Supplementary Material for 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

A Appendix

This is the supplementary document containing the additional results and details of our proposed Sparse Coding Architecture (SCA) formulations, as well as cluster details and additional preliminaries.

A.1 Reproducibility

In order to promote further research and standardize the evaluations of new defenses, we provide full cluster-ready PyTorch [23] implementations of SCA and all benchmarks as well as replication codes for all experiments on our project page at: https://sayantondibbo.github.io/SCA.

We provide full details of the cluster hardware and all parameter choices used in our experiments in *Appendix* A.3 and A.4, and in *Appendix* Tables 1 and 2.

A.2 Adapting Rozell LCA to Convolutional Networks

Although the original LCA formulation [25] was introduced for the non-convolutional case, it is based on the general principle of feature-similarity-based competition between neurons within the same layer, which can be adapted to the convolutional setting via only two minimal changes to Equation 1 [18, 29]. In Rozell's original formulation, $\Psi(t)$ can simply be recast from a matrix multiplication to a convolution between the input and dictionary. Second, the lateral interaction tensor, $\mathcal{G}$ in Equation 1, can also be recast from a matrix multiplication to a convolution between the dictionary and its transpose. Neuron membrane potential works as follows:

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

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 [18, 29].

A.3 Cluster Details

We run all our experiments using the slurm batch jobs on industry-standard high-performance GPU clusters with 40 cores and 4 nodes. Details of the hardware and architecture of our cluster are described in Table 1. We note that noise-based Gaussian and Titcombe et al. [30] defenses are typically fastest on this architecture (though they are the least-performant). We emphasize that our sparse coding implementations are 'research-grade', unlike the optimized torch GAN implementations available for [11]. See also *Appendix* G. Note that for large scale applications, SCA's sparse coding updates can be accelerated such that they can be computed extremely efficiently (see the training complexity discussion in the main paper body).

Table 1: Hardware Details of the Cluster in our Experiments.

Parameter	Measurements
Core	40
RAM	565GB
GPU	Tesla V100
Nodes	p01-p04
Space	1.5TB

A.4 Parameters and architecture of SCA

We implement SCA using two Sparse Coding Layers (SCL): One following the input image, and one following a downstream dense batch normalization layer. Finally, we follow these two pairs of dense-then-sparse layers with downstream fully connected (linear) layers before the classification layer. In the case of end-to-end network experiments, we use 5 downstream linear layers, which is a reasonable default. In the split network setting, we are careful to use 3 downstream fully connected layers in order to match the architectures used in the split network experimental setup of [30], and per our public codebase, we make every effort to make the benchmarks within each setting comparable in terms of architecture, aside from the obvious difference of SCA's sparse layers We train SCA's sparse layers with 500 iterations of lateral competitions during reconstructions in SCL layers. We emphasize that SCA can be made significantly more complex, either via the addition of more sparse-dense pairs of layers, or by adding additional (convolutional, linear) downstream layers before classification. We avoid such complexity in the experiments in order to compare more directly to benchmarks and because our goal is to study an architecture that captures the essence of SCA. We give all parameter and training details in Table 2.

Table 2: Architecture and Parameters of SCA implementation.

Parameter	VALUE
Sparse Layers	2
Batch Norm Layers	2
Fully Connected Layers	5
λ	0.5
Learning rate η	0.01
Time constant τ	1000
Kernel size	5
Stride	1,1
Lateral competition iterations	500

A.5 Attack details

In the Plug-&-Play attack experiments, we follow the authors' attack exactly [28], except we update their approach to use the latest **StyleGAN3** [17] for high-resolution image generation. For the end-to-end and split-network attacks, we consider a recent state-of-the-art surrogate model training attack optimized via SGD [1,39]. This attack works by querying the target model with an externally obtained dataset. To capture a well-informed 'worst-case' attacker, we set this dataset to a holdout set from the true training dataset. The attack then uses the corresponding model high-dimensional intermediate outputs to train an inverted surrogate model that outputs actual training data.

A.6 SCA sparsity vs. robustness

We vary the sparsity, i.e., λ parameter and run the SPARSE-STANDARD, as well as our SCA. We observe that increasing λ helps improve the robustness, without significant accuracy drops. For example, Table 3 shows this comparison for MNIST in the end-to-end setting.

B Model Inversion Attack Methodology: Additional discussion

Because privacy attacks are an emerging field, we feel it is relevant to include additional context and discussion here. Recent work has highlighted a variety of attack vectors targeting sensitive training data of machine learning models [3–7, 7,10,19,20,26,27,31–34,41,44]. These attacks not only target centralized models but also can make the federated learning models vulnerable to attacks [8, 13].

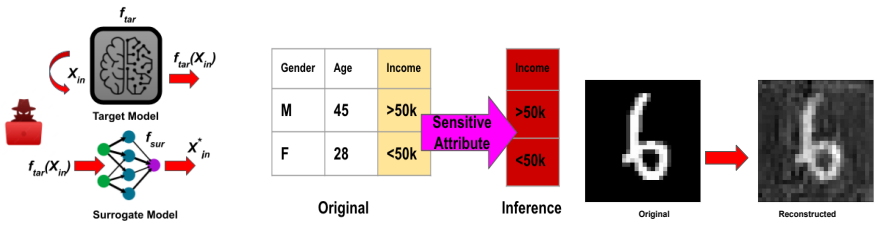
Table 3: SPARSE-STANDARD and SCA performance with $\lambda \in \{0.1, 0.25, 0.5, 0.75\}$

	PSNR↓↓		SSIM↓↓		FID ↑↑		Accuracy	
λ	SP-STD	SCA	SP-STD	SCA	SP-STD	SCA	SP-STD	SCA
0.1	23.45	19.54	0.650	0.502	111.5	178.5	0.984	0.984
0.25	21.34	18.81	0.438	0.340	142.9	174.1	0.986	0.983
0.5	22.16	17.85	0.598	0.164	136.9	**335.4**	0.985	0.977
0.75	22.39	**14.65**	0.593	**0.086**	142.0	214.1	0.981	0.971

Adversaries with different access (i.e., black-box, white-box) to these models perform different attacks leveraging a wide range of capabilities, e.g., knowledge about the target model confusion matrix and access to blurred images of that particular class [5, 9, 13, 16, 35]. Such attacks commonly fall under the umbrella of privacy attacks, which include specific attacker goals such as membership inference, model stealing, model inversion, etc. [14, 21, 37, 40]. Defending against privacy attacks is a core task of mainstream technology platforms ranging from public social networks to private medical research [2, 22, 38].

Our focus is model inversion attack, where an adversary aims to infer sensitive training data attributes X_s or reconstruct training samples X_{in}, a severe threat to the privacy of training data D_{Tr} [21, 30]. In Figure 1a, we present the pipelines of the model inversion attack. Depending on data types and purpose, model inversion attacks can be divided into two broader categories: (i) attribute inference (AttrInf) and (ii) image reconstruction (ImRec) attacks [6]. In AttrInf attacks, it is assumed the adversary can query the target model f_{tar} and design a surrogate model f_{sur} to infer some sensitive attributes X_s in training data D_{Tr}, with or without knowing all other non-sensitive attributes training data X_{ns} in the training data D_{Tr}, as presented in Figure 1b. In ImRec attacks the adversary reconstructs entire training samples D_{Tr} using the surrogate model f_{sur} with or without having access to additional information like blurred, masked, or noisy training samples D_s, as shown in Figure 1c [9, 41, 43]. To contextualize our SCA setting, recall that we suppose the attacker has only black-box access to query the model f_{tar} without knowing the details of the target model f_{tar} architecture or parameters like gradient information ∇_{Tr}. The attacker attempts to compute training data reconstruction (i.e., ImRec) attack without having access to other additional information, e.g., blurred or masked images D_s.

Two major components of the model inversion attack workflow are the target model f_{tar} and the surrogate attack model f_{sar} [6, 15, 42]. Training data reconstruction (i.e., ImRec) attack in the literature considers the target model f_{tar} to be either the split network [30] or the end-to-end network [11, 41]. In the split network f_{tar} model, the output of a particular layer l in the network, i.e., $a^{[l]}$, where $1 \leqslant l < L$ is accessible to the adversary, whereas, for the end-to-end network, the adversary does not have access to intermediate layer outputs; rather, the adversary only has access to the output from the last hidden layer before the classification layer $a^{[L]}$.



(a) Model Inversion Attack **(b)** Attribute Inference **(c)** Image Reconstruction

Fig. 1: Illustration of Model Inversion attack along with (a.) pipelines–an adversary queries target model f_{tar} with inputs $\mathcal{X}_{in}$ to obtain output $f_{tar}(X_{in})$. Then adversary trains a surrogate attack model f_{sar}, where the $f_{tar}(X_{in})$ is the input and $\mathcal{X}^*$ is the output; and (b.) categories, i.e., attribute inference (AttrInf) attack, where the adversary infers sensitive attribute $\mathcal{X}_s$ with or without knowing non-sensitive attribute values, i.e., $\mathcal{X}_{ns} \rightarrow \mathcal{X}_s$ and (c.) image reconstruction (ImRec) attack, where adversary reconstructs similar to original images, i.e., $\mathcal{X}_{in} \approx \mathcal{X}^*_{in}$.

C Results of extra {threat model, dataset} experiments

We experiment all 3 attack setups: *Plug-&-Play* model inversion attack [28], *end-to-end*, and *split* on three additional datasets: MNIST, Fashion MNIST, and CIFAR10. We experiment with all benchmarks and present the results on Tables 4, 5, and 6. In all of these additional datasets, SCA consistently outperforms all benchmarks.

D Additional baseline tuning

We also attempt to improve the Laplace noise-based defense of Titcombe et al. [30] by increasing the noise scale parameter b from $\mathcal{L}(\mu$=0, b=0.5) to $\mathcal{L}(\mu$=0, b=1.0). Tables 7, 8, and 9 compare these results to SCA for in all 3 attack settings. Observe that the additional noise significantly degrades classification accuracy in all but one case, yet it does not result in reconstruction metrics that rival those of SCA's. In Figure 2, we present the reconstructed images in the Split network attack setting on MNIST data. We also include the Laplace noise-based defense with higher noise parameter $\mathcal{L}(\mu$=0, b=1.0).

E Stability analysis of SCA

Tables 10 and 11 show mean metrics and std. deviation error bars taken over *multiple runs* of each defense. Observe that SCA is at least as stable (and in some cases significantly more stable) than alternatives.

Table 4: Experiments set 1 **Additional Datasets**: Performance in Plug-&-Play Model Inversion Attack [28] setting *(lower rows=better defense)*.

Dataset	Defense	PSNR ↓↓	SSIM ↓↓	FID ↑↑	Accuracy
MNIST	No-Defense	7.24	0.783	23.6	0.971
	Gaussian-Noise	6.94	0.686	31.22	0.958
	GAN	6.83	0.734	89.38	0.968
	Gong et al. [11]++	6.69	0.716	92.21	0.987
	Titcombe et al. [30]	6.34	0.744	131.8	0.980
	Gong et al. [11]	6.76	0.681	99.53	0.985
	Peng et al. [24]	6.89	0.704	283.8	0.941
	Hayes et al. [12]	7.03	0.672	396.1	0.871
	Wang et al. [36]	7.14	0.752	261.2	0.937
	Sparse-Standard	6.24	0.631	158.6	0.986
	SCA0.1	**6.19**	**0.633**	**287.9**	**0.984**
	SCA0.25	**5.83**	**0.607**	**289.3**	**0.983**
	SCA0.5	**5.74**	**0.604**	**299.6**	**0.977**
Fashion MNIST	No-Defense	8.91	0.147	235.5	0.886
	Gaussian-Noise	8.67	0.132	239.8	0.815
	GAN	8.66	0.147	243.3	0.883
	Gong et al. [11]++	8.73	0.130	220.2	0.906
	Titcombe et al. [30]	8.56	0.134	229.8	0.905
	Gong et al. [11]	8.57	0.143	244.3	0.888
	Peng et al. [24]	8.85	0.147	227.5	0.845
	Hayes et al. [12]	8.63	0.139	218.4	0.752
	Wang et al. [36]	8.90	0.119	210.3	0.880
	Sparse-Standard	8.71	0.135	223.3	0.879
	SCA0.1	**8.49**	**0.039**	**222.8**	**0.897**
	SCA0.25	**8.49**	**0.032**	**229.9**	**0.887**
	SCA0.5	**8.45**	**0.047**	**233.5**	**0.876**
CIFAR10	No-Defense	11.94	0.381	39.38	0.821
	Gaussian-Noise	11.88	0.365	77.92	0.626
	GAN	11.86	0.369	88.39	0.596
	Titcombe et al. [30]	10.89	0.346	79.19	0.792
	Gong et al. [11]++	11.06	0.339	78.48	0.773
	Gong et al. [11]	11.21	0.334	92.33	0.682
	Peng et al. [24]	11.96	0.354	120.5	0.752
	Hayes et al. [12]	11.12	0.342	142.1	0.626
	Wang et al. [36]	11.02	0.346	142.6	0.756
	Sparse-Standard	10.74	0.303	137.4	0.790
	SCA0.1	**10.59**	**0.305**	**144.1**	**0.787**
	SCA0.25	**10.27**	**0.279**	**189.9**	**0.772**
	SCA0.5	**10.23**	**0.276**	**189.7**	**0.744**

F Compute time

Our basic SCA research implementation completes in comparable or less compute time than highly optimized implementations of benchmarks. In the 'worst-

Table 5: Experiments set 2 **Additional Datasets**: Performance in *end-to-end* network setting *(lower rows=better defense)*.

Dataset	Defense	PSNR ↓↓	SSIM ↓↓	FID ↑↑	Accuracy
MNIST	No-Defense	40.87	0.982	16.31	0.971
	Gaussian-Noise	40.88	0.983	15.88	0.958
	GAN	40.69	0.981	16.59	0.968
	Titcombe et al. [30]	31.18	0.863	47.32	0.980
	Gong et al. [11]++	30.37	0.838	72.99	0.987
	Gong et al. [11]	29.05	0.817	75.39	0.985
	Peng et al. [24]	18.44	0.354	111.6	0.968
	Hayes et al. [12]	19.75	0.488	298.8	0.871
	Wang et al. [36]	27.26	0.862	72.66	0.962
	Sparse-Standard	21.34	0.439	142.9	0.986
	SCA0.1	**19.54**	**0.502**	**178.5**	**0.984**
	SCA0.25	**18.81**	**0.340**	**174.1**	**0.983**
	SCA0.5	**17.85**	**0.164**	**335.5**	**0.977**
Fashion MNIST	No-Defense	37.86	0.975	13.91	0.886
	Gaussian-Noise	36.54	0.969	16.49	0.815
	GAN	37.68	0.974	19.26	0.883
	Gong et al. [11]++	27.71	0.794	41.35	0.906
	Titcombe et al. [30]	26.66	0.759	53.76	0.905
	Gong et al. [11]	21.24	0.523	93.08	0.888
	Peng et al. [24]	17.98	0.368	70.53	0.880
	Hayes et al. [12]	21.13	0.297	223.3	0.752
	Wang et al. [36]	25.98	0.806	41.87	0.838
	Sparse-Standard	19.35	0.446	128.4	0.879
	SCA0.1	**17.92**	**0.209**	**196.1**	**0.897**
	SCA0.25	**17.03**	**0.186**	**195.2**	**0.887**
	SCA0.5	**14.51**	**0.069**	**423.2**	**0.876**
CIFAR10	No-Defense	21.17	0.477	70.96	0.821
	Gaussian-Noise	20.26	0.220	77.42	0.626
	GAN	19.71	0.259	132.0	0.596
	Titcombe et al. [30]	18.62	0.174	171.9	0.792
	Gong et al. [11]++	18.27	0.209	149.1	0.773
	Gong et al. [11]	19.10	0.150	133.8	0.682
	Peng et al. [24]	17.20	0.002	130.3	0.717
	Hayes et al. [12]	17.95	0.002	142.4	0.626
	Wang et al. [36]	17.08	0.002	136.1	0.793
	Sparse-Standard	18.01	0.003	168.6	0.790
	SCA0.1	**17.09**	**0.001**	**172.0**	**0.787**
	SCA0.25	**16.78**	**0.001**	**189.3**	**0.772**
	SCA0.5	**16.24**	**0.001**	**197.0**	**0.744**

case' across all of our experiments, SCA is faster than the best performing baseline (Peng et al. [24]) but slower than other baselines. Table 13 shows the compute times (in seconds) for this 'worst-case' experiment below (The MNIST dataset under the Plug-&-Play attack [28]).

Table 6: Experiments set 3 **Additional Datasets**: Performance in *split* network setting *(lower rows=better defense)*.

Dataset	Defense	PSNR ↓↓	SSIM ↓↓	FID ↑↑	Accuracy
MNIST	No-Defense	31.21	0.923	19.64	0.963
	Gaussian-Noise	31.07	0.922	23.27	0.972
	GAN	28.39	0.894	27.26	0.969
	Gong et al. [11]	28.30	0.806	69.38	0.986
	Titcombe et al. [30]	25.40	0.713	76.88	0.952
	Gong et al. [11]++	21.94	0.591	97.33	0.991
	Peng et al. [24]	16.90	0.475	103.2	0.960
	Hayes et al. [12]	17.23	0.030	288.1	0.856
	Wang et al. [36]	21.87	0.696	53.09	0.903
	Sparse-Standard	18.71	0.288	188.4	0.981
	SCA0.1	**16.17**	**0.109**	**227.4**	**0.988**
	SCA0.25	**17.40**	**0.058**	**301.6**	**0.980**
	SCA0.5	**14.98**	**0.044**	**307.7**	**0.975**
Fashion MNIST	No-Defense	29.66	0.911	14.33	0.868
	Gaussian-Noise	29.49	0.909	14.81	0.871
	GAN	26.03	0.849	19.33	0.885
	Gong et al. [11]	23.70	0.631	97.52	0.884
	Titcombe et al. [30]	20.48	0.565	81.01	0.872
	Gong et al. [11]++	25.77	0.726	57.72	0.908
	Peng et al. [24]	20.67	0.583	46.48	0.865
	Hayes et al. [12]	20.10	0.256	200.6	0.748
	Wang et al. [36]	24.53	0.588	81.79	0.881
	Sparse-Standard	19.54	0.405	200.5	0.882
	SCA0.1	**18.11**	**0.154**	**171.1**	**0.904**
	SCA0.25	**17.74**	**0.188**	**203.8**	**0.896**
	SCA0.5	**17.15**	**0.134**	**270.4**	**0.879**
CIFAR10	No-Defense	16.48	0.709	47.77	0.823
	Gaussian-Noise	14.79	0.311	149.5	0.598
	GAN	14.87	0.296	13.01	0.675
	Titcombe et al. [30]	14.68	0.244	157.3	0.779
	Gong et al. [11]++	13.32	0.003	162.4	0.691
	Gong et al. [11]	14.55	0.291	152.1	0.644
	Peng et al. [24]	17.18	0.002	169.1	0.707
	Hayes et al. [12]	15.44	0.005	204.5	0.596
	Wang et al. [36]	14.73	0.001	176.3	0.820
	Sparse-Standard	13.22	0.003	167.9	0.769
	SCA0.1	**13.18**	**0.002**	**174.2**	**0.758**
	SCA0.25	**13.07**	**0.002**	**181.2**	**0.742**
	SCA0.5	**12.88**	**0.002**	**375.3**	**0.739**

G Ablations: Tuning SCA

Observe that our SCA outperforms SOTA defense baselines in robustness even without any tuning of parameters. However, tuning the hyper-parameters can

Table 7: Experiments set 1: additional Laplace noise benchmark with larger 1.0 noise parameter: Performance in Plug-&-Play Model Inversion Attack [28] setting *(lower rows=better defense)*.

Dataset	Defense	PSNR ↓↓	SSIM ↓↓	FID ↑↑	Accuracy
MNIST	Titcombe et al. [30]-1.0	6.60	0.685	280.1	0.938
	SCA0.1	**6.19**	**0.633**	**287.9**	**0.984**
	SCA0.25	**5.83**	**0.607**	**289.3**	**0.983**
	SCA0.5	**5.74**	**0.604**	**299.6**	**0.977**
Fashion MNIST	Titcombe et al. [30]-1.0	8.72	0.1412	232.1	0.823
	SCA0.1	**8.49**	**0.039**	**222.8**	**0.897**
	SCA0.25	**8.49**	**0.032**	**229.9**	**0.887**
	SCA0.5	**8.45**	**0.047**	**233.5**	**0.876**
CIAFR10	Titcombe et al. [30]-1.0	10.75	0.335	112.7	0.779
	SCA0.1	**10.59**	**0.305**	**144.1**	**0.787**
	SCA0.25	**10.27**	**0.279**	**189.9**	**0.772**
	SCA0.5	**10.23**	**0.276**	**189.7**	**0.744**

Table 8: Experiments set 2 additional Laplace noise benchmark with larger 1.0 noise parameter: Performance in *end-to-end* network setting *(lower rows=better defense)*.

Dataset	Defense	PSNR ↓↓	SSIM ↓↓	FID ↑↑	Accuracy
MNIST	Titcombe et al. [30]-1.0	24.89	0.664	50.64	0.938
	SCA0.1	**19.54**	**0.502**	**178.5**	**0.984**
	SCA0.25	**18.81**	**0.340**	**174.1**	**0.983**
	SCA0.5	**17.85**	**0.164**	**335.5**	**0.977**
Fashion MNIST	Titcombe et al. [30]-1.0	20.21	0.567	80.55	0.823
	SCA0.1	**17.92**	**0.209**	**196.1**	**0.897**
	SCA0.25	**17.03**	**0.186**	**195.2**	**0.887**
	SCA0.5	**14.51**	**0.069**	**423.2**	**0.876**
CIFAR10	Titcombe et al. [30]-1.0	18.71	0.672	170.8	0.779
	SCA0.1	**17.09**	**0.001**	**172.0**	**0.787**
	SCA0.25	**16.78**	**0.001**	**189.3**	**0.772**
	SCA0.5	**16.24**	**0.001**	**197.0**	**0.744**

boost the accuracy further, e.g., we use kernel size as default 5 for all experiments. Increasing the kernel from 5 to 7 can improve SCA accuracies beyond. While our goal is to capture the essence of the SCA itself in terms of robustness, we explore a little bit on further possible improvements on accuracy scores. We consider the lowest robust SCA, i.e., SCA0.1 for the tuning of kernel size, and we present the comparisons of accuracies between SCA0.1 and TUNED SCA0.1 in Table 12.

Table 9: Experiments set 3: additional Laplace noise benchmark with larger 1.0 noise parameter: Performance in *split* network setting *(lower rows=better defense)*.

Dataset	Defense	PSNR ↓↓	SSIM ↓↓	FID ↑↑	Accuracy
MNIST	Titcombe et al. [30]-1.0	22.63	0.503	66.40	0.980
	SCA0.1	**16.17**	**0.109**	**227.4**	**0.988**
	SCA0.25	**17.40**	**0.058**	**301.6**	**0.980**
	SCA0.5	**14.98**	**0.044**	**307.7**	**0.975**
Fashion MNIST	Titcombe et al. [30]-1.0	18.36	0.408	80.80	0.878
	SCA0.1	**18.11**	**0.154**	**171.1**	**0.904**
	SCA0.25	**17.74**	**0.188**	**203.8**	**0.896**
	SCA0.5	**17.15**	**0.134**	**270.4**	**0.879**
CIAFR10	Titcombe et al. [30]-1.0	14.27	0.259	171.6	0.786
	SCA0.1	**13.18**	**0.002**	**174.2**	**0.758**
	SCA0.25	**13.07**	**0.002**	**181.2**	**0.742**
	SCA0.5	**12.88**	**0.002**	**375.3**	**0.739**

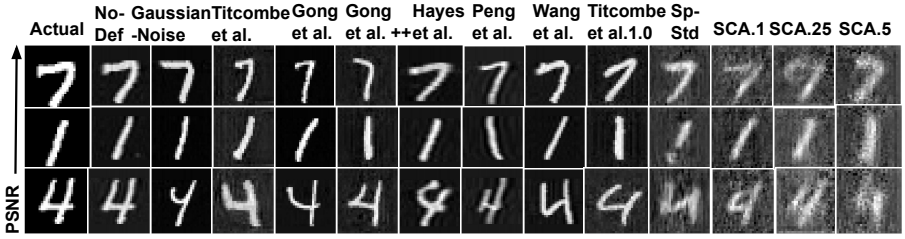


Fig. 2: Qualitative comparisons among actual and reconstructed images under SCA and additional Laplace noise defense benchmark with larger 1.0 noise parameter.

H Robustness of sparse coding layers: UMap

In Figure 3, we present the UMap representation of linear, convolutional, and sparse coding layers on the other datasets, i.e., CelebA and Medical MNIST datasets. Observe that, the data points are more scattered in the sparse coding layer UMap (Figure 3c and Figure 3f) representations compared to the linear (Figure 3a and Figure 3d) and convolutional layers (Figure 3b and Figure 3e), which provide more robustness to models with sparse coding layers, i.e., our proposed SCA, against the privacy attacks.

Table 10: Stability analysis 1: Performance comparison (mean± standard deviations) across multiple runs in Plug-&-Play Model Inversion Attack [28] setting *(lower rows=better defense)* on high-res CelebA dataset.

Dataset	Defense	PSNR ↓↓	SSIM ↓↓	FID ↑↑	Accuracy
CelebA	No-Defense	11.42 ± 2.44	0.613 ± 0.29	292.9 ± 81.5	0.721 ± 0.04
	Gaussian-Noise	10.87 ± 2.25	0.614 ± 0.30	296.5 ± 73.3	0.624 ± 0.03
	GAN	11.02 ± 1.82	0.600 ± 0.29	301.4 ± 92.4	0.613 ± 0.02
	Gong et al. [11]++	10.84 ± 1.94	0.556 ± 0.28	301.0 ± 81.4	0.658 ± 0.02
	Titcombe et al. [30]	10.76 ± 2.37	0.557 ± 0.24	345.5 ± 86.1	0.643 ± 0.01
	Gong et al. [11]	10.91 ± 1.88	0.560 ± 0.29	304.5 ± 82.5	0.616 ± 0.01
	Peng et al. [24]	10.17 ± 2.32	0.491 ± 0.24	399.1 ± 55.3	0.667 ± 0.04
	Hayes et al. [12]	10.16 ± 1.95	0.535 ± 0.25	320.8 ± 79.0	0.601 ± 0.02
	Wang et al. [36]	10.39 ± 2.55	0.505 ± 0.24	341.7 ± 74.2	0.669 ± 0.05
	Sparse-Std	9.78 ± 2.13	0.485 ± 0.24	367.3 ± 44.7	0.663 ± 0.03
	SCA0.1	**9.56** ± 2.30	**0.454** ± 0.25	**396.6** ± 45.0	**0.659** ± 0.04
	SCA0.25	**9.27** ± 2.06	**0.452** ± 0.25	**412.8** ± 53.7	**0.661** ± 0.05
	SCA0.5	**9.12** ± 2.68	**0.368** ± 0.24	**421.7** ± 49.9	**0.653** ± 0.04

Table 11: Stability analysis 2: Performance comparison (mean± standard deviations) across multiple runs in *end-to-end* network setting *(lower rows=better defense)* on Medical MNIST dataset.

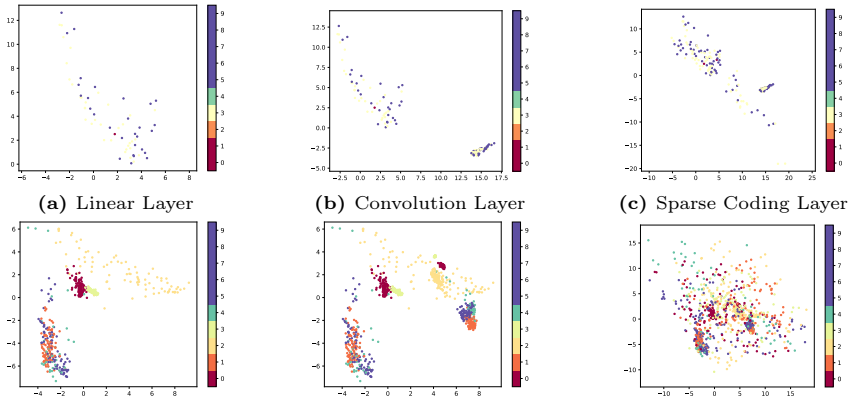
Dataset	Defense	PSNR ↓↓	SSIM ↓↓	FID ↑↑	Accuracy
Medical	No-Defense	30.17 ± 0.90	0.912 ± 0.01	12.40 ± 8.69	0.998 ± 0.01
MNIST	Gaussian-Noise	27.00 ± 1.30	0.828 ± 0.05	17.29 ± 11.9	0.886 ± 0.06
	GAN	25.05 ± 2.78	0.699 ± 0.03	29.08 ± 20.5	0.995 ± 0.01
	Gong et al. [11]++	20.37 ± 1.65	0.451 ± 0.03	44.68 ± 30.9	0.871 ± 0.01
	Titcombe et al. [30]	20.51 ± 0.28	0.574 ± 0.01	28.23 ± 1.65	0.805 ± 0.06
	Gong et al. [11]	23.65 ± 1.07	0.605 ± 0.09	37.16 ± 26.2	0.757 ± 0.03
	Peng et al. [24]	17.42 ± 2.87	0.519 ± 0.22	65.39 ± 32.8	0.866 ± 0.08
	Hayes et al. [12]	19.57 ± 1.08	0.003 ± 0.01	155.0 ± 92.5	0.847 ± 0.08
	Wang et al. [36]	17.89 ± 2.09	0.463 ± 0.08	101.8 ± 66.5	0.829 ± 0.08
	Sparse-Std	13.49 ± 0.29	0.158 ± 0.09	203.4 ± 92.2	0.865 ± 0.05
	SCA0.1	**12.46** ± 0.30	**0.006** ± 0.01	**231.8** ± 124	**0.858** ± 0.08
	SCA0.25	**11.89** ± 0.35	**0.008** ± 0.01	**254.1** ± 153	**0.850** ± 0.08
	SCA0.5	**11.19** ± 0.11	**0.001** ± 0.01	**276.9** ± 97.0	**0.841** ± 0.08

Table 12: Comparison of Accuracy Scores among our unoptimized SCA0.1 and Tuned SCA0.1 (kernel: 5→ 7) in all 3 setups on CelebA and Medical MNIST datasets.

Dataset	Setup	SCA0.1 ↑↑	Tuned SCA0.1 ↑↑
CelebA	plug and play	0.726	**0.730**
	end to end	0.748	**0.751**
	split	0.745	**0.759**
Medical	plug and play	0.888	**0.899**
MNIST	end to end	0.888	**0.996**
	split	0.946	**0.967**

Table 13

Model	Time (sec)
No-Defense	10555.3
Gaussian-Noise	12555.3
GAN	15762.4
Titcombe et al. [30]	14390.2
Gong et al. [11]	16061.8
Gong et al. [11]++	17521.8
Peng et al. [24]	18921.2
Hayes et al. [12]	16923.9
Wang et al. [36]	15229.9
Sparse-Standard	12327.5
SCA0.1	17009.8
SCA0.25	17181.2
SCA0.5	17912.9



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

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

Fig. 3: UMap 2D projections of input images' features by class after 2 linear layers, 2 conv. layers, or 2 sparse-coded layers on CelebA (top) & Medical MNIST (bottom).

References

1. Aïvodji, U., Gambs, S., Ther, T.: Gamin: An adversarial approach to black-box model inversion. arXiv preprint arXiv:1909.11835 (2019) 3
2. Breuer, A., Khosravani, N., Tingley, M., Cottel, B.: Preemptive detection of fake accounts on social networks via multi-class preferential attachment classifiers. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 105–116 (2023) 4
3. Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al.: Extracting training data from large language models. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 2633–2650 (2021) 3
4. 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) 3
5. Choquette-Choo, C.A., Tramer, F., Carlini, N., Papernot, N.: Label-only membership inference attacks. In: International conference on machine learning. pp. 1964–1974. PMLR (2021) 3, 4
6. 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) 3, 4
7. 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) 3
8. 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) 3
9. 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) 4
10. 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) 3
11. 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, 4, 6, 7, 8, 11, 12
12. Hayes, J., Mahloujifar, S., Balle, B.: Bounding training data reconstruction in dp-sgd. arXiv preprint arXiv:2302.07225 (2023) 6, 7, 8, 11, 12
13. 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) 3, 4
14. 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) 4
15. Jia, J., Gong, N.Z.: {AttriGuard}: A practical defense against attribute inference attacks via adversarial machine learning. In: 27th USENIX Security Symposium (USENIX Security 18). pp. 513–529 (2018) 4

16. Juuti, M., Szyller, S., Marchal, S., Asokan, N.: Prada: protecting against dnn model stealing attacks. In: 2019 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 512–527. IEEE (2019) 4
17. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: Proc. NeurIPS (2021) 3
18. 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) 1
19. 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) 3
20. Liu, Y., Wen, R., He, X., Salem, A., Zhang, Z., Backes, M., De Cristofaro, E., Fritz, M., Zhang, Y.: {ML-Doctor}: Holistic risk assessment of inference attacks against machine learning models. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 4525–4542 (2022) 3
21. 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) 4
22. Naveed, M., Ayday, E., Clayton, E.W., Fellay, J., Gunter, C.A., Hubaux, J.P., Malin, B.A., Wang, X.: Privacy in the genomic era. ACM Computing Surveys (CSUR) **48**(1), 1–44 (2015) 4
23. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019) 1
24. 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) 6, 7, 8, 11, 12
25. 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) 1
26. Sablayrolles, A., Douze, M., Schmid, C., Ollivier, Y., Jégou, H.: White-box vs black-box: Bayes optimal strategies for membership inference. In: International Conference on Machine Learning. pp. 5558–5567. PMLR (2019) 3
27. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE symposium on security and privacy (SP). pp. 3–18. IEEE (2017) 3
28. 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) 3, 5, 6, 7, 9, 11
29. 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) 1
30. 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, 5, 6, 7, 8, 9, 10, 11, 12
31. Tramèr, F., Shokri, R., San Joaquin, A., Le, H., Jagielski, M., Hong, S., Carlini, N.: Truth serum: Poisoning machine learning models to reveal their secrets. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 2779–2792 (2022) 3

32. Vhaduri, S., Cheung, W., Dibbo, S.V.: Bag of on-phone anns to secure iot objects using wearable and smartphone biometrics. IEEE Transactions on Dependable and Secure Computing (2023) 3
33. Vhaduri, S., Dibbo, S.V., Chen, C.Y.: Predicting a user's demographic identity from leaked samples of health-tracking wearables and understanding associated risks. In: 2022 IEEE 10th International Conference on Healthcare Informatics (ICHI). pp. 309–318. IEEE (2022) 3
34. Vhaduri, S., Dibbo, S.V., Cheung, W.: Hiauth: A hierarchical implicit authentication system for iot wearables using multiple biometrics. IEEE Access **9**, 116395–116406 (2021) 3
35. Wang, K.C., Fu, Y., Li, K., Khisti, A., Zemel, R., Makhzani, A.: Variational model inversion attacks. Advances in Neural Information Processing Systems **34**, 9706–9719 (2021) 4
36. 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) 6, 7, 8, 11, 12
37. 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) 4
38. Xu, T., Goossen, G., Cevahir, H.K., Khodeir, S., Jin, Y., Li, F., Shan, S., Patel, S., Freeman, D., Pearce, P.: Deep entity classification: Abusive account detection for online social networks. In: 30th {USENIX} Security Symposium ({USENIX} Security 21) (2021) 4
39. Xu, Y., Liu, X., Hu, T., Xin, B., Yang, R.: Sparse black-box inversion attack with limited information. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023) 3
40. 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) 4
41. 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) 3, 4
42. Zhao, B.Z.H., Agrawal, A., Coburn, C., Asghar, H.J., Bhaskar, R., Kaafar, M.A., Webb, D., Dickinson, P.: On the (in) feasibility of attribute inference attacks on machine learning models. In: 2021 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 232–251. IEEE (2021) 4
43. Zhao, X., Zhang, W., Xiao, X., Lim, B.: Exploiting explanations for model inversion attacks. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 682–692 (2021) 4
44. 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) 3