DISCO: Dynamic and Invariant Sensitive Channel Obfuscation for deep neural networks

Abhishek Singh[1], Ayush Chopra[1], Vivek Sharma[1,2], Ethan Garza[1], Emily Zhang[1], Praneeth Vepakomma[1], Ramesh Raskar[1]

[1] Massachusetts Institute of Technology, [2] Harvard Medical School

Abstract

Recent deep learning models have shown remarkable performance in image classification. While these deep learning systems are getting closer to practical deployment, the common assumption made about data is that it does not carry any sensitive information. This assumption may not hold for many practical cases, especially in the domain where an individual's personal information is involved, like healthcare and facial recognition systems. We posit that selectively removing features in this latent space can protect the sensitive information and provide better privacy-utility trade-off. Consequently, we propose DISCO which learns a dynamic and data driven pruning filter to selectively obfuscate sensitive information in the feature space. We propose diverse attack schemes for sensitive inputs & attributes and demonstrate the effectiveness of DISCO against state-of-the-art methods through quantitative and qualitative evaluation. Finally, we also release an evaluation benchmark dataset of 1 million sensitive representations to encourage rigorous exploration of novel attack schemes.

1. Introduction

Large deep neural network have resulted in breakthroughs across computer vision [1], speech recognition [2] and reinforcement learning [3] with their success largely attributed to their ability to efficiently learn complex patterns from data. The deployment of these algorithms in critical application domains such as healthcare and face-recognition has motivated a research focus on learning censored, unbiased and fair data representations to mitigate misuse by adversarial agents. Alternately, there can also be *sensitive* information in data which the user would like to keep private but the learned representations may inadvertently encode. This sensitive information may manifest as sensitive inputs or attributes. Consider a setup where citizens consent to usage of face recognition in public spaces for identifying criminals. During inference, feature representations are extracted for faces and identification is performed by matching in the feature space over an indexed database. While this may be a well-intended initiative, a malicious adversary may seek to intercept the feature representations to i) reconstruct the input face image or ii) extract personal attributes such as race, age, gender etc. The citizens did not consent to sharing this sensitive information which could be used to compromise their privacy and in a way that is biased or unfair to them. Exploring methods of improving privacy of the sensitive information (image, race, age, gender etc.) while preserving utility (identifying criminals) is the focus of this work.

Conventionally, research in privacy-aware machine learning has primarily focused on protecting training data from membership inference [4] and model inversion attacks [5], when i) training data is distributed over clients and ii) computation of training the model is out-sourced. For the former, distributed learning techniques such as federated learning [6, 7] and split learning [8, 9] are used, where clients communicate with a centralized server using weights and activations and the latter relies on homomorphic encryption [10, 11] and secure enclaves [12, 13]. Additionally, techniques such as multi-party computation [14, 15] and differential privacy [16, 17, 18, 19] have been employed to improve the privacy in federated-learning. While effective for training, scaling these methods for deployment at inference is a challenge for a variety of reasons. First, in several cases computational limitations and intellectual property considerations limit keeping the entire model on a client device. Secondly, cryptographic methods for training deep networks [20, 21, 22] are computationally very expensive operations which makes deploying models on the server infeasible when working with sensitive data. We posit that collaborative inference, where the inference network is distributed between client devices (client network) and a server (server network) which communicate via the *split activa-*

tions, presents a viable alternative. While amenable to scalability, it is important to encode explicit measures of security in the intermediate activations to protect privacy of the sensitive inputs and attributes.

While not motivating private collaborative inference, a few recent works [23, 24, 25] have attempted the related problem of attribute leakage [25, 26, 27] by focusing on adversarial representation learning (ARL). This couples together two entities, i) an adversarial network that seeks to extract a sensitive attribute from a given activation and, ii) a predictor network that intends to extract compact activations for accurate prediction of a task attribute (utility) while preventing the adversary from leaking the sensitive attribute (privacy). To balance this privacy-utility, [25] designed an objective to maximize entropy of the adversary network and [23, 24] to minimize likelihood of the predictor on the sensitive attributes.

Motivated by the above observations, in this work, we first examine existing ARL methods while reveals the presence of high redundancy in learned representations. We posit that selectively removing features in this latent space can protect the sensitive information and provide better privacy-utility trade-off. Consequently, we propose DISCO which learns a dynamic and data driven pruning filter to selectively to obfuscate sensitive information in the feature space. We validate DISCO and other baseline with multiple attacks on inputs and attributes. We observe that DISCO consistently achieves superior performance by disentangling representation learning from privacy using the pruning filter.

To this end, the contributions of this work can be summarized as follows:

- We introduce DISCO, a dynamic scheme for obfuscation of sensitive channels to protect sensitive information in collaborative inference. DISCO provides a steerable and transferable privacy-utility trade-off at inference without any retraining.
- We propose diverse attack schemes for sensitive inputs and attributes and achieve significant performance gain over existing state-of-the-art methods across multiple datasets.
- To encourage rigorous exploration of attack schemes for private collaborative inference, we release a benchmark dataset of **1 million** sensitive representations.

2. Related Work

Private Representation Learning [7, 8] propose mechanisms which allow for learning on data distributed across multiple agents with raw training data never leaving the corresponding client device. [28] further improves [7] by adding differentially private noise to weights of the trained model to prevent reconstruction of training data by inversion attacks. That said, techniques such as [28] are largely optimized to protect training data. In contrast, there is limited research on methods for privacy during inference via privatized activations. [29] introduced a distance correlation based regularization to decouple intermediate activations from input data while preserving performance on task attribute. [25] proposes to protect attribute leakage in classification tasks by minimizing mutual information of representations of private attribute and task attribute. In this work, we explore methods that seek to reduce redundancy and semantic integrity of activations to mitigate attacks on sensitive information.

Natural Pre-Image is a class of diagnostic techniques which are designed to reconstruct input image from intermediate activation and find utilization in computer vision tasks such as denoising, super-resolution etc. [30] leverages a randomly-initialized neural network and a hand crafted prior to invert deep neural representations and reconstruct the input. [31] seeks to train a decoder offline to learn to predict the input distribution. We leverage expected pre-image methods to formalize diverse attack schemes on sensitive inputs.

Bias in Machine Learning is a recent direction of ML research focused on two key problems: identifying and quantifying bias in datasets, and mitigating its harmful effects. The bias routinely manifests as some attributes of the input (eg. age, race, gender for faces) rather than the datapoints themselves (face images) and bias mitigation methods attempt to learn input (eg. face) representations that can decouple the task attribute (eg. gender) from the biased attribute (eg. age). A popular category of techniques involve adversarial representation learning [32, 23, 33] to mitigate the impact of the bias attribute on the task attribute. This family of adversarial mitigation techniques aligns with this work on selective privacy, with the private attribute analogous to the bias attribute, and a corresponding state-of-the-art [23] forms one baseline for our study.

Part-based Representation Learning involves splitting the image into several stripes to learn local representations and has achieved promising performance on computer vision tasks such as person re-identification which involves image retrieval under occlusions and partial observability. While sophisticated learning based partitioning methods have been explored [34, 35, 36, 37], methods such as [38] have achieved outstanding performance with trivial deterministic splitting. In this work, we adapt the static part-based techniques to decouple the intra-channel semantic consistency of convolutional activations for improving privacy-utility trade-offs in collaborative inference.

Channel Pruning is a prevalent technique for deep network compression to minimize computational complexity and accelerate inference [39]. While most methods interleave pruning with the training phase [40, 41, 42], there has been recent focus on pruning at inference [43]. [40] gradu-

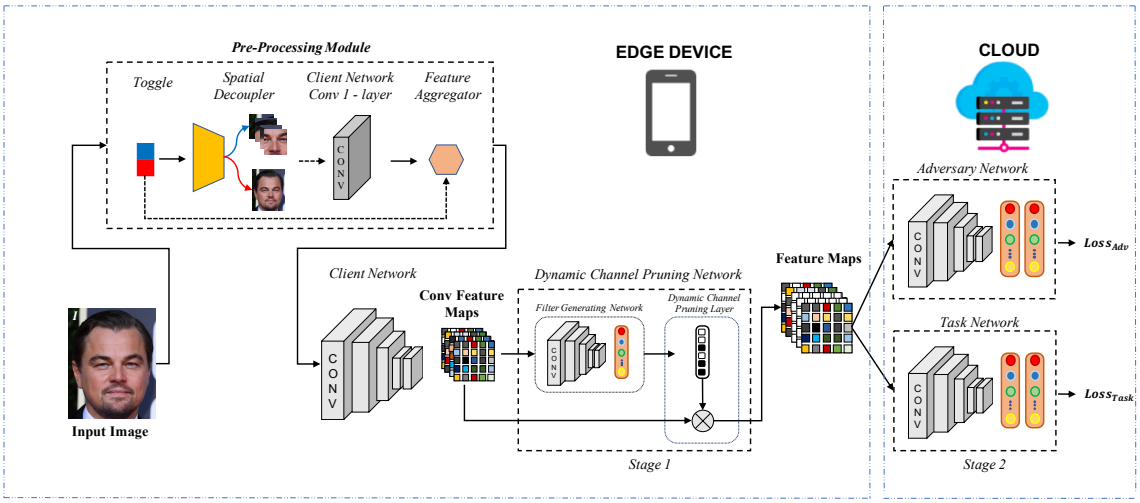

Figure 1: **DISCO for Privacy**. Input to the network is an image, as well as task labels and attribute labels to hide. The network is jointly optimized with a task objective to adaptively hide a given attribute without drop in performance of the target task.

ally prunes channels at fixed intervals during training using a feature relevance score to minimize compute cost. [43] propose dynamic feature boosting and suppression (FBS) to predictively amplify salient convolutional channels and skip unimportant ones at run-time for accelerated inference. In this work, our proposed method can be aligned with channel pruning but optimizes for a different objective of preventing leakage of sensitive information.

Filter Generating Networks (FGN) [44, 45], there is very limited literature on FGNs. One such module, the "Spatial Transformer" network, is proposed by Jaderberg et al. [45]. This spatial transformer module applies an affine transformation to feature maps to do translation and rotation for improved classification. Following [45], all these recent works [44, 46, 47] utilize the same concept to learn a steerable filter [44], weather prediction filter [46], an image enhancement filter [47], and a dynamic motion motion representation filter [48] using source-target image pairs. In contrast to these works, our focus is to learn dynamic filters that selectively prune channels which leak sensitive attributes without dropping in performance on the target task. The output of our dynamic channel pruning filters are binary (0 or 1) in nature, where 0 masks (or deactivates) channels that contribute to sensitive attributes, and 1 unmasks channels that contribute to the target task at hand. We clearly show that this form of channel pruning is quite effective in our experiments.

3. Methodology

First, we introduce the attack and threat models and then define the privacy considerations for our work. Finally, we formalize our privacy evaluation setup and delineate our proposed method DISCO: *Dynamic and Invariant Sensitive Channel Obfuscation* for protecting sensitive information in latent representation.

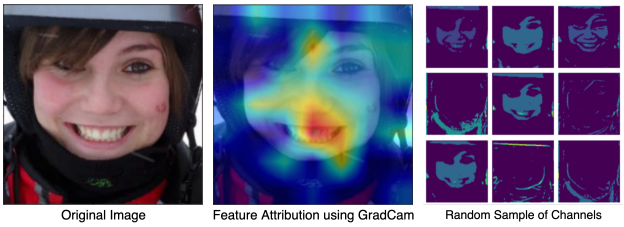

Figure 2: a) Input Image and Grad-CAM visualization from ResNet-18 classifier b) Corresponding convolution representations which encode inter-channel redundancy and preserve intra-channel semantic integrity.

3.1. Formulation

Setup. Consider a parameterized model $f(\theta;\cdot)$ trained to estimate the target attribute $y \in \mathcal{Y}_1$ for a given input image $x \in \mathcal{X}$. In many scenarios, x may be a sensitive input or have a sensitive attribute $\hat{y} \in \hat{\mathcal{Y}}$. Considerations for balancing compute feasibility and privacy has motivated private collaborative inference schemes [25, 24] that split $f(\theta;\cdot)$ into $f_1(\theta_1;\cdot)$ and $f_2(\theta_2;\cdot)$ where:

$$f_1(\theta_1;x) \in F_1 : \mathcal{X} \times \Theta_1 \rightarrow \mathcal{Z}$$

$$f_2(\theta_2;z) \in F_2 : \mathcal{Z} \times \Theta_2 \rightarrow \mathcal{Y}_1$$

such that $f_2 = (\theta_2; f_1(\theta_1;x))$ and $\theta = \{\theta_1, \theta_2\}$. We refer this vanilla collaborative inference scheme as *traditional* setup for collaborative inference. We formalize f_1 as the *client network* that is executed on a trusted device and f_2 as a *task network* which executes on an untrusted server using the *client activation* $z = f_1(\theta_1;x)$.

Threat Model. Under our threat model, a semi-honest [49] adversary exists on the side of the untrusted server.

The adversary would attempt to learn sensitive information about x by discovering an arbitrary sensitive attribute $\hat{y}$ or by reconstructing x itself. However, under this threat model, existence of the adversary should not alter the $\mathbb{P}_{\mathcal{Y}_1}(y_1)$. As a concrete example, x may be a face image with y as gender and $\hat{y}$ as racial identity. For the evaluation and algorithm design purposes, we build a proxy adversary that attempts to approximate the real world adversary. This proxy adversary is parameterized with an *adversarial network* $f_3(\theta_3; \cdot)$ that may intercept the payload z to extract the sensitive input x or the attribute $\hat{y}$. **Attack Model.** The adversary may utilize the activation z to perform a *reconstruction attack* to recover the sensitive input or a *leakage attack* to extract the sensitive attribute. We define the following attack models for the sensitive information z:

- *Supervised Decoder:* In this attack setting, the adversary leverages a small number of $(z, \hat{y})$ pairs to train a decoder network $\hat{f}(\hat{\theta})$ such that $\hat{y} = \hat{f}(z)$. The practical validity of this attack is in the scenarios where some finite number of pairs $(z, \hat{y})$ is obtained through a malicious or colluding client who is also participating in the collaborative inference setting. This attack scheme is inspired from [31, 52, 53]. Another practical scenario for this attack is where the pairs $(x, \hat{y})$ from the same distribution is publicly available, in such a case the client can train an auto-encoder and use the trained decoder for the attack. This technique can utilized for both *reconstruction attack* and *leakage attack*.

- *Likelihood Maximization:* Unlike the above scheme, here we do not require $(z, \hat{y})$ pairs to reconstruct the sensitive input, instead, the attacker uses the weights θ_1 of the client network and a randomly initialized network $\hat{f}(\hat{\theta}; \cdot)$ that generates an image $\hat{x}$ to produce $\hat{z} = f_1(\theta_1, \hat{x})$. Then the loss $\ell_2(\hat{z}, z)$ between random and sensitive activation is minimized by optimizing $\hat{\theta}$. This attack scheme is inspired by the deep image prior [30] for feature inversion. One drawback with this attack is that it is only applicable to the sensitive input protection and not sensitive attribute. This attack setting is stronger and harder to defend against because it does not require access to the $(z, \hat{y})$ pairs.

For the leakage attacks on sensitive attributes, we define a *Supervised Classifier* as the attack model that parameterizes the adversary as a vanilla neural network classifier to predict the leaked activation label pairs.

Privacy. Following the setup described in Hamm *et al.* [54], we define privacy as the expected loss over the estimation of sensitive information by the adversary. This privacy loss L_{priv}, given ℓ_p norm, for an adversary can be stated as:

$$L_{priv}(\theta_1, \theta_3) \triangleq E[\ell_p(f_3(f_1(x; \theta_1); \theta_3), \hat{y})]$$

Under this definition, releasing sensitive information while preserving privacy manifests as a min-max optimization between the data owner and the attacker. However, for training the model parameters, we use a proxy adversary from which gradients can be propagated. We refer the attack performed by this proxy adversary as *online attack*. We formalise our setup as an analogue but relax the non-invertibility assumption made by Hamm *et al.* [54] for the client f_1, following [55], to generalize for attacks on sensitive inputs with DNN learnt activations, as in our case. Additional details for the privacy framework are included in the *supplementary*.

3.2. Premise Validation

Adversarial representation learning is the existing state-of-the-art approach for performing inference [25, 24, 23] on sensitive data. Consider Figure 2 which visualizes the face image and the learned client activation in [23]. We note the following observations: a) the learned activations have high inter-channel redundancy, and b) individual feature maps preserve semantic integrity of the input image, especially with shallower client networks. Since gradient attribution in convnets is spatially localized [56], we posit that reducing this inter-channel redundancy and perturbing the intra-channel integrity of *client activations* can help achieve better privacy-utility trade-offs.

3.3. DISCO

DISCO, depicted in Figure 1, is composed of three key entities: a client, a predictor, and an adversary. The client transforms the input image to generate *client activations* which are communicated to the predictor for inferring the task attribute but can be collected by an adversary.

a) Client owns the sensitive information. Given an input $x \in R^{3\times H\times W}$, this entity participates in the collaborative inference and intends to achieve privacy in the *client activations* z it communicates.

Initially, x is passed through the *pre-processing module* where the *spatial decoupler* first decomposes it into d^2 disjoint spatial partitions $P_i \in R^{3\times \hat{H}\times \hat{W}}$ for $i = \{0, 1, 2, ..., d^2\}$ with $\hat{H} = H/d, \hat{W} = W/d$. Next, each of the partitions P_i is resized back to $H \times W$ and passed through a convolutional layer (with F filters) to generate $\hat{P}_i \in R^{F\times H'\times W'}$. Finally, the *feature aggregator* generates an aggregated representation $A \in R^{d^2\times H'\times W'}$ by averaging across channels and re-stacking each $\hat{P}_i$. A is then communicated to the client network. Here, we note that $d^2 = F$ in our pre-processing module so that the spatial decoupler can be easily bypassed (toggled-off) without altering the rest of the network architecture. We formalise the intuition and rational for this design choice in the *supplementary material*.

Next, the *client network* consumes the aggregated representation A and generates an intermediate activation $\hat{z} \in$

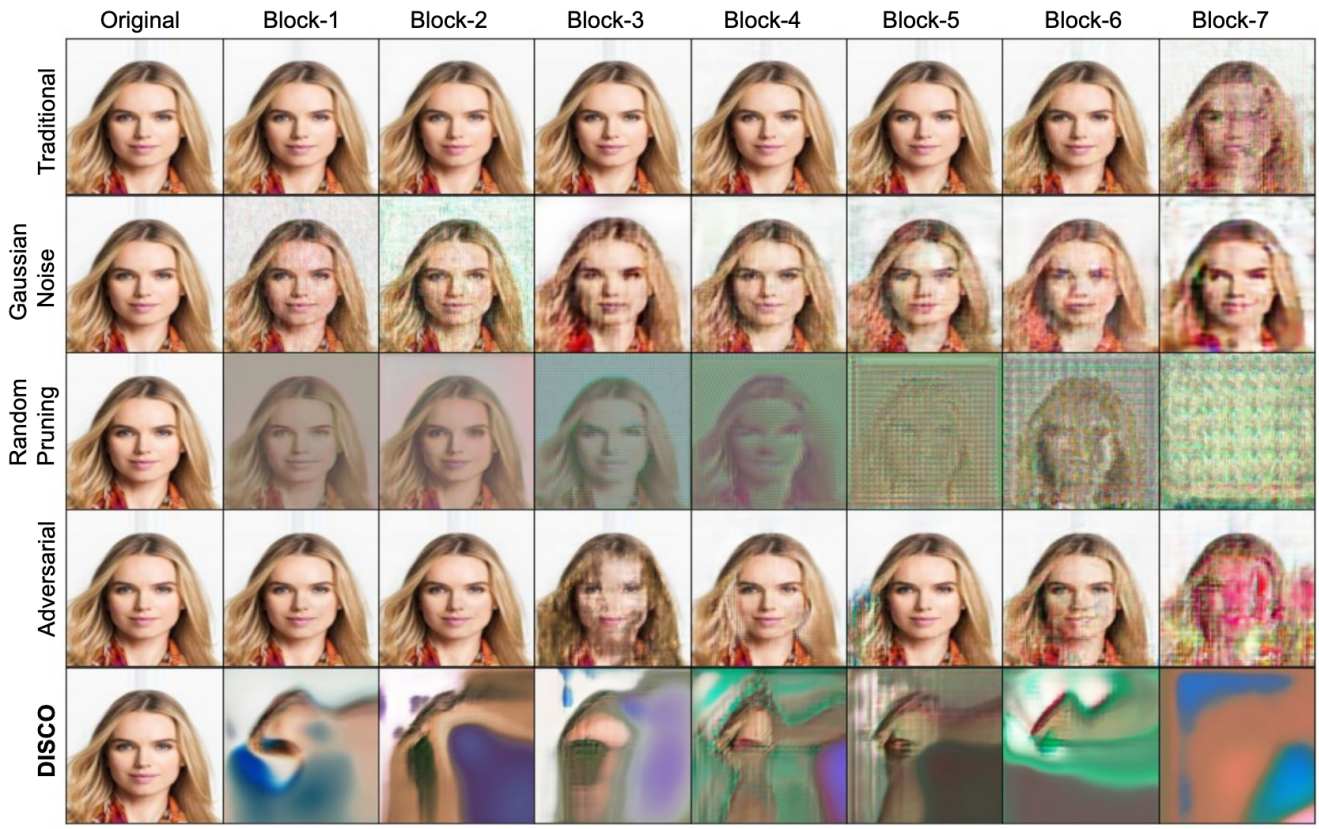

Figure 3: **Reconstruction results on CelebA [50]**: All of the reconstructed images are obtained from the activations using the likelihood maximization attack. We generate activations from the ResNet-18 [51] architecture where a set of convolution, batch normalization, and activation layers are grouped under a block. The first column shows the original sensitive input and remaining columns show its reconstruction across different blocks. For gaussian noise we use $\mu = -1., \sigma = 400$, this is the amount of noise at which the learning network gets utility close down to random chance. *Adversarial* refers to the set of techniques for filtering sensitive information using adversarial learning [24, 23]. For *DISCO* and *Random Pruning* we use a pruning ratio of $R = 0.6$.

$R^{C'' \times H'' \times W''}$. Finally, the *filter generating network* takes $\hat{z}$ as input and generates a feature map score $F \in R^{C''}$ for each channel in $\hat{z}$. The F channel pruning filters are weakly discretized using sigmoid with temperature (to avoid introducing discontinuity) and then thresholded to obtain a binary vector b. Then b is multiplied channel wise with $\hat{z}$ to produce a pruned feature volume z, the *client activation*, with channels leaking the sensitive information masked out (or deactivated) in the latent space. Note that F, the feature map score, is conditioned on $\hat{z}$ (hence x) and is thus generated dynamically on run-time per sample basis. A key idea of DISCO is to disentangle representation learning from privacy via the learned pruning filter. Specifically, the hyper-parameter pruning ratio R governs the number of active channels in the pruning filter and helps regulate the privacy-utility trade-off.

b) Predictor is an untrusted entity that receives the *client activations* (z) and executes the *task network* (f_2) to estimate the task attribute (y). The task network is optimized on the cross entropy loss (ℓ_{cce}) defined as:

$$loss_{task} = \ell_{cce}(f_2(\hat{z}), y) \tag{1}$$

c) Adversary manifests as a semi-honest entity that obtains the client activations (z) and executes the *adversarial network* (f_3) with the intent of extracting *sensitive* information - input or attribute. The adversary performs reconstruction attacks for obtaining the sensitive inputs or attribute leakage attacks to infer sensitive attributes. During the training, we design a proxy adversary that has access to the sensitive inputs (x) and attributes ($\hat{y}$). For reconstruction attacks, the adversarial network is a decoder module optimized using ℓ_1 loss against the input x. For attribute leakage attacks, the adversarial network is a convolutional classifier module optimized using ℓ_{cce} loss against the sensitive attribute $\hat{y}$. This loss can be summarized as :

$$loss_{adv} = \begin{cases} \ell_1(f_3(\hat{z}), x) & mode = SI \\ \ell_{cce}(f_3(\hat{z}), \hat{y})) & mode = SA \end{cases}$$

where, mode $\in$ [SI, SA] represents attack on sensitive input (SI) or sensitive attribute (SA).

3.4. Training

The utility of the task during inference depends upon parameters θ_1, ϕ, θ_2 learned during the training stage and can be expressed as

$$L_{util}(\theta_1, \phi, \theta_2) \triangleq E[\ell_u(f_2(g(f_1(x;\theta_1);\phi);\theta_2), y_1)]$$

As described previously we use a proxy adversary during the training and evaluation of our setup as described by the evaluation function L_{priv} to train the pruning network.

$$L_{priv}(\theta_1, \phi, \theta_3) \triangleq E[\ell_p(f_3(g(f_1(x;\theta);\phi);\theta_3), y_2)]$$

θ_3 is the parameters for the proxy adversary used during the training and evaluation. ℓ_u and ℓ_p is the loss function used for evaluating utility and privacy respectively. The adversary network and task network have access to supervised labels and attempt to minimize their losses L_{util} and L_{priv} respectively. The filter generating network is trained to minimize L_{util} and maximize L_{priv}, simulating an implicit min-max optimization for these two components. The client network parameters are only optimized to minimize L_{util}. We deliberately prohibit gradient flow from the adversary to the client network to ensure that the filter generating network generalizes and does not trivially utilize representations learned by the client network. This restricted gradient flow is one of the differentiating factor of our work from existing adversarial learning based methods [24, 23]. We posit that this facilitates the filter generating network to specialize at pruning by identifying the privacy leaking channels. This overall objective can be summarized as:

$$\min_{\phi}\left[\max_{\theta_3} -L_{priv}(\theta_1, \phi, \theta_3) + \min_{\theta}(-\rho \max_{w} -L_{util}(\theta_1, \phi, \theta_2))\right]$$

Here, ρ is chosen as a hyper-parameter to trade-off between accuracy and privacy. A high value of ρ would mean higher importance is placed on the utility during training.

3.5. Prediction

During the inference stage, computation for feature extraction $\hat{z} = f_1(x;\theta_1^*)$ and pruning $z = g(\hat{z};\phi^*;r)$ is performed on the trusted system and z is sent to the untrusted party that performs the remaining of the function evaluation. The value of pruning ratio r governs the total number of channels to be pruned from z and allows adjusting for the privacy and utility trade-off during runtime.

3.6. Generalization

The setup described in the main text is as follows for optimizing the parameters -

$$\min_{\phi}\left[\max_{\theta_3} -L_{priv}(\theta_1, \phi, \theta_3) + \min_{\theta}(-\rho \max_{w} -L_{util}(\theta_1, \phi, \theta_2))\right]$$

where $\theta_1, \phi, \theta_2, \theta_3$ are the parameters of the *filter generating network*, *client network*, and *server network* respectively. Let $\theta_1^*, \phi^*, \theta_2^*, \theta_3^*$ be the solution for the parameters we obtain by minimizing the expected loss. Let $\hat{\theta}_1, \hat{\phi}, \hat{\theta}_2, \hat{\theta}_3$ refers to the empirical minimizer of the above mentioned joint optimization. As noted before, we adapt to the setup described by Hamm [54]. However, a significant difference lies in the fact that θ_1 is not trained to minimize L_{priv} as this is to improve generalization of the ϕ across a different set of θ_1. The remaining parameters remain analogous to the min-max filters described in [54] .Following on that, we describe the joint loss as follows

$$L_J(\theta_1, \phi, \theta_2, \theta_3) = L_{util}(\theta_1, \phi, \theta_2) - \rho L_{priv}(\theta_1, \phi, \theta_3)$$

Let D be the original unknown data distribution and S be a set of samples obtained from the true distribution for calculating empirical loss then the empirical and expected loss can be bounded as follows, giving a generalization bound.

$$|E_D(L_J(\theta_1^*, \phi^*, \theta_2^*, \theta_3^*)) - E_S(L_J(\hat{\theta}_1, \hat{\phi}, \hat{\theta}_2, \hat{\theta}_3))| \leq 2 \sup_{\theta_1,\phi,\theta_2,\theta_3} |E_D(L_J(\theta_1, \phi, \theta_2, \theta_3)) - E_S(L_J(\theta_1, \phi, \theta_2, \theta_3))|$$

For more details, we refer the reader to the proof of theorem 1 shown in [54]. The equation above gives the bound on generalization error.

3.7. Effect of channel pruning on mutual information

We now study the effect of applying channel pruning of activations at the output of the client network with regards to the mutual information between the raw sample and the pruned activations. Inspired by the theoretical analysis in [57], we extend and adapt it to our setup of analyzing the reduction in mutual information between the *sensitive input* and *client activations* upon performing random pruning. We use the superscript notation $f_1^k(\theta_1^k; x)$ to denote the output of k'th layer of client network. We compare this with regards to no pruning and random pruning at the k'th layer of the client network as shown below.

Pre-pruning: The negative of the mutual information between the raw data and the output of 1'st layer prior to applying the pruning is given by

$$\begin{aligned} -\mathcal{I}(x; f_1^1(\theta_1^1; x)) &= -\mathcal{H}(f_1^1(\theta_1^1; x)) - \mathcal{H}(f_1^1(\theta_1^1; x)|x) \\ &= -\mathcal{H}(f_1^1(\theta_1; x)) \end{aligned}$$

as $-\mathcal{H}(f_1^1(\theta_1; x)|x) = 0$, due to $f_1^1(\cdot)$ being a deterministic function. Upon applying the data processing inequality, we have that the mutual information between the output of the k'th layer and the raw data satisfies:

$$\mathcal{I}(x; f_1^k(\theta_1^k; x)) \leq \mathcal{I}(x; f_1^{k-1}(\theta_1^k - 1; x)) \leq \ldots \leq \mathcal{I}(x; f_1^1(\theta_1^1; x))$$

where, we have the following relation $\mathcal{I}(x; f_1^k(\theta_1^k; x)) = \mathcal{H}(f_1^k(\theta_1^k; x))$.

Post-pruning: The mutual information after random pruning can be represented as a multiplication of the outputs at the k'th layer with a Bernoulli random variable $\mathcal{P}$ as $\mathcal{I}(x; f_1^k(x, \theta_1^k).\mathcal{P})$. In addition to the form of data processing inequality used in analysis of pre-pruning; there is an equivalent form of the classical data processing inequality given by

$$-\mathcal{I}(x; f_1^k(x, \theta_1^k).\mathcal{P}) \geq -\mathcal{I}(f_1^k(x, \theta_1^k); f_1^k(x, \theta_1^k).\mathcal{P})$$

Upon expanding this upper bound using entropy terms we get

$$\mathcal{I}(x; f_1^k(x, \theta_1^k).\mathcal{P}) \leq \mathcal{H}(f_1^k(\theta_1^k; x)) - \mathcal{H}(f_1^k(\theta_1^k; x)|f_1^k(\theta_1^k; x).\mathcal{P})$$

But $\mathcal{H}(f_1^k(\theta_1^k; x))$ is the mutual information in the case of pre-pruning as analyzed above. Therefore the decrease in information about raw data post-pruning is given by the term $\mathcal{H}(f_1^k(\theta_1^k; x)|f_1^k(\theta_1^k; x).\mathcal{P})$. Upon applying the Bayes rule (for conditional entropy), this term exactly equals:

$$\mathcal{H}(f_1^k(\theta_1^k; x).\mathcal{P}|f_1^k(\theta_1^k; x)) + \mathcal{H}(f_1^k(\theta_1^k; x)) - \mathcal{H}(f_1^k(\theta_1^k; x).\mathcal{P})$$

Since the term $f_1^k(\theta_1^k; x)$ is independent of the noise $\mathcal{P}$, the above can be further rearranged as

$$\mathcal{H}(f_1^k(\theta_1^k; x).\mathcal{P}|f_1^k(\theta_1^k; x)) + \mathcal{H}(f_1^k(\theta_1^k; x)) - \mathcal{H}(f_1^k(\theta_1^k; x)) - \mathcal{H}(\mathcal{P})$$

which simplifies to $\mathcal{H}(f_1^k(\theta_1^k; x).\mathcal{P}|f_1^k(\theta_1^k; x)) - \mathcal{H}(\mathcal{P})$. As we chose $\mathcal{H}(\mathcal{P})$ to be a Bernoulli random variable; upon considering its success probability to be p (lower-case) and probability of failure to be $q = 1 - p$, we have -$\mathcal{H}(\mathcal{P}) = plog(p) + qlog(q)$. Therefore, upon performing random pruning the decrease in mutual information amounts to

$$\mathcal{H}(f_1^k(\theta_1^k; x).\mathcal{P}|f_1^k(\theta_1^k; x)) - \mathcal{H}(f_1^k(\theta_1^k; x)) + plog(p) + qlog(q)$$

while the mutual information post-pruning is upper bounded by $\mathcal{H}(f_1^k(\theta_1^k; x).\mathcal{P}|f_1^k(\theta_1^k; x)) + plog(p) + qlog(q)$.

4. Discussion: Dynamic Design of *DISCO*

A key idea behind DISCO is the decoupling of privacy considerations from representation learning using the dynamic pruning filter. We analyse the dynamic formulation of this design along the following dimensions:

- Dynamic Private Representations: The filter generating network in DISCO estimates the pruning filter for each input, independently at run-time. Since different convolutional filters are known to activate differently [43], the dynamic channel pruning in DISCO enables more personalized identification of sensitive channels for each input resulting in better privacy-utility trade-offs.
- Dynamic Integration: We train DISCO in two phases as i) train the client and the predictor networks to maximize utility ii) train filter generating network with predictor and the (proxy) adversary to minimize privacy leakage and preserve utility. We note that deliberately prohibiting gradient flow from adversary to the client when training the filter generator equips DISCO for dynamic integration. Specifically, this decoupling of filter generation from the client network, enables private *expert filters* that can obfuscate sensitive attributes and be employed by a network running DISCO. For example, one can build a dictionary of DISCO modules for different sensitive attributes for faces such as race, gender, eyeglasses, and etc. can be trained and used by different vendors based on their context for privacy and utility.
- Dynamic Privacy Utility Trade-offs: All previous methods weight seek to balance privacy-utility during training by weighting the corresponding losses. However, once the model is trained, the privacy-utility trade-off is frozen. In contrast, DISCO is the *first* method where we can dynamically vary privacy-utility at inference by tweaking the pruning ratio (R). This dynamic adjustment enables one to continuously control the privacy offered by deployed systems without having to interrupt or retrain the machine learning service from scratch.

5. Experiments

In this section, we formalize the setup for our experiments with sensitive attributes and inputs. We delineate, in order, the datasets, implementation details, evaluation metrics, and baselines for our study.

Datasets We conduct experiments with the following datasets:

- Fairface [58] dataset consists of 108,501 images, with race, gender, and age groups. The dataset is designed with the emphasis of balanced race composition which we preserve in our experimental train and test sets. For our experiments, the task attribute is gender and the sensitive attribute is race.
- CelebA [50] consists of 202,599 celebrity face images across 10,177 identities, each with 40 attribute annotations. For our experiments, we define the task attribute as emotion and sensitive attribute as gender.
- CIFAR [59] consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. We manually label each of the 10 classes as living or non-living. For our experiments, the task attribute is the class label and sensitive attribute is living/non-living, as introduced in [25].

Implementation Details Experiments are implemented using Pytorch and conducted using NVIDIA Tesla V100 GPUs. The backbone network is ResNet-18 [51] with the *client activations* obtained from the block-4, unless specified otherwise. Additional details including hyperparameters can be found in the *supplementary*.

Evaluation Metrics We measure utility using top-1 accuracy on the task attribute. For attacks on sensitive inputs, we measure privacy using ℓ_1 loss, SSIM and PSNR [60] between reconstructed and input image and top-1 accuracy on the private attribute for attacks on sensitive attributes. The

Baselines For attacks on sensitive attributes, we baseline with [23, 25, 24] which are state-of-the-art on attribute leakage. For attacks on sensitive inputs, we baseline with [23, 24] and two randomized variants of DISCO where filter pruning is replaced with: i) random pruning ii) gaussian noise. Finally, for both sensitive inputs and attributes, we also compare with a vanilla CNN model, denoted as*traditional*, with no activation privacy.

6. Attack Benchmark

As we strive towards rigorous understanding of privacy for collaborative inference, we also release an evaluation benchmark for attack models on sensitive inputs and attributes. The benchmark consists of 1 million pairs of activations, model weights, and inputs images for online and offline attacks. The benchmark includes samples from 3 datasets: CelebA, CIFAR-10, and FairFace for multiple recent techniques focused on privacy in collaborative inference: DISCO, Max Entropy [25], and Learning not to learn [23]. The preliminary version of this benchmark can be found at http://tiny.cc/pci

7. Results

Sensitive Attribute. For sensitive attributes, we perform qualitative analysis and report performance in Table 1. We mention accuracy of the adversary on the sensitive attribute (i.e. *privacy*) and of the predictor on the task attribute (i.e. *utility*). We note that DISCO provides the best *privacy-utility* trade-off on each of these datasets. Specifically, on the CIFAR-10 dataset [59], without loss of utility, we improve on decreasing the adversary accuracy to **0.2282** from **0.3573** in [25], the most recent state-of-the-art.

Sensitive Input. For sensitive inputs, we perform both quantitative (Table 2) and qualitative analysis (Figure 4) for stronger *likelihood maximization* attacks. The visual results highlight that DISCO achieves significantly better obfuscation in the reconstructed input. This is corroborated by the quantitative results where DISCO obtains an SSIM of **0.38** and PNSR of **11.61** as against **0.68** and **20.49** for adversarial class of techniques [24, 23]. Please note that while other techniques may also provide some level of obfuscation in

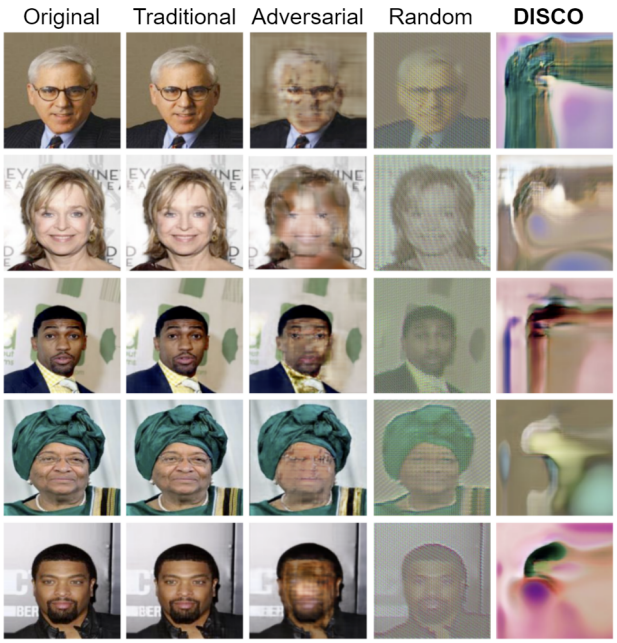

Figure 4: **Reconstruction attack qualitative evaluation**: We show the reconstruction quality across traditional collaborative inference, adversarial, and ours. Note that the utility performance on the target task of gender classification does not suffer from accuracy degradation in our task.

reconstruction, DISCO is the *only* technique which is able to additionally prohibit the *re-identification* of the input image. (compare columns 3 and 5 in Figure 4)

Next, Figure 3 presents the reconstructed outputs for varying depths of the client activations (from block-1 to block-7 of the ResNet-18 [51]). The results depict that all our baseline techniques depict a progressive worsening of performance as we move towards shallower client activations (lower resnet blocks). In contrast, we note that DISCO is still consistently protect the input from the likelihood maximization attack with its superiority over baselines higher with earlier client activations which have higher redundancy. This validates the motivation and formulation of the *pre-processing module* of DISCO.

8. Discussion

In this section, we present the motivation and analyse the implication of various design choices for *DISCO*.

i) Privacy-Utility for Correlated Attributes While users idealize high privacy-utility guarantees, we posit that what level can be empirically realized is conditioned on the similarity of the task and sensitive attribute. To corroborate this position, we conduct leakage attacks using DISCO and *traditional* with the following attribute configuration: corroborate this with observations from the following experiments on the celebA dataset:

Method	Privacy (Fairface) ↓	Utility (Fairface) ↑	Privacy (CelebA) ↓	Utility (CelebA) ↑	Privacy (CIFAR10) ↓	Utility (CIFAR10) ↑
[61]	0.319	**0.824**	0.729	**0.916**	0.912	0.498
DISCO	**0.190**	0.815	**0.612**	0.910	**0.223**	0.9198
[25]	0.236	0.802	0.780	0.880	0.358	0.915
[23]	0.193	0.815	0.675	0.905	0.526	**0.924**

Table 1: **Comparison for sensitive attribute leakage:** We compare our approach on sensitive attribute leakage with the existing works. For the fairface dataset, sensitive attribute is race and task attribute is gender. In the CelebA dataset, sensitive attribute is gender and task attribute is smiling. The adversary accuracy is reported on the supervised reconstruction attack as described in 3.1, for all the three methods, adversary accuracy is close to random chance, indicating that evaluation of privacy just by analyzing the adversary proxy during the training may give a false sense of privacy.

	SSIM ↓	PSNR ↓	ℓ_1 ↑	Utility ↑
Traditional [61]	0.88 ± 0.03	31.58 ± 2.44	108.82 ± 8.92	**97.35**
Adversarial [23]	0.68 ± 0.12	20.49 ± 5.94	123.33 ± 20.67	97.15
DISCO	**0.38±0.09**	**11.61±1.91**	**125.34±15.29**	95.66

Table 2: **Comparison for sensitive input leakage:** We compare our approach on sensitive input reconstruction task and compare with our baselines and the existing works.

Sensitive Attribute	Method	Privacy (↓)	Utility (↑)
Mouth Open (S1)	[61]	0.814	0.893
	DISCO	**0.783**	**0.907**
Big Nose (S2)	[61]	0.616	**0.896**
	DISCO	**0.559**	0.893

Table 3: Privacy-utility trade-offs is influenced by correlation of task and sensitive attribute. The task attribute here is *Smiling* (yes/no). Both sensitive attributes are binary.

- **S1**: Sensitive Attribute is *Mouth Open* (yes/no) and the Task Attribute is *Smiling* (yes/no)
- **S2**: Sensitive Attribute is *Nose Size* and Task Attribute is *Smiling* (yes/no)

Results in Table 3 indicate DISCO achieves near-perfect privacy and high utility in S2, the privacy-utility worsens for S1 where the sensitive attribute (*mouth open*) is strongly correlated with task attribute (*smiling*) due to spatial overlap of the corresponding regions of interest.

ii) Comparing with Activation Noise for Privacy Adding noise to the output of a statistical query (*client activations* in this case) is a well known mechanism for privatizing sensitive data. These mechanisms are sometimes built under the framework of differential privacy [16] or its variants [62, 63]. While we do not compare or operate under a strict differentially private mechanism, we posit that preventing sensitive input reconstruction requires a heavy amount of noise. To validate this, we design an experiment where we add Gaussian noise to the *client activations* and incrementally increase σ until the reconstruction is prevented. We also measure the difference in utility obtained by these noise based mechanisms. Compared to the learning based approaches like adversarial and ours, achieving privacy through random noise comes at a heavy cost of deteriorating utility. More precisely, we obtain final training and test accuracy close to random chance with noise that is capable of preventing reconstruction attack $\mu = -1, \sigma = 400$.

9. Conclusion

In this work, we focus on selective privacy of sensitive information in learned representations. We posit that selectively removing features in this latent space can protect the sensitive information and provide better privacy-utility trade-off. Consequently, we introduce DISCO, a dynamic scheme for obfuscation of sensitive channels to protect sensitive information in collaborative inference. DISCO provides a steerable and transferable privacy-utility trade-off at inference without any retraining. We propose diverse attack schemes for sensitive inputs and attributes and achieve significant performance gain over existing methods on multiple datasets. To encourage rigorous exploration of attack schemes for private collaborative inference, we also release a benchmark dataset of 1 million sensitive representations.

References

[1] LC Yan, B Yoshua, and H Geoffrey. Deep learning. nature, 521(7553):436–444, 2015. 1

[2] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In International conference on machine learning, pages 173–182, 2016. 1

[3] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A

brief survey. IEEE Signal Processing Magazine, 34(6):26–38, 2017. 1

[4] 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. 1

[5] 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. 1

[6] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977, 2019. 1

[7] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492, 2016. 1, 2

[8] Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. CoRR, abs/1810.06060, 2018. 1, 2

[9] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. arXiv preprint arXiv:1812.00564, 2018. 1

[10] Craig Gentry and Dan Boneh. A fully homomorphic encryption scheme, volume 20. Stanford university Stanford, 2009. 1

[11] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory (TOCT), 6(3):1–36, 2014. 1

[12] Fan Zhang, Ziyuan Liang, Cong Zuo, Jun Shao, Jianting Ning, Jun Sun, Joseph K Liu, and Yibao Bao. hpress: A hardware-enhanced proxy re-encryption scheme using secure enclave. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020. 1

[13] Andrew Ferraiuolo, Andrew Baumann, Chris Hawblitzel, and Bryan Parno. Komodo: Using verification to disentangle secure-enclave hardware from software. In Proceedings of the 26th Symposium on Operating Systems Principles, pages 287–305, 2017. 1

[14] Manoj M Prabhakaran and Amit Sahai. Secure multi-party computation, volume 10. IOS press, 2013. 1

[15] David Evans, Vladimir Kolesnikov, and Mike Rosulek. A pragmatic introduction to secure multi-party computation. Foundations and Trends® in Privacy and Security, 2(2-3), 2017. 1

[16] Cynthia Dwork. Differential privacy: A survey of results. In International conference on theory and applications of models of computation, pages 1–19. Springer, 2008. 1, 9

[17] Cynthia Dwork and Adam Smith. Differential privacy for statistics: What we know and what we want to learn. Journal of Privacy and Confidentiality, 1(2), 2010. 1

[18] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3-4):211–407, 2014. 1

[19] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 375–403. Springer, 2019. 1

[20] Ehsan Hesamifard, Hassan Takabi, and Mehdi Ghasemi. Cryptodl: Deep neural networks over encrypted data. arXiv preprint arXiv:1711.05189, 2017. 1

[21] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. {GAZELLE}: A low latency framework for secure neural network inference. In 27th {USENIX} Security Symposium ({USENIX} Security 18), pages 1651–1669, 2018. 1

[22] Karthik Nandakumar, Nalini Ratha, Sharath Pankanti, and Shai Halevi. Towards deep neural network training on encrypted data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 0–0, 2019. 1

[23] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019. 2, 4, 5, 6, 8, 9

[24] Ang Li, Jiayi Guo, Huanrui Yang, and Yiran Chen. Deepobfuscator: Adversarial training framework for privacy-preserving image classification, 2019. 2, 3, 4, 5, 6, 8

[25] Proteek Chandan Roy and Vishnu Naresh Boddeti. Mitigating information leakage in image representations: A maximum entropy approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019. 2, 3, 4, 7, 8, 9

[26] Yanai Elazar and Yoav Goldberg. Adversarial removal of demographic attributes from text data. arXiv preprint arXiv:1808.06640, 2018. 2

[27] John Martinsson, Edvin Listo Zec, Daniel Gillblad, and Olof Mogren. Adversarial representation learning for synthetic replacement of private attributes. arXiv preprint arXiv:2006.08039, 2020. 2

[28] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pages 308–318, 2016. 2

[29] Praneeth Vepakomma, Abhishek Singh, Otkrist Gupta, and Ramesh Raskar. Nopeek: Information leakage reduction to share activations in distributed deep learning, 2020. 2

[30] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Deep image prior. CoRR, abs/1711.10925, 2017. 2, 4

[31] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4829–4837, 2016. 2, 4

[32] Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8919–8928, 2020. 2

[33] Mohsan Alvi, Andrew Zisserman, and Christoffer Nellåker. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In Proceedings of the European Conference on Computer Vision (ECCV), pages 0–0, 2018. 2

[34] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification, 2018. 2

[35] Hao Liu, Jiashi Feng, Meibin Qi, Jianguo Jiang, and Shuicheng Yan. End-to-end comparative attention networks for person re-identification. IEEE Transactions on Image Processing, 26(7):3492–3506, Jul 2017. 2

[36] Liming Zhao, Xi Li, Jingdong Wang, and Yueting Zhuang. Deeply-learned part-aligned representations for person re-identification, 2017. 2

[37] Dong Yi, Zhen Lei, and Stan Z. Li. Deep metric learning for practical person re-identification, 2014. 2

[38] G. Wang, Y. Yuan, X. Chen, J. Li, and X. Zhou. Learning Discriminative Features with Multiple Granularities for Person Re-Identification. ArXiv e-prints, April 2018. 2

[39] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282, 2017. 2

[40] Sai Aparna Aketi, Sourjya Roy, Anand Raghunathan, and Kaushik Roy. Gradual channel pruning while training using feature relevance scores for convolutional neural networks. abs/2002.09958, 2020. 2

[41] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 2

[42] Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jinhui Zhu. Discrimination-aware channel pruning for deep neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems 31, pages 875–886. Curran Associates, Inc., 2018. 2

[43] Xitong Gao, Yiren Zhao, Łukasz Dudziak, Robert Mullins, and Cheng zhong Xu. Dynamic channel pruning: Feature boosting and suppression. In International Conference on Learning Representations, 2019. 2, 3, 7

[44] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In Advances in neural information processing systems, pages 667–675, 2016. 3

[45] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In NIPS, 2015. 3

[46] Benjamin Klein, Lior Wolf, and Yehuda Afek. A dynamic convolutional layer for short range weather prediction. In CVPR, 2015. 3

[47] Vivek Sharma, Ali Diba, Davy Neven, Michael S Brown, Luc Van Gool, and Rainer Stiefelhagen. Classification-driven dynamic image enhancement. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4033–4041, 2018. 3

[48] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. Dynamonet: Dynamic action and motion network. In Proceedings of the IEEE International Conference on Computer Vision, pages 6192–6201, 2019. 3

[49] Carmit Hazay and Yehuda Lindell. A note on the relation between the definitions of security for semi-honest and malicious adversaries. IACR Cryptol. ePrint Arch., 2010:551, 2010. 3

[50] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. Retrieved August, 15:2018, 2018. 5, 7

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. CoRR, abs/1512.03385, 2015. 5, 8, 13

[52] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. CoRR, abs/1412.0035, 2014. 4

[53] Alexey Dosovitskiy and Thomas Brox. Inverting convolutional networks with convolutional networks. CoRR, abs/1506.02753, 2015. 4

[54] Jihun Hamm. Minimax filter: Learning to preserve privacy from inference attacks. Journal of Machine Learning Research, 18(129):1–31, 2017. 4, 6

[55] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. Why are deep nets reversible: A simple theory, with implications for training. CoRR, abs/1511.05653, 2015. 4

[56] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. International Journal of Computer Vision, 128(2):336–359, Oct 2019. 4

[57] Fatemehsadat Mireshghallah, Mohammadkazem Taram, Prakash Ramrakhyani, Dean M. Tullsen, and Hadi Esmaeilzadeh. Shredder: Learning noise to protect privacy with partial DNN inference on the edge. CoRR, abs/1905.11814, 2019. 6

[58] Kimmo Kärkkäinen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age, 2019. 7

[59] Alex Krizhevsky and Geoff Hinton. Convolutional deep belief networks on cifar-10. Unpublished manuscript, 40(7):1–9, 2010. 7, 8

[60] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In 2010 20th international conference on pattern recognition, pages 2366–2369. IEEE, 2010. 8

[61] Seyed Ali Osia, Ali Taheri, Ali Shahin Shamsabadi, Kleomenis Katevas, Hamed Haddadi, and Hamid R. Rabiee. Deep private-feature extraction, 2018. 9

[62] Ilya Mironov. Rényi differential privacy. In 2017 IEEE 30th Computer Security Foundations Symposium (CSF), pages 263–275. IEEE, 2017. 9

[63] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? SIAM Journal on Computing, 40(3):793–826, 2011. 9

[64] Ning Qian. On the momentum term in gradient descent learning algorithms. Neural networks, 12(1):145–151, 1999. 13

Appendices

A. Hyper-parameters and Experimental Setup

All of the experimental setup is implemented in PyTorch and we will be releasing the codebase for all of different quantitative and qualitative experiments, with the random seeds used in all of the experiments.

Network architecture: We describe four distinct networks in the section 3, *client network*, *filter generating network*, *adversary network*, *task network*. We use ResNet-18 [51] as the base architecture for all of the four networks. For alignment of the architecture we experiment with the different blocks of the ResNet architecture and split the network such that output of the *client network* is fed to all three *filter generating network*, *adversary network*, and *task network*. The *filter generating network* has same number of neurons in the final fully connected layer as number of channels in the output produced by *client network*. The sigmoid temperature is 0.03 for the filter generating network. We adapt the ResNet backbone for *adversary network* when the protected attribute is sensitive input since it requires to build a generative model conditioned on *client activations*. We use a transpose convolution based architecture that upsamples the feature map to a higher dimensionality resulting in final image.

Pre-processing module described in the section 3.3.a is composed of a single convolution layer and a *spatial decoupler* that splits the feature-map into d^2 spatially disjoint partitions. For an image size of 112 and target d^2 to be 64, the resulting featuremap size is 14×14 that gets rescaled back to 112×112 using bilinear interpolation. We keep the value of the d^2 as 64 to make sure that the averaging in the channel space results in 64 distinct feature maps that can be fed into the remaining of the architecture, this allows compatibility of the *pre-processing module* with off the shelf architectures.

Optimizer: We use SGD optimizer with momentum [64] for all of the networks with a learning rate of 0.01

B. Reconstruction Results

We present more reconstruction results for the qualitative comparison. Our results indicate that supervised decoder based attack model performs significantly better than likelihood maximization attack for *DISCO*, however, for all other techniques, likelihood maximization attack provides much better reconstruction quality. The figure can be found on the next page.

Figure 5: Qualitative comparison for different techniques using the supervised decoder attack described in the Section 3.1. *DISCO (Off)* refers to DISCO with pre-processing module's toggle turned off. This technique results in a different yet realistic reconstruction for even *DISCO* compared to deep image prior results shown in the Figure 3.