Improving Robustness to Model Inversion Attacks via Sparse Coding Architectures

Sayanton V. Dibbo[1,2], Adam Breuer[1], Juston Moore[2], and Michael Teti[2]

[1] Dartmouth College, Hanover, NH 03755, USA
[2] Los Alamos National Laboratory, Los Alamos, NM 87545, USA
{f0048vh,adam.breuer}@dartmouth.edu
{jmoore01,mteti}@lanl.gov

Abstract. Recent model inversion attack algorithms permit adversaries to reconstruct a neural network's private and potentially sensitive training data by repeatedly querying the network. In this work, we develop a novel network architecture that leverages sparse-coding layers to obtain superior robustness to this class of attacks. Three decades of computer science research has studied sparse coding in the context of image denoising, object recognition, and adversarial misclassification settings, but to the best of our knowledge, its connection to state-of-the-art privacy vulnerabilities remains unstudied. In this work, we hypothesize that sparse coding architectures suggest an advantageous means to defend against model inversion attacks because they allow us to control the amount of irrelevant private information encoded by a network in a manner that is known to have little effect on classification accuracy. Specifically, compared to networks trained with a variety of state-of-the-art defenses, our sparse-coding architectures maintain comparable or higher classification accuracy while degrading state-of-the-art training data reconstructions by factors of 1.1 to 18.3 across a variety of reconstruction quality metrics (PSNR, SSIM, FID). This performance advantage holds across 5 datasets ranging from CelebA faces to medical images and CIFAR-10, and across various state-of-the-art SGD-based and GAN-based inversion attacks, including *Plug-&-Play* attacks. We provide a cluster-ready PyTorch codebase to promote research and standardize defense evaluations.

Keywords: Model Inversion Attack, Defense, Privacy Attacks, Sparse Coding.

1 Introduction

The popularization of machine learning has been accompanied by the widespread use of neural networks that were trained on private, sensitive, and proprietary datasets. This has given rise to a new generation of privacy attacks that seek to infer private information about the training dataset simply by inspecting the representation of the training data that remains encoded in the model's parameters [11, 15, 16, 20, 21, 31, 35, 41, 45, 58, 61, 68, 75, 76, 78].

Of particular concern is a devastating stream of privacy attacks known as model inversion. Model inversion attacks leverage the network's parameters or

classifications in order to reconstruct entire images or data that were used to train the network. Early work on model inversion focused on a white-box setting where the attacker has unfettered access to the model or auxiliary information about the training data [20, 30, 69, 71, 77]. However, recent work has shown that standard network architectures are vulnerable to model inversion attacks even when attackers have no knowledge of the model's architecture or parameters, and only have access to the model's classifications or its intermediate outputs, such as leaked outputs from a single hidden network layer [5, 22, 45, 46, 56, 74].

Are different network architectures robust to model inversion attacks?

Such attacks are feasible because each hidden layer of a standard network architecture captures a detailed representation of the training data. It is well-known that standard dense layers exhibit a tendency to memorize their inputs [10, 23, 54], so even a minimal leak of a network's class distribution output or a leak of its intermediate outputs from a single layer is often sufficient to train an inverse mapping for data reconstruction. More concretely, state-of-the-art inversion attacks work by submitting externally obtained images to the model, observing leaked outputs, then using this data to train a new 'inverted' neural network that reconstructs (predicts) an input image given a leaked output. This can be accomplished either directly via SGD, or by optimizing a GAN, and we consider both approaches here. Such attacks on standard network architectures can reconstruct private training images that are clearly recognizable by humans familiar with the training data [4, 18, 22, 26, 30, 33, 61, 71, 74].

Recent work has pursued a diverse array of defense strategies to mitigate these attacks. For example, [22] augments the training dataset with GAN-generated fake samples designed to inject spurious features into the trained network that mislead the gradients computed during inversion attacks. In contrast, multiple recent defenses add regularization terms during training that attempt to penalize training data memorization [53, 66]. Other recent defenses noise the network weights to obfuscate memorized data [2, 47, 64], or noise and clip training gradients via DP-SGD [25]. All such approaches are costly: data augmentation-based defenses entail the computational burden of building a GAN and applying sophisticated parameter tuning techniques during training; regularization-based defenses explicitly trade away classification accuracy for less memorization, and noise-based defenses are also known to impose significant accuracy costs. Until very recently, there were no known provable guarantees for model inversion defenses, and the current best-known guarantees require a DP-SGD-based training algorithm that imposes a significant computational burden and accuracy loss to obtain privacy guarantees that are impractically weak for these attacks [25].

Very little is known about how a network's architecture contributes to its robustness (or vulnerability). This is surprising since throughout three decades of research in other domains such as image denoising [3, 6, 9, 12, 19, 39, 50, 55], object recognition [24, 36, 51, 59], and adversarial misclassification [17, 37, 52, 62, 63], researchers seeking to control their model's representations of the data have heavily studied sparse coding-based architectures that prune unnecessary details and preserve only the information that is essential to the model objective.

Specifically, sparse coding seeks to approximately represent an image (or layer) with only a small set of basis vectors selected from an overcomplete dictionary [9, 19, 50]. While it is well-known that computing a sparse representation using a standard objective function is NP-hard in general [14, 32, 49], we now benefit from fast approximation algorithms that efficiently compute high-quality sparse representations [8, 13, 32, 36, 39, 40, 48, 55]. Sparse coding architectures leverage this technique by inserting a sparse network layer after a dense layer, such that the sparse layer reduces the dense layer's outputs to a sparse representation.

To our knowledge, sparse coding architectures have not been studied in the context of model inversion or privacy attacks. However, they suggest an advantageous means to prevent such attacks because they control the amount of irrelevant private information encoded in a model's intermediate representations in a manner that is known to have little effect on its accuracy, that can be computed efficiently during training, and that adds little to the trained model's overall parameter complexity. Put simply, sparsifying a network's representations is a natural means to preclude memorization of detailed information about its inputs that is unnecessary to obtain high accuracy, so even an idealized 'perfect attacker' could only hope to recover a sparsified, un-detailed training image.

Main contribution. We begin by showing that an off-the-shelf sparse coding preprocessing step offers performance advantages compared to state-of-the-art data augmentation, regularization, and noise based defenses in terms of robustness to model inversion attacks. We then refine this idea into a network architecture that achieves superior performance. Our main result is a novel sparse-coding architecture, SCA, that is robust to state-of-the-art model inversion attacks.

SCA is defined by pairs of alternating sparse coded and dense layers that jettison unnecessary private information in the input image and ensure that downstream layers do not e.g., reconstruct this information. We show that SCA maintains comparable or higher classification accuracy while degrading state-of-the-art training data reconstructions 1.1 to 11.7 times more than 8 state-of-the-art data augmentation, regularization, and noise-based defenses in terms of PSNR and FID metrics and 1.1 to 720 times more in terms of SSIM. This performance advantage holds across 5 datasets ranging from CelebA faces to medical images and CIFAR-10, and across various state-of-the-art SGD-based and GAN-based inversion attacks, including *Plug-&-Play* attacks. SCA's defense performance is also more stable than baselines across multiple runs. We emphasize that, unlike recent state-of-the-art defenses that require sophisticated parameter tuning to perform well, SCA obtains these results absent parameter tuning (i.e., using default sparsity parameters) because sparse coding naturally precludes networks from memorizing detailed representations of the training data.

More broadly, our results show a deep connection between state-of-the-art ML privacy vulnerabilities and three decades of computer science research on sparse coding for other application domains. We provide a comprehensive cluster-ready PyTorch codebase to promote research and standardize defense evaluation.

2 Threat models

We consider three threat models that span the diverse range of powerful and well-informed attackers considered in recent work. We emphasize that a defense that performs well in all three settings provides strong evidence of its privacy protections under weaker, more realistic threat models with real-world attackers.

1. Plug-&-Play threat model [61]. Plug-&-Play attacks are considered the most performant recent attacks. These attacks optimize the intermediate representation of StyleGAN's input vectors so that generated images maximize the target network's class prediction probability, which the attacker can query.

Separately, recent theoretical work on model inversion emphasizes that a strong threat model should capture 'worst-case' attackers with direct access to the information-rich, high-dimensional intermediate outputs of the target model that store private information about the training data, as well as ideal training data examples for training an inverted model [1, 25]. We consider two variants:

2. End-to-end threat model. We consider an attacker with access to all of the *last* hidden layer's raw, high-dimensional outputs, as well as a large number of ideal training data examples drawn from the true training dataset [60, 67].

3. Split network threat model (Federated Learning). We also consider the split network threat model described by [64]. There has been much recent interest in Federated Learning architectures that split the network across multiple agents [7, 18, 26, 38, 44], particularly for privacy-fraught domains such as medicine where legal requirements limit data sharing [34, 65]. These architectures are known to be susceptible to model inversion attacks, [18, 26, 64], and defenses are urgently needed. This threat model also allows us to capture a different view of a 'worst-case' threat model: Model inversion attacks are known to be more effective when the attacker has access to outputs from earlier layers that may exhibit a more direct representation of the input images [26]. To capture this 'worst-case', we consider the setting where the attacker has access to raw intermediate outputs from the *first* linear network layer. As before, we also assume the attacker has access to ideal training data examples drawn from the actual training datasets. Appendix B provides additional details about the model inversion threat models.

3 SCA architecture

We now describe the SCA architecture, which is defined by alternating pairs of Sparse Coding Layers (SCL) and dense layers, followed by downstream linear and/or convolutional layers.

Sparse Coding Layer (SCL). Sparse coding converts raw inputs to sparse representations where only a few neurons whose features are useful in reconstructing the inputs are active. Our Sparse Coding Layer (SCL) performs sparse coding to obtain a sparse representation of a previous dense layer's representation (if the SCL is not the first layer in the network) or of the inputs (if the SCL is the first layer in the network). Fig. 1 illustrates the working principle of SCL.

Formally, each SCL performs a reconstruction minimization problem to compute the sparse representation of its inputs (either a previous layer's representation or of the inputs to the network). Suppose the input to a (2D convolutional) SCL is $\mathcal{X} \in \mathbb{R}^{\mathcal{C}\times\mathcal{H}\times\mathcal{W}}$ with $\mathcal{H}$ height, $\mathcal{W}$ width, and $\mathcal{C}$ channels/features. The goal is to find the sparse representation $\mathcal{R}_x \in \mathbb{R}^{\mathcal{F}\times\lfloor\mathcal{H}/S_h\rfloor\times\lfloor\mathcal{W}/S_w\rfloor}$, where $\mathcal{R}_x$ has few active neurons and corresponds to a denoised version of the input $\mathcal{X}$, and S_w and S_h indicate convolutional strides across the width and height of the input, respectively. F is the number of convolutional features in the SCL layer's dictionary, $\Omega \in \mathbb{R}^{\mathcal{F}\times\mathcal{C}\times\mathcal{H}_f\times\mathcal{W}_f}$, where $\mathcal{H}_f$ and $\mathcal{W}_f$ are the height and width of each convolutional feature, respectively. Per Figure 1, the sparse coding layer starts with its input, $\mathcal{X}$, and dictionary of features, Ω, to produce $\mathcal{R}_x$ by solving the following sparse reconstruction problem:

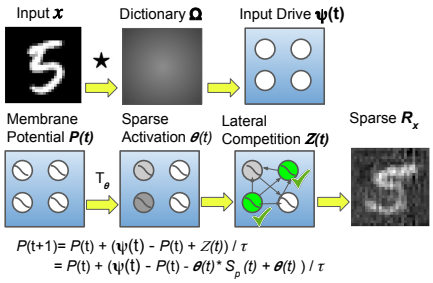


Fig. 1: Pipeline of neuron (membrane potential) dynamics in Sparse Coding Layer (SCL) with lateral competitions.

$$\min_{\mathcal{R}_x} \frac{1}{2}||\mathcal{X} - \mathcal{R}_x \circledast \Omega||_2^2 + \lambda||\mathcal{R}_x||_1 \tag{1}$$

where the first term represents how much information is preserved about $\mathcal{X}$ by $\mathcal{R}_x$ by measuring the difference between $\mathcal{X}$ and its reconstruction, $\mathcal{R}_x \circledast \Omega$, computed with a transpose convolution, $\circledast$. The second term measures how sparse $\mathcal{R}_x$ is, and λ is a constant which determines the trade-off between reconstruction fidelity and sparsity. Equation 1 is convex in $\mathcal{R}_x$, meaning we will always find the optimal $\mathcal{R}_x$ that solves Equation 1.

Among different techniques to perform sparse coding, we leverage the commonly used Locally Competitive Algorithm (LCA) [55]. LCA implements a recurrent network of leaky integrate-and-fire neurons that incorporates the general principles of thresholding and feature-similarity-based competition between neurons to solve Equation 1. While Rozell introduced LCA in the non-convolutional setting, it can be readily adapted to the convolutional setting (see Appendix A.2 for details). Specifically, each LCA neuron has an internal membrane potential $\mathcal{P}$ which evolves per the following differential equation:

$$\dot{\mathcal{P}}(t) = \frac{1}{\tau}[\Psi(t) - \mathcal{P}(t) - \mathcal{R}_x(t) * \mathcal{G}] \tag{2}$$

where τ is a time constant, $\Psi(t) = \mathcal{X} * \Omega$ is the neuron's bottom-up drive from the input computed by taking the convolution, $*$, between the input, $\mathcal{X}$, and the dictionary, Ω, and $-\mathcal{P}(t)$ is the leak term [37, 63]. Lateral competition between neurons is performed via the term $-\mathcal{R}_x(t) * \mathcal{G}$, where $\mathcal{G} = \Omega * \Omega - I$ is the similarity between each feature and the other $\mathcal{F}$ features ($-I$ prevents

self interactions). $\mathcal{R}_x$ is computed by applying soft threshold activation $T_\lambda(x) = \text{relu}(x - \lambda)$ to the neuron's membrane potential, which produces nonnegative, sparse representations. Overall, this means that in LCA neurons will compete to determine which ones best represent the input, and thus will have non-zero activations in $\mathcal{R}_x$, the output of the SCL that is passed to the next layer.

SCA architecture. The SCA architecture is defined by the use of multiple *pairs of sparse coding and dense (batch norm) layers* after the input image, which can then be followed by other (linear, convolutional) layers. Fig. 2 illustrates this design principle. The *key intuition* is that the first sparse layer jettisons unnecessary private information in the input image. Then, by alternating sparse-dense pairs of layers, we ensure that unnecessary information is also jettisoned from downstream layers. In this manner, downstream layers also do not convey unnecessary private information to the adversary, and they also do not e.g. learn to reconstruct private information jettisoned by the first sparse layer. In short, previous defenses work by trying to mislead attackers by pushing model features in a wrong direction, either randomly via noise or strategically via adversarial examples, or by disincentivizing memorization during training. In contrast, SCA directly removes the unnecessary private information. Training SCA is identical to training a standard network with one exception: after each backprop updates non-sparse layers, we perform a fast update on the sparse layers, except for the first sparse layer that sparse-codes the image input.[3]

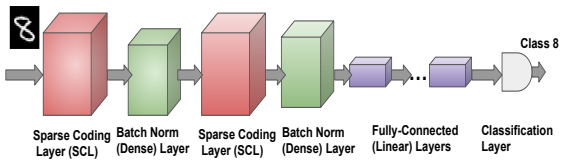


Fig. 2: Architecture of SCA.

SCA training complexity & large scale applications. While we focus on the neuron lateral competition approach to sparse coding as it is practically convenient and well-represented in recent work [63], we note that for large-scale machine learning applications, we now have practical parallel algorithms that learn the sparse coding dictionary near-optimally w.p. in parallel time (adaptivity) that is logarithmic in the size of the data [8, 13, 32]. Fast single-iteration heuristics are also available (see e.g. [72]). Thus, even for large-scale applications, computing sparse representations while training SCA adds little computational overhead compared to sophisticated optimization-based techniques necessary for recent defenses [22]. In practice, even our basic sparse coding research implementations (see Section 4 and Appendix A.3) are slightly faster than optimized Torch implementations of the best-performing recent defense [53].

[3] We can optionally also perform a backpropagation on sparse layers after updating them each iteration via sparse coding. We do this in our experiments.

4 Experiments

xOur goal in this section is to show that SCA performs well compared to state-of-the-art defenses as well as practical defenses used in leading industry models in terms of both classification accuracy and various reconstruction quality metrics. To evaluate its performance comprehensively, we test *all combinations* of 5 diverse datasets, 3 threat models, 9 defense baselines, plus multiple runs-per-experiment and various sparsity parameters λ. See Appendix A.1-A.6.

Five benchmark datasets. We test our performance on *all* **5** diverse datasets used to benchmark model inversion attacks across the recent literature: **CelebA** hi-res RGB faces, **Medical MNIST** medical images, **CIFAR-10** hi-res RGB objects, **MNIST** grayscale digits, and **Fashion MNIST** grayscale objects.

Three attacks. For each dataset, we conduct three sets of experiments corresponding to our three threat models. First, we test SCA and baselines' defenses against a state-of-the-art *Plug-&-Play attack [61] that leverages StyleGAN3* to obtain high-quality reconstructions. Second, we compare SCA networks to a variety of baselines in terms of their robustness to a state-of-the-art *end-to-end network attack that leverages leaked raw high-dimensional outputs from the networks' last hidden layer, as well as held-out training data drawn from the true training dataset.* This allows us to assess SCA's defenses in a realistic setting with a well-informed adversary. Our third set of experiments tests performance in a *split network setting of [64] where the attacker has access to leaked raw outputs from the* first *linear network layer*. Robustness in this setting is desirable because model inversion attacks are known to be more effective on earlier hidden layers [26], and also because an algorithm that is robust to such attacks would be an effective defense under novel security paradigms such as Federated Learning, which is vulnerable to inversion [64] (see Appendix A.5).

Nine defense baselines. We compare SCA to 9 baselines plus extra variants, including SOTA defenses and practical defenses used in leading industry models:

- **No-Defense**. The baseline target model with no added defenses.
- **Hayes et al.** [25]. We train a DP-SGD defense that noises and clips gradients during training. This is the only defense with provable guarantees.
- **Gong et al.** [22]. We train the very recent defense from [22] that uses sophisticated tuning and two types of GAN-generated images. We also try a '++' version that adds extra Continual Learning accuracy optimizations.
- **Peng et al.** [53]. We train the Bilateral Dependency defense that adds a loss function for redundant input memorization during training.
- **Wang et al.** [66]. We train a Mutual Information Regularization defense that penalizes dependence between inputs and outputs during training.
- **Titcombe et al.** [64]. We train a state-of-the-art Laplace $\mathcal{L}(\mu{=}0, b{=}0.5)$ noise defense as in [64]. We also try more noise—see Appendix D.
- **Sparse-Standard**. We train an off-the-shelf sparse coding architecture [63] with 1 sparse layer after the input image via lateral competition as in SCA.

- **GAN [common industry defense]**. We train a GAN for 25 epochs to generate fake samples, then train the target model with both original and GAN-generated samples. This defense is frequently used in industry.
- **Gaussian-Noise [common industry defense]**. We draw noises from $\mathcal{N}(\mu{=}0,\ \sigma{=}0.5)$ and inject them into intermediate dense layers post-training.

SCA* without *parameter tuning. In all experiments, we consider the simplest case of SCA architecture that contains SCA's alternating sparse-and-dense layer pairs followed by only linear layers. We note that adding downstream convolutional layers or more sophisticated downstream architectures is certainly possible, though we avoid this here in order to compare the essence of the SCA approach to the baselines. Appendix A.4 describes SCA details. In the split network setting, we are careful to use slightly shallower SCA architectures with fewer linear layers to match the split network experiments of [64].

Recent state-of-the-art defenses such as GAN-based defenses require sophisticated automatic parameter tuning techniques such as focal tuning and continual learning to obtain high performance [22]. To test whether SCA can be effective *absent* parameter tuning, we just run SCA with sparsity parameter λ set to $\mathbf{0.1}$, $\mathbf{0.25}$, or $\mathbf{0.5}$—the default values from various sparse coding contexts.

Performance metrics. We evaluate attackers' reconstructions using multiple standard metrics. Let X^*_{in} denote the reconstruction of training image X_{in}. Then:

- **Peak signal-to-noise ratio (PSNR)** [*lower=better*]. PSNR is the ratio of max squared pixel fluctuations from X_{in} to $X{*}_{in}$ over mean squared error.
- **Structural similarity (SSIM)** [70] [*lower=better*]. SSIM measures the product of luminance distortion, contrast distortion, & correlation loss: $SSIM(X_{in}, X^*_{in}) = l_{dis}(X_{in}, X^*_{in})c_{dis}(X_{in}, X^*_{in})c_{loss}(X_{in}, X^*_{in})$.
- **Fréchet inception distance (FID)** [28] [*higher=better*]. FID measures reconstruction quality as a distributional difference between X_{in} and X^*_{in}: $FID^2(X_{in}, X^*_{in}) = ||\mu_{X_{in}} - \mu_{X^*_{in}}||^2 + Tr(\mathrm{Co}_{X_{in}} + \mathrm{Co}_{X^*_{in}} - 2*\sqrt{\mathrm{Co}_{X_{in}} \cdot \mathrm{Co}_{X^*_{in}}})$

Target model. We focus on privacy attacks on linear networks because they capture the essence of the privacy attack vulnerability [20,29], and because there is broad consensus that a principled understanding of their emerging privacy (and security) vulnerabilities[4] is urgently needed [27, 42, 57, 73].

PyTorch codebase, replicability, and evaluation standardization. For all experiments, we consider the standard train test split of 70% and 30%. After training each defense model, we run attacks to reconstruct the entire training set and compare reconstruction performance. We run all the experiments on a standard industry production cluster with 4 nodes and DELL Tesla V100 GPUs with 40 cores. *We provide a cluster-ready PyTorch codebase on our project page at:* https://sayantondibbo.github.io/SCA.

[4] We also note that results on linear models may generalize better than results on more application-specific models, and linear models trained on private data remain ubiquitous among top industry products.

4.1 Experimental results overview

Defense. Across the 3 attack settings and 5 datasets, SCA maintains comparable or higher classification accuracy while degrading state-of-the-art training data reconstructions 1.1 to 11.7 times more than the 9 baselines in terms of PSNR & FID, and 1.1 to 720 times more in terms of SSIM. This performance gap exceeds the scale of improvements made by recent algorithms. SCA's defense is also more stable than baselines across multiple runs. This is because unlike for baselines, even an idealized 'perfect attacker' can only hope to recover a sparsified, un-detailed training image from SCA. We show results here for the 2 most privacy-sensitive datasets of medical images and CelebA in 3 threat models, and defer the 9 other {dataset, attack} combinations to Appendix C.

Accuracy. Typically, obtaining greater defense means trading away accuracy. However, in 6 of the 15 experiments (MNIST + FashionMNIST)×(Plug & Play + end-to-end + split networks), SCA *outperforms no-defense* ***and*** *all baselines' accuracy.* SCA also outperforms all baselines' accuracy on CelebA in Plug & Play, and a sparse approach is within 0.003 of the best accuracy on an 8th experiment. *No other single SOTA baseline wins on accuracy this consistently.* We emphasize that unlike baselines that do accuracy hyperparameter tuning, we obtain this result *absent* any such tuning. SCA drops accuracy on MedMNIST (which is the most imbalanced & has fewest training examples). However, tuning of SCA (kernel size 5 → 7) improves SCA's accuracy on MedMNIST in Table 3 from 94.6% to 97%—See Appendix G and Table 12.

Results of experiments set 1: Plug-&-Play attack.

Qualitative evaluations. Fig. 3 shows hi-res CelebA reconstructions generated by Plug-&-Play under various defenses. To avoid cherry-picking, Fig. 3 shows the 3 images with the highest (top row), median (middle), and lowest (bottom) PSNR reconstructions under No-Defense. Note that reconstructions under SCA totally differ from actual images (different race, hair, gender, child/adult), while those of other defenses are closer to actual images, indicating privacy leakage.

Metric evaluations. Table 1 reports reconstruction quality measures and accuracy for SCA and other baselines in the Plug-&-Play attack [61] setting (*lower rows = better defense performance*). In terms of PSNR and SSIM, training data

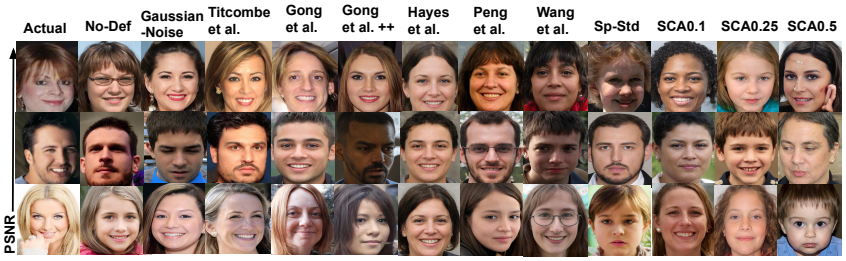


Fig. 3: Experiments set 1: Qualitative comparisons among actual and reconstructed images (Plug-&-Play Attack [61]) under SCA & baselines on hi-res CelebA dataset.

Table 1: Experiments set 1: Performance comparison under Plug-&-Play attack [61] setting *(lower rows=better defense)* on hi-res CelebA faces and Medical MNIST images.

Dataset	Defense	PSNR ↓↓	SSIM ↓↓	FID ↑↑	Accuracy
CelebA	No-Defense	11.35	0.718	256.4	0.779
	Gaussian-Noise	10.39	0.604	264.8	0.644
	GAN	10.15	0.613	289.7	0.635
	Titcombe et al. [64]	10.18	0.636	304.1	0.654
	Gong et al. [22]++	10.02	0.556	381.5	0.672
	Gong et al. [22]	10.11	0.595	350.5	0.614
	Peng et al. [53]	9.90	0.514	402.8	0.728
	Hayes et al. [25]	9.92	0.556	383.7	0.621
	Wang et al. [66]	9.85	0.527	402.8	0.742
	Sparse-Standard	9.84	0.539	374.7	0.728
	SCA0.1	**9.79**	**0.451**	**391.9**	**0.726**
	SCA0.25	**9.45**	**0.442**	**411.0**	**0.739**
	SCA0.5	**9.40**	**0.440**	**412.6**	**0.723**
Medical MNIST	No-Defense	22.04	0.396	196.1	0.998
	Gaussian-Noise	21.83	0.382	209.4	0.862
	GAN	21.77	0.427	219.0	0.998
	Gong et al. [22]++	21.50	0.359	273.1	0.894
	Titcombe et al. [64]	21.68	0.360	286.3	0.899
	Gong et al. [22]	21.75	0.477	249.1	0.770
	Peng et al. [53]	21.82	0.381	268.3	0.927
	Hayes et al. [25]	21.72	0.337	259.7	0.823
	Wang et al. [66]	21.71	0.322	211.7	0.937
	Sparse-Standard	20.97	0.086	239.3	0.907
	SCA0.1	**21.19**	**0.057**	**253.5**	**0.888**
	SCA0.25	**21.17**	**0.075**	**280.1**	**0.882**
	SCA0.5	**20.06**	**0.055**	**288.8**	**0.881**

reconstructions under the *least sparse* version SCA0.1 are degraded by factors of 1.01 to 6.7 and 1.01 to 5.6 compared to the regularization defenses of Peng et al. [53] and Wang et al. [66], respectively. Increasing SCA's sparsity λ to 0.5 widens the performance gap, increasing these factors to 1.1 to 6.9 and 1.02 to 5.9, respectively, and making SCA outperform baselines' FID. SCA0.1 also outperforms the noise-based approaches of Hayes et al. [25] and Titcombe et al. [64] on all metric by factors of 1.01 to 5.9 and 1.02 to 6.3. Increasing SCA's sparsity λ to 0.5 increases these factors to 1.1 to 6.1 and 1.1 to 6.5. Finally, SCA0.1 outperforms the data augmentation defense of [22] by factors of 1.01 to 8.4, which widens to 1.1 to 8.7 for SCA0.5. All baselines also outperform common GAN and Noise-based industry defenses.

Basic Sparse-Standard outperforms SOTA baselines. Our basic Sparse-Standard baseline outperforms the best baselines' PSNR on both CelebA and Medical MNIST, and also outperforms baselines' SSIM on the latter. SCA0.5 then outperforms Sparse-Standard on *all metrics*, obtaining SSIM and FID that are better by factors of 1.2 to 1.6 and 1.1 to 1.2, respectively, and slightly better PSNR. Thus, while Sparse-Standard offers an inferior defense vs. SCA, it still offers a fast practical defense for less privacy-critical domains.

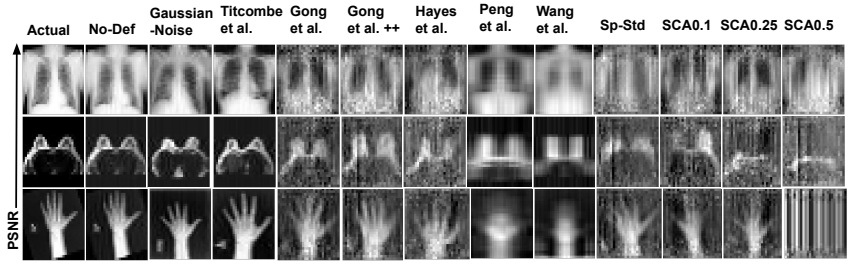


Fig. 4: Experiments set 2: Qualitative comparisons among actual & reconstructed images (*end-to-end* setting) under SCA & baselines on the Medical MNIST dataset.

Table 2: Experiments set 2: Performance comparison in *end-to-end* setting *(lower rows=better defense)* on hi-res CelebA faces and Medical MNIST images.

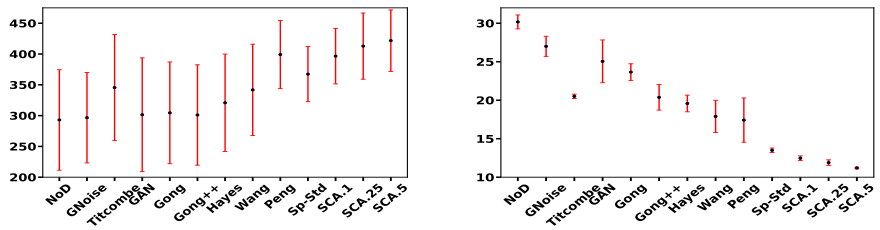
Dataset	Defense	PSNR ↓↓	SSIM ↓↓	FID ↑↑	Accuracy
CelebA	No-Defense	16.26	0.262	201.8	0.773
	Gaussian-Noise	16.08	0.262	220.4	0.638
	GAN	13.55	0.133	199.6	0.668
	Titcombe et al. [64]	15.13	0.191	197.7	0.695
	Gong et al. [22]++	13.10	0.032	204.8	0.704
	Gong et al. [22]	13.15	0.119	199.6	0.682
	Peng et al. [53]	13.78	0.141	218.8	0.716
	Hayes et al. [25]	14.10	0.004	199.0	0.664
	Wang et al. [66]	13.63	0.0011	203.2	0.744
	Sparse-Standard	13.09	0.002	222.1	0.749
	SCA0.1	**12.89**	**0.004**	**228.5**	**0.748**
	SCA0.25	**12.73**	**0.004**	**218.8**	**0.737**
	SCA0.5	**12.42**	**0.001**	**231.9**	**0.741**
Medical MNIST	No-Defense	31.48	0.935	10.66	0.998
	Gaussian-Noise	30.46	0.920	12.23	0.862
	GAN	27.34	0.480	33.77	0.998
	Gong et al. [22]++	18.37	0.353	81.52	0.894
	Titcombe et al. [64]	21.33	0.431	30.60	0.899
	Gong et al. [22]	21.52	0.436	64.88	0.770
	Peng et al. [53]	19.05	0.420	107.9	0.908
	Hayes et al. [25]	18.48	0.007	150.9	0.824
	Wang et al. [66]	20.48	0.549	30.01	0.946
	Sparse-Standard	14.79	0.119	250.6	0.907
	SCA0.1	**13.43**	**0.004**	**352.1**	**0.888**
	SCA0.25	**12.32**	**0.004**	**375.9**	**0.882**
	SCA0.5	**12.04**	**0.003**	**369.9**	**0.881**

Results of experiments set 2: End-to-end networks.

Qualitative evaluations. Fig. 4, shows Medical MNIST reconstructions in the end-to-end threat model. SCA's reconstructions are visually destroyed (esp. SCA0.5), whereas baselines admit noisy-but-recognizable reconstructions.

Metric evaluations. Table 2 reports performance in the end-to-end network setting. In this setting, SCA's performance advantage *widens*. In terms of PSNR & FID, training data reconstructions under SCA0.1 are degraded by factors of 1.1 to 3.3 and 1.1 to 11.7 compared to Peng et al. [53] and Wang et al. [66], respectively (but slightly worse SSIM vs. Wang et al. on CelebA). On all metrics, SCA0.1 outperforms Hayes et al. [25] and Titcombe et al. [64] by factors of 1.1 to 2.4 and 1.1 to 107.7, respectively. SCA0.1 also outperforms Gong et al. [22] by factors of 1.01 to 109. Increasing SCA's λ to 0.5 causes it to outperform the same baselines *on all metrics* by factors of 1.2 to 141 [53], 1.2 to 183 [66], 1.2 to 4.0 [25], 1.2 to 191 [64], and 1.1 to 145.3 [22], respectively.

Stability of SCA's defense vs. baselines. SCA's performance is also as stable or more stable than baselines' performance over multiple runs. For example, Fig. 5a plots the means & std. devs. of SCA & baselines' per-run FID (the standard metric for faces) over multiple runs for CelebA faces in the Plug-&-Play setting, and Fig. 5b plots this for PSNR (the standard metric for grayscale images) on Medical MNIST in the end-to-end setting. SCA obtains better performance while also exhibiting stability of defense performance on par with the best baseline (and better stability than most baselines). See Appendix E.



(a) FID↑↑ for CelebA faces under Plug-&-Play **(b)** PSNR↓↓ for Med.MNIST, end-to-end setting

Fig. 5: Stability of SCA & baselines' defense performance (mean $\pm$ std. dev.) of PSNR and FID across multiple runs on CelebA and Medical MNIST.

SCA's sparsity vs. performance. We also try varying SCA's and Sparse-Standard's sparsity parameters λ and recompute PSNR, SSIM, FID, and accuracy. Appendix A.6 shows that for each λ and defense metric, SCA significantly outperforms the off-the-shelf Sparse-Standard architecture for a small accuracy cost. Thus, for a given λ with Sparse-Standard, we can use a (smaller) λ with SCA to obtain better reconstruction *and* higher or equal (within 0.0017) accuracy. SCA is also amenable to more sophisticated tuning (and performance improvements) by tuning different λ per sparse layer (e.g., by having a sparser representation of inputs but less sparse reductions of downstream layers). We *avoid* such tuning here as it is unnecessary for good performance.

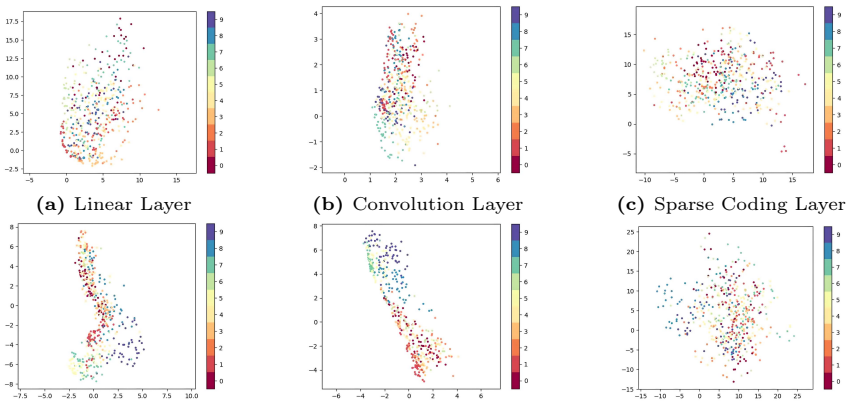
Table 3: Experiments set 3: Performance comparison in *split network* setting *(lower rows=better defense)* on hi-res CelebA faces and sensitive Medical MNIST images.

Dataset	Defense	PSNR ↓↓	SSIM ↓↓	FID ↑↑	Accuracy
CelebA	No-Defense	16.49	0.302	185.8	0.766
	Gaussian-Noise	15.44	0.227	191.1	0.753
	GAN	15.57	0.253	176.7	0.646
	Titcombe et al. [64]	14.99	0.144	194.2	0.725
	Gong et al. [22]++	15.06	0.038	190.5	0.756
	Gong et al. [22]	15.65	0.044	185.8	0.653
	Peng et al. [53]	16.23	0.211	198.6	0.717
	Hayes et al. [25]	15.06	0.005	178.8	0.672
	Wang et al. [66]	14.82	0.173	189.6	0.652
	Sparse-Standard	15.39	0.009	187.0	0.746
	SCA0.1	**15.05**	**0.005**	**178.7**	**0.745**
	SCA0.25	**14.76**	**0.003**	**191.1**	**0.743**
	SCA0.5	**14.71**	**0.003**	**206.1**	**0.739**
Medical MNIST	No-Defense	23.47	0.776	45.57	0.993
	Gaussian-Noise	21.93	0.722	44.72	0.811
	GAN	21.67	0.719	48.49	0.912
	Gong et al. [22]++	21.07	0.573	67.53	0.931
	Titcombe et al. [64]	21.35	0.704	48.82	0.961
	Gong et al. [22]	21.33	0.720	41.74	0.925
	Peng et al. [53]	18.98	0.426	124.8	0.914
	Hayes et al. [25]	21.46	0.442	137.4	0.850
	Wang et al. [66]	20.03	0.538	65.17	0.986
	Sparse-Standard	15.33	0.149	142.4	0.955
	SCA0.1	**13.95**	**0.008**	**244.9**	**0.946**
	SCA0.25	**12.31**	**0.008**	**255.3**	**0.928**
	SCA0.5	**12.27**	**0.001**	**285.3**	**0.909**

Results of experiments set 3: Split networks. Table 3 reports performance in the split network setting. As expected, all baselines and SCA perform slightly worse in this threat model compared to the end-to-end model (aside from Guassian and GAN heuristics). On all metrics, SCA's performance advantage remains consistent: SCA0.5 outperforms all baselines by factors of 1.1 to 720.

5 Empirical analysis of sparse coding robustness to attack

Sparse-coding layers' robustness to privacy attacks can be observed empirically. Consider that the attacker trains the attack to map leaked raw hidden layer outputs back to input images. Attacks are thus highly dependent on these outputs' distributions. Recall that UMAP projections compute a 2D visualization of the global structure of distances between different training images' features according to a particular layer [43]. Fig. 6 plots UMAP 2D projections of linear layer feature distributions of training inputs *after* either two linear layers (Figs. 6a & 6d), two convolutional layers (Figs. 6b & 6e), or two sparse coding layers (with



(a) Linear Layer **(b)** Convolution Layer **(c)** Sparse Coding Layer

(d) Linear Layer **(e)** Convolution Layer **(f)** Sparse Coding Layer

Fig. 6: UMap 2D projections of input images' features by class after 2 linear layers, 2 conv. layers, or 2 sparse-coded layers on MNIST (top) & Fashion MNIST (bottom).

interspersed dense layers – Figs. 6c & 6f). Importantly, observe that after two linear or two convolutional layers, points are clustered by color, i.e., input images' features are highly clustered by label. This class-clustered property leaves such layers vulnerable to model inversion attacks, as an attacker can 'home in on' examples from a specific class. In contrast, the goal in sparse coding is not to optimize the classification objective by separating classes, but rather to jettison unnecessary information. Here, this means that unnecessary information is jettisoned both from the input image and also the downstream dense layer. Per Figs. 6c & 6f, this tends to 'uncluster' remaining non-sparsified features of training examples from the same class, making it much harder for an attacker to compute informative gradients to home in on a training example.

6 Discussion & Conclusion

In this paper, we have provided the first study of sparse coding-based neural network architectures that are robust to model inversion attacks. Specifically, we have shown that the natural properties of sparse coded layers can control the extraneous private information about the training data that is encoded in a network without resorting to complex and computationally intensive parameter tuning techniques. Our work reveals a deep connection between state-of-the-art privacy vulnerabilities and three decades of computer science research on sparse coding for other application domains. Currently, our basic research implementation of SCA achieves compute times that in the worst-case are no better than some SOTA baselines (see Appendix F). However, given the rich theoretic body of work on fast algorithms and provable guarantees for sparse coding, we believe these aspects are opportune areas for future improvements.

Acknowledgements

We are grateful for generous support from OpenAI, as well as the Dartmouth College Cybersecurity Cluster Research. This work was partially funded by the Center for Nonlinear Studies and the Information Science and Technology Institute's Cyber Security Summer School at Los Alamos National Laboratory, as well as an award from the Department of Energy's Advanced Scientific Computing Research program (#77902).

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. pp. 308–318 (2016) 4
2. Abuadbba, S., Kim, K., Kim, M., Thapa, C., Camtepe, S.A., Gao, Y., Kim, H., Nepal, S.: Can we use split learning on 1d cnn models for privacy preserving training? In: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. pp. 305–318 (2020) 2
3. Ahmad, S., Scheinkman, L.: How can we be so dense? the benefits of using highly sparse representations. arXiv preprint arXiv:1903.11257 (2019) 2
4. Aïvodji, U., Gambs, S., Ther, T.: Gamin: An adversarial approach to black-box model inversion. arXiv preprint arXiv:1909.11835 (2019) 2
5. An, S., Tao, G., Xu, Q., Liu, Y., Shen, G., Yao, Y., Xu, J., Zhang, X.: Mirror: Model inversion for deep learning network with high fidelity. In: Proceedings of the 29th Network and Distributed System Security Symposium (2022) 2
6. Barlow, H.B.: The coding of sensory messages. Current problems in animal behavior (1961) 2
7. Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečnỳ, J., Mazzocchi, S., McMahan, B., et al.: Towards federated learning at scale: System design. Proceedings of machine learning and systems **1**, 374–388 (2019) 4
8. Breuer, A., Balkanski, E., Singer, Y.: The fast algorithm for submodular maximization. In: International Conference on Machine Learning. pp. 1134–1143. PMLR (2020) 3, 6
9. Candès, E.J., Donoho, D.L.: New tight frames of curvelets and optimal representations of objects with piecewise c2 singularities. Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences **57**(2), 219–266 (2004) 2, 3
10. Carlini, N., Jagielski, M., Zhang, C., Papernot, N., Terzis, A., Tramer, F.: The privacy onion effect: Memorization is relative. Advances in Neural Information Processing Systems **35**, 13263–13276 (2022) 2
11. Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., Wallace, E.: Extracting training data from diffusion models. In: 32nd USENIX Security Symposium (USENIX Security 23). pp. 5253–5270 (2023) 1
12. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM review **43**(1), 129–159 (2001) 2

13. Chen, Y., Dey, T., Kuhnle, A.: Best of both worlds: Practical and theoretically optimal submodular maximization in parallel. Advances in Neural Information Processing Systems **34**, 25528–25539 (2021) 3, 6
14. Davis, G., Mallat, S., Avellaneda, M.: Adaptive greedy approximations. Constructive approximation **13**, 57–98 (1997) 3
15. Dibbo, S.V.: Sok: Model inversion attack landscape: Taxonomy, challenges, and future roadmap. In: IEEE 36th Computer Security Foundations Symposium. pp. 408–425. IEEE Computer Society (2023) 1
16. Dibbo, S.V., Chung, D.L., Mehnaz, S.: Model inversion attack with least information and an in-depth analysis of its disparate vulnerability. In: 2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML). pp. 119–135. IEEE (2023) 1
17. Dibbo, S.V., Moore, J.S., Kenyon, G.T., Teti, M.A.: Lcanets++: Robust audio classification using multi-layer neural networks with lateral competition. arXiv preprint arXiv:2308.12882 (2023) 2
18. Fang, H., Chen, B., Wang, X., Wang, Z., Xia, S.T.: Gifd: A generative gradient inversion method with feature domain optimization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4967–4976 (2023) 2, 4
19. Field, D.J.: What is the goal of sensory coding? Neural computation **6**(4), 559–601 (1994) 2, 3
20. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. pp. 1322–1333 (2015) 1, 2, 8
21. Gong, N.Z., Liu, B.: You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors. In: 25th USENIX Security Symposium (USENIX Security 16). pp. 979–995 (2016) 1
22. Gong, X., Wang, Z., Li, S., Chen, Y., Wang, Q.: A gan-based defense framework against model inversion attacks. IEEE Transactions on Information Forensics and Security (2023) 2, 6, 7, 8, 10, 11, 12, 13
23. Haim, N., Vardi, G., Yehudai, G., Shamir, O., Irani, M.: Reconstructing training data from trained neural networks. Advances in Neural Information Processing Systems **35**, 22911–22924 (2022) 2
24. Hannan, D., Nesbit, S.C., Wen, X., Smith, G., Zhang, Q., Goffi, A., Chan, V., Morris, M.J., Hunninghake, J.C., Villalobos, N.E., et al.: Mobileptx: sparse coding for pneumothorax detection given limited training examples. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 15675–15681 (2023) 2
25. Hayes, J., Mahloujifar, S., Balle, B.: Bounding training data reconstruction in dp-sgd. arXiv preprint arXiv:2302.07225 (2023) 2, 4, 7, 10, 11, 12, 13
26. He, Z., Zhang, T., Lee, R.B.: Model inversion attacks against collaborative inference. In: Proceedings of the 35th Annual Computer Security Applications Conference. pp. 148–162 (2019) 2, 4, 7
27. Heredia, L.G., Negrevergne, B., Chevaleyre, Y.: Adversarial attacks for mixtures of classifiers. arXiv preprint arXiv:2307.10788 (2023) 8
28. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017) 8
29. Hidano, S., Murakami, T., Katsumata, S., Kiyomoto, S., Hanaoka, G.: Model inversion attacks for prediction systems: Without knowledge of non-sensitive attributes.

In: 2017 15th Annual Conference on Privacy, Security and Trust (PST). pp. 115–11509. IEEE (2017) 8
30. Hitaj, B., Ateniese, G., Perez-Cruz, F.: Deep models under the gan: information leakage from collaborative deep learning. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. pp. 603–618 (2017) 2
31. Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P.S., Zhang, X.: Membership inference attacks on machine learning: A survey. ACM Computing Surveys (CSUR) **54**(11s), 1–37 (2022) 1
32. Jiang, Z., Zhang, G., Davis, L.S.: Submodular dictionary learning for sparse coding. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3418–3425. IEEE (2012) 3, 6
33. Kahla, M., Chen, S., Just, H.A., Jia, R.: Label-only model inversion attacks via boundary repulsion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15045–15053 (2022) 2
34. Kaissis, G.A., Makowski, M.R., Rückert, D., Braren, R.F.: Secure, privacy-preserving and federated machine learning in medical imaging. Nature Machine Intelligence **2**(6), 305–311 (2020) 4
35. Kariyappa, S., Prakash, A., Qureshi, M.K.: Maze: Data-free model stealing attack using zeroth-order gradient estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13814–13823 (2021) 1
36. Kavukcuoglu, K., Ranzato, M., LeCun, Y.: Fast inference in sparse coding algorithms with applications to object recognition. arXiv preprint arXiv:1010.3467 (2010) 2, 3
37. Kim, E., Rego, J., Watkins, Y., Kenyon, G.T.: Modeling biological immunity to adversarial examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4666–4675 (2020) 2, 5
38. Konečnỳ, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492 (2016) 4
39. Krause, A., Cevher, V.: Submodular dictionary selection for sparse representation. In: International Conference on Machine Learning (ICML) (2010) 2, 3
40. Lee, H., Battle, A., Raina, R., Ng, A.: Efficient sparse coding algorithms. Advances in neural information processing systems **19** (2006) 3
41. Li, L., Xie, T., Li, B.: Sok: Certified robustness for deep neural networks. In: 2023 IEEE Symposium on Security and Privacy (SP). pp. 1289–1310. IEEE (2023) 1
42. Liu, G., Wang, C., Peng, K., Huang, H., Li, Y., Cheng, W.: Socinf: Membership inference attacks on social media health data with machine learning. IEEE Transactions on Computational Social Systems **6**(5), 907–921 (2019) 8
43. McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018) 13
44. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017) 4
45. Mehnaz, S., Dibbo, S.V., Kabir, E., Li, N., Bertino, E.: Are your sensitive attributes private? novel model inversion attribute inference attacks on classification models. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 4579–4596. USENIX Association, Boston, MA (Aug 2022) 1, 2
46. Melis, L., Song, C., De Cristofaro, E., Shmatikov, V.: Exploiting unintended feature leakage in collaborative learning. In: 2019 IEEE symposium on security and privacy (SP). pp. 691–706. IEEE (2019) 2

47. Mireshghallah, F., Taram, M., Ramrakhyani, P., Jalali, A., Tullsen, D., Esmaeilzadeh, H.: Shredder: Learning noise distributions to protect inference privacy. In: Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems. pp. 3–18 (2020) 2
48. Mirzasoleiman, B., Badanidiyuru, A., Karbasi, A., Vondrák, J., Krause, A.: Lazier than lazy greedy. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 29 (2015) 3
49. Natarajan, B.K.: Sparse approximate solutions to linear systems. SIAM journal on computing **24**(2), 227–234 (1995) 3
50. Olshausen, B.A., Field, D.J.: Sparse coding of sensory inputs. Current opinion in neurobiology **14**(4), 481–487 (2004) 2, 3
51. Olshausen, B.A., Field, D.J., et al.: Sparse coding of natural images produces localized, oriented, bandpass receptive fields. Submitted to Nature. Available electronically as ftp://redwood. psych. cornell. edu/pub/papers/sparse-coding. ps (1995) 2
52. Paiton, D.M., Frye, C.G., Lundquist, S.Y., Bowen, J.D., Zarcone, R., Olshausen, B.A.: Selectivity and robustness of sparse coding networks. Journal of vision **20**(12), 10–10 (2020) 2
53. Peng, X., Liu, F., Zhang, J., Lan, L., Ye, J., Liu, T., Han, B.: Bilateral dependency optimization: Defending against model-inversion attacks. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 1358–1367 (2022) 2, 6, 7, 10, 11, 12, 13
54. Rigaki, M., Garcia, S.: A survey of privacy attacks in machine learning. ACM Computing Surveys (2020) 2
55. Rozell, C.J., Johnson, D.H., Baraniuk, R.G., Olshausen, B.A.: Sparse coding via thresholding and local competition in neural circuits. Neural computation **20**(10), 2526–2563 (2008) 2, 3, 5
56. Salem, A., Bhattacharya, A., Backes, M., Fritz, M., Zhang, Y.: {Updates-Leak}: Data set inference and reconstruction attacks in online learning. In: 29th USENIX security symposium (USENIX Security 20). pp. 1291–1308 (2020) 2
57. Sannai, A.: Reconstruction of training samples from loss functions. arXiv preprint arXiv:1805.07337 (2018) 8
58. Sanyal, S., Addepalli, S., Babu, R.V.: Towards data-free model stealing in a hard label setting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15284–15293 (2022) 1
59. Schneiderman, H.: Feature-centric evaluation for efficient cascaded object detection. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. vol. 2, pp. II–II. IEEE (2004) 2
60. Song, L., Mittal, P.: Systematic evaluation of privacy risks of machine learning models. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 2615–2632 (2021) 4
61. Struppek, L., Hintersdorf, D., Correira, A.D.A., Adler, A., Kersting, K.: Plug & play attacks: Towards robust and flexible model inversion attacks. In: International Conference on Machine Learning. pp. 20522–20545. PMLR (2022) 1, 2, 4, 7, 9, 10
62. Sun, B., Tsai, N.h., Liu, F., Yu, R., Su, H.: Adversarial defense by stratified convolutional sparse coding. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11447–11456 (2019) 2
63. Teti, M., Kenyon, G., Migliori, B., Moore, J.: Lcanets: Lateral competition improves robustness against corruption and attack. In: International Conference on Machine Learning. pp. 21232–21252. PMLR (2022) 2, 5, 6, 7

64. Titcombe, T., Hall, A.J., Papadopoulos, P., Romanini, D.: Practical defences against model inversion attacks for split neural networks. arXiv preprint arXiv:2104.05743 (2021) 2, 4, 7, 8, 10, 11, 12, 13
65. Vepakomma, P., Gupta, O., Swedish, T., Raskar, R.: Split learning for health: Distributed deep learning without sharing raw patient data. arXiv preprint arXiv:1812.00564 (2018) 4
66. Wang, T., Zhang, Y., Jia, R.: Improving robustness to model inversion attacks via mutual information regularization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 11666–11673 (2021) 2, 7, 10, 11, 12, 13
67. Wang, X., Wang, W.H.: Group property inference attacks against graph neural networks. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 2871–2884 (2022) 4
68. Wang, Y., Qian, H., Miao, C.: Dualcf: Efficient model extraction attack from counterfactual explanations. In: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency. pp. 1318–1329 (2022) 1
69. Wang, Z., Song, M., Zhang, Z., Song, Y., Wang, Q., Qi, H.: Beyond inferring class representatives: User-level privacy leakage from federated learning. In: IEEE INFOCOM 2019-IEEE conference on computer communications. pp. 2512–2520. IEEE (2019) 2
70. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004) 8
71. Wei, W., Liu, L., Loper, M., Chow, K.H., Gursoy, M.E., Truex, S., Wu, Y.: A framework for evaluating gradient leakage attacks in federated learning. arXiv preprint arXiv:2004.10397 (2020) 2
72. Wu, P., Liu, J., Li, M., Sun, Y., Shen, F.: Fast sparse coding networks for anomaly detection in videos. Pattern Recognition **107**, 107515 (2020) 6
73. Wu, Y., Yu, N., Li, Z., Backes, M., Zhang, Y.: Membership inference attacks against text-to-image generation models. arXiv preprint arXiv:2210.00968 (2022) 8
74. Yang, Z., Zhang, J., Chang, E.C., Liang, Z.: Neural network inversion in adversarial setting via background knowledge alignment. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 225–240 (2019) 2
75. Yuan, X., Ding, L., Zhang, L., Li, X., Wu, D.O.: Es attack: Model stealing against deep neural networks without data hurdles. IEEE Transactions on Emerging Topics in Computational Intelligence **6**(5), 1258–1270 (2022) 1
76. Zhang, J., Peng, S., Gao, Y., Zhang, Z., Hong, Q.: Apmsa: adversarial perturbation against model stealing attacks. IEEE Transactions on Information Forensics and Security **18**, 1667–1679 (2023) 1
77. Zhang, Y., Jia, R., Pei, H., Wang, W., Li, B., Song, D.: The secret revealer: Generative model-inversion attacks against deep neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 253–261 (2020) 2
78. Zhong, D., Sun, H., Xu, J., Gong, N., Wang, W.H.: Understanding disparate effects of membership inference attacks and their countermeasures. In: Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security. pp. 959–974 (2022) 1