Vulnerability of Transfer-Learned Neural Networks to Data Reconstruction Attacks in Small-Data Regime

Tomasz Maciazek
School of Mathematics
University of Bristol
tomasz.maciazek@bristol.ac.uk

Robert Allison
School of Mathematics
University of Bristol

Abstract

Training data reconstruction attacks enable adversaries to recover portions of a released model's training data. We consider the attacks where a reconstructor neural network learns to invert the (random) mapping between training data and model weights. Prior work has shown that an informed adversary with access to released model's weights and all but one training data point can achieve high-quality reconstructions in this way. However, differential privacy can defend against such an attack with little to no loss in model's utility when the amount of training data is sufficiently large. In this work we consider a more realistic adversary who only knows the distribution from which a small training dataset has been sampled and who attacks a transfer-learned neural network classifier that has been trained on this dataset. We exhibit an attack that works in this realistic threat model and demonstrate that in the small-data regime it cannot be defended against by DP-SGD without severely damaging the classifier accuracy. This raises significant concerns about the use of such transfer-learned classifiers when protection of training-data is paramount. We demonstrate the effectiveness and robustness of our attack on VGG, EfficientNet and ResNet image classifiers transfer-learned on MNIST, CIFAR-10 and CelebA respectively. Additionally, we point out that the commonly used (true-positive) reconstruction success rate metric fails to reliably quantify the actual reconstruction effectiveness. Instead, we make use of the Neyman-Pearson lemma to construct the receiver operating characteristic curve and consider the associated true-positive reconstruction rate at a fixed level of the false-positive reconstruction rate.

1 Introduction

Machine Learning (ML) models are known to carry certain imprints of their training data [3, 44, 10, 46, 27]. If the training data contains sensitive information, it is important to train the corresponding ML model in a privacy-preserving way. Governments have recognised training data privacy risks as a crucial issue in the development of ML/AI systems and mention membership inference [44] and model inversion [20, 19] attacks explicitly in their official documents [15, 50]. Intuitively, the capacity of an ML model to memorise its training data grows with the complexity of the model and with the number of its parameters. In particular, neural networks have been shown to memorise global properties of their training data [21, 4], single datapoints [6] or even entire training datasets [24, 8, 35]. In this work we study training data reconstruction attacks that exploit this memorisation effect in neural networks.

One can order different types of attacks according to the amount of information they can extract from the attacked model. A membership inference attack (MIA) [44, 9] aims to predict whether a given data point was in the model's training set. If successful, it extracts only a single bit of information

Preprint. Under review.

about the training data. On the other end of the spectrum we have reconstruction attacks that extract an entire single datapoint [6] or large chunks of the training dataset [24, 8, 35]. Thus, a successful training data reconstruction attack requires leakage of many bits of information. In this paper we are concerned with the task of reconstructing several examples from the training dataset. We consider situations where one does not necessarily want to go as far as protecting against MIA but do want to protect against training data reconstruction.

Differential privacy (DP) [41] has become a popular tool for training ML models in a way that protects the individual records of a dataset (as opposed to the global information about the data which is not protected by DP [21]). It allows one to state formal privacy guarantees quantified in terms of the *privacy budget* denoted by $\epsilon \geq 0$. One of the most popular algorithms for DP-training of neural networks is the differentially private stochastic gradient descent DP-SGD [1]. In order to decide what value of ϵ is sufficient to defend against a given threat model, one needs to know how the DP-SGD guarantees translate to the success probability of the attack at hand. This can be formulated in terms of suitable tight upper bounds (called the reconstruction robustness bounds) for the success of any attack on a DP-SGD trained neural net under a given threat model. Recent work [6, 25] has provided tremendous advancements in this area under a strong threat model of an *informed adversary*. An informed adversary has access to intermediate gradients computed during DP-SGD and knowledge of all the training data except one data point which is the target of the reconstruction attack. Reconstruction robustness bounds have been derived in this setup [25] and shown experimentally to be nearly saturated by a certain gradient-based inversion attack (see [25] and references therein for mode details). In the more realistic *hidden state threat model* [17, 5, 54, 2, 11] the informed adversary does not have access to the intermediate gradients. There, the appropriate attack is the model-based attack that only uses the released weights of the model. The reconstruction success of the model-based attack is not well reflected by theoretical reconstruction robustness bounds derived for the threat model that assumes access to the intermediate gradients [6]. In other words, experimental evidence suggests that much lower levels of privacy are sufficient to defend against the model-based inversion attack in the hidden state threat model. However, derivation of suitably tight DP-SGD privacy guarantees and thus tight reconstruction robustness bounds in the hidden state threat model for neural networks is an open problem [2, 11]. In this work, we consider an even more realistic threat model (specified in detail below) which assumes the knowledge of the released model's weights and the distribution from which the training data has been sampled (rather than specific data points). Thus, we anticipate the existence of an even larger gap between the upper bounds for reconstruction success that are optimal for our threat model and the theoretical analysis from [25]. In realistic threat models theoretical analysis may not prove feasible, thus in order to draw practical conclusions about using DP in such cases one often needs to turn to empirical estimation of privacy leakage and robustness of DP-SGD [28, 37, 11].

A crucial practical aspect of any implementation of DP is the privacy-utility tradeoff [41]. Namely, achieving lower ϵ requires using more noise in a given DP algorithm which in turn degrades utility. The privacy-utility tradeoff is particularly severe when applying DP to small datasets [41]. This is also the case in DP-SGD [1, 49]. Our work extends these observations and shows that image classifiers transfer-learned on datasets containing few examples per class can suffer accuracy degradation of $10 - 30$ percentage points when trained with DP-SGD strong enough to defend against our data reconstruction attack. This is of particular relevance when working with data with significant class imbalances [12, 42, 51] where applying the common downsampling and reweigthing techniques [43, 7] can often lead to effective small training datasets containing few examples per (minority) class or tens of examples in total.

Our Threat Model (Weak Adversary) The adversary has access to the released model and their aim is to reconstruct some examples from the training dataset. Note that successful reconstruction of just a single data point can already be considered a privacy breach. More specifically, we assume the following threat model in which (only) the following hold.

- (A.1) Model's architecture and released parameters θ are known.
- (A.2) Model's training algorithm $\mathcal{A}$ is known.
- (A.3) Prior distribution, π, from which the training data has been sampled and the size of the training set N are known.

We call an adversary working within the framework of (A.1-3) the *weak adversary* in contrast to the *informed adversary* that has been considered in previous works [6, 25]. In the context of this

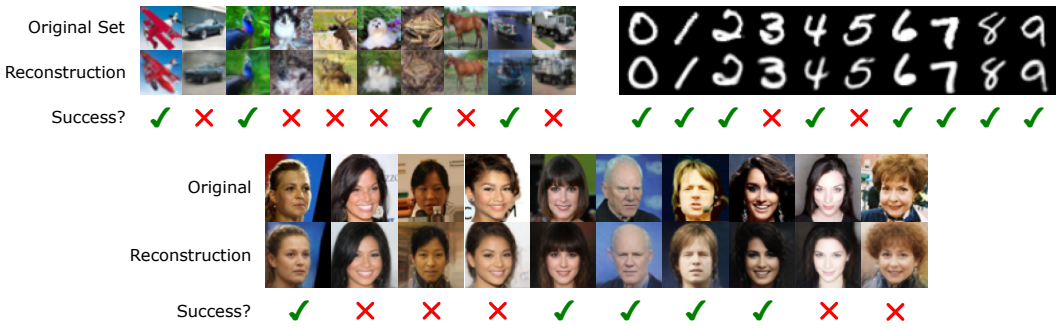


Figure 1: Examples of reconstructions obtained via our attack with CIFAR-10 [32], MNIST [16] (reconstructed in the 32×32-resolution) and CelebA [34] (reconstructed in the 64×64-resolution) datasets. The reconstruction "success" is determined via the nearest-neighbour MSE threshold criterion τ as defined in Section 2. Classifier training set size $N = 10$.

paper, our assumptions effectively mean that the trained model and its transfer-learning code is public. In particular, the adversary knows all the training hyper-parameters used in the transfer-learning process (number of epochs, learning rate, weight decay, batch size, the initialization distribution of the weights, etc.). However, we *do not* assume the knowledge of the random seed that has been used to initialize model's weights or to sample mini-batches during training. We also *do not* assume the knowledge of any gradients that were computed during the training of the model. We argue that this is indeed a realistic setting, since it is a widespread practice to publish trained models, but the exact random seeds used during the training or the computed intermediate gradients are often not known. This is especially common in the transfer-learning scenarios where the base model's architecture is typically one of the several standard and widely used architectures that has been pre-trained on a generic and public dataset and whose weights are publicly available. During the transfer-learning typically only the top layers of the pre-trained model are replaced and are usually initialized in a standardized way (e.g. using random normal, Xavier [22] or Kaiming initialization [26]). Thus, even without access to the exact model architecture or some of the training process details the adversary may need to explore only several likely possibilities via trial and error following generic published methods of transfer-learning (e.g., [13]). The size of the training set and the sizes of the classes being a global property of the data can be determined via an auxiliary property inference attack which cannot be defended against by DP-SGD [21].

Contributions

- We design a reconstruction attack that is able to retrieve examples from the training data under a realistic threat model of the weak adversary (see Section 3 and reconstruction examples in Fig. 1).
- In transfer-learning scenarios with small datasets we show experimentally that DP-SGD cannot defend against our attack without significant degradation in model's utility (see Section 4). This is in contrast to earlier reconstruction attack methods and therefore raises new significant concerns about the use of such models when protection of training data is paramount.
- We propose to measure the robustness against a reconstruction attack by considering true- and false-positive reconstruction rates. We use the receiver operating characteristic (ROC) curves to propose a test that can be used to practically evaluate a model's security (see Section 2).

Comparison with Prior Work We study the problem of (partially) inverting the mapping between the training set and trained model's weights. This is different than model inversion [19, 53] which reconstructs model's input from its output. The work [6] studies a similar problem to the one considered here by training a reconstructor NN. However, as explained above and specified in Section 3.2, our attack works under a more realistic threat model and under much more general conditions. Our attack relies on a reconstructor NN giving reconstructions that are semantically close to the original images as opposed to the recently proposed gradient-flow based attacks [24, 8, 35] which reconstruct the training data numerically as solutions to a certain algebraic minimization problem. Our attack is more flexible than the gradient-flow based methods which only work when the attacked model has been trained in a very specific way. For instance, they require long training with very

low learning rates until good (approximate) convergence to a critical/KKT point (we do not need any of these assumptions). As such, the gradient-flow based attacks will be disrupted even by small amounts of noise in DP-SGD. The importance of the true- and false-positive attack success rates and ROC curves to evaluating deep learning privacy has been emphasized in the context of MIA [9, 33], however it has not been considered in data reconstruction attacks, especially in realistic threat models such as the weak adversary model. A hypothesis testing approach (different from ours) has been recently used to derive tighter reconstruction robustness bounds in the informed adversary model [29].

2 Reconstruction Robustness Measures

We argue that in order to evaluate the effectiveness of a given reconstruction attack (or a model's robustness against these attacks) one needs to report both true- and false-positive reconstruction rates. The key object containing this information is the ROC curve corresponding to the attack at hand.

Notation Let $\mathcal{Z} \subset \mathbb{R}^d$ be the domain where the training data lives. We denote by π be the prior distribution of the training data over $\mathcal{Z}$ and by π_N the product-prior over $\mathcal{Z}^N$. The reconstruction error function will be denoted by $\ell : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}_{\geq 0}$. The weights of the neural network model will be denoted by $\theta \in \Theta$, where Θ is the range of a randomised training mechanism $M : \mathcal{Z}^N \to \Theta$.

Definition 1 (Reconstruction TPR and FPR)**.** *Let $\mathcal{R} : \Theta \to \mathcal{Z}$ be a reconstruction attack. For $\tau \geq 0$ define the true-positive reconstruction rate $TPR_\tau(\mathcal{R})$ and the false-positive reconstruction rate $FPR_\tau(\mathcal{R})$ with respect to π and ℓ as*

$$TPR_\tau(\mathcal{R}) := \mathbb{P}_{(Z_1,\ldots,Z_N)\sim\pi_N,\theta\sim M(Z_1,\ldots,Z_N)}\left[\min_{1\leq i\leq N} \ell\left(Z_i, \mathcal{R}(\theta)\right) \leq \tau\right], \quad (1)$$

$$FPR_\tau(\mathcal{R}) := \mathbb{P}_{(Z'_1,\ldots,Z'_N)\sim\pi_N,(Z_1,\ldots,Z_N)\sim\pi_N,\theta\sim M(Z_1,\ldots,Z_N)}\left[\min_{1\leq i\leq N} \ell\left(Z'_i, \mathcal{R}(\theta)\right) \leq \tau\right]. \quad (2)$$

Many prior data reconstruction attacks use incomplete evaluation methodology that only considers TPR – see the definition of reconstruction robustness in [6, 25] (note that the definition differs from ours due to the differences in the threat models). Unfortunately, in the context of this work such a definition of reconstruction robustness becomes less and less meaningful when N is large – it becomes easier and easier to devise a data reconstruction attack whose TPR approaches 1. To see this, consider the following *baseline attack* $\mathcal{R}_0$. For a given error threshold $\tau > 0$ let $Z_0 \in \mathcal{Z}$ be such that $\kappa_\tau(Z_0) := \mathbb{P}_{Z\sim\pi}\left[\ell(Z, Z_0) \leq \tau\right] > 0$. Assume also that the error threshold is chosen to be non-trivial in the sense that $\sup_{Z\in\mathcal{Z}} \mathbb{P}_{Z'\sim\pi}\left[\ell(Z', Z) \leq \tau\right] < 1$. The baseline attack is defined as $\mathcal{R}_0(\theta) = Z_0$ for any input weights θ. We immediately find that $TPR_\tau(\mathcal{R}_0) = 1 - (1 - \kappa_\tau(Z_0))^N$. Thus, when the amount of training data is large we have $TPR_\tau(\mathcal{R}_0) \xrightarrow{N\to\infty} 1$. In other words, the baseline attack can achieve great reconstruction TPR. However, it is not retrieving any information about the actual training set since it is merely returning an arbitrary member of $\mathcal{Z}$ – its *false-positive* reconstruction rate (FPR) tends to 1 as well. This shows that in order to make meaningful conclusions about the effectiveness of a reconstruction attack we need to consider both the true-positive and the false-positive reconstruction rates. For a general reconstruction attack, R, we define its FPR as the rate at which we obtain coincidental successful reconstructions $\mathcal{R}(\theta)$ obtained from weights θ coming from a different training set drawn from π_N. One can further formalize this idea by defining a modified version of the membership inference security game [9], see Appendix A.

Intuitively, FPR measures how good the attack R is at guessing successful reconstructions. Thus, from the perspective of the adversary it is desirable to achieve low FPR. The ROC curve shows how TPR changes as a function of FPR when varying τ. A relevant part of the ROC curve from the point of view of an adversary is the one that describes low FPR values e.g., $FPR < 0.01$. An adversary might reliably quantify their attack by finding the threshold τ for which $FPR = 0.01$ and calculate the corresponding TPR. This fact has also been emphasized in the context of MIA [9, 33]. In Appendix A.1 we compare the so-obtained ROC curves with the ones computed via the likelihood-ratio test according to the Neyman-Pearson hypothesis testing lemma [38] and show that in our experiments the both methods can yield extremely similar ROC curves. Since the Neyman-Pearson ROC curves describe the uniformly most powerful hypothesis test, results from Appendix A.1 justify our proposed method as being effectively near-optimal. We identify the reason why this

happens by finding a theoretical condition that, if satisfied exactly, causes the both ROC curves to be identical. In Appendix A.1 we also compare our attack against a simple attack that just draws the reconstructions randomly from the prior distribution π.

Selecting the nearest-neighbour threshold value τ Following other works [6, 25], we choose the error function $\ell = \ell_{MSE}$ to be the mean-squared error (MSE, averaged over the pixels and the image channels). To calculate an illustrative threshold τ which we call the *nearest-neighbour threshold* we: 1) split the entire dataset into train- and validation subsets, 2) for each element of the validation set find its nearest neighbour in the training set w.r.t. the MSE distance and 3) τ is the average nearest-neighbour MSE distance. Each image is normalized as $X \to X/127.5 - 1$. Our calculated thresholds τ were 0.0696, 0.1284 and 0.1217 for MNIST, CIFAR-10 and CelebA respectively. Unless stated otherwise, all the FPR and TPR values reported in this work are calculated using the nearest-neighbour threshold values. Crucially, one can lower our attack's FPR by varying τ and improve the TPR/FPR ratio, albeit at the expense of reducing the magnitude of TPR – this is illustrated by Table 1 and the ROC curves reported in Appendix A.

3 Our Reconstruction Attack and its Robustness

Transfer Learning Setup In this work, we aim to attack a neural network model (image classifier) that has been trained using the transfer learning technique [40]. More specifically, we consider the transfer learning where: 1) a base model is pre-trained to perform a certain general classification task on a large training dataset and 2) the pre-trained model is subsequently fine-tuned to do a different classification task where the available training dataset is much smaller. The fine-tuning that we apply relies on replacing the output layer of the pre-trained model with new layer(s) and training only the new layer(s) with the weights of all the other layers being frozen. In other words, the new layers are trained using the robust features that the model has leaned during the pre-training. We emphasize that in the transfer-learning scenarios the assumptions (A.1-3) describing the weak adversary refer to the transfer-learning data and training algorithm – we are not interested in reconstructing the data used in the pre-training step. In order to emulate such a situation, we set up the following experiments.

1. **MNIST** We pre-train a scaled-down VGG-11 neural net [45] on the EMNIST-Letters [14] classification task: $128K$ data, 26 classes, test accuracy of 94.6%. We transfer this base model to MNIST-classification [16] (10 classes) by replacing the 26 output neurons of the base model with 10 new neurons.
2. **CIFAR** We pre-train EfficientNet-B0 [47] on the CIFAR-100 [32] classification task: $50K$ data, 100 classes, test accuracy of 85.9%. We transfer this base model to CIFAR-10 classification by replacing the 100 output neurons of the base model with 10 new neurons.
3. **CelebA** We pre-train WideResNet-50 [56] on the CelebA [34] attributes classification task: $200K$ data, 40 binary attributes, test accuracy of 92.0%. We transfer this base model to a binary face recognition task (also using CelebA) by replacing the 40 output neurons of the base model with a two-layer NN of the size $4 - 1$ i.e., 4 neurons connected to 1 output neuron.

The transfer-learning dataset sizes that we consider in the above experiments range from $N = 10$ to $N = 40$. In the MNIST and CIFAR experiments the class sizes in the training set were balanced, while in the CelebA experiment we emulate transfer learning with unbalanced data – the positive class constituted only 10% of the training set. We use the cross-entropy as the training loss (with appropriate class reweighting in the unbalanced case). Care was taken to ensure that the transferred models attained high accuracy - see Appendix D.2.

3.1 Our Reconstruction Attack

We demonstrate that in the transfer learning setup it is indeed possible to reconstruct some of the training datapoints from the transfer-learning dataset with reconstruction TPR between $35 - 65\%$ at FPR between $1 - 10\%$, depending on the experiment (see Table 1 and examples in Fig. 1).

Reconstructor NN and the Shadow Training Method The reconstructor NN is trained using the shadow training method [44] that relies on creating a large number of "shadow models" that mimic the model to be attacked, see Algorithm 1. We split each dataset $\mathcal{D}$ (MNIST, CIFAR-10, CelebA) into disjoint $\mathcal{D}_{train}$ and $\mathcal{D}_{val}$ subsets and apply Algorithm 1 to the two splits separately to obtain

Table 1: True- and false-positive reconstruction rates for the MNIST, CIFAR and CelebA experiments depending on the training set size N of the attacked model. Asterisks * mark values calculated w.r.t. the nearest-neighbour MSE threshold, see Section 2.

	$N = 10$			$N = 40$		
Experiment	TPR^*	FPR^*	$TPR@FPR = 0.01$	TPR^*	FPR^*	$TPR@FPR = 0.01$
MNIST	0.636	0.018	0.431	0.374	0.098	0.052
CIFAR	0.388	0.063	0.217	0.580	0.383	0.045
CelebA	0.441	0.012	0.385	0.341	0.102	0.084

Algorithm 1 Shadow model training

Input: Large dataset $\mathcal{D}$ sampled from data prior distribution π, classifier training set size N, weight initialization distribution π_Θ, transfer-learning algorithm $\mathcal{A}$, number of shadow models N_{shadow}.
Initialize: $\mathcal{D}_{shadow} \leftarrow [\,]$ as an empty list.

1: **for** $k = 1$ `to` N_{shadow} **do**
2: `Sample the training dataset` $\mathcal{D}_k$ `from` $\mathcal{D}$, $\#\mathcal{D}_k = N$.
3: `Initialize the weights and biases of the trainable layers` $\theta_0 \sim \pi_\Theta$.
4: `Train the trainable layers` $(\theta_0, \mathcal{D}_k) \xrightarrow{\mathcal{A}} \theta_k$.
5: `Append the list` $\mathcal{D}_{shadow} \leftarrow \mathcal{D}_{shadow} + [(\theta_k, \mathcal{D}_k)]$.
6: **end for**

Output: Shadow dataset $\mathcal{D}_{shadow}$.

the disjoint shadow datasets $\mathcal{D}^{train}_{shadow}$ and $\mathcal{D}^{val}_{shadow}$. The reconstructor NN is trained on $\mathcal{D}^{train}_{shadow}$ and tested on $\mathcal{D}^{val}_{shadow}$. It takes as input the (flattened) trained weights and biases of the trainable layers of the attacked model and outputs a single image. In MNIST and CIFAR experiments we use the conditional reconstructor NN whose input consists of the trained parameters concatenated with the one-hot encoded class vector that determines the class of the output image. This way, we can reconstruct one image per class from the same training set.

The reconstructor NN can be viewed as an image generator if one treats the trained weights as random vectors in a high dimensional latent space. Thus, we can successfully borrow generator architectures that have been developed for generative adversarial networks. We have found that our minor modification of the deep convolutional residual architecture provided by [23, 52] has worked well for the purposes of this work. We describe this architecture in more detail in Appendix B.

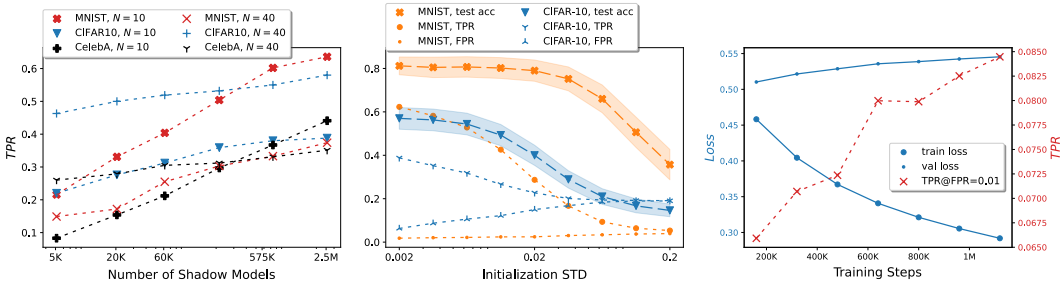
In the experiments we train the image classifiers on balanced training sets i.e., $N = C \cdot M$, where C is the number of classes. In the CIFAR and MNIST experiments M is the (fixed) number of examples per image class and $C = 10$. In the CelebA experiment the models are binary classifiers trained to recognize the face of a given person. There, the training set of a classifier consists of M images of that person and $9M$ images of other randomly selected people. The selected identities vary between the shadow models. In each experiment we prepare $N^{train}_{shadow} = 2.56 \times 10^6$ training shadow models and $N^{val}_{shadow} = 10^3/10^4$ validation shadow models for the CIFAR, MNIST/CelebA experiments respectively. We subsequently calculate the trained weights statistics

$$\overline{\theta} = \frac{1}{N^{train}_{shadow}} \sum_{k=1}^{N^{train}_{shadow}} \theta_k, \quad \mathrm{Cov}_\theta = \frac{1}{N^{train}_{shadow}} \sum_{k=1}^{N^{train}_{shadow}} \left(\theta_k - \overline{\theta}\right) . \left(\theta_k - \overline{\theta}\right)^T,$$

It is important to normalize the input of the reconstructor NN as $\theta_{k,i} \to (\theta_{k,i} - \overline{\theta}_i)/\sqrt{[\mathrm{Cov}_\theta]_{i,i}}$ due to the possible variation in the trained weights' magnitude. In estimating FPR of the reconstructor NN we approximate the sampling from the distribution of $\theta \sim M(Z_1, \ldots, Z_N)$ with $(Z_1, \ldots, Z_N) \sim \pi_N$ from Definition 1 by sampling from a multivariate random normal distribution. This is done by feeding the reconstructor NN with weights sampled from $\mathcal{N}(0, \widetilde{\mathrm{Cov}_\theta})$, where $\widetilde{\mathrm{Cov}_\theta}$ is the covariance of the normalized weights. For the precise description of TPR and FPR estimation procedures, see Algorithm 2 and Algorithm 3 in Appendix A.

Reconstructor NN Loss Function Let $(\mathcal{D}_k, \theta_k) \in \mathcal{D}_{shadow}$ and let $\mathcal{D}^c_k = \{Z^c_1, \ldots, Z^c_M\}$ be the subset of $\mathcal{D}_k$ consisting of images of the class $c \in \{0, \ldots, C-1\}$. Given a reconstruction $R(\theta_k)$ we

Figure 2: Examples of false-positive reconstructions based on the nearest-neighbour MSE threshold for CelebA and MNIST. Shadow model training set size $N = 40$.



a) b) c)

Figure 3: a) Reconstruction attack TPR as a function of the number of shadow models. b) Influence of weight initialization STD on reconstruction TPR, FPR and attacked classifier accuracy for the training dataset size of $N = 10$. c) Training and validation loss and TPR changes at $FPR = 0.01$ during the reconstructor NN training in the CelebA experiment with $N = 40$. Note: a) and b) use the nearest-neighbour MSE threshold values given in Section 2.

define the reconstruction loss of the conditional reconstructor (for CelebA we always take $c = 1$, the class corresponding to one specific individual) via the following *softmin* function.

$$loss_{rec}\left(\mathcal{D}_k^c, R(\theta_k)\right) = \frac{\sum_{i=1}^{M} \ell_i \exp\left(-\alpha\, \ell_i\right)}{\sum_{i=1}^{M} \exp\left(-\alpha\, \ell_i\right)}, \quad \alpha > 0, \tag{3}$$

where $\ell_i = \left(\ell_{MSE}\left(Z_i^c, R(\theta_k)\right) + \ell_{MAE}\left(Z_i^c, R(\theta_k)\right)\right)/2$ with ℓ_{MAE} being the mean absolute error. In the CIFAR and CelebA experiments we also add the LPIPS loss [58] to ℓ_i (also used in [6]). In our experiments we have worked with $\alpha = 100$. For the Reconstructor NN training we use Adam optimizer [30] with the learning rate 0.0002 and batch size 32. Early stopping after about 10^6 gradient steps helps prevent overfitting. This takes up to 72 hours to train on a GeForce RTX 3090 GPU.

Examples of reconstructions together with TPR and FPR are presented in Table 1 and in Fig. 1. Examples of false-positive reconstructions are shown in Fig. 2. For more reconstruction examples and further technical details for each of the discussed experiments see Appendix C and Appendix D.

3.2 Factors Affecting Reconstruction

Our attack remains robust in a wide range of circumstances which we describe in detail below.

1) Number of Shadow Models. To obtain high reconstruction TPR it is necessary to produce as many shadow models as possible given the computational resources at hand – see Fig. 3a. It is relatively easy to produce several millions of shadow models – this task is massively parallelizable and training a single shadow model is fast.

2) Attacked Model Training Set Size. If the number of images in the classifier's training set grows, the reconstruction FPR of our attack becomes higher (see Table 1). Intuitively, this means that it becomes easier for the reconstructor to "guess" a close match to one of the target images by outputting a generic representative of the target class. We have noticed that this can be mitigated to some extent by training the reconstructor for longer as measured by our attack's TPR at $FPR = 0.01$ (see Fig. 3c). However, training for too long may reduce TPR due to overfitting, especially when there is not enough shadow models to train on or when the image dataset is small and there are many overlaps between the shadow training datasets.

3) Conditional/Non-conditional Reconstructor NN. For the same reasons as in point 2) above non-conditional reconstructor NNs have higher FPR. In the CIFAR experiment with the non-conditional

reconstructor and $N = 10$ we got reconstruction $TPR = 54.3\%$ and $FPR = 28.1\%$. As with GANs, the non-conditional reconstructor is susceptible to the mode collapse - its output images tend to be generated from only one or two of the classes. Thus, conditional reconstructor NNs can retrieve much more information about the training dataset.

4) Reconstructor conditioning mismatch. We can successfully reconstruct training data when the reconstructor is conditioned on a different set of classes than the attacked classifier is trained to distinguish. We have attacked binary classifiers (even/odd for MNIST and animal/vehicle for CIFAR-10) with reconstructors conditioned on ten classes. We have obtained $TPR = 46.1\%$, $FPR = 1.8\%$ for MNIST and $TPR = 28.2\%$, $FPR = 9.5\%$ for CIFAR-10 with the training set size $N = 10$.

5) Training Algorithm $\mathcal{A}$: SGD/Adam. Results from Table 1 concern attacks on classifier models trained with SGD. We can also successfully attack models trained with Adam [30], although with slightly lower success. We have obtained reconstruction $TPR = 44.3\%$, $FPR = 2.2\%$ for MNIST and $TPR = 36.2\%$, $FPR = 11.1\%$ for CIFAR with training set size $N = 10$. We have not observed significant changes in reconstruction TPR and FPR when comparing full-batch SGD/Adam vs. mini-batch SGD/Adam with batch size $B = N/2$.

6) Attacked Model Weight Initialization. We initialize the weights of the classifier according to i.i.d. normal distribution $\mathcal{N}(0, \sigma^2)$ and initialize the biases to zero. Unless stated otherwise, we choose $\sigma = 0.002$ for the MNIST and CIFAR experiments and $\sigma = 0.0002$ for the CelebA experiments. Fig. 3b shows how the reconstruction TPR, FPR and classifier accuracy depend on σ for the training dataset size of $N = 10$. Higher values of σ decrease reconstruction TPR and increase FPR eventually destroying our attack. However, in order to completely prevent our reconstruction attack one needs to use values of σ that result in suboptimal classifier training and highly reduced accuracy. In contrast, reconstruction attacks in the threat model of the informed adversary require access to the exact initialized weights and fail otherwise [6].

7) Attacked Model Underfitting/Overfitting. We have trained the attacked models for 1, $32/48$ (optimal) and 512 epochs in the MNIST/CIFAR experiments respectively with training set size $N = 10$. We have not noticed any significant differences in the corresponding reconstruction success rates.

8) Out-of-distribution (OOD) Data. CIFAR-100 is often used for OOD benchmark tests for CIFAR-10 [18]. To study the effectiveness of the reconstructor NN trained on OOD data, we have trained the conditional reconstructor NN on shadow models (classifiers) trained on $N = 10$ CIFAR-100 images that were randomly assigned the class labels $c \in \{0, 1, \ldots, 9\}$. We have subsequently used such a reconstructor to attack classifiers that were trained on CIFAR-10 images (with original labels) obtaining reconstruction $TPR = 17.1\%$ and $FPR = 4.4\%$. This shows that our attack can be effective even with limited information about the prior training data distribution π.

4 Our Reconstruction Attack Under DP-SGD

DP-SGD adds random noise to model's gradients during training to provably protect the training data against reconstruction [1]. As we explained in the Introduction, DP-SGD privacy guarantees and the related reconstruction robustness privacy guarantees are derived under a much stronger threat model than the one considered in this work. In this work we study situations where one aims to defend against the (realistic) weak adversary defined in the Introduction. Thus, one can expect that a much higher privacy budget ϵ and hence less noise will be sufficient to defend against our attack. In this section, we show that despite these relaxed privacy requirements, successful defence against the weak adversary in the small-data regime requires using very noisy DP-SGD training which in turn significantly degrades trained model's utility. This effect is known as the privacy-utility tradeoff [41]. In Fig. 4 we show the privacy-utility tradeoff and summarize the effectiveness of our reconstruction attack in the MNIST and CIFAR experiments. The setup was the same as in Section 3, except that the models were trained with mini-batch DP-SGD with Gaussian noise and gradients clipped to a given maximum l_2-norm, C. The trained models were (ϵ, δ)-differentially private with $\delta = N^{-1.1}$ as recommended in [41]. We have made sure that the DP-SGD training hyper-parameters were chosen to have as small an impact on trained model's accuracy as possible. To this end, we have followed the hyper-parameter selection procedure outlined in [41]. In particular, because of the way noise is accumulated in DP-SGD it is beneficial to use the mini-batch size, B, which is as large as possible [41, 1]. We have worked with $B = N - 1$ i.e., the sampling probability of $q = (N - 1)/N$, which is the largest possible while still making use of the privacy amplification enabled by the mini-batch

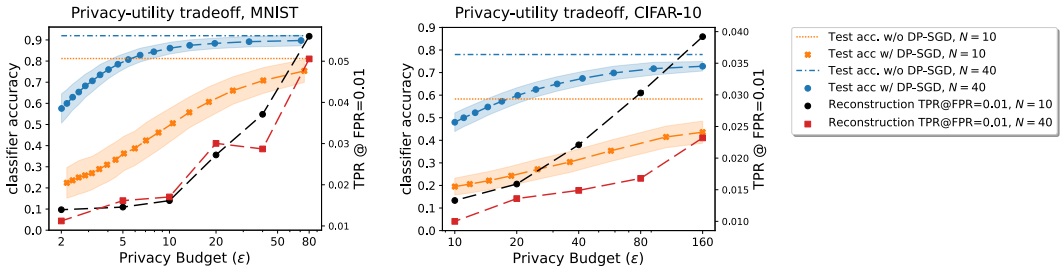


Figure 4: Privacy-utility tradeoff for the MNIST and CIFAR experiments. All models suffer severe accuracy degradation even for relatively large values of ϵ – see discussion below.

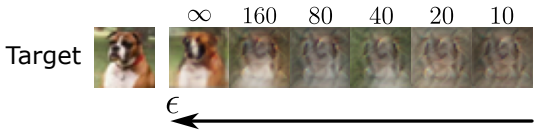


Figure 5: Reconstruction example for CIFAR-10 and (ϵ, δ)-DP models for varying values of ϵ and training set size $N = 10$. Training without DP-SGD means $\epsilon = \infty$.

Poisson sampling [1]. The optimal clipping norms C were found to be 4.0 and 1.2 for the MNIST and CIFAR experiments respectively. The other training details are specified in Appendix E.

The TPR-criterion at low FPR To find the value of ϵ which is sufficient to defend against our reconstruction attack, we look at TPR at low FPR (the "low" value of FPR is somewhat arbitrary – here, we take $FPR = 0.01$). As plots in Fig. 4 indicate, the TPR drops when ϵ grows. Thus, to decide if our attack has been defended against, one can select a threshold $\gamma > 0$ and test the criterion $TPR \leq \gamma$ at $FPR = 0.01$. Smaller γ means stricter defence criterion. In our experiments we have found that $\gamma = 0.015$ is sufficient to remove most of the details of the original data from the reconstructions. By interpolating data from Fig. 4, this translates to the maximal privacy budget of $\epsilon \lesssim 7.2$ and $\epsilon \lesssim 6.0$ for MNIST with $N = 10$ and $N = 40$ respectively. For CIFAR we get $\epsilon \lesssim 15.5$ and $\epsilon \lesssim 41.8$ for $N = 10$ and $N = 40$ respectively. As can be read off from the test accuracy plots in Fig. 4, defending a training set of the size $N = 10$ against our attack causes model's accuracy to drop by over 30 percentage points both for MNIST and CIFAR. When $N = 40$ the accuracy drop is much smaller, but still notable – slightly above 10 percentage points both for MNIST and CIFAR. Fig. 5 illustrates how different values of ϵ affect a CIFAR-10 reconstruction for $N = 10$. One can see that at the threshold value (ϵ between 10 and 20) the reconstruction becomes very blurry. For more reconstruction examples under DP-SGD see Appendix E.

5 Summary, Limitations and Future Work

We have described a data reconstruction attack against a transfer-learned classifier network built from a small quantity of training data. The attack can be implemented by a determined adversary who has a realistic level of knowledge about the classifier and it cannot be defended against by DP-SGD without severely damaging classifier accuracy. This flags-up major new concerns about the use of such transfer-learned classifiers when privacy of training-data is paramount, particularly bearing in mind the increasing use and range of applications of transfer learning. For limitations, see Appendix F. Since this is a newly introduced method, there remain several research avenues for developing our findings further: firstly, there is scope for developing the attack into a more fully operational and widely applicable one (Appendix F). Secondly it may be of interest to develop theoretical bounds for reconstruction success under the threat model of the weak adversary. This would require finding new DP-privacy guarantees in the hidden state threat model for non-convex functions which is known as a hard open problem [2, 11], but our work may inform future theory in this direction. Thirdly, there is the challenge of finding new methods of defending data privacy against these attacks. Perhaps, for example, one could produce a more tailored version of DP or replace the original dataset with a distilled one [39]. Defences against these attacks will be a focus of our future work.

Acknowledgments and Disclosure of Funding

We would like to thank His Majesty's Government for fully funding Tomasz Maciazek and for contributing toward Robert Allison's funding during the course of this work. This work was carried out using the computational facilities of the Advanced Computing Research Centre, University of Bristol - `http://www.bristol.ac.uk/acrc/`. We also thank Tom Lovett from the Oxford University Mathematical Institute for useful discussions, comments and suggestions relating to this work.

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 308–318, New York, NY, USA, 2016. Association for Computing Machinery.

[2] J. M. Altschuler and K. Talwar. Privacy of noisy stochastic gradient descent: more iterations without more privacy loss. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.

[3] D. Arpit, S. Jastrzundefinedbski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 233–242. JMLR.org, 2017.

[4] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *Int. J. Secur. Netw.*, 10(3):137–150, Sept. 2015.

[5] B. Balle, G. Barthe, M. Gaboardi, and J. Geumlek. Privacy amplification by mixing and diffusion mechanisms. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

[6] B. Balle, G. Cherubin, and J. Hayes. Reconstructing training data with informed adversaries. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1138–1156, 2022.

[7] W. Bankes, G. Hughes, I. Bogunovic, and Z. Wang. REDUCR: Robust data downsampling using class priority reweighting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[8] G. Buzaglo, N. Haim, G. Yehudai, G. Vardi, Y. Oz, Y. Nikankin, and M. Irani. Deconstructing data reconstruction: Multiclass, weight decay and general losses. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 51515–51535. Curran Associates, Inc., 2023.

[9] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914, 2022.

[10] N. Carlini, C. Liu, U. Erlingsson, J. Kos, and D. Song. The secret sharer: evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, page 267–284, USA, 2019. USENIX Association.

[11] T. I. Cebere, A. Bellet, and N. Papernot. Tighter privacy auditing of DP-SGD in the hidden state threat model. In *The Thirteenth International Conference on Learning Representations*, 2025.

[12] P. K. Chan and S. J. Stolfo. Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, KDD'98, page 164–168. AAAI Press, 1998.

[13] F. Chollet. Complete guide to transfer learning & fine-tuning in Keras. `https://keras.io/guides/transfer_learning/`. Accessed: 2025-04-30.

[14] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926, 2017.

[15] Council of European Union and European Parliament. Regulation (EU) 2024/1689, 2024. Retrieved from `http://data.europa.eu/eli/reg/2024/1689/oj`.

[16] L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[17] V. Feldman, I. Mironov, K. Talwar, and A. Thakurta. Privacy Amplification by Iteration . In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 521–532, Los Alamitos, CA, USA, Oct. 2018. IEEE Computer Society.

[18] S. Fort, J. Ren, and B. Lakshminarayanan. Exploring the limits of out-of-distribution detection. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[19] M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1322–1333, New York, NY, USA, 2015. Association for Computing Machinery.

[20] 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 *Proceedings of the 23rd USENIX Conference on Security Symposium*, SEC'14, page 17–32, USA, 2014. USENIX Association.

[21] 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 *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 619–633, New York, NY, USA, 2018. Association for Computing Machinery.

[22] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[23] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of Wasserstein gans. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5769–5779, Red Hook, NY, USA, 2017. Curran Associates Inc.

[24] N. Haim, G. Vardi, G. Yehudai, O. Shamir, and M. Irani. Reconstructing training data from trained neural networks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 22911–22924. Curran Associates, Inc., 2022.

[25] J. Hayes, B. Balle, and S. Mahloujifar. Bounding training data reconstruction in DP-SGD. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 78696–78722. Curran Associates, Inc., 2023.

[26] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification . In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, Los Alamitos, CA, USA, Dec. 2015. IEEE Computer Society.

[27] D. Ippolito, F. Tramer, M. Nasr, C. Zhang, M. Jagielski, K. Lee, C. Choquette Choo, and N. Carlini. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In C. M. Keet, H.-Y. Lee, and S. Zarrieß, editors, *Proceedings of the 16th International Natural Language Generation Conference*, pages 28–53, Prague, Czechia, Sept. 2023. Association for Computational Linguistics.

[28] M. Jagielski, J. Ullman, and A. Oprea. Auditing differentially private machine learning: How private is private SGD? In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22205–22216. Curran Associates, Inc., 2020.

[29] G. Kaissis, J. Hayes, A. Ziller, and D. Rueckert. Bounding data reconstruction attacks with the hypothesis testing interpretation of differential privacy, 2023.

[30] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *ICLR (Poster)*, 2015.

[31] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby. Big transfer (bit): General visual representation learning. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*, page 491–507, Berlin, Heidelberg, 2020. Springer-Verlag.

[32] A. Krizhevsky. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

[33] B. Kulynych, J. F. Gomez, G. Kaissis, F. du Pin Calmon, and C. Troncoso. Attack-aware noise calibration for differential privacy. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 134868–134901. Curran Associates, Inc., 2024.

[34] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[35] N. Loo, R. Hasani, M. Lechner, A. Amini, and D. Rus. Understanding reconstruction attacks with the neural tangent kernel and dataset distillation. In *The Twelfth International Conference on Learning Representations*, 2024.

[36] S. Marcel and Y. Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, page 1485–1488, New York, NY, USA, 2010. Association for Computing Machinery.

[37] M. Nasr, J. Hayes, T. Steinke, B. Balle, F. Tramèr, M. Jagielski, N. Carlini, and A. Terzis. Tight auditing of differentially private machine learning. In *Proceedings of the 32nd USENIX Conference on Security Symposium*, SEC '23, USA, 2023. USENIX Association.

[38] J. Neyman, E. S. Pearson, and K. Pearson. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

[39] T. Nguyen, Z. Chen, and J. Lee. Dataset meta-learning from kernel ridge-regression. In *International Conference on Learning Representations*, 2021.

[40] E. S. Olivas, J. D. M. Guerrero, M. M. Sober, J. R. M. Benedito, and A. J. S. Lopez. *Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques - 2 Volumes*. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA, 2009.

[41] N. Ponomareva, S. Vassilvitskii, Z. Xu, B. McMahan, A. Kurakin, and C. Zhang. How to dp-fy ml: A practical tutorial to machine learning with differential privacy. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 5823–5824, New York, NY, USA, 2023. Association for Computing Machinery.

[42] P. Radivojac, N. V. Chawla, A. K. Dunker, and Z. Obradovic. Classification and knowledge discovery in protein databases. *J. of Biomedical Informatics*, 37(4):224–239, Aug. 2004.

[43] M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4334–4343. PMLR, 10–15 Jul 2018.

[44] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership Inference Attacks Against Machine Learning Models . In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, Los Alamitos, CA, USA, May 2017. IEEE Computer Society.

[45] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[46] C. Song and V. Shmatikov. Overlearning reveals sensitive attributes. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[47] M. Tan and Q. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019.

[48] M. Tomasz and A. Robert. Data reconstruction attacks. `https://github.com/tmaciazek/training-data-reconstruction.git`, 2025.

[49] F. Tramer and D. Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2021.

[50] UK Department for Science, Innovation and Technology. Code of Practice for the Cyber Security of AI, 2025. Retrieved from `https://www.gov.uk/government/publications/ai-cyber-security-code-of-practice`.

[51] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. The iNaturalist Species Classification and Detection Dataset . In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8769–8778, Los Alamitos, CA, USA, June 2018. IEEE Computer Society.

[52] Y. Wu, P. Zhou, A. G. Wilson, E. P. Xing, and Z. Hu. Improving gan training with probability ratio clipping and sample reweighting. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[53] Z. Yang, J. Zhang, E.-C. Chang, and Z. Liang. Neural network inversion in adversarial setting via background knowledge alignment. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 225–240, New York, NY, USA, 2019. Association for Computing Machinery.

[54] J. Ye and R. Shokri. Differentially private learning needs hidden state (or much faster convergence). In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.

[55] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao, G. Cormode, and I. Mironov. Opacus: User-friendly differential privacy library in PyTorch. *CoRR*, abs/2109.12298, 2021.

[56] S. Zagoruyko and N. Komodakis. Wide residual networks. In R. C. Wilson, E. R. Hancock, and W. A. P. Smith, editors, *Proceedings of the British Machine Vision Conference 2016 (BMVC 2016)*. BMVA Press, 2016.

[57] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. Deep sets. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

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

A Membership Inference Security Game, Reconstruction FPR, TPR and ROC Curves

The game proceeds between a challenger and an adversary.

1) The challenger samples a training dataset $D_n = (Z_1, \ldots, Z_N) \sim \pi_N$ and applies the training mechanism $D_n \xrightarrow{M} \theta$.
2) The challenger randomly chooses $b \in \{0, 1\}$, and if $b = 0$, samples a new dataset $D'_n \sim \pi_N$ such that $D_n \cap D'_n = \emptyset$ and applies the training mechanism $D'_n \xrightarrow{M} \theta'$. Otherwise, the challenger selects $\theta' = \theta$.
3) The challenger sends D_n and θ' to the adversary.
4) The adversary gets query access to the distribution π and the mechanism M and outputs a bit $\hat{b}$ given the knowledge of D_n and θ'.
5) Output 1 if $\hat{b} = b$, and 0 otherwise.

Note that if the training mechanism M is deterministic then the above game is trivial – the adversary can simply compute $M(D_n)$ and compare it with θ'. Otherwise, the adversary can apply the standard MIA to some $z_0 \in D_n$ to decide whether z_0 is a member of the training set that corresponds to θ'. In this sense, a reconstruction attack can also be used as a MIA. Namely, given a reconstruction attack $R : \Theta \to \mathcal{Z}$ and a threshold $\tau \geq 0$ the adversary outputs $\hat{b} = 1$ if $\min_{Z \in D_n} \ell(Z, \mathcal{R}(\theta)) \leq \tau$ and $\hat{b} = 0$ otherwise. We define the false-positive reconstruction rate as the probability of the false-positive outcome in the above game, i.e. $\hat{b} = 1$ and $b = 0$. Under the assumption that the probability of sampling two non-disjoint training sets from π_N is zero this is equivalent to Definition 1. Similarly, the true-positive reconstruction rate is the probability of the true-positive outcome in the above game, i.e. $\hat{b} = 1$ and $b = 1$.

Algorithm 2 and Algorithm 3 describe the TPR and FPR estimation procedures that we have used to calculate the ROC curves. For explanation of the notation, see Section 3 and Algorithm 1 in particular.

Algorithm 2 TPR, FPR estimation for unconditional reconstructor NN

Input: Reconstructor NN $\mathcal{R}$, shadow validation dataset $\mathcal{D}^{val}_{shadow}$, the covariance matrix of the normalized weights $\widetilde{\mathrm{Cov}}_\theta$, MSE threshold τ.
Initialize: $N_{TP} \leftarrow 0$ and $N_{FP} \leftarrow 0$.

for $(\mathcal{D}_k, \theta_k)$ `in` $\mathcal{D}^{val}_{shadow}$ **do**
  if $\min_{Z \in \mathcal{D}_k} \ell_{MSE}(Z, \mathcal{R}(\theta_k)) \leq \tau$ **then**
    $N_{TP} \leftarrow N_{TP} + 1$
  end if
  `Sample` $\theta \sim \mathcal{N}(0, \widetilde{\mathrm{Cov}}_\theta)$.
  if $\min_{Z \in \mathcal{D}_k} \ell_{MSE}(Z, \mathcal{R}(\theta)) \leq \tau$ **then**
    $N_{FP} \leftarrow N_{FP} + 1$
  end if
end for
$TPR \leftarrow N_{TP}/N^{val}_{shadow}$
$FPR \leftarrow N_{FP}/N^{val}_{shadow}$

Output: Reconstruction TPR, FPR.

Figure 6 shows the success rates of our reconstruction attack on a log-scale receiver operating characteristic curve (ROC curve). The ROC curve shows how TPR and FPR changes for all the threshold values τ. In particular, the ROC curve shows how our attack performs at low FPR.

A.1 Evaluating Data Reconstruction Attacks via Hypothesis Testing

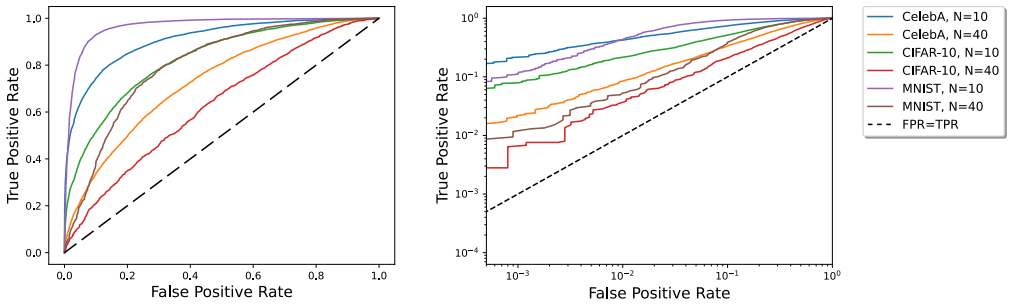
When deploying a trained reconstructor NN $\mathcal{R} : \Theta \to \mathcal{Z}$, one needs to understand how reliable it is. In Section 2 we have argued that a reconstructor NNs can make up its outputs by generating false-positive reconstructions. In terms of the modified membership security game, given a reconstruction, $\mathcal{R}(\theta)$,

Algorithm 3 TPR, FPR estimation for conditional reconstructor NN

Input: Reconstructor NN $\mathcal{R}$, shadow validation dataset $\mathcal{D}^{val}_{shadow}$, the covariance matrix of the normalized weights $\widetilde{\mathrm{Cov}}_\theta$, MSE threshold τ, reconstruction classes $c \in \{0, \ldots, C-1\}$.
Initialize: $N_{TP} \leftarrow 0$ and $N_{FP} \leftarrow 0$.

for $(\mathcal{D}_k, \theta_k)$ `in` $\mathcal{D}^{val}_{shadow}$ **do**
  `Sample` $\theta \sim \mathcal{N}(0, \widetilde{\mathrm{Cov}}_\theta)$.
  for $c = 0$ `to` $C-1$ **do**
    `Select` $\mathcal{D}^c_k \subset \mathcal{D}_k$, `the subset of images of the class` c.
    `Set` $\theta^c_k \leftarrow$ `the vector` θ_k `appended with the one-hot encoded class vector of the class` c.
    `Set` $\theta^c \leftarrow$ `the random vector` θ `appended with the one-hot encoded class vector of the class` c.
    if $\min_{Z \in \mathcal{D}^c_k} \ell_{MSE}(Z, \mathcal{R}(\theta^c_k)) \leq \tau$ **then**
      $N_{TP} \leftarrow N_{TP} + 1$
    end if
    if $\min_{Z \in \mathcal{D}^c_k} \ell_{MSE}(Z, \mathcal{R}(\theta^c)) \leq \tau$ **then**
      $N_{FP} \leftarrow N_{FP} + 1$
    end if
  end for
end for
$TPR \leftarrow N_{TP} / \left(N^{val}_{shadow} \times C\right)$
$FPR \leftarrow N_{FP} / \left(N^{val}_{shadow} \times C\right)$

Output: Reconstruction TPR, FPR.



a) *b)*

Figure 6: The (cumulative) ROC curves of our reconstruction attack for the reconstructor NNs obtained from the experiments described in Section 3. a) Linear scale plot showing the ROC at high FPR and b) log-scale plot showing the ROC at low FPR.

and a training set $\mathcal{D}_N$ this raises the problem of deciding whether $\mathcal{R}(\theta)$ is indeed a reconstruction of some element of $\mathcal{D}_N$ or not.

As presented in Section 2, one possible criterion is to select a MSE threshold τ and say that $\mathcal{R}(\theta)$ is a successful reconstruction if

$$\ell_{min\text{-}MSE}(\mathcal{D}_N, \mathcal{R}(\theta)) := \min_{Z \in \mathcal{D}_N} \ell_{MSE}(Z, \mathcal{R}(\theta)) \leq \tau. \tag{4}$$

Consequently, one can calculate the corresponding reconstruction TPR and FPR according to Definition 1 and generate the ROC curves which we present in Fig. 6. We will call the ROC curves computed in this way the *cumulative* ROC curves. This is because the relevant FPR_τ and TPR_τ are the cumulative distribution functions of the random variable $\ell_{min\text{-}MSE}$.

In this section, we compare the cumulative ROC curves with the ROC curves obtained via the Neyman-Pearson hypothesis test [38]. The Neyman-Pearson hypothesis test is also the base for the powerful LiRA membership inference attack [9] which uses similar approach to the one presented here. The Neyman-Pearson ROC curves are provably optimal in the sense that they give the best possible TPR at each fixed FPR. To calculate them, we recast the reconstructor NN evaluation

problem as hypothesis testing. Namely, we treat the observable $\ell_{min\text{-}MSE}\left(\mathcal{D}_N, \mathcal{R}(\theta)\right)$ as a random variable supported on $[0; 4]$ (recall that we normalize the images to $[-1, 1]$) which can come from different distributions. More specifically, we study the following hypotheses.

- The null hypothesis H_0 states that θ are the weights of a model trained on $\mathcal{D}_N$ i.e.,
$$H_0: \quad \ell_{min\text{-}MSE} = \min_{Z \in \mathcal{D}_N} \ell_{MSE}\left(Z, \mathcal{R}(\theta)\right) \quad \text{with} \quad \mathcal{D}_N \sim \pi_N,\ \theta \sim M(\mathcal{D}_N),$$
where M is the known randomized training mechanism.
- The alternative hypothesis H_1 states that θ are the weights of a model trained on another training set $\mathcal{D}'_N$ i.e.,
$$H_1: \quad \ell_{min\text{-}MSE} = \min_{Z \in \mathcal{D}_N} \ell_{MSE}\left(Z, \mathcal{R}(\theta)\right) \quad \text{with} \quad \mathcal{D}_N \sim \pi_N,\ \mathcal{D}'_N \sim \pi_N,\ \theta \sim M(\mathcal{D}'_N).$$
- The alternative hypothesis H_1^* states that the reconstruction was randomly drawn from the prior distribution π i.e.,
$$H_1^*: \quad \ell_{min\text{-}MSE} = \min_{Z \in \mathcal{D}_N} \ell_{MSE}\left(Z, Z'\right) \quad \text{with} \quad \mathcal{D}_N \sim \pi_N,\ Z' \sim \pi.$$

Let us denote the respective probability distributions of $\ell_{min\text{-}MSE}$ as ρ_0, ρ_1 and ρ_1*. When deciding between H_0 and H_1, the Neyman-Pearson hypothesis testing criterion accepts H_0 for a given value of $\ell_{min\text{-}MSE}$ when the likelihood-ratio
$$\log\left(\frac{\rho_0(\ell_{min\text{-}MSE})}{\rho_1(\ell_{min\text{-}MSE})}\right) > C \tag{5}$$
for some threshold value $C \in \mathbb{R}$. The TPR in this test is the probability of accepting H_0 for $\ell_{min\text{-}MSE} \sim \rho_0$ and the FPR the probability of accepting H_0 for $\ell_{min\text{-}MSE} \sim \rho_1$. Analogous criteria are applied when deciding between H_0 and H_1^*. The Neyman-Pearson ROC curves are obtained by varying the threshold C. Unfortunately, in our case we do not have access to the analytic expressions for ρ_0, ρ_1, ρ_1*, so we cannot calculate the likelihoods in Equation 5 exactly. In practice, we will approximate the distributions ρ_0, ρ_1 and ρ_1* by something more tractable. To this end, we will look at the distribution of the random variable
$$\phi = \log\left(\frac{\ell_{min\text{-}MSE}/4}{1 - \ell_{min\text{-}MSE}/4}\right) \in \mathbb{R}.$$
This is because our experimental evidence shows that ϕ is approximately normally distributed, as opposed to $\ell_{min\text{-}MSE}$. The relevant histograms for the CelebA-reconstructor NN are shown in Fig. 7. The histograms have two important properties: i) the separation between the distribution of $\ell_{min\text{-}MSE}$ corresponding to H_0 and H_1 or H_1^* decreases with N making it more difficult to distinguish between the hypotheses, ii) there is more separation between the distributions of $\ell_{min\text{-}MSE}$ corresponding to H_0 and H_1^* than between H_0 and H_1 showing that the reconstructor NN is better at generating false-positive reconstructions than simple random guessing.

In Fig 8 we compare the cumulative- and the Neyman-Pearson ROC curves for the reconstructor NNs trained in the CelebA-experiment. The plots show that both ROC curves are extremely close to each other. Below, we prove that under certain conditions they are exactly the same and derive an analytic formula for these ROC curves. The analytic formula is compared with our experimental ROC curves in Fig. 9.

Theorem 1. *Let ρ_0 and ρ_1 be the distributions of the observable ω supported on a connected domain $\Omega \subset \mathbb{R}$ under hypotheses H_0 and H_1 respectively. Define the cumulative- and the Neyman-Pearson ROC via*
$$TPR_{cum}(\tau) := \mathcal{P}_{\omega\sim\rho_0}\left[\omega < \tau\right], \quad FPR_{cum}(\tau) := \mathcal{P}_{\omega\sim\rho_1}\left[\omega < \tau\right],$$
$$TPR_{NP}(C) := \mathcal{P}_{\omega\sim\rho_0}\left[\rho_0(\omega) > C\rho_1(\omega)\right], \quad FPR_{NP}(C) := \mathcal{P}_{\omega\sim\rho_1}\left[\rho_0(\omega) > C\rho_1(\omega)\right],$$
Assume that there exists a strictly increasing differentiable transformation $\Phi : [0, 1] \to \mathbb{R}$ such that
$$\Phi(\omega) \sim \mathcal{N}\left(\mu_0, \sigma_0^2\right) \quad \text{if} \quad \omega \sim \rho_0, \quad \Phi(\omega) \sim \mathcal{N}\left(\mu_1, \sigma_1^2\right) \quad \text{if} \quad \omega \sim \rho_1.$$
Then, the cumulative-ROC and the Neyman-Pearson ROC are identical and given by the following formula
$$TPR = \frac{1}{2} - \frac{1}{2}\operatorname{erf}\left(\frac{\mu_0 - \mu_1}{\sqrt{2}\sigma_0} + \frac{\sigma_1}{\sigma_0}\operatorname{erf}^{-1}\left(1 - 2FPR\right)\right), \tag{6}$$
where erf *is the error function.*

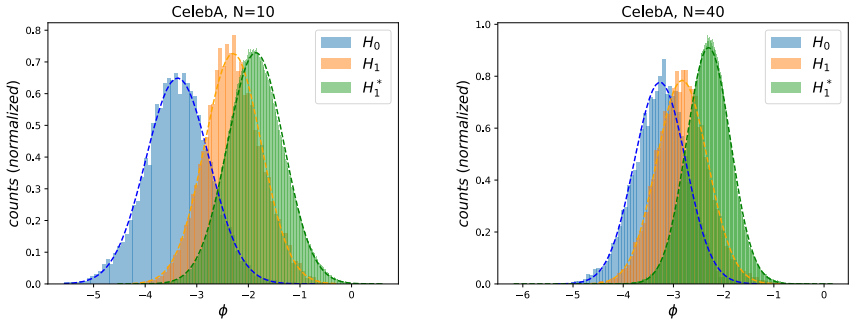


Figure 7: The histograms illustrating the approximately normal distributions of ϕ for the different hypotheses. The sample sizes were 10^4, 10^4 and 10^6 for H_0, H_1 and H_1^* respectively. The dashed curves show the Gaussians fitted according the maximum likelihood.

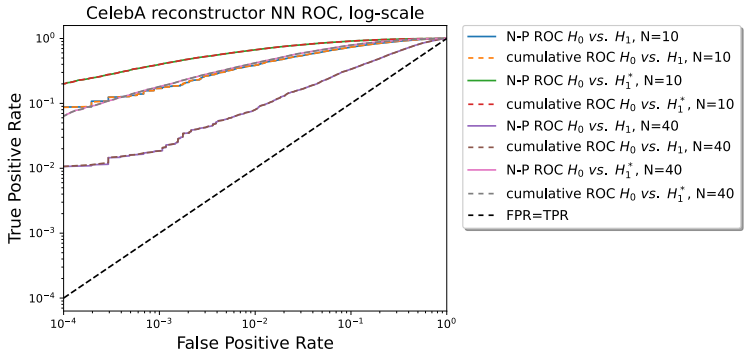


Figure 8: The comparison between the cumulative- and Neyman-Pearson (N-P) ROC curves for different pairs of hypotheses in the CelebA-experiment. There cumulative- and N-P ROC overlap almost exactly.

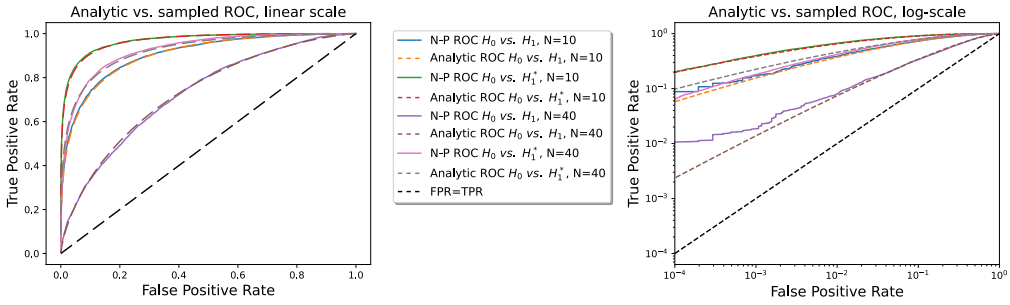


Figure 9: The comparison between the analytic ROC described by Equation (6) and Neyman-Pearson (N-P) ROC curves for different pairs of hypotheses in the CelebA-experiment. There is a good agreement between the analytic expressions and the experiment, but the curves often diverge a bit at low FPR. We suppose this may be due to insufficient sampling at this extreme of the ROCs.

Proof. (Sketch.) Since the transformation Φ is strictly increasing, we get that $\omega < \tau$ if and only if $\Phi(\omega) < \Phi(\tau)$. Thus,

$$TPR_{cum}(\tau) := \mathcal{P}_{\phi\sim\mathcal{N}\left(\mu_0,\sigma_0^2\right)}\left[\phi < \Phi(\tau)\right], \quad FPR_{cum}(\tau) := \mathcal{P}_{\phi\sim\mathcal{N}\left(\mu_1,\sigma_1^2\right)}\left[\phi < \Phi(\tau)\right]. \tag{7}$$

The above probabilities can be calculated analytically.

$$TPR_{cum}(\tau) := \frac{1}{2}\left(1 - \operatorname{erf}\left(\frac{\mu_0 - \Phi(\tau)}{\sqrt{2}\sigma_0}\right)\right), \tag{8}$$

$$FPR_{cum}(\tau) := \frac{1}{2}\left(1 - \operatorname{erf}\left(\frac{\mu_1 - \Phi(\tau)}{\sqrt{2}\sigma_1}\right)\right). \tag{9}$$

It is a matter of a straightforward calculation to extract $\Phi(\tau)$ from Equation (9) and plug it into Equation (8) to obtain the cumulative ROC curve given by Equation (6). In the remaining part of this proof we argue that the same ROC curve is obtained from the Neyman-Pearson criterion.

The transformation Φ is differentiable and strictly increasing, so $\left(\Phi^{-1}\right)'(\phi) > 0$ for all $\phi \in \mathbb{R}$. Thus, $\rho_0(\omega) > C\rho_1(\omega)$ for some $\omega = \Phi^{-1}(\phi)$ if and only if

$$\rho_0\left(\Phi^{-1}(\phi)\right)\left(\Phi^{-1}\right)'(\phi) > C\rho_1\left(\Phi^{-1}(\phi)\right)\left(\Phi^{-1}\right)'(\phi).$$

Recall that the transformation Φ is chosen so that

$$\rho_i\left(\Phi^{-1}(\phi)\right)\left(\Phi^{-1}\right)'(\phi) = \tilde{\rho}_i(\phi) := \frac{1}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(\phi-\mu_i)^2}{2\sigma_i^2}\right), \quad i = 0, 1.$$

Thus, using the monotonicity of the logarithm we can write equivalently that

$$TPR_{NP}(\tilde{C}) := \mathcal{P}_{\phi\sim\tilde{\rho}_0}\left[\log\frac{\tilde{\rho}_0(\phi)}{\tilde{\rho}_1(\phi)} > \tilde{C}\right], \quad FPR_{NP}(\tilde{C}) := \mathcal{P}_{\phi\sim\tilde{\rho}_1}\left[\log\frac{\tilde{\rho}_0(\phi)}{\tilde{\rho}_1(\phi)} > \tilde{C}\right]. \tag{10}$$

Plugging in the Gaussian form of $\tilde{\rho}_0$ and $\tilde{\rho}_1$ we get that the log-likelihood-ratio is given by the following quadratic equation.

$$\log\frac{\tilde{\rho}_0(\phi)}{\tilde{\rho}_1(\phi)} = \log\frac{\sigma_1}{\sigma_0} - \frac{(\phi-\mu_0)^2}{2\sigma_0^2} + \frac{(\phi-\mu_1)^2}{2\sigma_1^2}. \tag{11}$$

In what follows, we will assume that $\sigma_0 > \sigma_1$ which is the case in the CelebA experiments. The case $\sigma_0 < \sigma_1$ can be treated fully analogously. If $\sigma_0 > \sigma_1$, then the log-likelihood-ratio is described by a convex parabola. The Neyman-Pearson criterion leads to two roots whenever

$$\tilde{C} > \tilde{C}_{\min} = \log\frac{\sigma_1}{\sigma_0} - \frac{1}{2}\frac{(\mu_0-\mu_1)^2}{\sigma_0^2-\sigma_1^2}.$$

The two roots are of the form $r_0 \pm \delta$, where

$$r_0 = \frac{\mu_1\sigma_0^2 - \mu_0\sigma_1^2}{\sigma_0^2-\sigma_1^2}, \quad \delta\left(\tilde{C}\right) = \frac{\sigma_0\sigma_1\sqrt{(\mu_0-\mu_1)^2 + 2\left(\tilde{C} + \log\frac{\sigma_0}{\sigma_1}\right)(\sigma_0^2-\sigma_1^2)}}{\sigma_0^2-\sigma_1^2} > 0. \tag{12}$$

Hence, we can compute

$$\begin{aligned} TPR_{NP}\left(\tilde{C}\right) &= 1 - \mathcal{P}_{\phi\sim\tilde{\rho}_0}\left[\log\frac{\tilde{\rho}_0(\phi)}{\tilde{\rho}_1(\phi)} < \tilde{C}\right] = 1 - \frac{1}{\sqrt{2\pi}\sigma_0}\int_{r_0-\delta}^{r_0+\delta}\exp\left(-\frac{(\phi-\mu_0)^2}{2\sigma_0^2}\right)d\phi \\ &= 1 - \frac{1}{2}\left(\operatorname{erf}\left(\frac{\mu_0 - r_0 + \delta\left(\tilde{C}\right)}{\sqrt{2}\sigma_0}\right) - \operatorname{erf}\left(\frac{\mu_0 - r_0 - \delta\left(\tilde{C}\right)}{\sqrt{2}\sigma_0}\right)\right) \end{aligned}$$

Similarly, we get

$$FPR_{NP}\left(\tilde{C}\right) = 1 - \frac{1}{2}\left(\operatorname{erf}\left(\frac{\mu_1 - r_0 + \delta\left(\tilde{C}\right)}{\sqrt{2}\sigma_1}\right) - \operatorname{erf}\left(\frac{\mu_1 - r_0 - \delta\left(\tilde{C}\right)}{\sqrt{2}\sigma_1}\right)\right).$$

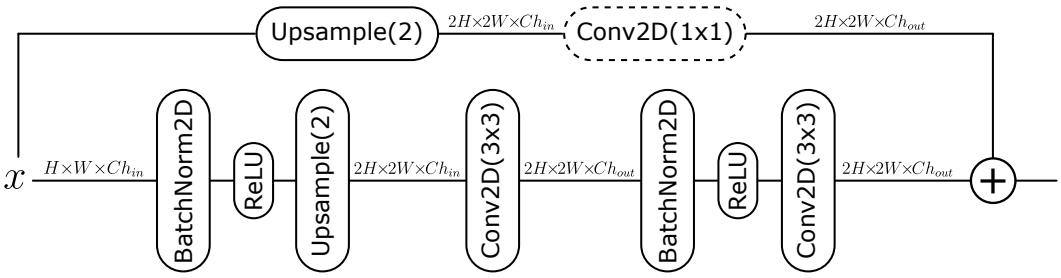


Figure 10: The residual block of the reconstructor NN. The "+"-operation means element-wise addition.

To verify that the above expressions yield the ROC curve described by Equation (6), we plug them into the formula. After some algebra, we get that this is true if and only if the following identity concerning the error function holds.

$$\begin{aligned} 0 =& \sqrt{2}\Delta\mu + 2\sigma_1 \operatorname{erf}^{-1}\left(\operatorname{erf}\left(\frac{\sigma_1\,\Delta\mu}{\sqrt{2}\Delta\sigma^2} + \frac{\delta}{\sqrt{2}\sigma_1}\right) - \operatorname{erf}\left(\frac{\sigma_1\,\Delta\mu}{\sqrt{2}\Delta\sigma^2} - \frac{\delta}{\sqrt{2}\sigma_1}\right) - 1\right) \\ & - 2\sigma_0 \operatorname{erf}^{-1}\left(\operatorname{erf}\left(\frac{\sigma_0\,\Delta\mu}{\sqrt{2}\Delta\sigma^2} + \frac{\delta}{\sqrt{2}\sigma_0}\right) - \operatorname{erf}\left(\frac{\sigma_0\,\Delta\mu}{\sqrt{2}\Delta\sigma^2} - \frac{\delta}{\sqrt{2}\sigma_0}\right) - 1\right), \end{aligned} \tag{13}$$

where $\delta > 0$ and $\Delta\mu := \mu_0 - \mu_1$ and $\Delta\sigma^2 := \sigma_0^2 - \sigma_1^2 > 0$. We do not currently have a proof of this identity, but we have verified it numerically for a range of the relevant parameters $\Delta\mu < 0$, $\sigma_1 > \sigma_2$ and $\delta > 0$. □

B Reconstructor NN architecture

The reconstructor is a residual network. Each residual block uses ReLU activations and consists of: 1) 2D batch norm layer followed by ReLU whose output is upsampled via the nearest-neighbours algorithm, 2) a pre-activation (3×3)-convolutional layer with 1-padding and stride 1 followed by another 2D batch norm layer and ReLU, 3) another pre-activation (3×3)-convolutional layer with 1-padding and stride 1. The bypass connection contains upsampling via the nearest-neighbours algorithm and, if the number of output channels of the residual block is different than the number of its input channels, then the bypass-upsampling is followed by a pre-activation (1×1)-convolution layer with 0-padding and stride 1. See Fig. 10. This residual block has the same architecture as the one used in [23, 52], except for the bypass-convolution which we need to allow for the number of channels to change. Table 2 shows the full architecture which is also slightly modified relative to the original version from [23, 52] – the first layer is convolutional and we change the number of channels between some of the consecutive residual blocks. For MNIST and CIFAR the output image has 32×32 resolution while for CelebA the output image has 64×64 resolution. For further details, please refer to our open-source implementation on GitHub [48].

C Further reconstruction examples

In this section, we present the following reconstruction examples.

- Figures 11 and 12 show reconstruction examples for the CelebA experiment with $N = 10$ and $N = 40$ respectively. Each training set contains $N/10$ images of positive class corresponding to a given person. The bottom rows contain the reconstructions and the top rows contain their closest match out of the $N/10$ images of the positive class from the original training set. Successful reconstructions (with MSE below the nearest-neighbour threshold) are marked by the tick-signs.
- Figure 13 shows reconstruction examples from the conditional reconstructor for the MNIST experiment with $N = 10$ and $N = 40$. Every training set contains $N/10$ images of each class. The bottom rows contain the reconstructions and the top rows contain their closest

Table 2: Summary of the reconstructor NN architecture. D_Θ is the number of trainable parameters in the attacked model that are accessed by the reconstructor. We have D_Θ equal to 2570, 12810 and 8201 for the MNIST-, CIFAR- and CelebA-experiments respectively. S is reconstructor's internal size parameter. In the MNIST-experiment we take $S = 256$ and in the CIFAR- and CelebA-experiments we take $S = 512$. $Ch_{out} = 1$ for MNIST and $Ch_{out} = 3$ for CIFAR, CelebA.

Reconstructor $R(\theta)$			
	Kernel Size	Output Shape	Notes
θ	–	$1 \times 1 \times (D_\Theta + 10)$	–
ConvTranspose2D	4×4	$4 \times 4 \times 4S$	$str. = 1, pad. = 0$
Residual Block	3×3	$8 \times 8 \times 2S$	–
Residual Block	3×3	$16 \times 16 \times S$	–
Residual Block	3×3	$32 \times 32 \times S$	–
Residual Block	3×3	$64 \times 64 \times S$	only for CelebA
BatchNorm2D	–	$32 \times 32 \times S$/$64 \times 64 \times S$	–
ReLU	–	$32 \times 32 \times S$/$64 \times 64 \times S$	–
Conv2D	3×3	$32 \times 32 \times Ch_{out}$/$64 \times 64 \times Ch_{out}$	$str. = 1, pad. = 1$
Tanh	–	$32 \times 32 \times Ch_{out}$/$64 \times 64 \times Ch_{out}$	–

match from the original training set. Successful reconstructions (with MSE below the nearest-neighbour threshold) are marked by the tick-signs.

- Figure 14 shows reconstruction examples from the conditional reconstructor for the CIFAR experiment with $N = 10$ and $N = 40$. Every training set contains $N/10$ images of each class. The bottom rows contain the reconstructions and the top rows contain their closest match from the original training set. Successful reconstructions (with MSE below the nearest-neighbour threshold) are marked by the tick-signs. Note that the reconstructions in Fig. 14b) are more generic and often lack details. This is due to the high reconstruction FPR for $N = 40$ (see Table 1).
- Figure 15 shows examples of false-positive reconstructions for MNIST and CIFAR-10. The bottom rows contain the random outputs of the conditional reconstructor NN and the top rows contain their closest match from the original (random) training set of the size $N = 40$. False-positive reconstructions (with MSE below the nearest-neighbour threshold) are marked by the tick-signs.

D Technical Details of the Experimental Setups

In this section we present details of the experiments presented in Section 3. For implementation details please refer to our code on GitHub [48].

D.1 Image Classifier Pre-training

Below, we specify the pre-training steps that we have done in each of the experiments. The goal is to pre-train the base models on a sufficiently general task so that they develop robust features which can be successfully transferred to other more specialized classification tasks where much smaller amounts of data are available.

1. **MNIST** We pre-train a scaled-down VGG-11 neural net [45] on the EMNIST-Letters [14] classification task: $128K$ data, 26 classes. Our implemented VGG-11 differs from the original one in [45] only by the number of channels which we divide by the factor of 16 (e.g. the first (3×3)-convolution has 4 output channels instead of the original 64). Similarly, the sizes of the final three fully connected layers are $256 - 256 - 26$ instead of the original $4096 - 4096 - 1000$ that has been designed for the ImageNet-1K classification. We initialize the weights of our VGG-11 randomly, rescale the input images to the 32×32 resolution and normalize them according to $X \to X/127.5 - 1$. We train for 50 epochs with the learning rate of 10^{-4} and mini-batch size 256 using Adam optimizer and cross-entropy loss. We obtain the base model test accuracy of 94.6%.

Figure 11: Reconstruction examples for the CelebA experiment with $N = 10$.

Figure 12: Reconstruction examples for the CelebA experiment with $N = 40$.

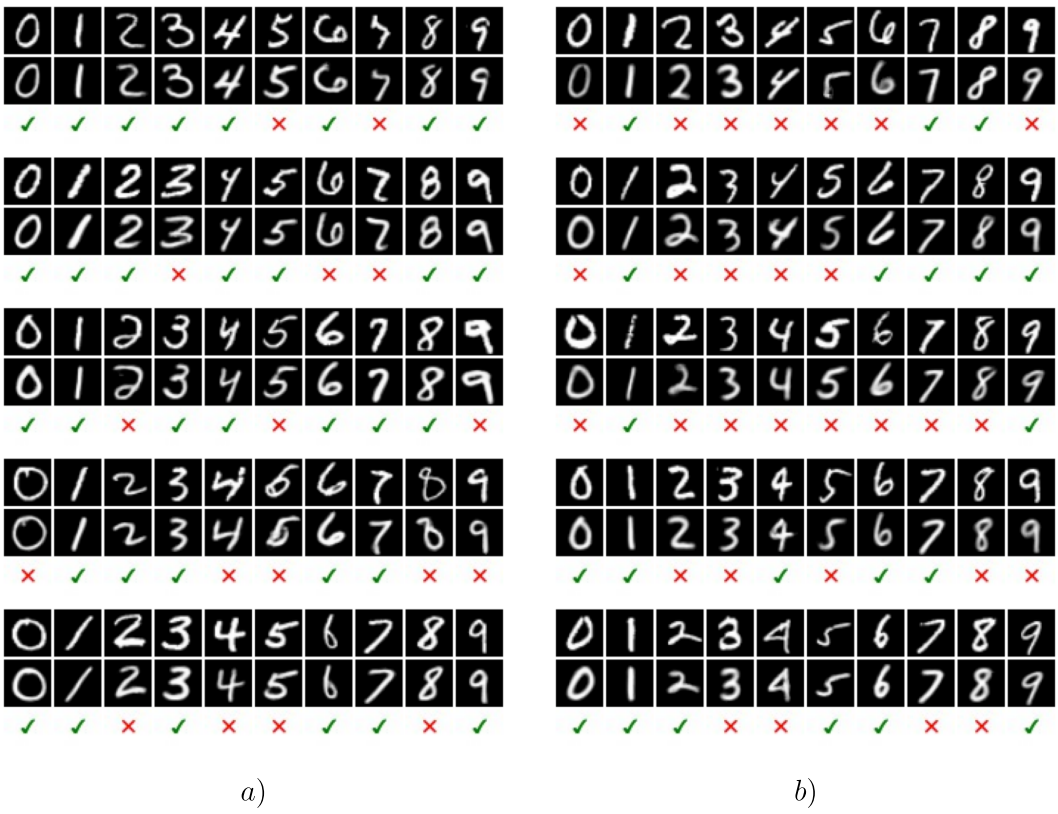


Figure 13: Reconstruction examples from the conditional reconstructor for the MNIST experiment with a) $N = 10$ and b) $N = 40$.

2. **CIFAR** We pre-train EfficientNet-B0 [47] on the CIFAR-100 [32] classification task: $50K$ data, 100 classes. We use the implementation of EfficientNet-B0 provided by Torchvision [36]. For the weight initialization we use the weights that were pre-trained on the ImageNet-1K classification (available in Torchvision). We subsequently remove the 1000 output neurons from the top fully connected layer and replace them with 100 neurons that we initialize with Pytorch's default initialization. The input images are resized to the 224×224 resolution using the bicubic interpolation. The pixel values are subsequently rescaled to $[0, 1]$ and normalized as $X \to (X - \mu)/\sigma$, where $\mu = [0.485, 0.456, 0.406]$ and $\sigma = [0.229, 0.224, 0.225]$ for each channel. We pre-train in two stages. In both stages we use the Adam optimizer and the cross-entropy loss. In the first stage we freeze all the parameters except for the parameters of the output layer and train the output layer for 20 epochs with learning rate 10^{-4}, weight decay 10^{-4} and batch size 256. In the second stage we unfreeze all the parameters except for the batch-norm layers and train the entire network for 200 epochs with learning rate 10^{-6} and batch size 64. We obtain the base model test accuracy of 85.9% which is close to some of the reported benchmarks for EfficientNet-B0, see [47].
3. **CelebA** We pre-train WideResNet-50 [56] on the CelebA [34] attributes classification task: $200K$ data, 40 binary attributes. We use the implementation of WideResNet-50-2 provided by Torchvision [36]. For the weight initialization we use the weights that were pre-trained on the ImageNet-1K classification (available in Torchvision). We subsequently remove the 1000 output neurons from the top fully connected layer and replace them with 40 neurons that we initialize with Pytorch's default initialization. The input images are resized to the 232×232 resolution using the bicubic interpolation and center-cropped to the size 224×224. The pixel values are subsequently rescaled to $[0, 1]$ and normalized as $X \to (X - \mu)/\sigma$, where $\mu = [0.485, 0.456, 0.406]$ and $\sigma = [0.229, 0.224, 0.225]$ for each channel. We pre-train in two stages. In both stages we use Adam optimizer and binary

$a)$ $b)$

Figure 14: Reconstruction examples from the conditional reconstructor for the CIFAR experiment with a) $N = 10$ and b) $N = 40$.

Figure 15: Examples of false-positive reconstructions for MNIST and CIFAR-10 ($N = 40$).

Table 3: The head-NN size/architecture and training details in the transfer-learning step. $FC(k)$ denotes the fully-connected layer with k neurons. In each experiment the head-NNs are trained via the standard gradient descent (full batch, no momentum) with the learning rate lr and weight decay λ_{WD}. The initial weights of the head-NN are drawn from $\mathcal{N}(0, \sigma^2_{init})$ and the biases are initialized to 0. The training loss is always the cross-entropy loss. N is the training set size of a given shadow model.

Experiment	Head-NN Architecture	σ_{init}	lr	λ_{WD}	Epochs
MNIST	Input $\rightarrow FC(10) \rightarrow$ Output	0.002	0.01	10^{-5}	$26 + 3N/5$
CIFAR	Input $\rightarrow FC(10) \rightarrow$ Output	0.002	0.01	10^{-4}	$38 + N$
CelebA	Input $\rightarrow FC(4) \xrightarrow{ReLU} FC(1) \rightarrow$ Output	0.0002	0.02	10^{-2}	100

cross-entropy loss. In the first stage we freeze all the parameters except for the parameters of the output layer and train the output layer for 5 epochs with learning rate 10^{-4}, weight decay 10^{-4} and batch size 256. In the second stage we unfreeze all the parameters except for the batch-norm layers and train the entire network for 30 epochs with learning rate 10^{-6} and batch size 128. We obtain the base model test accuracy of 92.0% in the pre-training task.

Note that we could have skipped the above pre-training with CIFAR-100 or the attribute classification with CelebA and just use the weights pre-trained on ImageNet-1K. However, note that our goal is to obtain possibly highly accurate classifiers in the subsequent transfer-learning step. Thus, it is beneficial to pre-train the base model on another public dataset which is more similar to the one used in the ultimate transfer-learning task.

D.2 Image Classifier Transfer-learning

During the transfer-learning step we realize the following general procedure.

1. We remove the output neurons from each of the pre-trained neural nets described in the Subsection D.1. The outputs of the neural net obtained this way are the post-activation outputs of the penultimate layer of the original neural net.
2. For all the images coming from the datasets (MNIST, CIFAR-10 and CelebA) we pre-compute their corresponding deep features vectors. This is done by feeding the neural nets from point 1 above by the images from the corresponding dataset and saving the obtained (penultimate post-activation) outputs. The input images are transformed according to the transformations specified in the Subsection D.1. This way, we replace the image datasets with their corresponding deep-feature datasets.
3. In each of the experiments we define a new head-NN which takes the pre-computed deep features as inputs and outputs the predictions. Due to the different nature of each of the transfer-learning tasks, in each experiment the head-NN has a slightly different size/architecture. We also train the head-NNs as part of the shadow model training (both for the training shadow model sets and the validation shadow model sets), see Algorithm 1. The head-NN architecture and training details are summarized in Table 3.

In the MNIST- and CIFAR-10 experiments we use balanced training sets i.e., $N = C \times M$ with $C = 10$ being the number of classes and M the number of training examples per class. In the CelebA-experiment we emulate transfer-learning with unbalanced data. There, we have M examples of the positive class and $9M$ examples of the negative class. In the CelebA training loss we upweight the positive class with the weight 10. Table 4 shows the mean and the standard deviation of the test accuracy of the resulting transfer-learned image classifiers. Because in this work we are evaluating the impact of DP-SGD on the classifier accuracy, it was important to optimize classifier training to obtain reasonably good accuracies. Indeed, the accuracies obtained for transfer-learning in the CIFAR-10 experiment with the training set sizes $N = 10$ and $N = 40$ are comparable with accuracies reported in the literature, see Figure 2 in [31]. Note, however, that work [31] uses data augmentation to improve transfer-learning test accuracy with small datasets, so their accuracies are generally better. We have not used data augmentation in our transfer-learning experiments to keep the setup simple.

Table 4: The mean and the standard deviation of the test accuracy of the transfer-learned image classifiers. Calculated over a sample of 5000 classifiers.

	$N = 10$	$N = 40$
Experiment	Test Accuracy	Test Accuracy
MNIST	0.812 ± 0.041	0.920 ± 0.017
CIFAR-10	0.587 ± 0.049	0.794 ± 0.022

Table 5: The mean and the standard deviation of the classification test accuracy, classification true-positive rate (TPR, correct classification of the minority class) and classification true-negative rate (TNR, correct classification of the majority class) of the transfer-learned face-recognition classifiers trained on unbalanced data in the CelebA-experiment. Calculated over a sample of 5000 classifiers.

N	Classification Acc.	Classification TPR	Classification TNR
10	0.914 ± 0.037	0.434 ± 0.286	0.967 ± 0.035
40	0.935 ± 0.030	0.645 ± 0.216	0.967 ± 0.022

In the face recognition experiment with CelebA data it has been challenging to train good image classifiers on small datasets. The true-positive rate of the classifiers (corresponding to the correct classification of the minority class) in this experiment has varied significantly between the shadow models, see Table 5.

D.3 Factors Affecting Reconstruction - Experimental Details

Here, we provide some more details concerning the experiments described in Section 3.2. Unless stated otherwise, the hyper-parameter and architecture configuration in each experiment from Section 3.2 has been the same as described in Section D.2.

In point 4 of Section 3.2 (reconstructor conditioning mismatch) we have used the hyper-parameters and head-NN architectures for the binary image classifiers as specified in Table 6.

In point 6 of Section 3.2 (attacked model weight initialization) we have varied σ_{init} and kept all the other hyper-parameters fixed to the values specified in Table 3. This way, we have produced the plots presented in Fig. 3b.

E Our Reconstruction attack under DP-SGD

We have made sure that the DP-SGD training hyper-parameters were chosen to have as small an impact on trained image classifier's accuracy as possible. To this end, we have followed the hyper-parameter selection procedure outlined in [41]. To keep this paper self-contained, we summarize this procedure below. The hyper-parameters in DP-SGD are the gradient clipping norm C, noise multiplier σ_{noise}, poisson sampling rate $q = B/N$ (B is the mini-batch size and N is the size of the training set), the number of training epochs and the learning rate. We have trained the shadow models with weight decay.

Table 6: The head-NN size/architecture and training details in the transfer-learning step in the binary classification tasks. $FC(k)$ denotes the fully-connected layer with k neurons. In each experiment the head-NNs are trained via the standard gradient descent (full batch, no momentum) with the learning rate lr and weight decay λ_{WD}. The initial weights of the head-NN are drawn from $\mathcal{N}(0, \sigma^2_{init})$ and the biases are initialized to 0. The training loss is always the binary cross-entropy loss. N is the training set size of a given shadow model.

Experiment	Head-NN Architecture	σ_{init}	lr	λ_{WD}	Epochs
MNIST	Input $\to FC(8) \xrightarrow{ReLU} FC(1) \to$ Output	0.002	0.1	10^{-4}	$26 + 3N/5$
CIFAR	Input $\to FC(8) \xrightarrow{ReLU} FC(1) \to$ Output	0.002	0.05	10^{-4}	$38 + N$

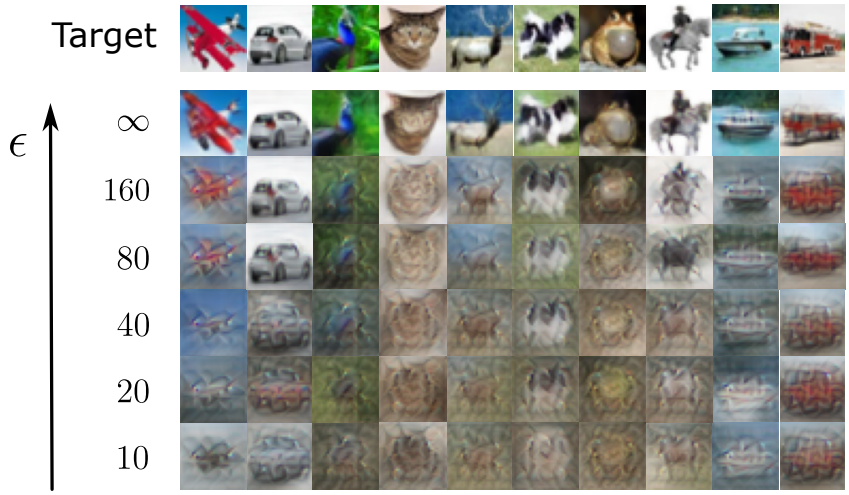


Figure 16: Reconstruction examples for CIFAR-10 and (ϵ, δ)-DP models for different values of ϵ. Training without DP-SGD means $\epsilon = \infty$. Classifier training set size $N = 10$.

1. The batch size B should be as large as computationally feasible. We choose $B = N - 1$ (the largest possible allowing non-trivial mini-batch Poisson sampling). Another factor to take into account is the fact that training for too many epochs will negatively impact model's accuracy due to overfitting. In line with some transfer-learning guidelines [13] we have worked with the optimal n_E (number of training epochs) specified in Table 3 in the column "Epochs".
2. Tune the other hyper-parameters (learning rate and weight decay) in the non-private setting (without DP-SGD) with the chosen B and n_E. The optimal learning rate and weight decay we have found are the same as specified in 3.
3. Choose the clipping norm C by training with DP-SGD with $\sigma_{noise} = 0$ and keeping all the parameters selected in previous steps fixed. This can be done by one or more parameter swipes in the log-scale. Small values of C adversely affect the accuracy of the classifier. C should be chosen so that it is small, but at the same time has only slight effect on classifier's accuracy when compared to the accuracy of the non-private classifier.
4. Compute the noise multiplier σ_{noise} that makes the model (ϵ, δ)-DP via the privacy accountant [1] with C and B found in the previous steps. In all experiments we take $\delta = N^{-1.1}$.
5. Do the final learning rate adjustment with C, B, σ_{noise} and weight decay found in the previous steps.

By performing the above hyper-parameter selection procedure we have found that the optimal training hyper-parameters for $B = N - 1$ have been the same as the ones specified in Table 3. For simplicity, in our experiments we have not performed the final adjustment of the learning rate. However, we have found that the learning rate values specified in Table 3 have been close to optimal in all experiments and that the model accuracy has not been very sensitive to small changes of the learning rate around these values. The optimal clipping norms C were found to be 4.0 and 1.2 for the MNIST and CIFAR experiments respectively. In all the DP-SGD experiments we use the same weight initialization as specified in Table 3. For DP-SGD training we have used the *Opacus* library [55].

Figure 16 shows some further examples of reconstructions for models trained with DP-SGD.

F Limitations and Future Work

We claim that the prototype attack we have presented is already sufficient to demonstrate the necessity of new protective measures for transfer-learned neural networks trained on small datasets. Nevertheless, there is considerable scope for developing this attack capability further (and those responsible for security should assume that a determined adversary would do so). For example, research directions in this area include:

- Developing automated processes for recovering information on the training set which, in the absence of any prior knowledge, is sufficient to generate effective shadow models.
- Showing additional robustness of the attack to mismatches between the actual and the assumed transfer learning process. This would allow for the possibility that non-published training methods might have been applied.
- Extending the attacks to recover more substantial subsets of the training data.

One of the clear limitations of our data reconstruction attack method is the fact that it becomes less effective when the size of the attacked model's training set increases. More specifically, when using the conditional reconstructor NN we can reliably reconstruct only the training data examples that come from the classes that are represented by only a few data instances. When the class from which we draw the reconstructions becomes larger then the reconstruction FPR increases making our attack unreliable. Thus, we suppose that the scope of real life situations where one can apply our attack reliably is restricted to those where the amount of training data was very small or the data was highly imbalanced. As we argue in the Introduction, this can commonly happen in practice in transfer-learning with imbalanced datasets.

Another limitation of our current approach concerns the architecture of the reconstructor NN and the related amount of compute and memory required to train it. Recall that the input of the reconstructor NN consists of the concatenated and flattened weights and biases of the image classifier layers that have been trained during the transfer learning. This is typically only one or two final layers, however as explained in Table 2 the resulting input dimensions D_Θ can easily be of the order of 10^4. This means that most of the parameters of the reconstructor NN are contained in its first convolutional layer. The exact number of parameters in this layer is equal to $4S(16D_\Theta + 1)$ where S is the internal size of the reconstructor NN (equal to 256 or 512). Thus, the first convolutional layer contains $\sim 10^8$ parameters. This makes the training relatively inefficient. In future work we will work on more efficient ways of processing reconstructor NN input. For instance, instead of concatenating all the weights of the trainable layers one could take inspiration from Deep Sets [57] and process each neuron separately by a sequence of convolutions and sum the result over the neurons. Similar approach has been used to construct NNs used for property inference attacks [21].

In Appendix A we have studied ROC curves coming from considering the distributions of the $min\text{-}MSE$ observable (see Equation 4). However, it is possible that there exist more informative observables that would yield better ROC curves. For instance, one could consider the components of the loss function that has been used for training the reconstructor NN i.e., the sum of MSE, MAE and the $LPIPS$-loss. It would also be interesting to use an MIA attack alongside our reconstruction attack to decide whether the reconstruction is an element of the image classifier's training set. This could reduce the FPR of our attack.