Measuring Model Inversion Defences in Edge–Cloud Collaborative Inference Systems

Anonymous Author(s)
Affiliation
Address
email

Abstract

The edge-cloud collaborative inference systems are designed to speed up the prediction processes in edge-cloud scenarios, where the local devices and the cloud system work together to run a complex deep learning model. However, those edge-cloud collaborative inference systems are vulnerable to emerging Model Inversion (MI) attacks, where malicious cloud service providers are able to recover the edge-side users' private data. To defend against such attacks, several countermeasures have been recently introduced. Unfortunately, little is known about the robustness of those defence countermeasures. In this paper, we take the first step towards measuring the robustness of those state-of-the-art defence countermeasures with respect to MI attacks. We show the trade-offs of privacy and utility of these countermeasures and propose a novel anti-defence method called Sensitive Feature Distillation (SFD) to restore sensitive information from the protected feature representations. Our experiments show that SFD is able to break through defence mechanisms in model partitioning scenarios, demonstrating the inadequacy of existing defence mechanisms as a privacy-preserving technique against MI attacks. We hope our findings encourage researchers to pursue more robust defence mechanisms against MI attacks for edge-cloud collaborative inference systems.

1 Introduction

With the rapid development of deep neural network technology, the deployment of deep learning (DL) applications on edge devices is becoming more widespread. However, on the one hand, resource-constrained edge devices are often difficult to accommodate computationally intensive DL models, and on the other hand, cloud-based DL inference applications require devices to upload users' private data to cloud service providers, which poses a serious privacy threat. Collaborative DL system has recently emerged as a good alternative to address these challenges, where the local devices and the cloud system work together to run a complex neural network [Eshratifar et al., 2019; Banitalebi-Dehkordi et al., 2021; Li et al., 2021]. The DL models can be split such that only the first few layers are executed on the local edge device while the remaining layers are executed on the remote cloud. And the end user does not need to upload her raw data to the cloud, so that privacy concerns can be addressed to some extent.

This paper focuses on Model Inversion (MI) attacks, where existing research has shown that adversaries can easily recover sensitive data with the output of the middle layers [He et al., 2020]. To address this problem, many defence approaches have been proposed for the collaborative edge-cloud inference systems. [Xiao et al., 2020] demonstrates that adversarial learning improves the robustness of the model against MI attacks. [Mireshghallah et al., 2020] and [Titcombe et al., 2021] proposes learning additive noise distributions to protect the privacy of inference data. [He et al., 2020] uses

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

dropout to defend against MI attacks. However, existing work does not provide a systematic security measurement of the various defences, as well as the trade-off between model accuracy and privacy risks. This leads to our main research question: *are those privacy protection techniques against MI attacks for collaborative edge-cloud inference systems really effective?*

In this paper, we summarise the various defences, systematically evaluate their performance in defending against MI attacks, and present their trade-offs regarding data utility and privacy leakage. We propose a novel anti-defence technique to break down existing defences, which can achieve improved attack performance for a variety of datasets, target models, and defence algorithms. We focus on the privacy of the inference data [He et al., 2019]. Since MI protectors obfuscate the intermediate feature representation to limit the redundant information about the model input, our idea is to learn to distill obfuscated features into sanitised features for better inversion. Specifically, we introduce the Sensitive Feature Distillation (SFD) to extract the image structure knowledge of sensitive features; the proper formulation of these prior knowledge will be helpful to solve the ill-posed inversion problem. We show that this vulnerability is inevitable even when different defence countermeasures are utilized.

We summarize our contributions as follows:

- We systematically evaluate the effectiveness of three state-of-the-art defence mechanisms against model inversion attacks in edge-cloud collaborative inference systems. We show the privacy and utility trade-offs of these defence methods.
- To further evaluate the robustness of existing MI defences, we propose a novel anti-defence technique called Sensitive Feature Distillation (SFD) to transfer obfuscated features into sanitised features so as to solve the ill-posed inversion problem.
- We conduct comprehensive evaluation on three representative image classification tasks using different model architectures. Compared with the baseline, our evaluation results demonstrate that our anti-defence method is more effective. We show that the existing defence mechanisms are of little use in protecting against our proposed method. This highlights that current defences still carry significant privacy risks.

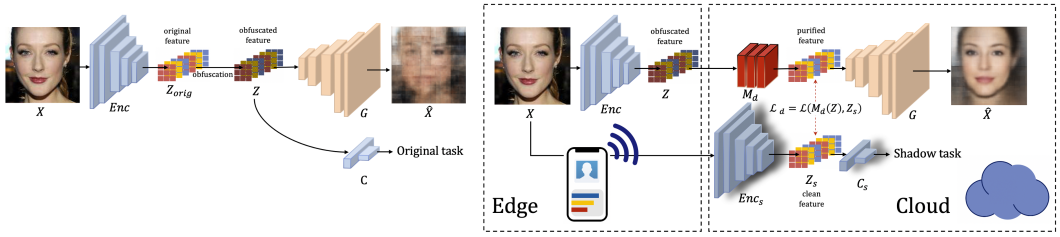
2 Background and Related Work

This paper focuses on the most common DL task: image classification. A Deep Neural Network (DNN) model can be characterized as $F : \mathcal{X} \mapsto \mathcal{Y}$ which maps an input data $x \in \mathcal{X}$ into an output result $y \in \mathcal{Y}$.

Edge-Cloud Collaborative Inference Systems. Running the entire deep learning inference application on the edge device can be time- and resource-consuming. A common scaling approach is to partition the DNN model where the first few layers of the model are executed within the local edge device while the remaining layers are offloaded to the remote cloud [Kang et al., 2017; Eshratifar et al., 2019; Banitalebi-Dehkordi et al., 2021; Li et al., 2021]. This scheme can reduce the computational burden of edge devices on the one hand, and can mitigate privacy leaks of inference raw data on the other [Wang and Gong, 2018; Osia et al., 2020]. Collaborative inference makes it feasible and efficient to deploy large-scale intelligent workloads on today's edge platforms.

In the collaborative inference system, a model is split into two parts: a on-device part and a remote cloud-based part. Each part will contain several layers of the model. From now on, we reformulate the model $F = C \circ Enc$ as the Encoder module Enc and the Classifier module C. The edge device deploys the first part of the model Enc, it feeds the users' inference data x to the feature extractor to get an intermediate feature representation $z = Enc(x)$, and then sends z to the cloud. The cloud hosts the second part C, it calculates the prediction output $y = C(z)$ when the intermediate value z is received from the edge device, then it returns expected result y to the edge side.

Model Inversion Attacks. Model inversion attacks aim to infer the input data given the access to a target machine learning model. [Fredrikson et al., 2014] proposed MI attacks for the first time, they presented a general inversion algorithm which intended to find the optimal data for the given model output and auxiliary information. The basic idea of the algorithm is to formulate the MI attack as an gradient-based optimization problem in the data space. Then they extended the algorithm to some



(a) Defence against MI attack (b) MI attack based on SFD

Figure 1: Model inversion defence and attack in Edge-Cloud collaborative inference systems. (a) shows the main idea of defences. (b) shows the proposed Model Inversion (MI) attack based on Sensitive Feature Distillation (SFD).

more complex models [Fredrikson et al., 2015]. However, this method was shown to be ineffective especially for the complex model structure such as convolutional neural network [Hitaj et al., 2017].

Unlike developing optimization algorithms to perform MI attacks, recent research introduce training-based approach which suggest to inverse the model by training an inverse network. [Zhang et al., 2020] focused on the white-box setting and proposed a GAN-based inversion model, which can extract general knowledge from public datasets, to improve the realism of reconstructed images. [Yang et al., 2019] studied black-box MI and proposed to construct the input and corresponding output of the target model as the training set of the attack model for inversion training. [Salem et al., 2020] applied a specific attack algorithm for online learning. [He et al., 2019] improved black-box MI attacks by using mid-level representations in the context of the edge-cloud collaborative inference systems, where the adversary has access to the cloud part of the target model, the target is to reconstruct images from intermediate feature representations of the inference data. Their experimental results showed that the inversion accuracy and restored image quality are significantly improved compared with previous works. This is because the inversion model can capture richer semantics in the data space from intermediate features than from the output confidence scores of the target model.

3 Problem Setup: Defences against MI attacks

3.1 Adversary: Learn to Inverse

The goal of the adversary is to reconstruct the private input data sent by the user to the edge-cloud DL inference services. Following existing works on MI attacks [He et al., 2019, 2020], we consider a collaborative inference system between a local, edge-device part $\mathbb{E}$ and a remote, cloud-based part $\mathbb{C}$. For a partitioned model $F = C \circ Enc$, $\mathbb{E}$ performs the earlier layers Enc and $\mathbb{C}$ performs C. We consider the $\mathbb{E}$ is trusted, when an input is fed into Enc, $\mathbb{E}$ correctly processes it and computes a representation. $\mathbb{E}$ also attempts to prevent the input data from being reconstructed for privacy, so $\mathbb{E}$ applies the defences to the representation and sends it to the cloud. The cloud $\mathbb{C}$, i.e. the adversary, is untrusted to try to steal the input.

3.2 Protector: Learn "NOT" to Inverse

For privacy, the edge part of the model typically obfuscates the intermediate feature representations before sending them to the cloud. The goal of the specially designed obfuscation is to encode the input x into a representation z that does not leak the characteristics of x, but whose expressive power is sufficient to predict the task label y. Several defence mechanisms have been proposed to mitigate the risks of MI. Figure 1 (a) shows the main idea of the defences.

Adversarial Learning. The most straightforward defence against MI is to apply adversarial training [Goodfellow et al., 2014] to significantly improve the robustness of the model's intermediate feature representation. [Xiao et al., 2020] presents the training objective of the target model, which aims at learning a privacy-preserving and task-oriented representation.

The protector formalize the joint privacy and classification objectives in the following min-max optimization problem:

$$\underbrace{\min_{Enc,C}(\mathcal{L}_{task} + \lambda \underbrace{\max_{Dec} \mathcal{G}_{\text{adv}}}_{\text{optimal inversion}})}_{\text{optimal privacy-preserving classification}}. \tag{1}$$

, where the Enc aims to fool the adversary's inversion model by minimizing the reconstruction gain $\mathcal{G}_{\text{adv}}$. This results in designing the optimal privacy mechanism, which is also utility-maximizing.

Noise Mask. Differential privacy has been introduced to protect model inference Mireshghallah et al. [2020] through adding random noise to the inference data. Titcombe et al. [2021] suggests defending against MI attacks by adding additive Laplacian noise to the intermediate feature representation. Formally, it calculates $z^{noise} = Enc(x)+\epsilon$, the random noise ϵ is sampled from a Laplace distribution parameterized by location μ and scale b each time the model is used. The partitioned model, no longer a one-to-one function, obscures the data communicated between the edge and the cloud.

Adding noise directly to intermediate data in prediction can introduce a significant performance loss. As a result, the protector would prefer to perform noisy training, enhancing the robustness of the model to noise. The noisy training loss is defined as:

$$\mathcal{L}_{noisy} = \alpha\mathcal{L}(C(z), y) + (1-\alpha)\mathcal{L}(C(z^{\epsilon}), y) \tag{2}$$

, where the losses on both the clean and the noisy representations are combined, and uses λ to control the trade-off.

Dropout Defence. He et al. [2020] introduce dropout defence to defeat MI attacks. Specifically, the protector generate random masks M to protect the intermediate data during the inference phase of the model:

$$z^{dropout} = Enc(x) \otimes M \tag{3}$$

, where M denotes a randomly generated mask, $\otimes$ denotes the element-wise multiplication. To allocate each element of M to two values at random: one of them is 0 with probability r and the other is 1 with probability $1-r$. Note that the dropout defence differs from noise obfuscation in that dropout hides partial sensitive information in the feature representation, whereas noise disturbs all the information of the intermediate data.

4 Methodology

The main goal of this paper is to study the potential privacy leakage points of existing defences in edge-cloud collaborative inference systems. Specifically, we propose an anti-defence method and describe how to exploit Sensitive Feature Distillation (SFD) for more aggressive inversion through the carefully designed algorithm.

4.1 Adversary's Capability

We consider black-box scenario where an adversary is allowed to query the target model's encoder module Enc to make inferences of their collected data samples, and obtain the corresponding intermediate features. For adversary, this is the most challenging attack setting [Salem et al., 2018]. The Feature Distiller module of our attack model is defined as a supervised DL model, which needs ground truth data for model training. So we adopt shadow training techniques following previous works [Shokri et al., 2017; Salem et al., 2020; Zhou et al., 2021]. We consider the adversary trains a shadow model M_s, which is designed to imitate the behavior of the target model. We also assume that the shadow model has the same model architecture and hyper-parameters as the target model. In order to achieve this in practice, the adversary can perform model extraction attacks and model hyper-parameter stealing attacks [Sha et al., 2022; Pal et al., 2020; Wang et al., 2018; Oh et al., 2019]. Moreover, to train the shadow model, we further assume that the adversary has a local dataset, denoted by D_s, comes from the the same distribution as the target model's training set. Once the shadow model has been trained, the adversary can rely on the shadow model to derive the ground truth clean feature to train her attack model. Note that most of the experiments related to shadow models' training by previous works make the same assumptions. In experiment section, we show these assumptions can be relaxed as well.

4.2 Anti-defence Inversion Algorithm

Our general attack pipeline contains three phases. In the first phase, the adversary constructs a shadow model locally, which is used to generate training data for the next stage. In the second phase, the Feature Distiller module of our attack model learns to distill obfuscated features into sanitized features. In the last phase, the Inverse-Network module of our attack model takes the sanitized features to produce reconstructed images. The second and third steps are performed alternately in the concrete attack flow. Figure 1 (b) provides a schematic view of our attack pipeline.

Shadow Model. The Feature Distiller module of our attack model needs to be trained in a supervised manner. Due to our minimal assumptions in Section 4.1, the adversary can only obtain the obfuscated intermediate features by querying the target model. Nevertheless, we also need ground truth data, i.e. clean intermediate features, to train the Feature Distiller. To overcome this problem, we rely on shadow training, the shadow model is applied to mimic an unshielded target model. The adversary first establishes a shadow model $M_s = C_s \circ Enc_s$ with the same structure as the target model, and then uses its own dataset to train M_s. She can use SGD or ADAM to optimize the cross-entropy loss function of the shadow model. If the adversary's goal is to find the inversion of the DNN for different tasks, then other more appropriate loss functions can be utilized. Once the shadow model M_s is obtained, the adversary can use it to generate pure intermediate features that are not obfuscated.

Sensitive Feature Distillation. A straightforward inversion attack may not succeed if the intermediate data z is obfuscated to reduce the amount of information it reveals about input x. To address this issue, we borrow the idea from the feature-based knowledge distillation [Romero et al., 2014; Zagoruyko and Komodakis, 2016]. But unlike these efforts, which use the intermediate feature mappings of the teacher model to help student model achieve better results in the original classification task, we use the teacher model (shadow model) to help the student model (Feature Distiller) distill sensitive information to improve the effectiveness of the model inversion attack.

To successfully reconstruct a user's private image x, our approach utilizes a convolutional neural network, termed Feature Distiller, to extract missing sensitive regions in the obfuscated intermediate features. In our settings, the Feature Distiller needs to use clean feature data as the training targets. Specifically, the adversary first uses her local training set X (X are sampled from the adversary's datasets $\mathcal{D}_s$) to query both the target model M and the shadow model M_s, and observes the corresponding intermediate outputs $Z = Enc(X)$ and $Z_s = Enc_s(X)$. Next, the adversary can directly train her Feature Distiller using Z as the training inputs and Z_s as the training targets. We take Sensitive Feature Distillation as an optimization problem with feature space L_2 loss:

$$\mathcal{L}_d = \sum_{z \in Z, z_s \in Z_s} \|M_d(z) - z_s\|_2^2 \tag{4}$$

, where M_d is the Feature Distiller that the adversary wants to learn. The possibility of training with a feature-space loss has been demonstrated in [Nguyen et al., 2016; Shen et al., 2019]. We match obfuscated features Z and clean features Z_s in order to encourage the Feature Distiller to identify the critical mappings in sensitive spatial information. After training, we can use the Feature Distiller to distill the latent vector $\hat{z}$ that is most similar to the clean feature from the target obfuscated feature z.

Inversion Attack. At this phase, we use the same attack scheme proposed in [He et al., 2019] : the Inverse-Network. Conceptually, the Inverse-Network is the approximated inverse function of Enc. However, our approach differs from the previous one in that the training input to the Inverse-Network is no longer observed data $Enc(x)$, but purified data $M_d(Enc(x))$ produced by the Feature Distiller. Meanwhile, the Inverse-Network G and the Feature Distiller M_d are jointly trained, which leads to better inversion quality. We show the detailed description of our proposed model inversion attack using sensitive feature distillation in Appendix.

Similar to the previous phase, the adversary generates a stack of training samples of labeled (z, x) pairs. She can then use $M_d(z)$ as the sanitized latent vectors of z to train the inversion model. We leverage L2 norm in the pixel space as the loss function of G:

$$\mathcal{L}_G = \sum_{z \in Z, x \in X} \|G(M_d(z)) - x\|_2^2 \tag{5}$$

, G is optimized to reconstruct data from sanitized latent vectors. It is worth noting that the structure of G is not necessarily correlated with the target Encoder module Enc. For simplicity, in our case, we use the same Inverse-Network architecture as [He et al., 2019]. After G and M_d are trained, the adversary can feed any observed intermediate-level value $z^\star = Enc(x^\star)$ to the attack model and obtain the reconstructed sensitive input $\hat{x}^\star = G(M_d(z^\star))$.

5 Experiment

In this section, we systematically measure the effectiveness of the existing defences against MI. We have conducted detailed experiments on three representative image classification tasks, each of which uses a different architecture of the target DNN model. We showcase the privacy leakage and data utility trade-offs of these defences. Then, we evaluate the inversion performance of the proposed method and compare it with existing work.

5.1 Experiment Setup

Table 1: Experiment Configurations

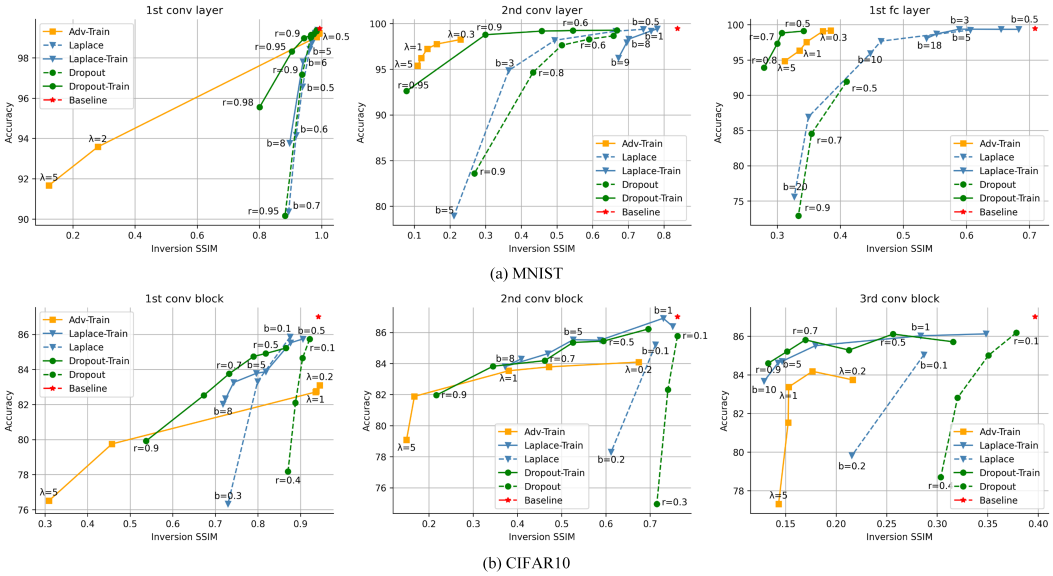
Dataset	MNIST	CIFAR10	CelebA
Target Model	LeNet	VGG-16	ResNet-50
Split point	1st conv layer (8*8) 2nd conv layer(4*4) 1st fc layer(64)	1st conv block (16*16) 2nd conv block (8*8) 3rd conv block (4*4)	2nd conv block (16*16) 3rd conv block (8*8) 4th conv block (4*4) avgpool (1*1) 1st fc layer (512)

Datasets. To perform experimental evaluation, we use three standard benchmark image recognition datasets: MNIST [LeCun, 1998], CIFAR10 [Krizhevsky et al., 2009] and CelebA [Liu et al., 2015]. These datasets are also the benchmark datasets for various ML security and privacy tasks. In most of our experiment, we split the dataset into three parts, target models' private dataset $\mathcal{D}_p$, adversary's shadow dataset $\mathcal{D}_s$ and the test set $\mathcal{D}_t$ with the rest. We detail each dataset in the following: MNIST contains of 70,000 grey-scale images. We set $\mathcal{D}_p$, $\mathcal{D}_s$ and $\mathcal{D}_t$ to 50,000, 10,000 and 10,000 respectively; CIFAR10 consists of 60,000 color images. We separate $\mathcal{D}_p$, $\mathcal{D}_s$ and $\mathcal{D}_t$ to 40,000, 10,000 and 10,000 respectively; CelebA is a dataset composed of 202,599 images of celebrities. For simplicity, the task is to predict gender. The $\mathcal{D}_p$, $\mathcal{D}_s$ and $\mathcal{D}_t$ are set to 101,299, 81,040 and 20,260 accordingly.

Models. For MNIST, We experiment with LeNet [LeCun et al., 1998]. For CIFAR10, we apply VGG-16 [Simonyan and Zisserman, 2014]. For CelebA, we adopt ResNet-50 [He et al., 2016]. We split each target model at different layers, with the main partition being the convolutional layer. For Inverse-Network G, following [He et al., 2019; Yang et al., 2019], we use up-sampling blocks to decode target intermediate feature representations to pixel images. For Feature Distiller M_d, we choose neural network using multiple convolutional layers or full connection layers, with the number of hidden units set to the dimension of the feature data. Specifically, in our approach, the Inverse-Network would be combined with the Feature Distiller as a component in order to operate directly on top of the Feature Distiller. When training the target and shadow models, we choose the SGD optimizer with cosine annealing learning rate scheduler [Loshchilov and Hutter, 2016]. The optimizations of the Inverse-Network and Feature Distiller are performed by the ADAM optimizer with learning rate $10^{-3}, \beta_1 = 0.5, \beta_2 = 0.999$.

Compared Baselines. In an edge-cloud collaborative inference system scenario, since there is no other prior method exploiting feature-level data, we evaluate the effectiveness of the existing defences by comparing the proposed attack with Existing Model Inversion (EMI) attack against all three defence methods.

Experiment Details and Evaluation Metrics. Table 1 gives details of the experimental configurations. These configurations are practical in an edge-cloud scenario, as some of the network layers,



(a) MNIST

(b) CIFAR10

Figure 2: Utility-Privacy trade-off of different defences. (a) shows the result of MNIST dataset. (b) shows the result of CIFAR10 dataset.

which are much larger in terms of computation as well as number of parameters, are offloaded to the cloud. To quantify the privacy risks of model inversion attacks, we follow the setting in most of previous work [He et al., 2019; Xiao et al., 2020; Zhao et al., 2021]. We adopt two metrics, Mean Squared Error (MSE) and Structural Similarity Index (SSIM) [Wang et al., 2004]. We use different evaluation metrics between the reconstructed images and the ground truth to gather multiple evidences of how well the inversion reconstructs the private target image.

5.2 Performance of Defence Methods

Through performing the EMI attack, we verify the performance of defences. Figure 2 shows the quantitative analysis of data utility (y-axis) and recovered image quality (x-axis) on MNIST and CIFAR10 datasets and partitioned layers of different target models, respectively. Higher accuracy represents better utility, while smaller SSIM represent better privacy. We observe that larger defence hyperparameters result in a more significant loss of model accuracy and thus better privacy-preserving, demonstrating the trade-offs between utility and privacy.

Adversarial Learning. The adversarial learning is represented as orange lines. The Enc trained with this method maintains a good task performance with a decrease in the inverse image quality. However, this method also results in a more significant loss of model accuracy for better privacy-preserving. Our results suggest that defending MI with adversarial learning may require the regularization factor $\lambda \geq 5.0$. As a trade-off, such a large factor would introduce the accuracy loss of around 5% for MNIST and 10% for CIFAR10.

Noise Mask. The noise mask with noisy training and adding noise directly to the intermediate feature is represented as solid and dotted blue lines. The model accuracy degradation can be mitigated to some extent by using noisy training instead of adding noise directly, however, the reconstruction is still recognizable (SSIM > 0.3 [He et al., 2019, 2020]). This suggests that noise may not be a practical defence to achieve the desired protective effect.

Dropout Defence. The dropout defence with training and applying dropout to the intermediate data are represented as solid and dotted green lines, respectively. Similar to noise obfuscation, adding dropout masks to the target layer during the training phase provides better privacy. We also find that dropout defence is superior to the other two defence options at the deeper layers. Dropout exploits

Table 2: Inversion results against all three defences, the data in brackets represents the attack performance improvement of MI attacks based on SFD compared to the baseline EMI attacks. Defence hyper-parameters are set $\lambda = 2.0$ for adversarial learning, $b = 10.0$ for noise mask and $r = 0.9$ for dropout defence.

Dataset	Split Point	Vanilla		Adversarial		Noise		Dropout	
		inv loss	ssim	inv loss	ssim	inv loss	ssim	inv loss	ssim
MNIST	1st conv layer	0.002	0.994	0.442-(0.310)	0.123+(0.451)	0.035-(0.013)	0.894+(0.039)	0.022-(0.006)	0.943+(0.013)
	2nd conv layer	0.057	0.838	0.324-(0.191)	0.136+(0.485)	0.212-(0.127)	0.363+(0.394)	0.231-(0.110)	0.299+(0.358)
	1st fc layer	0.103	0.709	0.215-(0.017)	0.303+(0.006)	0.199-(0.010)	0.349+(0.020)	0.235-(0.002)	0.239+(0.007)
CIFAR10	1st conv block	0.006	0.942	0.230-(0.137)	0.457+(0.221)	0.045-(0.022)	0.723+(0.116)	0.075-(0.030)	0.537+(0.116)
	2nd conv block	0.026	0.762	0.220-(0.069)	0.167+(0.164)	0.112-(0.024)	0.409+(0.077)	0.173-(0.044)	0.217+(0.106)
	3rd conv block	0.100	0.397	0.230-(0.021)	0.150+(0.022)	0.212-(0.005)	0.130+(0.019)	0.223-(0.016)	0.132+(0.035)
CelebA	2nd conv block	0.010	0.785	0.124-(0.092)	0.370+(0.248)	0.172-(0.082)	0.289+(0.211)	0.088-(0.036)	0.416+(0.175)
	3rd conv block	0.020	0.698	0.189-(0.045)	0.331+(0.121)	0.206-(0.038)	0.332+(0.083)	0.155-(0.021)	0.370+(0.090)
	4th conv block	0.038	0.625	0.216-(0.023)	0.369+(0.048)	0.219-(0.018)	0.364+(0.034)	0.202-(0.015)	0.375+(0.048)
	avgpool	0.067	0.551	0.222-(0.084)	0.384+(0.070)	0.231-(0.012)	0.372+(0.015)	0.230-(0.008)	0.371+(0.025)
	1st fc layer	0.124	0.452	0.216+(0.005)	0.399-(0.002)	0.232-(0.006)	0.234+(0.003)	0.231-(0.003)	0.382+(0.007)

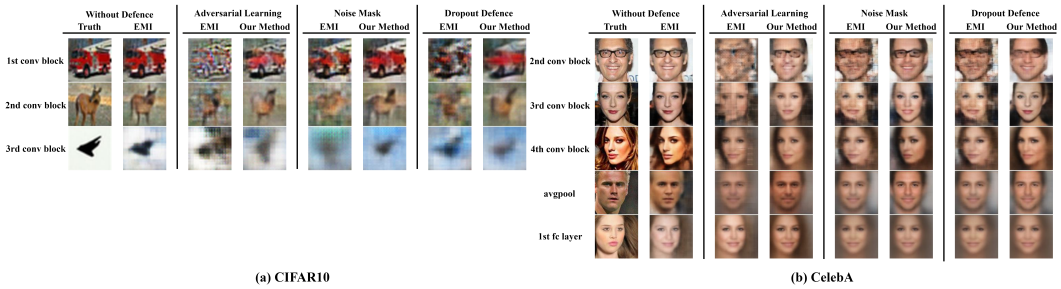


Figure 3: Qualitative comparison of the MI attack based on SFD with the EMI attack. Each row corresponds to a different category. In both (a) and (b), the first column depicts the original images, and the second column depicts the result derived from attacking unprotected features, which is the upper bound for our approach. The third to eighth columns show the results EMI and our method under different defence mechanisms.

the parameter redundancy of DNN model (fully connected layer would introduce greater redundancy) to remove sensitive information, while the other two approaches cause greater accuracy loss by disrupting clean intermediate features.

5.3 Performance of MI attack based on SFD

First, Table 2 shows the inversion performance of the proposed counter defence approach. We report the difference values (values in brackets) between the inversion loss of MI attack based on SFD and the baseline EMI attack, and we also report the difference values between the SSIM of the reconstructed image generated from our method and from the baseline, both of which represent performance gains. Averagely, the proposed method provides a great improvement in inversion result against all three defences. We owe this to SFD used in our method, which can capture sensitive spatial information for more aggressive attacks. Note that the vanilla results show the inversion effect of using EMI when the target model is without defence. Here, we just use vanilla results to show the best possible results for the attack. The observations is that our approach still provides a competitive performance. From Table 2, we can conclude that the improvements of our approach is most significant on feature representations that have been trained with adversarial learning, the improvements are smaller against noise obfuscation and dropout defence. This is due to the fact that both noise and dropout perform perturbation operations via introducing the randomly generated mask, which makes it more difficult for our Feature Distiller to learn the connection between clean and obfuscated features. Also the defense effect of noise mask is inherently weak, leaving little room for improvement.

Secondly, we compare the results of attack performance at different split points of the target model. Generally, we observe that our approach works better at early layers than at later layers. It is consistent with the intuition that features are becoming more and more abstract during the forward propagation process of network, so the quality of reconstructed images deteriorates as the layer goes deeper. Besides, we also observe a significant deterioration in image quality on deeper layers (especially

Table 3: Evaluation for the impact of different defence hyper-parameters on CIFAR10 dataset.

Defence	Param	1st conv block		2nd conv block		3rd conv block	
		acc	ssim	acc	ssim	acc	ssim
Adversarial	1.0	83.1	0.945+(0.005)	83.5	0.380+(0.117)	83.4	0.153+(0.014)
	2.0	79.8	0.457+(0.221)	81.8	0.167+(0.165)	81.5	0.150+(0.022)
	5.0	76.5	0.309+(0.181)	79.0	0.149+(0.123)	77.3	0.143+(0.023)
Noise	1.0	85.5	0.876+(0.053)	86.9	0.729+(0.019)	86.0	0.283+(0.010)
	5.0	83.7	0.796+(0.089)	85.5	0.526+(0.062)	85.9	0.144+(0.015)
	10.0	82.3	0.723+(0.116)	84.3	0.409+(0.077)	84.9	0.130+(0.019)
Dropout	0.5	84.9	0.818+(0.058)	85.4	0.594+(0.022)	84.9	0.256+(0.015)
	0.7	83.7	0.732+(0.089)	84.2	0.461+(0.061)	86.1	0.169+(0.027)
	0.9	79.9	0.537+(0.116)	81.9	0.217+(0.106)	84.6	0.132+(0.035)

Table 4: Results of shadow models for different structure. The accuracy of shadow models using VGG-16, VGG-13 and ResNet-18 is 72%, 65% and 61% respectively.

CIFAR10	split point	acc	inv loss				ssim			
			none	VGG-16	VGG-13	ResNet-18	none	VGG-16	VGG-13	ResNet-18
Adversarial	1st conv block	79.8	0.220	0.084	0.082	0.078	0.457	0.678	0.681	0.671
	2nd conv block	81.8	0.221	0.152	0.154	0.153	0.167	0.331	0.330	0.302
	3rd conv block	81.5	0.219	0.203	0.206	0.198	0.156	0.173	0.177	0.182

the fully connected layer). It is because the size of feature mapping on deeper layers is typically smaller than shallower layers, and the fully connected layer even mix up previous visual information, keeping only the semantic information. So it is relatively harder to distill sensitive features from small obfuscated representations.

Finally, Figure 3 visualize some randomly sampled reconstructed images by our attack on CIFAR10 and CelebA. As shown in the figure, our method is able to effectively distill sensitive features and reconstruct images that are visually similar to the original samples.

Effect of Defence Hyper-parameters. We further evaluate the inversion performance under different defence hyper-parameters. We report the results in Table 3. Strong defence does not help, the performance of our attack increases along with the hyper-parameter increases, which demonstrates that our method can effectively extract sensitive information from feature spaces that are protected by stronger defences (more heavily obfuscated features).

Effect of Shadow Model. To further relax the assumption that the adversary has knowledge of the target model, we let the adversary train a shadow model with different structure from the target model. For simplicity, we used shadow model with the same size intermediate feature representations as the target model but with different architecture. In our experiments on CIFAR10, we use VGG-16 mentioned previously in Sec.5.1 for target model, and use VGG-13 and ResNet-18 for the shadow models respectively. Table 4 depicts the evaluation results. The results of the attack on the intermediate features of different sizes show both worse and better performance, but the variation is small(i.e., less than 3% on average), indicating that our attack is robust to variations in the structure of shadow model. When repeating the experiments on other datasets or other, we observed similar results and omit them due to space constraints.

6 Conclusion

In this paper, we track the potential privacy leakage points of defence mechanisms against model inversion attacks in edge-cloud collaborative inference systems. We systematically evaluate the effectiveness of three defences. And we show the privacy leakage and data utility trade-offs of these defence methods. We propose a counter defence technique, termed sensitive feature distillation, where shadow models are trained using public datasets and Feature Distiller is designed to purify sensitive feature representations that are obfuscated by defence mechanisms, so as to better extract sensitive information. We evaluate the proposed method on three standard image classification tasks with different popular pre-trained DL models, and show that our method can achieve significantly performance improvement at different split point, demonstrating the inadequacy of existing defence mechanisms as a privacy-preserving technique against MI attacks.

References

Amin Banitalebi-Dehkordi, Naveen Vedula, Jian Pei, Fei Xia, Lanjun Wang, and Yong Zhang. Auto-split: a general framework of collaborative edge-cloud ai. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2543–2553, 2021.

Amir Erfan Eshratifar, Mohammad Saeed Abrishami, and Massoud Pedram. Jointdnn: an efficient training and inference engine for intelligent mobile cloud computing services. *IEEE Transactions on Mobile Computing*, 20(2):565–576, 2019.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An {End-to-End} case study of personalized warfarin dosing. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 17–32, 2014.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Zecheng He, Tianwei Zhang, and Ruby B Lee. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 148–162, 2019.

Zecheng He, Tianwei Zhang, and Ruby B Lee. Attacking and protecting data privacy in edge–cloud collaborative inference systems. *IEEE Internet of Things Journal*, 8(12):9706–9716, 2020.

Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 603–618, 2017.

Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, and Lingjia Tang. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. In *Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '17, page 615–629, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450344654. doi: 10.1145/3037697.3037698. URL `https://doi.org/10.1145/3037697.3037698`.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Min Li, Yu Li, Ye Tian, Li Jiang, and Qiang Xu. Appealnet: An efficient and highly-accurate edge/cloud collaborative architecture for dnn inference. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 409–414. IEEE, 2021.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Fatemehsadat Mireshghallah, Mohammadkazem Taram, Prakash Ramrakhyani, Ali Jalali, Dean Tullsen, and Hadi Esmaeilzadeh. 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*, pages 3–18, 2020.

Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in neural information processing systems*, 29, 2016.

Seong Joon Oh, Bernt Schiele, and Mario Fritz. Towards reverse-engineering black-box neural networks. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 121–144. Springer, 2019.

Seyed Ali Osia, Ali Shahin Shamsabadi, Sina Sajadmanesh, Ali Taheri, Kleomenis Katevas, Hamid R Rabiee, Nicholas D Lane, and Hamed Haddadi. A hybrid deep learning architecture for privacy-preserving mobile analytics. *IEEE Internet of Things Journal*, 7(5):4505–4518, 2020.

Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, and Vinod Ganapathy. Activethief: Model extraction using active learning and unannotated public data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 865–872, 2020.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.

Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. {Updates-Leak}: Data set inference and reconstruction attacks in online learning. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1291–1308, 2020.

Zeyang Sha, Xinlei He, Ning Yu, Michael Backes, and Yang Zhang. Can't steal? cont-steal! contrastive stealing attacks against image encoders. *arXiv preprint arXiv:2201.07513*, 2022.

Zhiqiang Shen, Zhankui He, and Xiangyang Xue. Meal: Multi-model ensemble via adversarial learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4886–4893, 2019.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Tom Titcombe, Adam J Hall, Pavlos Papadopoulos, and Daniele Romanini. Practical defences against model inversion attacks for split neural networks. *arXiv preprint arXiv:2104.05743*, 2021.

Binghui Wang and Neil Zhenqiang Gong. Stealing hyperparameters in machine learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 36–52. IEEE, 2018.

Ji Wang, Jianguo Zhang, Weidong Bao, Xiaomin Zhu, Bokai Cao, and Philip S Yu. Not just privacy: Improving performance of private deep learning in mobile cloud. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2407–2416, 2018.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Taihong Xiao, Yi-Hsuan Tsai, Kihyuk Sohn, Manmohan Chandraker, and Ming-Hsuan Yang. Adversarial learning of privacy-preserving and task-oriented representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12434–12441, 2020.

Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. Neural network inversion in adversarial setting via background knowledge alignment. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 225–240, 2019.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 253–261, 2020.

Xuejun Zhao, Wencan Zhang, Xiaokui Xiao, and Brian Lim. Exploiting explanations for model inversion attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 682–692, 2021.

Junhao Zhou, Yufei Chen, Chao Shen, and Yang Zhang. Property inference attacks against gans. *arXiv preprint arXiv:2111.07608*, 2021.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section **??**
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section **??**
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Sections **??** and **??**
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section **??**, last paragraph
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Please find the URL in Section **??**, last paragraph
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section **??**
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section **??**
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]