches which do not share the same global property with the original mini-batch, which can therefore obfuscate the attacker's inference.

9 Conclusion

In this paper, we provide the first analytic study which explores the security boundary of data reconstruction from gradient via the lens of neuron exclusivity states. Specifically, we determine and prove the boundary condition of insecure exclusivity states by constructing an attack algorithm with guaranteed accuracy. Moreover, we prove the impossibility of unique reconstruction for the exclusivity states satisfying the lack of exclusivity condition. With our proposed simple yet effective exclusivity reduction strategy as a preliminary step, we hope our study would arouse more research interests and efforts in investigating and strengthening the privacy properties of model gradient via its intrinsic interaction with the underlying mechanism of deep learning.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments and input to improve our paper. We would like to thank the anonymous reviewers for their insightful comments that helped improve the quality of the paper. This work was supported in part by the National Key Research and Development Program (2021YFB3101200), National Natural Science Foundation of China (61972099, U1736208, U1836210, U1836213, 62172104, 62172105, 61902374, 62102093, 62102091), Natural Science Foundation of Shanghai (19ZR1404800). Min Yang is a faculty of Shanghai Institute of Intelligent Electronics & Systems, Shanghai Institute for Advanced Communication and Data Science, and Engineering Research Center of Cyber Security Auditing and Monitoring, Ministry of Education, China. Mi Zhang and Min Yang are the corresponding authors.

References

[1] M. Abadi, P. Barham, J. Chen *et al.*, "Tensorflow: A system for large-scale machine learning," *OSDI*, 2016.

[2] M. Abadi, A. Chu *et al.*, "Deep learning with differential privacy," *CCS*, 2016.

[3] S. Bubeck, "Convex optimization: Algorithms and complexity," *Foundations and Trends in Machine Learning*, 2015.

[4] N. Carlini, C. Liu, Ú. Erlingsson *et al.*, "The secret sharer: Evaluating and testing unintended memorization in neural networks," *USENIX Security*, 2019.

[5] C. Chen, B. Wu *et al.*, "Nebula: A scalable privacy-preserving machine learning system in ant financial," *CIKM*, 2020.

[6] F. Chin-Lung and SaundersMichael, "LSMR: An iterative algorithm for sparse least-squares problems," *SIAM Journal on Scientific Computing*, 2011.

[7] Y. Dauphin, R. Pascanu *et al.*, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," *NIPS*, 2014.

[8] V. Duddu, D. Samanta, D. V. Rao *et al.*, "Stealing neural networks via timing side channels," *ArXiv*, vol. abs/1812.11720, 2018.

[9] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," *CCS*, 2015.

[10] M. Fredrikson, E. Lantz, S. Jha *et al.*, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," *USENIX Security*, 2014.

[11] K. Ganju, Q. Wang, W. Yang *et al.*, "Property inference attacks on fully connected neural networks using permutation invariant representations," *CCS*, 2018.

[12] J. Geiping, H. Bauermeister, H. Dröge *et al.*, "Inverting gradients - how easy is it to break privacy in federated learning?" *NeurIPS*, 2020.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[14] D. Gutman, N. C. F. Codella, M. E. Celebi *et al.*, "Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging," *International Symposium on Biomedical Imaging*, 2018.

[15] B. Hanin and D. Rolnick, "Complexity of linear regions in deep networks," *ICML*, 2019.

[16] B. Hitaj, G. Ateniese, and F. Pérez-Cruz, "Deep models under the gan: Information leakage from collaborative deep learning," *CCS*, 2017.

[17] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin *et al.*, "High accuracy and high fidelity extraction of neural networks," *USENIX Security*, 2020.

[18] Y. Ji, X. Zhang, S. Ji *et al.*, "Model-reuse attacks on deep learning systems," *CCS*, 2018.

[19] P. Kairouz, H. B. McMahan *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, 2021.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ArXiv*, vol. abs/1412.6980, 2015.

[21] J. Konecný, H. B. McMahan, F. X. Yu *et al.*, "Federated learning: Strategies for improving communication efficiency," *ArXiv*, vol. abs/1610.05492, 2016.

[22] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Master's thesis, University of Toronto*, 2009.

[23] A. Krizhevsky, I. Sutskever *et al.*, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, 2012.

[24] H. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, 1955.

[25] T. Laurent and J. von Brecht, "The multilinear structure of ReLU networks," *ICML*, 2018.

[26] K. Leino and M. Fredrikson, "Stolen memories: Leveraging model memorization for calibrated white-box membership inference," *USENIX Security*, 2020.

[27] D. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, 1989.

[28] L. Lyu, H. Yu *et al.*, *Threats to Federated Learning*. Springer International Publishing, 2020.

[29] L. Melis, C. Song, E. D. Cristofaro *et al.*, "Exploiting unintended feature leakage in collaborative learning," *S&P*, 2019.

[30] G. Montúfar, R. Pascanu *et al.*, "On the number of linear regions of deep neural networks," *ArXiv*, vol. abs/1402.1869, 2014.

[31] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," *S&P*, 2019.

[32] H. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," *ICIP*, 2014.

[33] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," *CVPR*, 2019.

[34] X. Pan, M. Zhang, S. Ji, and M. Yang, "Privacy risks of general-purpose language models," *S&P*, 2020.

[35] X. Pan, M. Zhang, Y. Lu, and M. Yang, "TAFA: A task-agnostic fingerprinting algorithm for neural networks," *ESORICS*, 2021.

[36] A. Paszke, S. Gross, F. Massa *et al.*, "PyTorch: An imperative style, high-performance deep learning library," *NIPS*, 2019.

[37] H. Robbins, "A stochastic approximation method," *Annals of Mathematical Statistics*, 2007.

[38] D. Rolnick and K. P. Kording, "Reverse-engineering deep ReLU networks," *ICML*, 2020.

[39] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, 1986.

[40] O. Russakovsky, J. Deng, H. Su *et al.*, "ImageNet large scale visual recognition challenge," *IJCV*, 2015.

[41] A. Salem, A. Bhattacharyya, M. Backes *et al.*, "Updates-leak: Data set inference and reconstruction attacks in online learning," *ArXiv*, vol. abs/1904.01067, 2019.

[42] A. Salem, Y. Zhang, M. Humbert *et al.*, "ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models," *NDSS*, 2019.

[43] R. Shokri, M. Stronati, C. Song *et al.*, "Membership inference attacks against machine learning models," *S&P*, 2017.

[44] R. Shokri, M. Strobel, and Y. Zick, "Privacy risks of explaining machine learning models," *ArXiv*, vol. abs/1907.00164, 2019.

[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ArXiv*, vol. abs/1409.1556, 2015.

[46] F. Tramèr, F. Zhang, A. Juels *et al.*, "Stealing machine learning models via prediction apis," *USENIX Security*, 2016.

[47] D. Ulyanov, A. Vedaldi *et al.*, "Deep image prior," *CVPR*, 2018.

[48] B. Wang and N. Gong, "Stealing hyperparameters in machine learning," *S&P*, 2018.

[49] Z. Wang, M. Song, Z. Zhang *et al.*, "Beyond inferring class representatives: User-level privacy leakage from federated learning," *ICCC*, 2019.

[50] J. Yang, R. Shi, and B. Ni, "MedMNIST classification decathlon: A lightweight AutoML benchmark for medical image analysis," *ArXiv*, vol. abs/2010.14925, 2020.

[51] Q. Yang, Y. Liu *et al.*, "Federated machine learning: Concept and applications," *TIST*, 2019.

[52] B. Zhao, K. R. Mopuri, and H. Bilen, "iDLG: Improved deep leakage from gradients," *ArXiv*, vol. abs/2001.02610, 2020.

[53] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *NeurIPS*, 2019.

A Details of Scenarios

Table A.1: Scenarios covered in experiments.

Dataset	Task	Input Size (d_0)	# Classes (K)
CIFAR-10 [22]	Object Classification	$3 \times 32 \times 32$	10
FaceScrub [32]	Face Recognition	$3 \times 32 \times 32$	20
RetinaMNIST [50]	Iris Diagnosis	$3 \times 28 \times 28$	5
DermaMNIST [50]	Dermatology	$3 \times 28 \times 28$	7
OrganMNIST [50]	Pathology	$1 \times 28 \times 28$	11
ImageNet [23]	Object Classification	$3 \times 224 \times 224$	1000
ISIC [14]	Skin Cancer Diagnosis	$3 \times 224 \times 224$	7

Table A.1 summarizes the general information of the datasets we cover in our experiments. In the following, we provide more details.

Academic Benchmarks. We choose the standard benchmark image datasets, i.e., CIFAR-10 [22] and ImageNet [23], which are considered in previous data reconstruction attacks. These two datasets originate from the machine learning community and are widely used as computer vision benchmarks for image classification and many other tasks. These two datasets mainly cover daily objects and show incremental complexity in various aspects (e.g., total pixels, color channels, class number).

Medical Scenarios. We consider three real-world medical imaging datasets made public by [50], namely, RetinaMNIST, DermaMNIST, OrganMNIST. These three datasets corresponds to the tasks of intelligent diagnosis of iris-related, skin-related and organ-related pathology. We choose these three datasets out of the 9 datasets from [50] based on its diversity in color channels and image variance. Besides, we use the ISIC skin cancer dataset [14], which consists of more high-resolution skin cancer images for evaluating our hybrid attack on deep CNNs.

Identity-Related Scenario. We consider a face recognition system built with a subset of the Facescrub dataset [32], which consists of portraits of 20 celebrities randomly selected from the full dataset.

B Algorithm Details

In this part, we provide the algorithmic descriptions of the key procedures in our proposed data reconstruction attacks on FCNs in Algorithm B.1, B.2 & B.3.

Algorithm B.1 Determine $\{(\overline{g}_c^m)_{c=1}^K\}_{m=1}^M$.

```
Input: The gradient of $W_H$, i.e., $\overline{G}_H$.
Output: Reconstructed labels $\{Y_1, \ldots, Y_M\}$ and loss vectors $\{(\overline{g}_c^m)_{c=1}^K\}_{m=1}^M$.
Compute $r_c := [\overline{G}_H]_c / [\overline{G}_H]_1$ for every $c$ in $1, \ldots, K$.
Find all the disjoint index groups $\{I^m\}_{m=1}^M$ where $(r_2)_j$ is constant whenever $j \in I^m$.    ▷ $M$ is hence the inferred batch size and $I^m$ is the index set of the exclusively activated neurons at the last ReLU layer.
for all $c$ in $1, \ldots, K$ do
    for all $m$ in $1, \ldots, M$ do
        Select an arbitrary index $j$ from $I^m$.
        $\overline{g}_c^m / \overline{g}_1^m \leftarrow [r_c]_j$.
    end for
end for
for all $m$ in $1, \ldots, M$ do
    $Y_m \leftarrow$ Apply Algorithm B.2 to $(\overline{g}_c^m)_{c=1}^K$.
    Estimate the upper bound of feasible range of $\overline{g}_1^m$ as $\delta_m \leftarrow \overline{g}_1^m / \overline{g}_{Y_m}^m$
    Fix $\overline{g}_1^m = 2 \times \delta_m / 3$.    ▷ This is practiced in all our experiments.
    Calculate each $\overline{g}_c^m$ according to the ratio.
end for
```

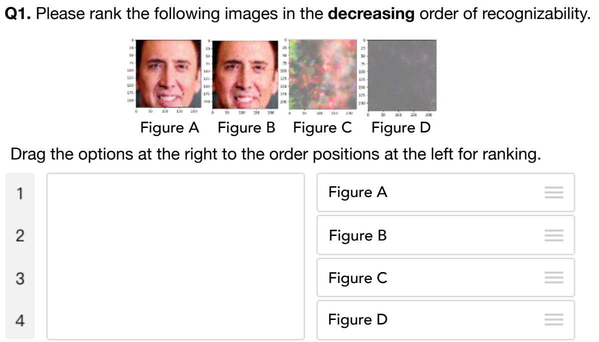
Algorithm B.2 Exact label reconstruction from the loss vector.

```
Input: The loss vector for the $m$-th sample $(\overline{g}_c^m)_{c=1}^K$.
Output: Reconstructed label $Y_m$.
if $(\overline{g}_c^m)_{c=1}^K$ have one negative element then
return $Y_m \leftarrow$ The index of the negative element
else
return $Y_m \leftarrow 1$
end if
```

Algorithm B.3 Determine activation patterns $\{(D_i^m)_{i=1}^H\}_{m=1}^M$.

```
Input: The gradients $(\overline{G}_i)_{i=0}^H$ at each layer, the index sets $(I_H^m)_{m=1}^M$ of exclusively activated neurons at the last ReLU layer and the reconstructed $\{(\overline{g}_c^m)_{c=1}^K\}_{m=1}^M$
Output: Reconstructed activation patterns. $\{(D_i^m)_{i=1}^H\}_{m=1}^M$.
$I_{\text{cur}} \leftarrow \{I_H^m\}_{m=1}^M$.
for all $i$ in $H-1, \ldots, 1$ do
    for all $m$ in $1, \ldots, M$ do
        Select an arbitrary index $j$ from $I_{\text{cur}}^m$.
        $\text{diag}(D_i^m) \leftarrow ([\overline{G}_i]_{:,j} \neq 0)$
    end for
    Construct the index sets $\{I_i^m\}_{m=1}^M$ of exclusively activated neurons at the $i$-th layer from $\{D_i^m\}_{m=1}^M$.
    $I_{\text{cur}} \leftarrow \{I_i^m\}_{m=1}^M$.
end for
Solve $D_H^m$ from the binary equation $\frac{1}{M}\sum_{m=1}^M \sum_{c=1}^K \overline{g}_c^m [W_H]_c^T D_H^m I_{d_H} = \frac{\partial \ell}{\partial b_{H-1}}$.
```

C More Evaluation Results



Q1. Please rank the following images in the **decreasing** order of recognizability.

Figure A Figure B Figure C Figure D

Drag the options at the right to the order positions at the left for ranking.

1 Figure A
2 Figure B
3 Figure C
4 Figure D

Figure C.1: A sample question from our survey for human evaluation of reconstruction quality.

Omitted Results on Other Datasets. In Fig. C.5, we present the omitted results accompanying Fig. 7 in the main text. In Fig. C.4, we present the omitted results accompanying Fig. 9

in the main text. In Fig. C.2, we present the omitted results accompanying Fig. 11 in the main text.

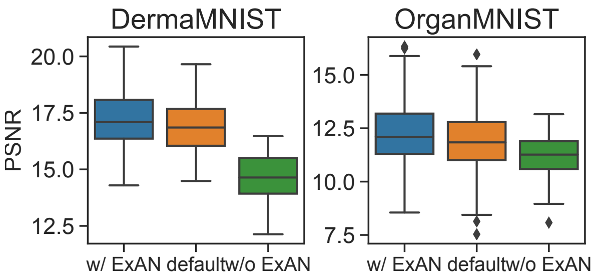


Figure C.2: Omitted results on other datasets for Fig. 11.

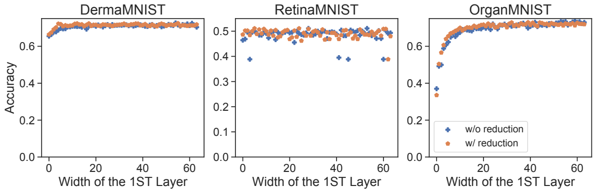


Figure C.3: The accuracy of three-layer FCNs on the main task of DermaMNIST, RetinaMNIST and OrganMNIST, with and without exclusivity reduction. The width of the first ReLU layer varies in $[1,64]$.

Omitted Visualization Results. We present the omitted results accompanying Fig. 6 & 8 and other visualization results in the following link: `https://tinyurl.com/2p8pvyra`.

More Details on Human Evaluation. First, we collect a group of reconstruction results of DLG, Inverting and our attack on the same batch for each of the following 8 test cases, namely, FCN on Facescrub & CIFAR-10, LeNet-5 on Facescrub & CIFAR-10, AlexNet on ImageNet & Facescrub, and VGG-13 on ImageNet & Facescrub, where the batch size is always set as 8. We do not include the reconstruction results on ISIC skin cancer dataset because the images may be inappropriate for all of our participants to view. With the collected reconstruction results, we prepare a survey composed of 24 questions in the same format. As shown in Fig. C.1, each question shows 4 images in a line (i.e., 3 reconstruction results for the same ground-truth image and the corresponding ground-truth image, positioned in a random order). For each question, the participants are required to rank the 4 images in a decreasing order of recognizability.

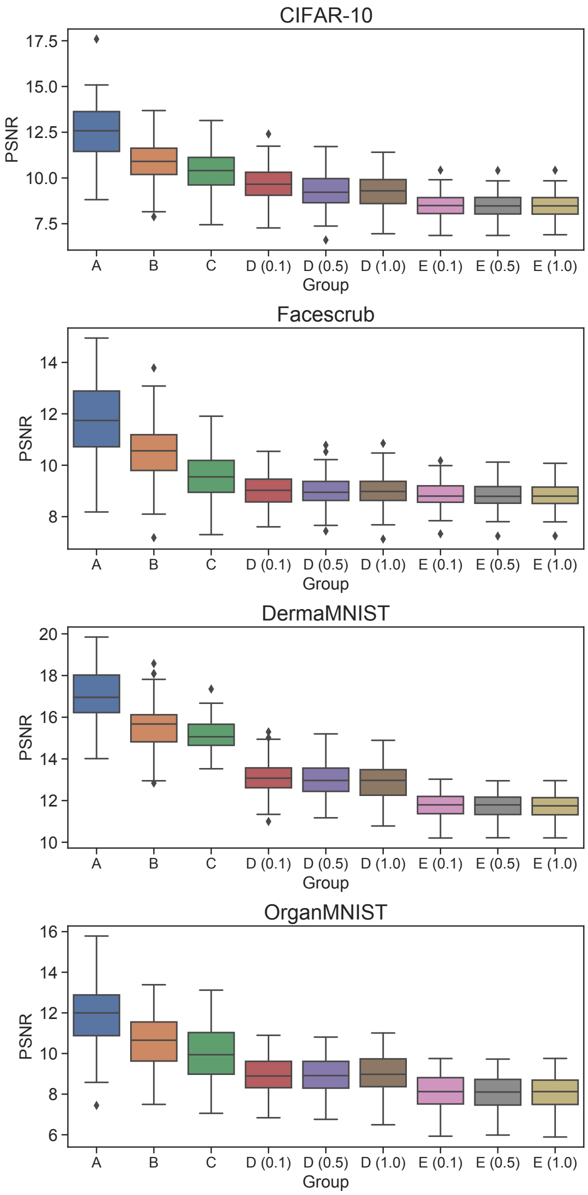


Figure C.4: Omitted results on other datasets for Fig. 9.

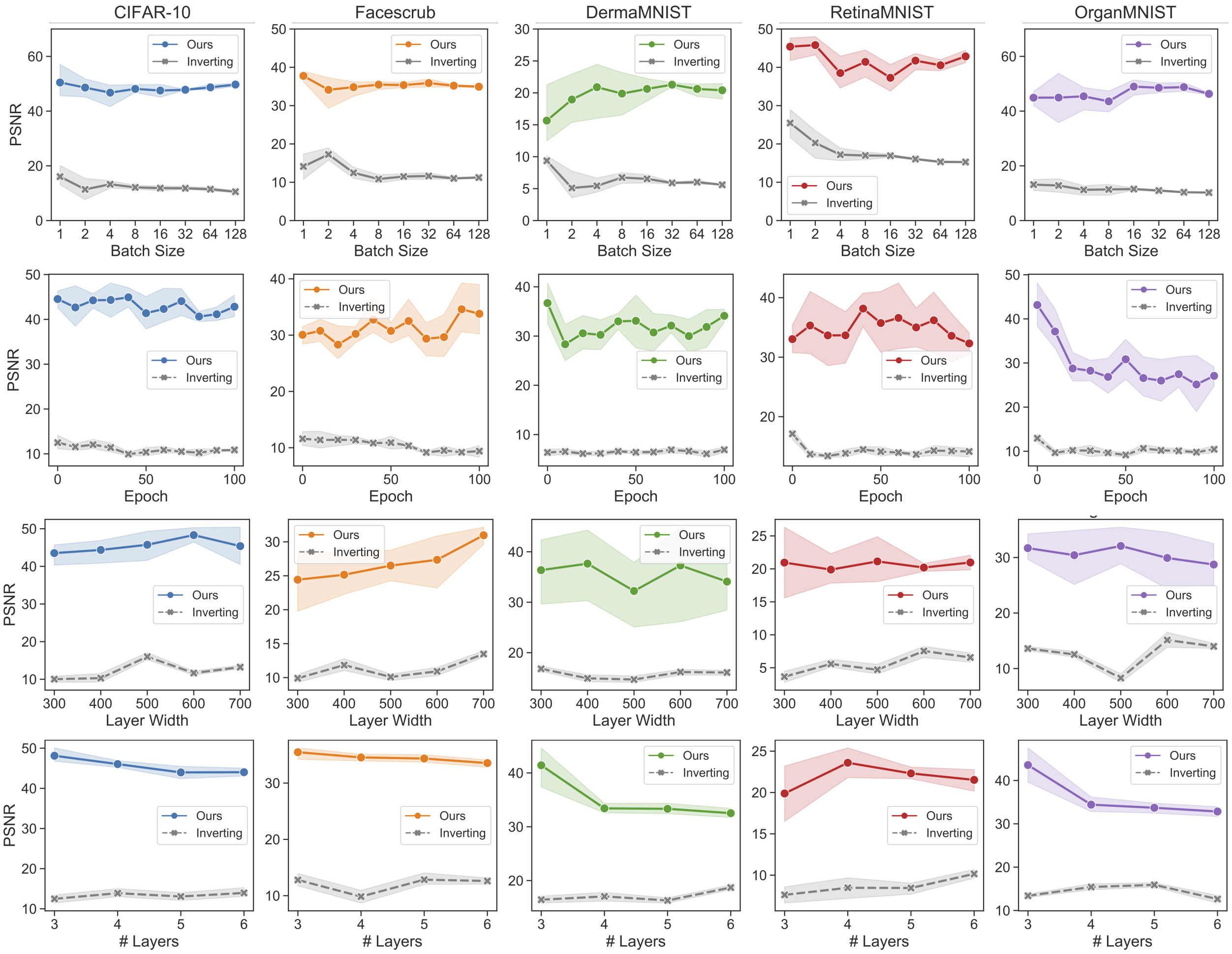


Figure C.5: Omitted results on other datasets for Fig. 7.