Reconstructing Training Data with Informed Adversaries

Borja Balle*
DeepMind

Giovanni Cherubin*†
Microsoft Research

Jamie Hayes*
DeepMind

2022 IEEE Symposium on Security and Privacy (SP) | 978-1-6654-1316-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/SP46214.2022.9833677

***Abstract*—Given access to a machine learning model, can an adversary reconstruct the model's training data? This work studies this question from the lens of a powerful informed adversary who knows all the training data points except one. By instantiating concrete attacks, we show it is feasible to reconstruct the remaining data point in this stringent threat model. For convex models (e.g. logistic regression), reconstruction attacks are simple and can be derived in closed-form. For more general models (e.g. neural networks), we propose an attack strategy based on training a reconstructor network that receives as input the weights of the model under attack and produces as output the target data point. We demonstrate the effectiveness of our attack on image classifiers trained on MNIST and CIFAR-10, and systematically investigate which factors of standard machine learning pipelines affect reconstruction success. Finally, we theoretically investigate what amount of differential privacy suffices to mitigate reconstruction attacks by informed adversaries. Our work provides an effective reconstruction attack that model developers can use to assess memorization of individual points in general settings beyond those considered in previous works (e.g. generative language models or access to training gradients); it shows that standard models have the capacity to store enough information to enable high-fidelity reconstruction of training data points; and it demonstrates that differential privacy can successfully mitigate such attacks in a parameter regime where utility degradation is minimal.**

***Index Terms*—machine learning, neural networks, reconstruction attacks, differential privacy**

I. Introduction

Machine learning (ML) models have the capacity to memorize their training data [1], and such memorization is sometimes unavoidable while training highly accurate models [2, 3, 4]. When the training data is sensitive, sharing models that exhibit memorization can lead to privacy breaches. To design mitigations enabling privacy-preserving deployment of ML models we must understand how these breaches arise and how much information they leak about individual data points.

Membership leakage is considered the gold standard for privacy in ML, both from the point of view of empirical privacy evaluation (e.g., via membership inference attacks (MIA) [5]) as well as mitigation (e.g., differential privacy (DP) [6]). Membership information represents a minimal level of leakage: it allows an adversary to infer a single bit determining if a given data record was present in the training dataset. Models trained on health data represent a prototypical application where membership can be considered sensitive: the presence of an individual's record in a dataset might itself be indicative of whether they were tested or treated for a medical condition.

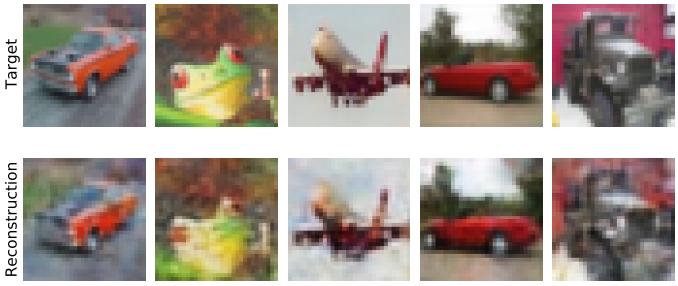


Fig. 1: Examples of training data points reconstructed from a 55K parameter CNN classifier trained on CIFAR-10.

Reconstruction of training data from ML models sits at the other extreme of the individual privacy leakage spectrum: a successful attack enables an adversary to reconstruct all the information about an individual record that a model might have seen during training. The possibility of extracting training data from models can pose a serious privacy risk even in applications where membership information is not directly sensitive. For example, reconstruction of individual images from a model trained on pictures that were privately shared in a social network can be undesirable even if that individual's membership in the social network is public information.

Existing evidence of the feasibility of reconstruction attacks is sparse and focuses on specialized use cases. For example, recent work on generative language models highlights their capacity to memorize and regurgitate some of their training data [7, 8], while works on gradient inversion show that adversaries with access to model gradients (e.g. in federated learning (FL) [9]) can use this information to reconstruct training examples [10]. Similarly, attribute inference attacks reconstruct a restricted subset of attributes of a training data point given the rest of its attributes [11], while property inference attacks infer global information about the training distribution rather than individual points [12, 13].

Our work proposes a general approach to study the feasibility of reconstruction attacks against ML models without assumptions on the type of model or access to intermediate gradients, and initiates a study of mitigation strategies capable of preventing this kind of attacks. The starting point is the instantiation of an *informed adversary* that, knowing all the records in a training data set except one, attempts to recon-

*Equal contribution
†Work done while at the Alan Turing Institute

DOI 10.1109/SP46214.2022.00127

struct the unknown record after obtaining white-box access to a released model. This choice of adversary is inspired by the (implicit) threat model in DP [14].

Working with such a powerful, albeit unrealistic, adversary enables us to demonstrate the feasibility of reconstruction, both in theory against convex models as well as experimentally against standard neural network architectures for image classification. Furthermore, the use of an informed adversary makes our work relevant for provable mitigations: effective defenses against optimal informed adversaries will also protect against attacks run by less powerful and more realistic adversaries.

A. Overview of Contributions and Paper Outline

We start by introducing and motivating the informed adversary threat model (Section II). Our first contribution is a theoretical analysis of reconstruction attacks against simple ML models like linear, logistic, and ridge regression (Section III). We show that for a broad class of generalized convex linear models, access to the maximum likelihood solution enables an informed adversary to recover the target point exactly.

In the convex setting, the attack relies on solving a simple system of equations. Extending reconstruction attacks to neural networks requires a different approach due to the inherent non-convexity of the learning problem. In Section IV, we propose a generic approach to reconstruction attacks based on *reconstructor networks* (RecoNN): networks that are trained by the adversary to output a reconstruction of the target point when given as input the parameters of a released model.

Our second contribution is to show that it is feasible to attack standard neural network classifiers using reconstructor networks; we present effective RecoNN architectures and training procedures, and show they can extract high-fidelity training images from classifiers trained on MNIST[1] and CIFAR-10. Figure 1 provides an illustration of reconstructions produced by a RecoNN-based attack against a convolutional neural network (CNN) classifier trained on CIFAR-10. These experiments provide compelling evidence that image classification models can store in their weights enough information to reconstruct individual training data points.

Section VI describes our third contribution: an in-depth analysis around what factors affect the success of our RecoNN-based attack. These include hyper-parameter settings in the model training pipeline, degree of access to model parameters, and quality and quantity of side knowledge available to the adversary. We also explore how different levels of knowledge about the internal randomness of stochastic gradient descent (SGD) affect reconstruction; we observe that knowing the model's initialization significantly improves the quality of reconstructions, while knowing the randomness used for mini-batch sampling is not necessary for good reconstruction.

As part of our experiments, we also investigate the use of DP-SGD [15] as a mitigation to protect against reconstruction attacks. We find that large values of ϵ suffice to defend against our best RecoNN-based attacks – in fact, values that are much larger than what is necessary to protect against membership inference attacks by informed adversaries [14]. Section VII supports this observation by introducing a definition of *reconstruction robustness*, analyzing its relation to the (Rényi) DP parameters of the training algorithm, and showing that, under mild conditions on the adversary's side knowledge, $\epsilon = o(d)$ suffices to prevent reconstruction of d-dimensional data records.

[1]A minimal implementation of our reconstruction attack on MNIST is available at https://github.com/deepmind/informed_adversary_mnist_reconstruction.

TABLE I: Summary of notation

Model Developer		Reconstruction Adversary	
$\mathcal{Z}$	Data domain	$D_{\text{-}}$	Training dataset minus target point
Θ	Model domain	z	Target point
D	Training dataset	R	Reconstruction algorithm
n	size of training set (includes target point)	aux	Side knowledge about z
A	Training algorithm	$\hat{z}$	Candidate reconstruction
θ	Released model	ℓ	Reconstruction error

II. Reconstruction with Informed Adversaries

We start by instantiating and justifying the *informed adversary* threat model for reconstruction attacks against ML models, and by comparing it to related attacks in the literature. Notation for the most important concepts introduced in this section is summarized in Table I. At its core, our threat model assumes a powerful adversary with white-box access to a model released by a *model developer*. The developer owns a dataset $D \in \mathcal{Z}^n$ of n training records from some domain $\mathcal{Z}$, and a (possibly randomized) training algorithm $A : \mathcal{Z}^n \to \Theta$. They train (the parameters of) a model $\theta = A(D)$, and then release it as part of a system or service. For example, records in D may be feature-label pairs in standard supervised learning settings, and A may implement an optimization algorithm (e.g. SGD or Adam) for a loss function associated with D and Θ.

A. Threat Model

A reconstruction adversary with access to the released model aims to infer enough information about its training data to reconstruct one of the examples in D. In this paper, we consider a powerful adversary who already has full knowledge about all but one of the training points. Formally, they have access to the following information to carry out the attack.

Definition 1 (Informed reconstruction adversary)**.** *Let θ be a model trained on dataset D of size n using algorithm A. Let $z \in D$ be an arbitrary training data point and $D_{\text{-}} = D \setminus \{z\}$ denote the remaining $n-1$ points; we refer to z as the* target point*. An* informed reconstruction adversary *has access to:*

a) The fixed dataset $D_{\text{-}}$*;*
b) The released model*'s parameters θ;*
c) The model's training algorithm A;
d) (Optional) Side knowledge aux *about the target point.*

We first discuss each piece of knowledge we give to our attacker, and then analyze in depth how our adversary relates to other threat models arising in other privacy attacks.

a) Fixed dataset: Arguably, the assumption that gives our attacker the greatest advantage is knowing all the training data except for the target point. There are two main reasons to consider such a stringent threat model. First, since our ultimate goal for studying ML vulnerabilities is to design effective mitigations, by evaluating the resilience of ML models in this strong threat model we ensure their resilience against weaker (and more realistic) attackers. Second, our setup captures the implicit threat model used in the DP definition; indeed, DP bounds the ability of a mechanism at preventing the disclosure of membership information about one data record from an adversary who knows all the other records in the database.

b) White-box model access: White-box access to the model is motivated by several real-world scenarios. First, the practice of publishing models online (e.g. to facilitate their use or favor public scrutiny) is increasingly widespread. Second, proprietary models shipped as part of hardware or software components can be vulnerable to reverse-engineering; it would be naive to assume that sufficiently motivated adversaries will never obtain white-box access to such models. Finally, FL settings may give real-world attackers access to similar information to the one we capture in our threat model.

c) Training algorithm: Privacy (and security) through obscurity is generally regarded as a bad practice. Thus, we assume the adversary has access to the model developer's training algorithm A, including any associated hyper-parameters (e.g. learning rate, regularization, batch size, number of iterations, etc). Access to A can be in the form of a concrete (e.g. open source) implementation. Nevertheless, black-box access (e.g. through a SaaS API) suffices for the general reconstruction attack presented in Section IV. In cases where A is randomized, we will evaluate attacks with and without knowledge of the different sources of randomness used when training the released model. In stochastic optimization algorithms these typically include model initialization and mini-batch sampling. Knowledge of A's internal randomness could come from the model developer using a hard-coded random seed in a public implementation. Alternatively, knowledge about the model's initialization will also be available whenever the released model is obtained by fine-tuning a publicly available model (e.g. in transfer learning scenarios), or in FL settings where the adversary has successfully compromised an intermediate model by taking part in the training protocol.

d) Side knowledge about target point: Privacy attacks do not happen in a vacuum, so adversaries will often have some prior information about the target point before observing the released model. For starters, knowledge of $D_{\text{-}}$ and A provides the adversary with syntactic and semantic context for a learning task in which the model developer deemed it useful to include the target point. In our investigations, we often consider adversaries with additional side knowledge abstractly represented by aux. From a practical perspective, the attack presented in Section IV takes aux to be a dataset $\bar{D}$ of points disjoint from $D_{\text{-}}$. For example, these could come from a public academic dataset or from scraping relevant websites. Our experiments in Section VI-B show that these additional points do not necessarily need to come from the same distribution as the training data. In our theoretical investigation (Section VII), we model the adversary's side knowledge as a probabilistic prior π from which the target is assumed to be sampled.

B. Reconstruction Attack Protocol and Error Metric

Algorithm 1 formalizes the interaction between model developer and reconstruction adversary in our threat model. After the model θ is trained on $D = D_{\text{-}} \cup \{z\}$, the adversary runs their attack algorithm R using all the information discussed in the previous section, and produces a *candidate* reconstruction $\hat{z}$ for the target point z. The protocol returns a measure of the attack's success based on a *reconstruction error* function ℓ; smaller error means the reconstruction is more faithful.

Algorithm 1 Reconstruction attack with an informed adversary. (Auxiliary side knowledge aux is optional).

procedure RECONSTRUCTION($A, R, D_{\text{-}}, z; \mathsf{aux}$)
  $\theta \leftarrow A(D_{\text{-}} \cup \{z\})$
  $\hat{z} \leftarrow R(\theta, D_{\text{-}}, A; \mathsf{aux})$
  return $\ell(z, \hat{z})$

Privacy expectations are contextual, and depend on the information content and modality of the sensitive data. Perfect reconstruction may not be necessary for the user to claim their privacy has been violated; e.g., a privacy breach may occur if the image of a car's license plate is revealed via an attack, even if the reconstructed background is inaccurate. In particular, the error function ℓ can encode not only proximity between the feature representations of the target and candidate points, but also the correctness with which an attack can recover a (private) property of interest about the target. Our experiments on image classifiers use the MSE between pixels as a measure of reconstruction, as well as the similarity between outputs of machine learning models on z and $\hat{z}$ (through the LPIPS and KL metrics cf. Section V-B). In general, an appropriate choice of ℓ and a threshold for declaring successful reconstruction is a policy question that will depend on the particular application: it should capture the minimum level of leakage that would cause a significant harm to the involved individual.

C. Relation to Attribute Inference

Reconstruction can be seen as a generalization of attribute inference attacks (AIA) [11, 16, 17, 18], also sometimes referred to as model inversion attacks. In AIA, an attacker that knows part of a data record z aims to reconstruct the entire record by exploiting (white-box or black-box) access to a model θ whose training dataset contained z. It is also common for the attack goal of a model inversion attack to try and reveal training data information in aggregate, possibly isolated to a specific target label. Although no individual training records are reconstructed through this attack, privacy can be leaked if aggregated training information with respect to a target label is sensitive (e.g. facial recognition where each label is associated with an identity). The standard threat model in AIA does not include an informed adversary, but we can get a more

direct comparison with our model by considering an *informed* AIA adversary. Such an adversary is identical to Definition 1 but also receives as input partial information about the target point z, which we denote by $\eta(z)$. This can be incorporated in Definition 1 via the side knowledge aux, showing that informed AIA corresponds to reconstruction in our model with a particular type of side knowledge. We conclude that any investigation into mitigating general reconstruction attacks in our threat model will also be useful in protecting against informed AIA, and, by extension, standard AIA.

D. Relation to Membership Inference

In membership inference attacks (MIA) [5, 17, 19, 20], an attacker with access to a released model θ and a *challenge example* $z \in \mathcal{Z}$ guesses if z was part of the model's training data. Like in AIA, standard MIA does not assume an informed adversary. Introducing an *informed* MIA adversary yields a model matching the adversary in the threat model behind DP [14]. This adversary is identical Definition 1, with the exception that it also receives two candidates $z_0, z_1 \in \mathcal{Z}$ for the additional data point that was used for training the model, and the developer decides which one to use uniformly at random. The corresponding interaction protocol between model developer and adversary is summarized in Algorithm 2, where the adversary uses a MIA algorithm M and the result provides a bit representing whether it guessed correctly.

Algorithm 2 Informed Membership Inference Attack

procedure INFORMED-MIA(A, M, D_{-}, z_0, z_1)
  $b \leftarrow \mathrm{Unif}(\{0,1\})$
  $\theta \leftarrow A(D_{-} \cup \{z_b\})$
  $\hat{b} \leftarrow M(\theta, D_{-}, A, z_0, z_1)$
  return $b = \hat{b}$

We remark that this attacker is much more powerful than the one in standard MIA. In particular, if the model's training algorithm A is deterministic, then there is a trivial strategy: the attacker trains models on $D_{-} \cup \{z_0\}$ and $D_{-} \cup \{z_1\}$ and checks which of the two matches the released model θ. This is coherent with the observation that randomized algorithms are necessary to (non-trivially) provide DP. Note also that accurate reconstruction provides an informed MIA. Indeed, assume, for example, that ℓ satisfies the triangle inequality and reconstruction succeeds at achieving error less than $\ell(z_0, z_1)/2$. Then the reconstruction adversary uses θ to obtain a candidate $\hat{z}$, and then guess z_0 if $\ell(\hat{z}, z_0) < \ell(\hat{z}, z_1)$ and z_1 otherwise.

The contrapositive implication of the above is that if this powerful notion of MIA is not possible, then accurate reconstruction is also not possible. Furthermore, the existence of a standard MIA attacker implies the existence of an informed one. This argument indicates that protecting against informed MIA will protect against both standard MIA and accurate reconstruction, thus motivating the use of DP – a mitigation against informed MIA – as a strong privacy protection. The experiments in Section VI and the theoretical investigation developed in Section VII will, however, illustrate that values of the DP parameter ϵ that are too large to protect against informed MIA can still protect against accurate reconstruction.

E. Further Related Work

Attacks for reconstructing training data have been studied in the context of generative language models (LM). Carlini et al. [7] proposed a *targeted* black-box reconstruction attack where the adversary knows part of a training example (i.e. a text prompt) and infers the rest (e.g. a credit card number). Their attack assumes partial knowledge of the target record (as with AIA) and a threat model where the adversary has significant computational power but no additional knowledge of the training data. An *untargeted* version of this attack was later performed against GPT-2 [21] by repeatedly sampling from the model and comparing the samples with the training data [8]. Both works crucially exploit the generative aspect of LMs to carry out reconstruction; our attacks are more general and require no such assumptions, making them suitable to attack standard image classification models.

Many works have investigated what an attacker can infer from inspecting the intermediate gradients in FL settings or multiple model snapshots during training [22, 23, 24, 25, 26]. These attacks focus on inferring training points, their labels, or related properties. The task our reconstruction adversary has to solve is harder: whilst a gradient leakage adversary has access to information involving only a mini-batch of training points, our attacks needs to invert the entire training procedure.

Finally, *property inference attacks* (PIA) are a generalization of AIA where the adversary infers properties about the training set [12, 13]. These attacks are effective at recovering overall statistics (e.g. the percentage of training records coming from a minority group, the average value of a feature across the data) but in general do not compromise the privacy of individuals.

III. Reconstruction in Convex Settings

In this section, we focus on attacking convex supervised learning models. We discuss a general reconstruction attack strategy against a broad family of convex models when the empirical risk minimization (ERM) problem has a unique minimum and is solved to optimality. Specifically, we show there exists a closed form solution to perform reconstruction attacks against Generalized Linear Models (GLMs) without any additional side knowledge about the target point. This attack applies to popular models such as linear regression, ridge regression, and logistic regression.

A. Reconstruction Strategy for Convex Models

Consider an ML model θ trained by exactly solving the ERM problem. Formally, let $C(\hat{\theta}) = \sum_{z \in D} c(z, \hat{\theta})$ be a risk function for some loss c, and let $\theta \in \mathrm{argmin}_{\hat{\theta} \in \Theta} C(\hat{\theta})$. If the loss is strictly convex, this optimization admits a unique global minimum. Further, if the loss is differentiable and there is no constraint on the parameters (i.e. $\Theta = \mathbb{R}^{d'}$), then the optimum is characterized by the system of equations $\nabla C(\theta) = 0$.

This simplified scenario enables a direct strategy to perform a reconstruction attack. Recall the adversary has white-box

access to the released model θ and knowledge of the fixed dataset $D_{\text{-}}$. This allows them to write the following system of equations which will be satisfied by the target point z:

$$\nabla_\theta c(z,\theta) = -\sum_{z' \in D_{\text{-}}} \nabla_\theta c(z',\theta) \ . \tag{1}$$

Since in supervised training every point $z = (x, y)$ is represented by a feature vector $x \in \mathbb{R}^d$ and a label $y \in \mathbb{R}$, this provides d' equations from which the adversary wants to recover $d+1$ unknowns (d features plus the label). Note that this strategy is independent of the algorithm that was used for training the model as long as the model was trained to optimality. Next we show a closed-form solution for this attack exists in the case of GLMs fitted with an intercept term.

B. Closed-Form Reconstruction Against GLMs

Consider fitting a GLM derived from a canonical exponential family with canonical link function g (see, e.g. [27]). The GLM parameters are trained via (regularized) ERM by minimizing the maximum likelihood objective $C(\hat{\theta}) = -\sum_{(x,y)\in D} \Big(b(\langle x,\hat{\theta}\rangle) - \langle x,\hat{\theta}\rangle y\Big) + \lambda\|\hat{\theta}\|^2$, where b is a function satisfying $b' = g^{-1}$, and $\lambda \geq 0$ is a regularization parameter. For example, g^{-1} is the identify function for linear regression and the sigmoid function for logistic regression. This optimization admits a unique minimum when either $\lambda > 0$, b is strictly concave (as in the examples above) or the data is in general position [28]. In any of these cases (1) connects the unknown $z = (x, y)$ with θ and $D_{\text{-}}$. Assuming the model is trained with an intercept parameter (i.e. the first coordinate of each feature vector is equal to 1) this results in a system of d equations with d unknowns. The following solution for this system gives an effective reconstruction attack.

Theorem 1 (Reconstruction attack against GLMs)**.** *Let θ be the unique optimum of $C(\hat{\theta})$ and $D_{\text{-}}$ the training data set except for one point $z = (x, y)$. Suppose $\bar{X} \in \mathbb{R}^{(n-1)\times d}$ contains as rows the features of all points in $D_{\text{-}}$ where its first column satisfies $\bar{X}_1 = \vec{1}$, and similarly for the labels $\bar{Y} \in \mathbb{R}^{n-1}$. Then taking $B = g^{-1}(\bar{X}\theta) - \bar{Y}$ we get:*

$$x = \frac{\bar{X}^\top B + \lambda\theta}{\bar{X}_1^\top B + \lambda\theta_1} \ , \qquad y = g^{-1}(\langle x,\theta\rangle) + \lambda \bar{X}_1^\top B\theta_1 \ .$$

We defer all proofs to the appendix. Two important takeaways from this result are: 1) an informed adversary needs no additional side knowledge about z to effectively attack a GLM trained with intercept; and, 2) whether the model overfits the data or generalizes well plays no role in the attack's success.

IV. A General Reconstruction Attack

We describe a reconstruction attack against general ML models. Intuitively, our attack stems from the observation that the influence of the target point z on the released model θ is similar to the influence an alternative point $\bar{z}$ would have on the model $\bar{\theta} = A(D_{\text{-}} \cup \{\bar{z}\})$. By repeatedly training models on different points, our attack collects enough information about the mapping from training points to model parameters to invert it at the model of interest θ. We give a high-level introduction to our attack strategy using *reconstructor networks* (RecoNN).

A. General Attack Strategy

Let us use the shorthand notation $A_{D_{\text{-}}} : \mathcal{Z} \to \Theta$ with $A_{D_{\text{-}}}(z) = A(D_{\text{-}} \cup \{z\})$ to emphasize that, from the point of view of an informed adversary, when $D_{\text{-}}$ is fixed A effectively becomes a mapping from target points to model parameters. An ideal reconstruction attack would invert the training procedure and output $\hat{z} = A^{-1}_{D_{\text{-}}}(\theta)$; whenever A is easy to invert, this will produce a perfect reconstruction as in the setting analyzed in Section III. In general, however, the training process is not (easily) invertible, due to the non-convexity of the optimization problem solved by A, or to the presence of randomness in the training process. In such settings, our general reconstruction attack relies on *approximately* solving this inverse problem by producing a function $\phi : \Theta \to \mathcal{Z}$ that associates model weights to a guess for the target point in a similar way to the (ideal) inverse mapping $A^{-1}_{D_{\text{-}}}$. Note that the adversary in this threat model is extremely powerful; for example, they could enumerate (a fine discretization of) $\mathcal{Z}$ and pick the candidate $\hat{z}$ that produces the model $\hat{\theta} = A_{D_{\text{-}}}(\hat{z})$ closest to θ. However, for high-dimensional data this enumerative approach is infeasible, so we focus on attacks that can be executed in practice.

In this paper, we instantiate the search for ϕ as a learning problem, effectively using "neural networks to attack neural networks". To solve this learning problem, we first design a RecoNN architecture for neural networks whose inputs lie in the parameter space Θ of the released model and outputs lie in the domain $\mathcal{Z}$ of the training data; typically we can encode both using numerical vectors. The adversary then uses its knowledge of $D_{\text{-}}$ and A, together with side knowledge in the form of *shadow target* points $\bar{D}$ disjoint from $D_{\text{-}}$, to generate a collection of *shadow models*. These shadow model and target pairs comprise the training data for the RecoNN, which is then applied to the released model to obtain a candidate reconstruction $\hat{z}$ for the (previously unseen) target point z.

B. Training Reconstructor Networks

Consider an informed adversary in our threat model (Definition 1). As side knowledge about z, we assume the attacker has k additional shadow targets $\bar{D} = \{\bar{z}_1, \ldots, \bar{z}_k\}$ from $\mathcal{Z}$. Ideally, if we think that the attack's success will depend on the RecoNN's ability to exhibit statistical generalization, these points would be sampled from the same distribution as the target point z. Nonetheless, we will see in our experimental evaluation that this requirement is not strictly necessary to achieve good reconstructions (Section VI-B). The general reconstruction attack proceeds as follows (see also Figure 2):

1) For $i = 1, \ldots, k$, train model $\bar{\theta}_i = A_{D_{\text{-}}}(\bar{z}_i)$ on the fixed dataset plus the ith shadow target from the adversary's side knowledge pool $\bar{D}$. Together, we refer to the collection of shadow model-target pairs $S = \{(\bar{\theta}_i, \bar{z}_i)\}_{i=1}^k$ as the *attack training data*.
2) Train a RecoNN ϕ using S as examples of successful reconstrutions. Abusing our notation, we use R to denote the training algorithm used by the adversary: $\phi = R(S)$.
3) Obtain a reconstruction candidate by applying the RecoNN to the target model: $\hat{z} = \phi(\theta)$.

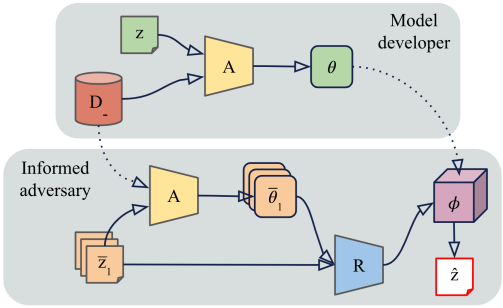


Fig. 2: Overview of RecoNN-based attack.

In all our experiments, we consider classification tasks where $z = (x, y) \in \mathcal{X} \times \mathcal{Y}$ with $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y}$ is a finite set of labels. We also make the simplifying assumption that y can be inferred from x, and focus only on reconstructing x.

Related work: The idea of using "neural networks to attack neural networks" has been used in the literature to implement a number of attacks, including (black-box and white-box) membership inference [5, 19, 20], model inversion [18], and property inference [12, 13]. Our use of RecoNNs is related to [12], where an invariant representation of a released neural network parameters is fed into another neural network to perform a PIA, although the output of our attack is often a high-dimensional object (e.g. an image) instead of single scalar. In preliminary experiments we did not see an improvement from using this invariant representation as a pre-processing step; standard normalization was sufficient for a successful attack. Similarly, the use of shadow models trained by the adversary to imitate the behavior of the released model is a common approach in MIA and AIA, although most works do not consider an informed adversary with knowledge of $D_{\text{-}}$. Despite the attack being an instantiation of the shadow model technique, it is not a foregone conclusion that this approach will work for reconstruction attacks. Reconstruction is a more difficult task than membership inference, and it entails a considerable amount of engineering, data curation, and ML training insight to carry out, as we will discuss.

V. Experimental Setup

We discuss the default experimental settings, and how we will evaluate reconstruction attacks.

A. Default Settings

We evaluate our reconstruction attacks on the MNIST and CIFAR-10 datasets using fully connected (i.e. multi-layer perceptron) and convolutional neural networks (CNN) as the released (and shadow) models. Our experiments investigate the influence that training hyperparameters for A have on the effectiveness of reconstruction. Default model architectures and hyperparameters for both released and reconstructor models are summarized in Table IV. Most of these choices are standard and were selected based on preliminary experiments. In the following we highlight the most important details.

a) Dataset splits: We split each dataset into three disjoint parts: fixed dataset ($D_{\text{-}}$), shadow dataset ($\bar{D}$), and test targets dataset; the latter contains $1K$ points, both for MNIST and CIFAR-10. We train one released model per test target and report average performance of our attack across test targets.

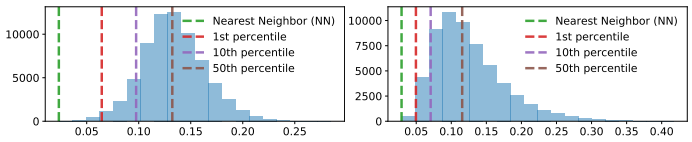
b) Released model training: The training algorithm for released and shadow models is standard gradient descent with momentum. By default, we use full batches (i.e. no mini-batch sampling) to keep the algorithm deterministic. Additionally, by default we assume the adversary knows the model initialization step, so both released and shadow models are trained from the same starting point. We explore the effect of mini-batching and random initialization separately in Section VI-B.

The architecture is an MLP for MNIST and a CNN for CIFAR-10. On average, the released models achieve over 94% accuracy on MNIST and 40% on CIFAR-10 without significant overfitting (generalization gap is less 1% on MNIST and 5% on CIFAR-10). The reason for the subpar performance on CIFAR-10 is partially[2] because the models are trained with only 10% of the data used in standard evaluations – this constraint comes from the need to reserve a large disjoint set of shadow points to train RecoNN. We experiment with a larger CIFAR-10 fixed set size ($50K$) in Section VI-B; in this setting the released models achieve $\sim 50\%$ test accuracy.

We expect reconstructing CIFAR-10 targets will be a more challenging task than MNIST. CIFAR-10 images have a richer, more complex structure, and so capturing and reconstructing the intricacies of such an image may be difficult. Additionally, the underlying released model is larger; hence: 1) a larger reconstructor network is required, which comes with higher computational costs for the adversary; 2) the shadow dataset may need to be larger, to facilitate learning on high dimensional data (i.e. on the shadow models' weights).

c) Reconstructor network training: When training the reconstructor, shadow model parameters across layers are flattened and concatenated together. We also re-scale each coordinate in this representation to zero mean and unit variance; we found this pre-processing step to be important, as some of the parameters can be extremely small. For MNIST, we use a mean absolute error (MAE) + mean squared error (MSE) loss between shadow targets and reconstructor outputs as the training objective. For CIFAR-10 we modify the reconstructor training objective by adding an LPIPS loss [29] and a GAN-like Discriminator loss to improve visual quality of reconstructed images. We use a patch-based Discriminator [30] with the architecture given in Table VII, and train it using mean squared error loss [31] and a learning rate of 10^{-5}. The patch-based discriminator aims to distinguish shadow targets from reconstructor generated candidates. At a high-level, we can view the reconstructor network as a generative model with a latent space defined over a distribution of shadow models; this enables us to apply ideas from Generative Adversarial Networks (GANs) training. Our discriminator training set-up

[2]Training without random mini-batches, no regularization and a small CNN architecture also contribute to this effect.



(a) MNIST. $|\bar{D} \cup D_{\text{-}}| = 69K$. (b) CIFAR-10. $|\bar{D} \cup D_{\text{-}}| = 59K$.

Fig. 3: For each test target compute the MSE to all points in adversary's available pool of data ($\bar{D} \cup D_{\text{-}}$) in the default setting. Plot the histogram of MSEs (averaged over all test targets) along with some highlighted order statistics.

is as in [30] – we alternate between one gradient descent step on the discriminator, and one step on the reconstructor network. From visual inspection, we found using a discriminator improves sharpness of CIFAR-10 reconstructed images, even if it does not strictly improve the MSE metric.

B. Criteria for Attack Success

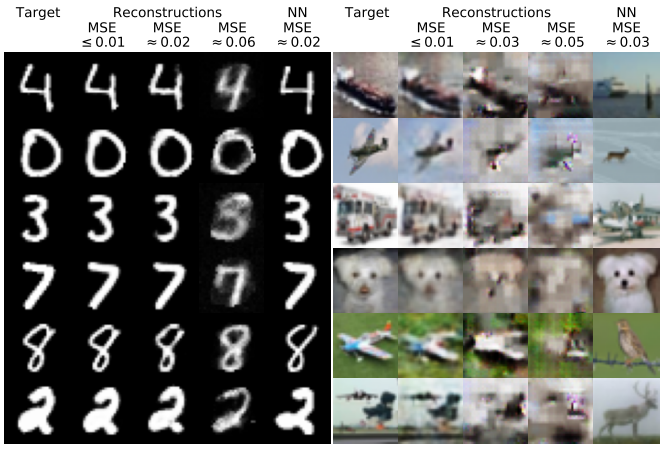
In our experiments, we use several evaluation metrics ℓ to capture various aspects of information leakage from reconstruction attacks. When reporting an average metric we measure performance of a single reconstructor network on $1K$ released model and target point pairs.

a) Mean squared error (MSE): We report the MSE between a target and its reconstruction. In the context of images, while discovery of private information does not necessarily perfectly coincide with a decreasing MSE between the original and reconstructed training point, in general the two are correlated (Section VI-A).

b) LPIPS: We report the LPIPS metric [29] as it has been shown to be closer to the human's visual systems determination of image similarity in comparison to the MSE distance. LPIPS is measured by comparing deep feature representations from visual models trained with similarity judgements made by human annotators.

c) KL: After running the attack, a real-world adversary may need to post-process the reconstructed image; e.g. if they wanted to extract a license plate from the reconstructed image, they may need to run a downstream image classifier. We therefore include a similarity metric between the outputs of a highly accurate classifier on the target and reconstructed image based on the Kullback–Leibler (KL) divergence between predicted class probabilities. For MNIST, we use a LeNet classifier [32] achieving 99.4% test accuracy, and for CIFAR-10 use a Wide ResNet [33] achieving 94.7% test accuracy.

d) Nearest Neighbor Oracle: To contextualize MSE reconstruction metrics we consider an oracle that exploits all the data available to the adversary in the default setting and guesses the point $\hat{z} \in D_{\text{-}} \cup \bar{D}$ that has the smallest MSE distance to z. The MSE distance between z and its nearest neighbor $\hat{z}$ serves as a conservative threshold for successful reconstruction: although faithful reconstructions with larger MSE are certainly possible, falling below the threshold means the reconstruction is closer to the target than to any other point previously available to the adversary, so the attack must have



(a) MNIST (b) CIFAR-10

Fig. 4: Visualization of reconstructions for six random targets selected from the test set. The first column shows the targets, the second shows the default reconstruction attack, the third shows reconstructions around the same MSE provided by the NN oracle. The fourth column corresponds to reconstructions with distance approximately equal to the 1st percentile (Figure 3). The last column shows the NN oracle.

extracted unique information about the target point from the released model. Figure 3 provides average histograms (over 1K test targets) of MSEs between a target point and all points in $D_{\text{-}} \cup \bar{D}$. The green line corresponds to the average MSE to the nearest neighbor across all test targets (0.0232 on MNIST and 0.0291 on CIFAR-10); if reconstructions have a smaller MSE than this distance we will judge the target to have been successfully reconstructed. For reference, we also highlight the 1st, 10th and 50th percentile MSEs, which will be helpful to contextualize experiments throughout Section VI.

VI. Empirical Studies in Reconstruction

We now conduct extensive experiments investigating how the released model architecture and its training hyperparameters impact reconstruction quality. We first demonstrate the feasibility of the reconstruction attack against models trained on our default experimental setup. Then we discuss an in-depth study on which factors, such as training set size or released model's hyperparameters, affect the success of reconstruction. Finally, we investigate DP as a mitigation against reconstruction attacks. Our findings are summarized in Table II.

A. Feasibility of Reconstruction Attacks

We first carry out the general reconstruction attack under the default experimental settings (cf. Section V).

Figure 4 shows examples of targets and respective reconstructions; we use the nearest neighbor (NN) oracle as a baseline. We observe a good overall reconstruction quality on both datasets. Running the attack against 1K test targets, we observe an average reconstruction MSE of 0.0089 (MNIST) and 0.0049 (CIFAR-10). These numbers, compared to the NN oracle baselines, demonstrate our attack is effective. To

TABLE II: Effect of different factors on the success of reconstruction attacks.

Factor	Description	MNIST		CIFAR-10	
		MSE	Success	MSE	Success
—	Nearest neighbor (NN) oracle	0.0232	–	0.0291	–
—	Default hyper-parameters and architectures (cf. Section V)	0.0089	✓	0.0049	✓
Fixed set size	Change size of fixed set to: $1K$ (MNIST) $50K$ (CIFAR-10 + shadows from CIFAR-100)	0.0094	✓	0.0039	✓
Size & architecture	Larger MLP (MNIST) and CNN (CIFAR-10)	0.0079	✓	0.0047	✓
Released layers	Restrict attack to use subset of released model layers	0.0124	✓	0.0257	✓
Epochs	Increase number of released model training epochs: 250 (MNIST) 200 (CIFAR-10)	0.0121	✓	0.0094	✓
Activation	Change released model activations to ReLU	0.0182	✓	0.0324	✗
Learning rate	Decrease released model learning rate: 0.01 (MNIST) 0.001 (CIFAR-10)	0.0049	✓	0.0055	✓
Random initialization	Adversary does not know initial released model parameters	0.0695	✗	0.0931	✗
Model access	Only allow logit-based black-box access to released model	0.0110	✓	0.0198	✓

account for the variance across experimental runs (e.g. different random selections of fixed sets across experiments), we repeated this experimental procedure ten times with differing fixed sets, initial released model parameters, and evaluation sets. We saw minimal variance in results; importantly, reconstructions were consistently better than the NN oracle.

To help the reader calibrate MSE values to reconstruction quality, Figure 4 shows poor reconstructions with MSE close to the oracle NN's MSE (third column) and to its 1st percentile (fourth column); these reconstructions were obtained in preliminary experiments with weaker RecoNN instances.

Relation between reconstruction metrics: With the same experimental setup as above, we also evaluate results across our other metrics (Section V-B) on MNIST. We observe that MSE and LPIPS are strongly correlated (Figure 5a). Figure 5b also shows that a small MSE implies a small KL but the converse is not true; in other words, it is possible for two images that are not identical to have similar predictions. Since these metrics exhibit significant correlations, we only report a subset of them in subsequent experiments, focusing mostly on comparing the MSE metric with the NN oracle. We observe similar trends on CIFAR-10, although MSE vs LPIPS correlation is weaker; this partially motivated including the LPIPS loss when training RecoNNs on CIFAR-10.

B. What Factors Affect Reconstruction

We study which factors may improve or impact reconstruction success; these are summarized in Table II.

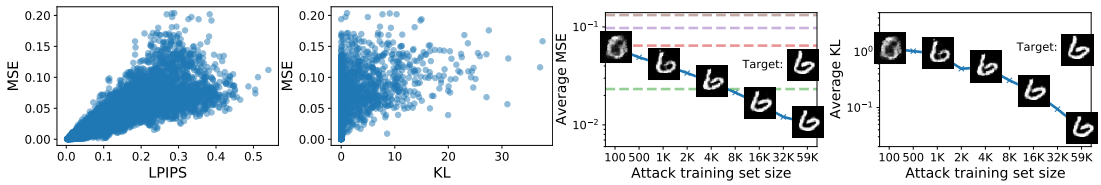
a) Attack training set size: Recall that the general reconstruction attack assumes the attacker has access to k shadow data points $\bar{D}$ from the same distribution as the target point. From this knowledge, the attacker generates a collection of shadow model-target pairs (the attack training data), which is used to train the RecoNN. Note that the size of the attack training data depends both on the knowledge of the attacker (simply, the attacker may not have access to many examples), and on their computational power: they need to train one shadow model per data point to create the attack training data.

We explore the fidelity of reconstruction on MNIST as the amount of attack training data k ranges from 100 to $59K$. Figure 5c shows the average MSE between reconstructions over the $1K$ released model targets as k varies. Clearly, the attack becomes better as more training data is available. However, high fidelity reconstructions occur already with $1K$ shadow models; in our plots we include reconstructed examples at different values of k illustrating this. Reconstructions that are (on average) better than the NN oracle only require $8K$ shadow models. Because the correlation between MSE and KL is not symmetric, we also plot the average KL against attack training set size and observe a similar monotonic decrease (Figure 5d). We observe similar trends on CIFAR-10 when increasing the attack training set size; $5K$ shadow models is enough to generate reconstructions below the 1st percentile oracle MSE ($\sim$0.05) and $10K$ shadow models will generate reconstructions below the NN oracle MSE ($\sim$0.03). See Appendix B for full results on CIFAR-10.

b) Out-of-distribution (OOD) data on CIFAR-10: The previous experiment indicates that reconstructions are poor when an adversary has relatively little side-information ($< 1K$ points) to create shadow models. We now investigate if these additional points must come from the same distribution as the fixed set and target sample. If the attack succeeds even when $\bar{D}$ comes from a different distribution, they can potentially create a larger pool of shadow targets for the attack. In addition, when reasonable OOD data is scarce or not available, the attacker could instead use $D_{\text{-}}$ to train a generative model and use it to generate shadow targets from a similar distribution.

To relax the assumption that shadow targets come from the same distribution as the released model's training data we use CIFAR-100, a standard OOD benchmark for CIFAR-10 [34], to construct the adversary's side knowledge. In this experiment, the fixed data $D_{\text{-}}$ and the 1K test targets are still selected from CIFAR-10, but the shadow targets in $\bar{D}$ are OOD points corresponding to images sampled from CIFAR-100 annotated with a random CIFAR-10 label. We measure attack success on the 1K released models with in-distribution CIFAR-10 targets. As we observe a negligible difference in MSE between the two cases, we conclude that the success of the attack does not require having access to the correct prior distribution. We exploit OOD data in later experiments when evaluating how the size of the fixed set affects reconstruction.

c) Influence of training hyper-parameters: Table II summarizes what factors in training affect reconstruction. The appendix expands upon these and gives empirical insights.
Fixed set size. We measure the role of the fixed set size by reducing from $10K$ to $1K$ (MNIST) and increasing from $5K$



(a) Correlation between MSE and LPIPS. (b) Correlation between MSE and KL metric. (c) Average MSE against attack training set size. (d) Average KL against attack training set size.

Fig. 5: Correlation between metrics, and quality of reconstruction as a function of the number of shadow models.

to $50K$ (CIFAR-10). We observe almost no difference in MSE in both cases; e.g. CIFAR-10 target points can be reconstructed even if there are $50K$ other points in the training set.

Model size and architecture. We assess whether the size and architecture of the released model affect reconstruction. For MNIST, we increase the size of the hidden layer from 10 to 100; this increases the number of trainable parameters tenfold. For CIFAR-10, we double the size from $50K$ to $100K$ by increasing the width of the first linear layer. The rest of the architecture is kept to the defaults (Table V). We observe almost no difference in reconstruction success when attacking these larger released models. Nevertheless, this attack has a bigger computational cost: the size of the RecoNN for CIFAR-10 increases from $226M$ to over $400M$ parameters.

Layers. Instead of allowing the RecoNN to process all parameters from a released model, we restrict to only the second layer for MNIST and convolutional layers for CIFAR-10. This significantly reduces the input size to the reconstructor network, by 98% on MNIST and 84% on CIFAR-10. We observe that this does not substantially affect the reconstruction fidelity, demonstrating that memorization of training points is not localized to a specific layer or small group of neurons.

Epochs. The number of epochs has a small impact on reconstruction. For both MNIST and CIFAR-10 there is a slight increase in MSE if we more than double the number of training epochs, although targets are still successfully reconstructed. We investigate this relationship in more detail in Appendix F.

Activation. One may wonder why we used ELU activations in the released model instead of the more common ReLUs. We noticed that released models with ReLU activations tend to be harder to attack in comparison to other activation functions, resulting in poor quality reconstructions on CIFAR-10 (i.e. MSE larger than the NN oracle). It is well known that ReLUs induce sparse gradients; we observed that $> 60\%$ of weights are not updated during training when the loss is computed with respect to the target. We suspect this is why RecoNN is less effective against ReLU activated models: there is less mutual information between the model parameters and the target in comparison to models trained with other activations. We discuss this in further detail in Appendix G.

Learning rate. Decreasing the learning rate of the released model did not affect the attack in the deterministic training setting. If randomness is introduced via mini-batch sampling, we will see that the learning rate impacts reconstruction.

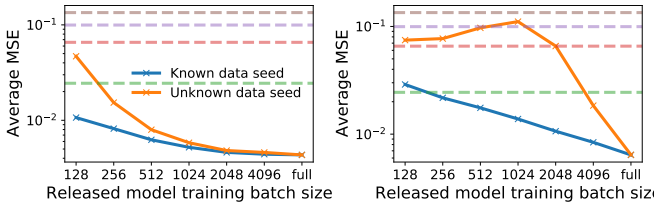


Fig. 6: MSE and released model training batch size when the adversary knows/does not know the data sub-sampling random seed. MSE is sensitive to the learning rate and momentum. Learning rate: 0.01 (left), 0.2 (right). Momentum: 0 (both).

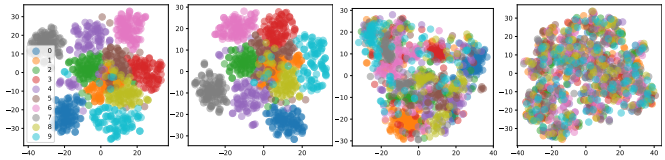


Fig. 7: TSNE embeddings of 1K released models trained with 1024 batch-size. From left to right (known sub-sampling seed, learning rate): (Yes, 0.01), (No, 0.01), (Yes, 0.2), (No, 0.2).

d) Randomness from data sub-sampling: We explore how randomness stemming from data sub-sampling affects the attack on MNIST, by removing the assumption that the released model is trained with full batch gradient descent. We consider settings where the adversary knows the random seed used to shuffle the data (this corresponds to SGD but with no randomness), and settings where the adversary does not know the random seed. Results in Figure 6 indicate that when the adversary knows the data shuffling seed, reconstruction attacks are successful even for small batch sizes. Without knowing the seed, attack success depends on the training hyper-parameters, such as the choice of the learning rate. It appears that attacking models with randomness from sub-sampling is more difficult than determinstically trained released models, and that larger learning rates also increase the hardness of the reconstruction task. Loss landscapes of neural networks are extremely non-convex and contain many local optima [35]; if more randomness is introduced, this will increase the opportunity for different shadow models to reach different optima. This increases the difficulty of reconstruction as these shadow models will not be representative of the optima attained by the released model, and training with a larger learning rate will exacerbate this issue. In Figure 7, we

show plot TSNE embeddings of parameters for all $1K$ released models for each of the two learning rates given in Figure 6 and the two randomness settings (known and unknown seed) for a batch size of 1024. We represent each released model with a color depending on the label of the respective target. For a small learning rate, labels are grouped together in both known and unknown seed settings, implying the local optima these models realize are similar; this makes it easier for the RecoNN to learn and subsequently generalize to the released model. Conversely, in the large learning rate setting there is a stark difference between known and unknown seed settings: if the seed is known, groupings of labels still happen, and a successful attack is possible; however, if the seed is unknown, the local optima reached by each released model has less structure that the reconstructor network can learn on.

e) Randomness from model initialization: We explore how initialization randomness can affect the attack on MNIST. Firstly, we remove the assumption that the adversary knows the initial parameters of the released model; in practice, this means training each released and shadow model with a new random seed controlling the model's initial parameters. By default, each linear and convolutional layer is initialized with Lecun Normalization, which is the default in the Haiku library [36]. In our experiments, we evaluated other common initialization procedures (e.g., Glorot, He), which did not change any of our findings; we omit these results. We refer the reader to Figure 4 for visual inspection of reconstructions at the two error rates reported in Table II, and conclude that the attack is unable to successfully reconstruct without knowledge of initialization, as they are far larger than the NN oracle described in Section V-B.

One may conjecture that the current attack pipeline is not suitable for this setting: we only train a single shadow model per shadow target, which may fail to capture the variance in shadow model parameters over different initializations for the same shadow target. For this reason, we further created an attack training set of $5M$ shadow model-target pairs, consisting of $10K$ shadow targets, where each target has 500 shadow models all differing in initial parameters. Even so, this approach did not improve the MSE reported in Table II. In Appendix A, we discuss evidence suggesting that reconstruction may not be possible without knowing the initial released model parameters. A similar observation was made by Jagielski et al. [37], who run attacks to find lower bounds of the privacy budget ϵ in DP-SGD. They observed that the bounds become tighter with less randomness from model initialization.

C. Black-box Access to Released Model

We design a black-box attack by limiting the adversary's access to only the logits predicted by the released model. For each shadow model, using a set of 200 (500) images from $\bar{D}$ for MNIST (CIFAR-10), the adversary collects the logit outputs of each image, concatenates them together, and uses this as the feature representation of the model, instead of the flattened weights. This reduces the dimensionality of the feature vector from 8K to 2K for MNIST and 55K to 5K for CIFAR-10. The average MSE using this logit representation

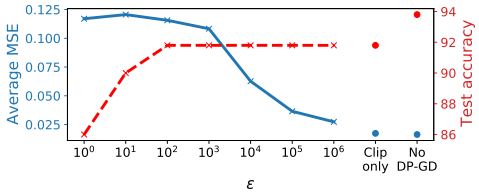


Fig. 8: Average MSE of reconstructions and test accuracy of released model using (ϵ, δ)-DP on the MNIST dataset.

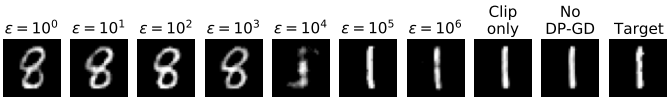


Fig. 9: Example of MNIST reconstructions under DP.

approach is 0.011 for MNIST and 0.0198 for CIFAR-10, which is only marginally worse than the MSE of white-box attacks with default settings, and still much better that the NN oracle. We conclude that black-box reconstruction attacks are feasible and have comparable performance to white-box ones.

D. Released Model Trained with Differential Privacy

Having discussed what factors help and hinder reconstruction, we now evaluate on MNIST the resilience of models trained with DP. The released model training set-up is identical to before (Section IV-B), except we train with full batch DP gradient descent (DP-GD) with clipped gradients [15]. Gradients are clipped to have a maximum ℓ_2 norm of 1, and Gaussian noise (unknown to the adversary) is added to make the model (ϵ, δ)-DP with $\delta = 10^{-5}$. Figure 8 shows that even a large ϵ successfully mitigates reconstruction attacks, and that in these ϵ regimes the reduction in utility (measured by test accuracy) is negligible (Appendix E reports similar results on CIFAR-10). Interestingly, for high levels of privacy, the reconstruction attack generates realistic but wildly incorrect reconstructions (Figure 9). These findings motivate our theoretical investigation into what level of DP is sufficient to protect against reconstruction attacks.

VII. Towards Formal Guarantees Against Reconstruction Attacks

Mitigations that (provably) protect released models against reconstruction attacks can (and should) be implemented within the training algorithm used by the model developer. Protections that defend against effective reconstruction by informed adversaries will also protect against attacks by weaker, more realistic adversaries. In this section, we propose a definition of *reconstruction robustness* against informed adversaries, and compare it to the privacy guarantees afforded by DP. As will soon become apparent, the strength of mitigations against reconstruction is necessarily going to be *relative* to the strength of the prior information available to the adversary.

A. Reconstruction Robustness

Our main definition focuses on bounding the success probability of achieving accurate reconstruction by any (informed)

adversary. The definition is parameterized by the side information available to the adversary, captured by a probabilistic prior π from which the target z is sampled, and by the adversary's goal expressed as a measure of reconstruction error ℓ.

Definition 2. *Let π by a prior over $\mathcal{Z}$ and $\ell : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}_{\geq 0}$ a reconstruction error function. A randomized mechanism $M : \mathcal{Z}^n \to \Theta$ is (η, γ)-ReRo (*reconstruction robust*) with respect to π and ℓ if for any dataset $D_{-} \in \mathcal{Z}^{n-1}$ and any reconstruction attack $R : \Theta \to \mathcal{Z}$ we have*

$$\mathbb{P}_{Z\sim\pi,\theta\sim M(D_{-}\cup\{Z\})}[\ell(Z, R(\theta)) \leq \eta] \leq \gamma \ . \tag{2}$$

Suppose M is an (η, γ)-ReRo mechanism. The definition prevents any reconstruction attack with knowledge[3] of the prior π, the dataset D_{-} and the output $\theta = M(D_{-} \cup \{Z\})$ to attain a reconstruction error lower than η on an unknown target $Z \sim \pi$ with probability larger than γ. A good ReRo mechanism is one with large η and very small γ, i.e. one where even "decent" reconstructions are impossible with high probability. In practice a tension between these two parameters is expected, at least for mechanisms providing some form of utility when computing a function depending on all the inputs.

Definition 2 assumes the reconstruction attack is deterministic. We could consider randomized attacks instead, but note that determinism is not a limitation when trying to capture worst-case attacks: the R that maximizes $\mathbb{P}[\ell(Z, R(\theta)) \leq \eta]$ is given by the (deterministic) *maximum a posteriori* attack:

$$R^*(\theta) = \mathrm{argmax}_{\hat{z}\in\mathcal{Z}}\, \mathbb{P}_{Z\sim\pi}[\ell(Z, \hat{z}) \leq \eta | M(D_{-} \cup \{Z\}) = \theta] \ .$$

Similarly, the definition protects against adversaries with full knowledge of the prior π. Since the optimal attack run by an adversary with a wrong prior is necessarily weaker than the optimal attack with a correct prior, assuming the adversary knows π is preferable when designing mitigations.

Our main results provide two connections between reconstruction robustness and DP. The first observation is that DP implies reconstruction robustness. Quantitatively, we show that the ReRo parameters of a Rényi DP (RDP) mechanism depend in a simple way on its privacy parameters and another quantity capturing the relation between π and ℓ. The second observation is that any mechanism that is robust against exact reconstruction with respect to a sufficiently rich family of priors supported on pairs of points must satisfy DP. Together, both results stress the importance of correctly modelling an adversary's prior knowledge in effectively protecting against reconstruction attacks. In particular, we show that very weak DP guarantees suffice to protect against reconstruction when the adversary has limited knowledge about the target point.

B. From DP to ReRo

We now show that differentially private mechanisms provide reconstruction robustness. Let us recall the definitions of approximate and Rényi DP.

[3]Knowledge of π and D in the attack is implicit through the fact that (2) has to hold for any reconstruction attack.

Definition 3 ([6, 38, 39])**.** *Let $M : \mathcal{Z}^n \to \Theta$ be a randomized mechanism, $\epsilon > 0$, $\delta \in [0, 1]$ and $\alpha > 1$. We say that:*

1) *M is (ϵ, δ)-DP if for any datasets $D, D' \in \mathcal{Z}^n$ differing in a single record and any event $E \subseteq \Theta$ we have*

$$\mathbb{P}[M(D) \in E] - e^{\epsilon}\mathbb{P}[M(D') \in E] \leq \delta \ .$$

   When $\delta = 0$ we simply say the mechanism is ϵ-DP.
2) *M is (α, ϵ)-RDP if for any datasets $D, D' \in \mathcal{Z}^n$ differing in a single record we have*

$$\mathbb{E}_{\theta\sim M(D')}\left[\left(\frac{\mathbb{P}[M(D) = \theta]}{\mathbb{P}[M(D') = \theta]}\right)^{\alpha}\right] \leq e^{(\alpha-1)\epsilon} \ .$$

The effect of the prior on ReRo bounds obtained from DP is through an anti-concentration property. For prior π, error function ℓ and error threshold η, define the *baseline error* as

$$\kappa_{\pi,\ell}(\eta) = \sup_{z_0\in\mathcal{Z}} \mathbb{P}_{Z\sim\pi}[\ell(Z, z_0) \leq \eta] \ .$$

When π, ℓ or η are clear from the context we may drop them to unclutter our notation. Whenever ℓ is a metric on $\mathcal{Z}$, an upper bound on κ provides a measure of *anti-concentration* of the prior by guaranteeing that no single point has too much of probability mass concentrated around it; bounds for κ for some prior distributions are given in Section VII-D. Another interpretation of κ is as the success probability of the best *oblivious* reconstruction attack that ignores the output of M. By this interpretation, the next theorem says that if a mechanism is RDP, the best reconstruction attack cannot have success probability much larger than the best oblivious attack.

Theorem 2. *Fix π, ℓ and $\eta > 0$, and let $\kappa = \kappa_{\pi,\ell}(\eta)$. If a mechanism M satisfies (α, ϵ)-RDP then it also satisfies (η, γ)-ReRo with respect to π and ℓ with $\gamma = (\kappa \cdot e^{\epsilon})^{\frac{\alpha-1}{\alpha}}$.*

Taking $\alpha \to \infty$ and recalling that (∞, ϵ)-RDP is equivalent to ϵ-DP [39] we obtain the following corollary.

Corollary 3. *Fix π, ℓ and $\eta > 0$, and let $\kappa = \kappa_{\pi,\ell}(\eta)$. If a mechanism M satisfies ϵ-DP then it also satisfies (η, γ)-ReRo with respect to π and ℓ with $\gamma = \kappa \cdot e^{\epsilon}$.*

Another way to interpret Theorem 2 is through the lens of zero-concentrated DP (zCDP) [40]. A mechanism is ρ-zCDP if it satisfies $(\alpha, \alpha\rho)$-RDP for every $\alpha > 1$. This definition provides a natural and convenient way to express the privacy afforded by the ubiquitous Gaussian mechanism [38]. Applying Theorem 2 to a ρ-zCDP mechanism and optimizing α to minimize the upper bound yields the following.

Corollary 4. *Fix π, ℓ and $\eta > 0$, and let $\kappa = \kappa_{\pi,\ell}(\eta)$. If a mechanism M satisfies ρ-zCDP with $\rho < \log(1/\kappa)$ then it also satisfies (η, γ)-ReRo with respect to π and ℓ with $\gamma = e^{-(\sqrt{\log(1/\kappa)} - \sqrt{\rho})^2}$.*

C. From ReRo to DP

Next we investigate the reverse implication: does a strong enough level of reconstruction robustness imply a standard definition of privacy protection like DP? We show that this is

indeed the case if one insists on protecting against *exact* reconstruction simultaneously for a family of priors concentrated on pairs of data points. From this lens, the result says that as soon as a mechanism exhibits strong enough reconstruction robustness to prevent membership inference it must necessarily satisfy DP.

Before stating the result we introduce the following notation. Given $p \in (0,1)$ and $z, z' \in \mathcal{Z}$, $z \neq z'$, let $\pi_{p,z,z'}$ denote the prior over $\mathcal{Z}$ that assigns probability p to z and $1-p$ to z'. We also let $\ell_{0/1}(z,z') = \mathbf{1}[z \neq z']$.

Theorem 5. *Fix $\epsilon \geq 0$, $\eta \in (0,1)$ and $\gamma \in [0,1]$. Let $\Pi_\epsilon = \{\pi_{p,z,z'} : z, z' \in \mathcal{Z}, z \neq z'\}$ be the class of all priors on $\mathcal{Z}$ concentrated on pairs of points with $p = \frac{1}{e^\epsilon+1}$. If a mechanism $M : \mathcal{Z}^n \to \Theta$ is (η,γ)-ReRo with respect to $\ell_{0/1}$ and every prior $\pi \in \Pi_\epsilon$, then M satisfies (ϵ,δ)-DP with $\delta = \max\{0, (e^\epsilon+1)\gamma - e^\epsilon\}$.*

D. ReRo Against High-Dimensional, High-Uncertainty Priors

A standard "rule of thumb" says that DP only provides a meaningful protection when ϵ is a small constant. On the other hand, our experiment on models trained with DP-SGD (Section VI-D, Appendix E) shows that much larger values of ϵ are successful at mitigating our RecoNN-based attack. This could be interpreted as a limitation of our attack in the presence of weak levels of DP protection. An alternative explanation is that DP with large values of ϵ can protect against reconstruction attacks if the reconstruction target is high-dimensional and the adversary's prior knowledge contains a large degree of uncertainty. We formalize this intuition by instantiating the bounds from Section VII-B on two natural priors where κ is easy to bound: uniform and Gaussian priors. A similar analysis in the context of local DP was presented in [41] (see Section VII-F for a detailed comparison).

a) Uniform priors: Suppose training data points in $\mathcal{Z}$ are represented by d-dimensional real vectors and all the adversary knows about the target point z is a norm bound of the form $\|z\|_2 \leq 1$. Then it makes sense for the adversary to take as prior the uniform distribution $\mathcal{U}(B_1^d(0))$ over the Euclidean d-dimensional unit ball $B_1^d(0)$ centered at zero. For simplicity, suppose also that reconstruction error is measured in terms of the Euclidean distance ℓ_2. Then we have the following.

Proposition 6. *Fix a constant $\eta \in (0,1)$. Suppose M is a mechanism satisfying ϵ-DP with $\epsilon = o(d)$ or ρ-zCDP with $\rho = o(d)$. Then M is (η,γ)-ReRo with respect to $\mathcal{U}(B_1^d(0))$ and ℓ_2 with $\gamma = e^{-\Omega(d)}$.*

This result shows that, in high-dimensional settings where an informed adversary's knowledge about the target datapoint is only in the form a syntactic constraint like $\|z\|_2 \leq 1$, privacy parameters sub-linear in the dimension suffice to make the reconstruction success probability negligible.

b) Gaussian priors: Another natural prior to consider is a (d-dimensional, isotropic) Gaussian distribution $\mathcal{N}(w, \sigma^2 I_d)$ specifying the adversary's prior knowledge about the location w of the target point with some degree of uncertainty controlled by σ. Taking again ℓ_2 as the measure of reconstruction error, we obtain the following.

Proposition 7. *Fix a constant $\eta > 0$. Suppose M is a mechanism satisfying ϵ-DP with $\epsilon = o(d)$ or ρ-zCDP with $\rho = o(d)$. Then M is (η,γ)-ReRo with respect to $\mathcal{N}(w, \sigma^2 I_d)$ and ℓ_2 with $\gamma = e^{-\Omega(d)}$ as long as $\sigma \geq \frac{2\eta}{\sqrt{d}}$.*

The idea that large values of ϵ can protect against reconstruction when the adversary's prior contains significant uncertainty (i.e. it is diffused) was previously noticed in [41] in the context of local DP (LDP) with priors close to uniform. Inspired by FL applications where adversaries get access to LDP gradients, the authors propose a notion of protection against *reconstruction breaches* that is more stringent than Definition 2: it asks that the adversary cannot effectively reconstruct a particular feature of interest about the target point no matter what the output of the mechanism is – in contrast, ReRo uses an *average-case* requirement over the outputs of the mechanism. Technically, [41, Lemma 2.2] shows that the bound in Corollary 3 also holds for this worst-case notion of protection against reconstruction.[4] Such worst-case guarantees, however, are not attainable under relaxations of ϵ-DP like RDP because the latter does not enforce an almost sure bound on the privacy loss: instead, it just guarantees that the privacy loss will be small with high probability. Thus, Theorem 2 and Proposition 6 are natural generalizations of the results from [41] to RDP, which is the default notion of privacy provided by DP-SGD and other popular private ML algorithms [42, 43].

E. Is Reconstruction Robustness Useful in Practice?

To deploy the bounds from Theorem 2 two things are necessary: the description of a criterion for reconstruction error ℓ with an associated threshold η, and an understanding of the success rate of η-approximate reconstruction by the adversary prior to the release. Equipped with ℓ and η, one can then engage in a conversation with stakeholders and domain experts to determine what success rate of reconstruction is reasonable to adjudicate to a potential adversary before the release is made. An interesting feature of Theorem 2 is that it reduces adversarial modelling to a question about determining a *single number* $\kappa_{\pi,\ell}(\eta)$. Furthermore, it is possible that one does not need to be overly conservative in estimating this number. After all, the theorem bounds the success probability of the wost-case adversary which, in particular, knows all the fixed dataset. Realistic adversaries will often have less knowledge of the fixed dataset, so it might be possible to trade-off knowledge of the fixed dataset with the amount of diffusion required from the prior. We leave this question for future work.

F. Further Related Work

a) Threat modelling and privacy semantics: The use of informed adversaries in formal privacy analyses can be tracked back to the *sub-linear queries* (SuLQ) framework [44]. SuLQ was later subsumed by DP [6], where mentions to a

[4] Although the bound in [41] is stated in terms of ϵ-LDP, it is easy to see that the same holds for central ϵ-DP in the presence of an informed adversary.

concrete adversary were expressly avoided in the definition that is widely used nowadays [45]. Nonetheless, [6, Appendix A] provides a "semantically flavored" definition equivalent to DP which involves the likelihood ratio between the prior and posterior beliefs of an informed adversary about any property of the target data point. The adversarial model put forward in Section II uses the same notion of informed adversary.

In other frameworks where the adversary is not (necessarily) informed (e.g. Pufferfish privacy [46] and inferential privacy [47]), side knowledge about the whole dataset is encoded in a probabilistic prior capturing information about the individual entries in the dataset as well as their statistical dependencies. These frameworks extend the semantic approach to DP by replacing the prior-vs-posterior condition with an odds ratio condition – such modification is motivated by the observation that prior-vs-posterior bounds cannot hold in general for uninformed adversaries unless the prior distribution over the dataset assumes the records are mutually independent. Alternatively, [48] provides posterior-vs-posterior semantics for DP in the presence of an uninformed adversary with an arbitrary prior. In the definition of reconstruction robustness, our use of an informed adversary with a prior over the target data point circumvents the complications arising from dependencies between points in the training data: the prior captures the adversary's *residual* uncertainty about the target point after observing the fixed dataset. On the opposite direction, several authors have proposed approaches where the adversary's uncertainty with respect to the input data of a mechanism is leveraged to increase the privacy provided to individuals [49, 50, 51, 52]. Implicitly, these works assume a less powerful adversary than the one considered in this paper.

Most of the semantic definitions we discussed formalize the privacy protection goal without assuming the adversary is interested in a particular inference task; that is, protection applies simultaneously to all possible inferences about the target point(s). In contrast, the use of an explicit reconstruction error ℓ makes the definition of reconstruction robustness syntactic in nature. Section II-B briefly discusses how the problem of designing an appropriate error function for each application can be approached. A similar dilemma arises in location privacy, where distortion-based notions include an explicit measure of reconstruction error [53, 54]. Nonetheless, as Theorem 5 shows, by considering a very stringent reconstruction goal and a set of sufficiently informative priors one can recover semantic privacy notions from reconstruction robustness.

The connection between DP and protection against membership inference is perhaps best understood via its hypothesis testing interpretation [55, 56]. A comprehensive discussion of the adversary *implicit* in the definition of DP from the hypothesis testing standpoint can be found in [14]. Interestingly, [57] shows that, unlike standard DP, RDP does not admit a hypothesis testing interpretation. A semantic (Bayesian) interpretation of RDP in terms of moment bounds on the odds ratio is presented in [39]. Theorem 2 provides an alternative characterization of the privacy protection afforded by RDP in terms of resilience to reconstruction attacks.

b) DP and protection against reconstruction: How standard DP offers concrete protection against reconstruction attacks has been studied in other contexts. Indeed, one of the original motivations for the definition of DP was to defeat database reconstruction attacks in the context of interactive query mechanisms [58, 59, 60, 61, 62]. In such attacks, the adversary receives (noisy) answers to a sequence of specially crafted queries against a database and, if the noise is small enough, uses the answers to (partially) reconstruct every record in the database. The success of these attacks is contingent on the adversary's ability to control these queries; in contrast, in ML applications like the ones we consider the computation performed by the mechanism is completely under the model developer's control.

The quantitative information flow literature seeks to provide information-theoretic bounds on data leakage in information processing systems [63, 64]. When applied to differentially private mechanisms, these ideas yield bounds on the protection against *exact* reconstruction when $\mathcal{Z}$ is finite. In particular, when specialized to informed adversaries and translated into our terminology, [65, Theorem 3] shows that any ϵ-DP mechanism is (η, γ)-ReRo with $\eta \in (0, 1)$ with respect to $\ell_{0/1}$ and any prior π with $\gamma \leq \frac{|\mathcal{Z}|\kappa e^{\epsilon}}{|\mathcal{Z}|+e^{\epsilon}-1}$. Taking $|\mathcal{Z}| \to \infty$ recovers the bound from Corollary 3 in the case of $\ell_{0/1}$. Our results can thus be interpreted as a generalization of this line of work where no assumptions about $\mathcal{Z}$ are necessary.

VIII. Conclusions

Our work provides compelling evidence that standard ML models can memorize enough information about their training data to enable high-fidelity reconstructions in a very stringent threat model. By instantiating an informed adversary that learns to map model parameters to training images, we successfully attacked standard MNIST and CIFAR-10 classifiers with up to $100K$ parameters, and showed the attack is significantly robust to changes in the training hyper-parameters. Two aspects of our attack we would like to improve in future work are its data and computational efficiency, and its scalability to larger, more performant released models. This would not lead to real-world adversaries mounting practical attacks due to the nature of our threat model, but it would enable model developers to assess potential privacy leakage in models before deployment. Extending our attacks to reconstruct $N > 1$ targets simultaneously would also be interesting, but we expect this to be substantially harder. For example, in this setting our attacks against convex models lead to a problem with more unknowns than equations. On the defenses side, we empirically showed that DP training with large values of ϵ can effectively mitigate our reconstruction attacks. Our theoretical discussion, stemming from a new definition of reconstruction robustness and a study of its connection to (R)DP, shows this is a general phenomenon: informed reconstruction attacks can be prevented with large values of ϵ under some assumptions on the adversary. Validating such assumptions in particular applications would open the door to practical models which are accurate and resilient against reconstruction attacks.

ACKNOWLEDGMENT

The authors would like to thank: Leonard Berrada, Adrià Gascón and Shakir Mohamed for feedback on an earlier version of this manuscript; Brendan McMahan for suggesting the idea that random initialization in SGD might make privacy attacks harder which inspired some of our experiments; and Olivia Wiles for discussions on how to improve reconstructor network training on CIFAR-10. This work was done while G.C. was at the Alan Turing Institute.

REFERENCES

[1] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *International Conference on Learning Representations (ICLR)*, 2017.

[2] V. Feldman, "Does learning require memorization? a short tale about a long tail," in *ACM Symposium on Theory of Computing (STOC)*, 2020.

[3] V. Feldman and C. Zhang, "What neural networks memorize and why: Discovering the long tail via influence estimation," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[4] G. Brown, M. Bun, V. Feldman, A. D. Smith, and K. Talwar, "When is memorization of irrelevant training data necessary for high-accuracy learning?" in *ACM Symposium on Theory of Computing (STOC)*, 2021.

[5] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE Symposium on Security and Privacy (SP)*, 2017.

[6] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference (TCC)*, 2006.

[7] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *USENIX Security Symposium*, 2019.

[8] N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. B. Brown, D. Song, Ú. Erlingsson, A. Oprea, and C. Raffel, "Extracting training data from large language models," in *USENIX Security Symposium*, 2021.

[9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[10] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[11] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *USENIX Security Symposium*, 2014.

[12] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov, "Property inference attacks on fully connected neural networks using permutation invariant representations," in *ACM Conference on Computer and Communications Security (CCS)*, 2018.

[13] A. Suri and D. Evans, "Formalizing and estimating distribution inference risks," *arXiv:2109.06024*, 2021.

[14] M. Nasr, S. Song, A. Thakurta, N. Papernot, and N. Carlini, "Adversary instantiation: Lower bounds for differentially private machine learning," in *IEEE Symposium on Security and Privacy (SP)*, 2021.

[15] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *ACM Conference on Computer and Communications Security (CCS)*, 2016.

[16] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *ACM Conference on Computer and Communications Security (CCS)*, 2015.

[17] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *IEEE Computer Security Foundations Symposium (CSF)*, 2018.

[18] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[19] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models," in *Network and Distributed System Security Symposium (NDSS)*, 2019.

[20] 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," in *IEEE Symposium on Security and Privacy (SP)*, 2019.

[21] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[22] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE Conference on Computer Communications (INFOCOM)*, 2019.

[23] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients - how easy is it to break privacy in federated learning?" in *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[24] A. Wainakh, F. Ventola, T. Müßig, J. Keim, C. G. Cordero, E. Zimmer, T. Grube, K. Kersting, and M. Mühlhäuser, "User label leakage from gradients in federated learning," *arXiv:2105.09369*, 2021.

[25] S. Z. Béguelin, L. Wutschitz, S. Tople, V. Rühle, A. Paverd, O. Ohrimenko, B. Köpf, and M. Brockschmidt, "Analyzing information leakage of updates to natural language models," in *ACM Conference on Computer and Communications Security*

(CCS), 2020.

[26] A. Salem, A. Bhattacharya, M. Backes, M. Fritz, and Y. Zhang, "Updates-leak: Data set inference and reconstruction attacks in online learning," in *USENIX Security Symposium*, 2020.

[27] P. McCullagh and J. A. Nelder, *Generalized linear models*. Routledge, 2019.

[28] R. W. Wedderburn, "On the existence and uniqueness of the maximum likelihood estimates for certain generalized linear models," *Biometrika*, 1976.

[29] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[30] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[31] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[33] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *British Machine Vision Conference (BMVC)*, 2016.

[34] S. Fort, J. Ren, and B. Lakshminarayanan, "Exploring the limits of out-of-distribution detection," *arXiv:2106.03004*, 2021.

[35] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[36] T. Hennigan, T. Cai, T. Norman, and I. Babuschkin, "Haiku: Sonnet for JAX," 2020. [Online]. Available: http://github.com/deepmind/dm-haiku

[37] M. Jagielski, J. Ullman, and A. Oprea, "Auditing differentially private machine learning: How private is private sgd?" *Advances in Neural Information Processing Systems*, 2020.

[38] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, 2006.

[39] I. Mironov, "Rényi differential privacy," in *IEEE Computer Security Foundations Symposium (CSF)*, 2017.

[40] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Theory of Cryptography Conference (TCC)*, 2016.

[41] A. Bhowmick, J. C. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, "Protection against reconstruction and its applications in private federated learning," *arXiv:1812.00984*, 2018.

[42] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson, "Scalable private learning with PATE," in *International Conference on Learning Representations (ICLR)*, 2018.

[43] I. Mironov, K. Talwar, and L. Zhang, "Rényi differential privacy of the sampled gaussian mechanism," *arXiv:1908.10530*, 2019.

[44] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: the SuLQ framework," in *ACM Symposium on Principles of Database Systems (PODS)*, 2005.

[45] F. McSherry, "I suspect the "Discovery" had a different feel for the various involved people. I personally spent a lot of time trying to remove explicit references to adversaries and assumptions about them." Jan 2021. [Online]. Available: https://twitter.com/frankmcsherry/status/1354789417727234049

[46] D. Kifer and A. Machanavajjhala, "Pufferfish: A framework for mathematical privacy definitions," *ACM Trans. Database Syst.*, 2014.

[47] A. Ghosh and R. Kleinberg, "Inferential privacy guarantees for differentially private mechanisms," in *Innovations in Theoretical Computer Science Conference (ITCS)*, 2017.

[48] S. P. Kasiviswanathan and A. D. Smith, "On the 'semantics' of differential privacy: A bayesian formulation," *J. Priv. Confidentiality*, 2014.

[49] Y. Duan, "Privacy without noise," in *ACM Conference on Information and Knowledge Management (CIKM)*, 2009.

[50] R. Bhaskar, A. Bhowmick, V. Goyal, S. Laxman, and A. Thakurta, "Noiseless database privacy," in *International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT)*, 2011.

[51] R. Bassily, A. Groce, J. Katz, and A. D. Smith, "Coupled-worlds privacy: Exploiting adversarial uncertainty in statistical data privacy," in *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2013.

[52] D. Desfontaines, E. Mohammadi, E. Krahmer, and D. Basin, "Differential privacy with partial knowledge," *arXiv:1905.00650*, 2019.

[53] R. Shokri, J. Freudiger, M. Jadliwala, and J. Hubaux, "A distortion-based metric for location privacy," in *ACM Workshop on Privacy in the Electronic Society (WPES)*, 2009.

[54] R. Shokri, G. Theodorakopoulos, J. L. Boudec, and J. Hubaux, "Quantifying location privacy," in *IEEE Symposium on Security and Privacy (SP)*, 2011.

[55] L. Wasserman and S. Zhou, "A statistical framework for differential privacy," *Journal of the American Statistical Association*, 2010.

[56] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," in *International Conference on Machine Learning (ICML)*, 2015.

[57] B. Balle, G. Barthe, M. Gaboardi, J. Hsu, and T. Sato, "Hypothesis testing interpretations and renyi differential privacy," in *International Conference on Artificial Intel-*

ligence and Statistics (AISTATS), 2020.
[58] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *ACM Symposium on Principles of Database Systems (PODS)*, 2003.
[59] C. Dwork, F. McSherry, and K. Talwar, "The price of privacy and the limits of LP decoding," in *ACM Symposium on Theory of Computing (STOC)*, 2007.
[60] C. Dwork, A. Smith, T. Steinke, and J. Ullman, "Exposed! A survey of attacks on private data," *Annual Review of Statistics and Its Application*, 2017.
[61] A. Cohen and K. Nissim, "Linear program reconstruction in practice," *J. Priv. Confidentiality*, 2020.
[62] A. Cohen, S. Nikolov, Z. Schutzman, and J. Ullman, "Reconstruction attacks in practice," 2020. [Online]. Available: https://differentialprivacy.org/diffix-attack/
[63] G. Smith, "On the foundations of quantitative information flow," in *International Conference on Foundations of Software Science and Computational Structures (FOSSACS)*, 2009.
[64] M. S. Alvim, K. Chatzikokolakis, A. McIver, C. Morgan, C. Palamidessi, and G. Smith, *The Science of Quantitative Information Flow*. Springer, 2020.
[65] E. ElSalamouny, K. Chatzikokolakis, and C. Palamidessi, "Generalized differential privacy: Regions of priors that admit robust optimal mechanisms," in *Horizons of the Mind. A Tribute to Prakash Panangaden - Essays Dedicated to Prakash Panangaden on the Occasion of His 60th Birthday*, 2014.
[66] B. Balle, G. Cherubin, and J. Hayes, "Reconstructing training data with informed adversaries," *arxiv:2201.04845*, 2022.
[67] M. Shaked and J. G. Shanthikumar, *Stochastic orders*. Springer Science & Business Media, 2007.
[68] S. Dasgupta and A. Gupta, "An elementary proof of a theorem of johnson and lindenstrauss," *Random Struct. Algorithms*, 2003.
[69] M. Johnson, "add gpu determinism note," Nov 2020. [Online]. Available: https://github.com/google/jax/pull/4824
[70] "JAX activations," https://jax.readthedocs.io/en/latest/jax.nn.html, accessed: 2022-03-25.

APPENDIX
PROOFS

We provide proof sketches for the main theoretical results of the paper. Full proofs can be found on the arXiv version of the paper [66].

Proof sketch of Theorem 1. For a GLM model θ trained to convergence, (1) takes the form

$$x(g^{-1}(\langle x,\theta\rangle)-y) = -\bar{X}^\top(g^{-1}(\bar{X}\theta)-\bar{Y}) - \lambda\theta \ .$$

When the model contains an intercept parameter, this yields d equations with d unknowns $(x_2,\dots,x_d,y)$ because $x_1=1$. From the equation corresponding to this coordinate we obtain $g^{-1}(\langle x,\theta\rangle)-y = \bar{X}_1^\top B + \lambda\theta_1$, which we can plug in the rest of equations to obtain the desired expression for x. Once we have x we plug it back into the first equation to recover y. □

Proof sketch of Theorem 2. Fix $R:\Theta\to\mathcal{Z}$ and $D_-\in\mathcal{Z}^{n-1}$, and let $Z\sim\pi$, $D_Z = D_-\cup\{Z\}$ and $\theta\sim M(D_Z)$. We write $p_M(\theta|z) = \mathbb{P}[M(D_z)=\theta]$ for the output density of M on input D_z. First we take an arbitrary $z_0\in\mathcal{Z}$ and show the probability $\mathbb{P}[\ell(Z,R(\theta))\le\eta]$ equals

$$\int_\Theta\left(\int_{\mathcal{Z}}\mathbf{1}[\ell(z,R(\theta))\le\eta]\frac{p_M(\theta|z)}{p_M(\theta|z_0)}\pi(dz)\right)p_M(d\theta|z_0) \ .$$

Next we take $\alpha' = \frac{\alpha}{\alpha-1}$ and through a standard application of Hölder's inequality bound the inner integral above by:

$$\kappa^{1/\alpha'}\cdot\left(\int_{\mathcal{Z}}\left(\frac{p_M(\theta|z)}{p_M(\theta|z_0)}\right)^\alpha\pi(dz)\right)^{1/\alpha} \ .$$

Plugging this bound into the expression for $\mathbb{P}[\ell(Z,R(\theta))\le\eta]$ and re-arranging terms, we use Jensen's inequality and the RDP assumption on M to obtain:

$$\begin{aligned}&\left(\frac{\mathbb{P}[\ell(Z,R(\theta))\le\eta]}{\kappa_\pi(\eta)^{1/\alpha'}}\right)^\alpha \le\\ &\le\int_{\mathcal{Z}}\left(\int_\Theta\left(\frac{p_M(\theta|z)}{p_M(\theta|z_0)}\right)^\alpha p_M(d\theta|z_0)\right)\pi(dz)\\ &\le\sup_z\int_\Theta\left(\frac{p_M(\theta|z)}{p_M(\theta|z_0)}\right)^\alpha p_M(d\theta|z_0)\\ &\le e^{(\alpha-1)\epsilon} \ .\end{aligned}$$

□

Proof sketch of Theorem 5. Fix arbitrary $D_-\in\mathcal{Z}^{n-1}$, $z,z'\in\mathcal{Z}$, $z\neq z'$, and $E\subseteq\Theta$, and let $\pi=\pi_{p,z,z'}$. Define the reconstruction mapping R_E mapping θ to z is $\theta\in E$ and to z' otherwise. By the ReRo assumptions on M we have $\mathbb{P}_{Z\sim\pi,\theta\sim M(D_Z)}[R_E(\theta)=Z]\le\gamma$. On the other hand, by definition of π and R_E, $\mathbb{P}_{Z\sim\pi,\theta\sim M(D_Z)}[R_E(\theta)=Z]$ equals

$$\frac{\mathbb{P}[M(D_z)\in E]-e^\epsilon\mathbb{P}[M(D_{z'})\in E]+e^\epsilon}{e^\epsilon+1} \ .$$

Upper bounding by γ and re-arranging completes the proof. □

Proof of Proposition 6. Let $\pi=\mathcal{U}(B_1^d(0))$ and write $\mathsf{Vol}(A)$ to denote the Eucliean volume of a set $A\subset\mathbb{R}^d$. By definition of the baseline error, for $\eta\in(0,1)$ we have

$$\kappa_{\pi,\ell_2}(\eta) = \sup_{z_0}\frac{\mathsf{Vol}(B_1^d(0)\cap B_\eta^d(z_0))}{\mathsf{Vol}(B_1^d(0))} = \eta^d = e^{-\Omega(d)} \ ,$$

where the calculation follows by the standard volume formula for d-dimensional Euclidean balls. Plugging this expression in Corollary 3 shows that any ϵ-DP mechanism with $\epsilon=o(d)$ provides (η,γ)-ReRo with respect to π and ℓ with $\gamma=e^{-\Omega(d)}$. A similar claim follows from Corollary 4 applied to ρ-zCDP mechanisms with $\rho=o(d)$. □

Proof sketch of Proposition 7. Let $Z\sim\mathcal{N}(0,I)$ and $F_\eta(z_0)=\mathbb{P}[\|Z+z_0\|^2\le\eta^2]$. First we show that $\operatorname{argmax}_{z_0}F_\eta(z_0)=0$. The proof of this intuitive fact relies on extending a 1-dimensional stochastic domination property

of Gaussian random variables [67, Example 1.A.27] to d dimensions using an orthogonal decomposition of Z along the space spanned by z_0 and its orthogonal complement. Then we show that for $\nu = \mathcal{N}(w, \sigma^2 I)$ this claim implies $\kappa_{\nu,\ell_2}(\eta) = F_{\eta/\sigma}(0)$. Next we use a tail lower bound for chi-squared random variables [68, Lemma 2.2] to get

$$\kappa_{\nu,\ell_2}(\eta) \leq e^{\frac{d}{2}\left(1-\frac{\eta^2}{\sigma^2 d}+\log\frac{\eta^2}{\sigma^2 d}\right)} \quad .$$

In particular, for $\sigma \geq \frac{2\eta}{\sqrt{d}}$ we get $\kappa_{\nu,\ell_2}(\eta) \leq e^{-\Omega(d)}$. The remaining of the proof follows the same argument as in Proposition 6. □

APPENDIX
ADDITIONAL EXPERIMENTAL RESULTS

We provide additional experiment results here. The interested reader can find a more expansive set of findings in [66], where we discuss in more detail the role of the released model hyperparameters, size, initialization, and gradient norm of the target on the reconstruction MSE.

A. Randomness from Released Model Initialization

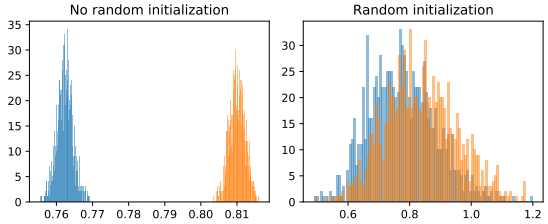
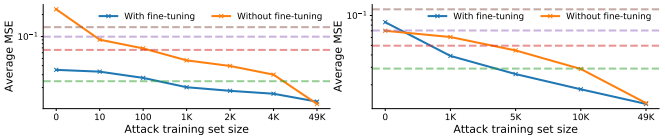


Fig. 10: On the MNIST dataset, given a target point z we train $1K$ released models with (blue) and without (orange) this point included in two settings: when each model is initialized with a new random seed, and when each model has the same initialization. We plot the distribution of losses on this target point in these two settings. Clearly, when there is no model randomization the distributions are perfectly separable and so membership is easy to infer, while in the random setting, the distributions nearly perfectly overlap implying membership may be more difficult.

As observed in Section VI-B, the attack will fail when the adversary does not have knowledge of the initial parameters of the released model and so must instantiate each shadow model used to train the attack with a new seed that controls the selection of initial parameters. We provide evidence that it may not be possible to perform a reconstruction attack in this setting by appealing to a simpler task of inferring membership, and demonstrating this problem is also difficult without knowledge of the initial parameters. We instantiate an informed MIA as described in Section II on the MNIST dataset. Specifically, given a target point z we train $1K$ released models with and without this point included (but with the same fixed set) in two settings: when each model is initialized with a new seed (differing initial parameters), and when each model is initialized with the same seed (identical initial parameters). In Figure 10, we plot the distribution of losses on this target point in these two settings. Clearly, when there is no initial parameter randomization the distributions are perfectly separable and so membership is easy to infer, while in the random setting, the distributions nearly perfectly overlap implying membership may be more difficult, if not impossible. Note that if released model training was fully deterministic, the distribution of losses on the target point in the setting with no random initialization would collapse to a point distribution. However, all our models are trained with JAX on GPUs that compile with non-deterministic reductions, introducing a small source of randomness [69].

B. Transfer Learning from a Reconstructor Network Trained on a Different Fixed Set



(a) MNIST, $|D_{\text{-}}| = |D'_{\text{-}}| = 10K$, leaving a maximum $49K$ shadow models.

(b) CIFAR-10, $|D_{\text{-}}| = |D'_{\text{-}}| = 5K$, leaving a maximum $49K$ shadow models.

Fig. 11: Fine-tuning the reconstruction network for a new target. The reconstruction network is initially trained to attack a released model trained with fixed dataset $D_{\text{-}}$, and then fine-tuned for a new released model trained with fixed dataset $D'_{\text{-}}$. Interestingly, the reconstructor network can do zero-shot learning on MNIST images, despite being trained on entirely separate data (i.e. $D'_{\text{-}} \cap D_{\text{-}} = \emptyset$).

Given a reconstructor network, ϕ, trained to attack released models of the form $\theta = A_{D_{\text{-}}}(z)$, can the adversary amortize the cost training a new ϕ' that aims to attack a released model $\theta' = A_{D'_{\text{-}}}(z)$, where $D'_{\text{-}} \cap D_{\text{-}} = \emptyset$? On both MNIST and CIFAR-10, in Figure 11 we show that fine-tuning the reconstructor ϕ on only a small number of shadow models can reach comparative performance to a reconstructor trained from scratch on substantially more data.

C. Adversary Knowledge of Starting Point: Initialization vs Near Convergence

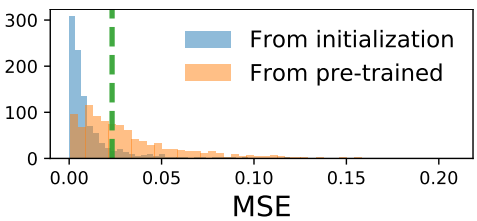
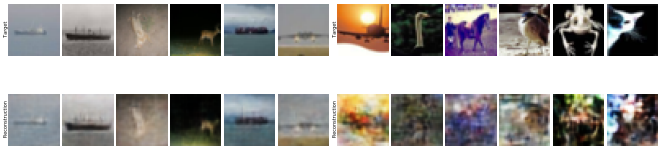


Fig. 12: A histogram of MSE for $1K$ released model targets for an adversary that observes the initial parameters compared to first observing a pre-trained released model near convergence. We also give the NN oracle for reference.

By default we assume the adversary knows the initial released model parameters, motivated by scenarios where the random seed used to generate initial parameters is made public or is leaked. Another motivating example is that of federated learning, where an adversary participates in the learning protocol. However, in such a setting, it is not guaranteed the adversary will observe a model at it's initial state. If the adversary is only included in the protocol after a sufficient number of time steps, the state at which they first observe released model parameters may be close to convergence. Here, we measure how reconstructions are affected by this subtle assumption. We pre-train a released model on $10K$ MNIST images (this model already achieves $> 92\%$ MNIST test set accuracy), and then following the experimental set-up reported in Section V on the remaining MNIST data, and compare to a released model in the standard setting where no pre-training occurs. Figure 12 shows the MSE for each 1K released model target in both settings. Clearly there is a difference in reconstruction fidelity that depends on the step at which the adversary first observes the released model parameters. A model that has nearly converged may be less dependent / not memorize it's newly seen training data, making reconstructions more challenging.

D. Visualization of Easy and Hard CIFAR-10 Reconstructions



(a) Smallest MSE (b) Largest MSE

Fig. 13: Example of the six smallest and size largest MSE reconstructions for CIFAR-10.

In Figure 13 we show the six reconstructed CIFAR-10 examples with smallest MSE and six examples with largest MSE out the $1K$ targets used for evaluation. The easiest targets to reconstruct correspond to structurally simple images with a constant background, while the most difficult often have complex background and color schemes.

E. Reconstructing Against a Released Model Trained with DP on CIFAR-10

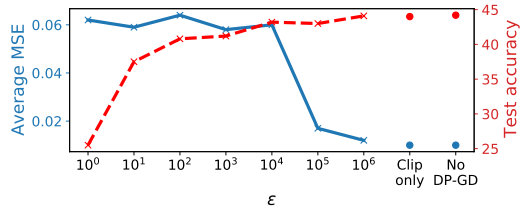


Fig. 14: Average MSE of reconstructions and test accuracy of released model using (ϵ, δ)-DP on the CIFAR-10 dataset.

We perform analogous DP experiments as in Section VI-D for CIFAR-10. Gradients are clipped to have a maximum ℓ_2 norm of 10, and Gaussian noise is added such that the model is $(\epsilon, \delta = 10^{-5})$-DP. In Figure 14 we see that again, a large ϵ in (ϵ, δ)-DP successfully mitigates against reconstruction attacks while preserving test accuracy in comparison to non-DP training.

F. Fine-Grained Analysis of CIFAR-10 Reconstructions over Released Model Training Epochs

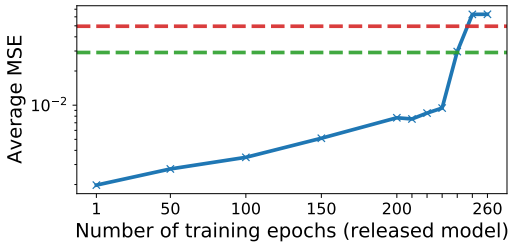


Fig. 15: How average MSE increases with the number of training epochs of the released model for CIFAR-10.

Reconstructing CIFAR-10 images is sensitive to the number of training epochs of the released model. We perform a fine-grained analysis to inspect at what epoch the attack becomes unsuccessful. This can be seen in Figure 15, where we plot average MSE over 1K released model targets as a function of the number of training epochs. MSE slowly increases with number of epochs up until approximately 240-250 epochs, at which point we observe that "reconstructability" undergoes a phase transition. Initially, we conjectured this was due to non-determinism from GPU training increasing the variance of shadow model parameters for a larger number of training epochs. However, when we implemented shadow model training in a deterministic set-up (using TPUs) we observed no difference in experimental outcomes. We leave a more in-depth investigation into the relationship between reconstruction success and number of training epochs for future work.

G. ReLU Activations in Released Model

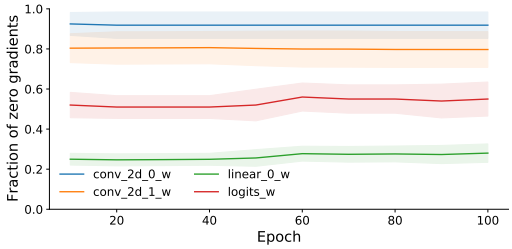


Fig. 16: Evidence on CIFAR-10 reconstruction task that ReLU activations make reconstruction attacks harder. For the target, z, we plot $\frac{\partial \ell(z)}{\partial \theta}$ for each layer in the released model θ, throughout training. A large fraction of these gradients are zero, implying less influence of this additional point on the trained model, in comparison to other activations that have non-zero gradients everywhere.

TABLE III: Comparison of reconstructions for different released model activations on MNIST. Please refer to [70] for a description of each activation function.

Activation	Average MSE over 1K test targets
ReLU	0.0182
$\max(-0.5, x)$	0.0096
ELU	0.0089
Sigmoid	0.0085
Softplus	0.0083
Swish	0.0091
Leaky ReLU	0.0092
Tanh	0.0086
CELU	0.0077
SELU	0.0083
GELU	0.0088
Identity	0.0085

As we saw in Section VI-B, released models with ReLU activations tend to be harder to attack in comparison to other activation functions with non-zero gradients almost everywhere, and result in poor quality reconstructions (an MSE larger than the NN oracle distance). We conjecture that this is caused by a large fraction of parameters receiving zero gradients at each step of training, thereby diminishing the mutual information shared between model parameters and the unknown target training point. In Figure 16, for each layer of the released model, we show the fraction of parameters that received zero gradient when computing the loss of the unknown training point. Over 80% of the parameters in the convolutional layers have zero gradients. Additionally, in Table III we compare reconstructions against released models that employ different activation functions, and find that ReLU remains the outlier. Note that we also reconstruct against a released model that uses a modified version of ReLU that has zero gradient for $x < -0.5$, and find that allowing a small negative signal is enough to reach parity with reconstruction MSE on smooth activations or activations that contain a non-zero signal almost everywhere.

TABLE IV: Experimental setup.

		MNIST	CIFAR10
Data	Resolution	28×28 (grayscale)	32×32 (RGB)
	Size	$70K$	$60K$
	Fixed size	$10K$	$5K$
	Shadow size	$59K$	$54K$
	Test targets	$1K$	$1K$
$\theta, \bar{\theta}$	Type	MLP	CNN
	Architecture	1-hidden layer, width 10	Table V
	Activations	ELU	ELU
	Parameters	$8K$	$55K$
ϕ	Type	MLP	Transposed CNN
	Architecture	2-hidden layers, width $1K$	Table VI
	Activations	ReLU	ReLU
	Parameters	$9.7M$	$226M$
A	Algorithm	GD+Momentum	GD+Momentum
	Loss	Cross-entropy	Cross-entropy
	Learning rate	0.2	0.01
	Momentum	0.9	0.9
	Epochs	100	100
R	Algorithm	RMSProp	Adam
	Loss	MAE+MSE	+LPIPS+Discriminator
	Learning rate	0.001	0.0001
	Weight decay	0	0.0001
	Batch size	128	128
	Epochs	100	1000

TABLE V: CIFAR-10 released model, θ.

Layer	Parameters
Convolution	16 filters of 4×4, strides 2
Convolution	32 filters of 4×4, strides 1
Fully connected	10 units
Softmax	10 units

TABLE VI: CIFAR-10 reconstructor network, ϕ.

Layer	Parameters
Fully connected	4096 units
Reshape	64×64
Transposed convolution	32 filters of 5×5, strides 2
Transposed convolution	3 filters of 5×5, strides 2

TABLE VII: CIFAR-10 attack PatchGAN Discriminator model.

Layer	Parameters
Convolution	64 filters of 4×4, stride 2
Convolution	128 filters of 4×4, stride 2
Convolution	256 filters of 4×4, stride 2
Convolution	512 filters of 4×4, stride 1
Convolution	1 filter of 4×4, stride 1