# *NoPeek-Infer: Preventing face reconstruction attacks in distributed inference after on-premise training*

**Citation:** Vepakomma, Praneeth, Singh, Abhishek, Zhang, Emily, Gupta, Otkrist and Raskar, Ramesh. 2021. "NoPeek-Infer: Preventing face reconstruction attacks in distributed inference after on-premise training." 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021).

**As Published:** 10.1109/FG52635.2021.9667085

**Publisher:** Institute of Electrical and Electronics Engineers (IEEE)

**Persistent URL:** https://hdl.handle.net/1721.1/146535

**Version:** Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

# NoPeek-Infer: Preventing face reconstruction attacks in distributed inference after on-premise training

Praneeth Vepakomma[1], Abhishek Singh[1], Emily Zhang[1], Otkrist Gupta[2], Ramesh Raskar[1]
[1] Massachusetts Institute of Technology
[2] Lendbuzz

***Abstract*— For models trained on-premise but deployed in a distributed fashion across multiple entities, we demonstrate that minimizing distance correlation between sensitive data such as faces and intermediary representations enables prediction while preventing reconstruction attacks. Leakage (measured using distance correlation between input and intermediate representations) is the risk associated with the reconstruction of raw face data from intermediary representations that are communicated in a distributed setting. We demonstrate on face datasets that our method is resilient to reconstruction attacks during distributed inference while maintaining information required to sustain good classification accuracy. We share modular code for performing NoPeek-Infer at http://tiny.cc/nopeek along with corresponding trained models for benchmarking attack techniques.**

## I. Introduction

Data sharing and distributed computation while preserving privacy and safety has been identified amongst important current trends in the adoption of computer vision and machine learning technologies. In this setting with several client-server entities interacting in a distributed fashion, there is a need for privacy preserving technologies to handle face and gesture data such that attackers residing in one or more entities cannot reconstruct face data belonging to genuine clients. This would help to deploy powerful face recognition technologies such as biometric authentication, facial expression analysis and consumer attention/engagement analysis in a truly distributed fashion across a wide array of device types while maintaining privacy.

We now elaborate on the sub-problem of private collaborative inference that is the setting in which this paper proposes a method to prevent face reconstruction attacks. With rapid advances in computing, organizations are now able to train ultra-large machine learning models on huge data sets with massive computing resources. This opens up a new set of problems for external clients that intend to predict with these models on their query test data. The client would not like to download these large models in their entirety on their device given that they often have billions of parameters. Generation of predictions with these trained models is computationally intensive if solely performed on-device by the client.

In this setting, we propose a method (NoPeek-Infer) for the client to share activations from a chosen intermediate layer such that reconstruction attacks of raw face data can be prevented while the rest of the prediction after this layer is performed on a server. We test this on several face datasets to measure the efficacy of NoPeek-Infer on preventing face reconstruction attacks within the task of distributed predictive inference.

### A. Motivation

**Activation sharing:** This setting of distributed learning with communication of intermediate activations upon splitting the deep learning model such that some layers lie with the client and the rest with the server is popular in *split learning* [12], [44], an important variant of *federated learning* [19], [28], [18]. Sharing of activations from intermediate layers is also relevant in distributed learning approaches of local parallelism [21], features replay [17], divide and conquer quantization [11] and in task-independent privacy-respecting data crowdsourcing [22]. The client's data records on which the predictions need to be obtained are private and therefore the model's intermediate representations (or activations) that are communicated in this setting need to be desensitized to prevent reconstruction attacks. This opens up the relatively new problem of private collaborative inference (PCI) where the model is split across the client and server.

This is in contrast to an alternate setting of federated learning with considerable existing work where the server intends to privately share the weights of a trained model, the privacy desired is with regards to the server's own training data. Traditionally two standard modes of machine learning deployment exist for practical applications: a.) on-device prediction and b.) machine learning as a service (MLaaS). In the MLaaS setup, the service provider is assumed to be trusted by the client using the service. The assumption is not valid if the client's data is sensitive.

The following issues motivate the design of practical PCI algorithms and systems for on-device prediction:

1) **Computation efficiency:** Recent state-of-the-art models require a lot of computation even during inference. These models cannot fit into hardware limited devices such as smartphones and other edge/IoT devices.

2) **Secrecy of the models:** Parameters of a model or architecture can be a secret or intellectual property of the server. In such cases, it is not possible to ship the models locally.

3) **Shipping updates to the model:** To update the model parameters, the server needs to apply updates to all clients in on-device machine learning. Within PCI, server can update server-side parameters and treat the client's model as frozen.

**Privacy preserving ML for faces:** Recent privacy preserving machine learning techniques applied to face data include blurring techniques such as [33]. We compare the performance of NoPeek-Infer against this method in the experimental section. Earlier works on blurring such as [32] have shown how earlier approaches of blurring fail to preserve privacy in home-based video conferencing setups. Other recent baselines we compare our method against include siamese embedding based privacy [34] and adversarial baselines such as DeepObfuscator [23] that was originally benchmarkd on faces and with Privacy Adversarial Networks [26] that was benchmarked on non face images. We note that recent works such as [6] were applied to faces for preventing reconstruction of specifically chosen attributes about the face as opposed to altogether preventing reconstruction of the entire face. We note that NoPeek-Infer deals with this latter problem of preventing face reconstruction attacks as opposed to any attribute specific reconstruction. In addition, we also compare against other face reconstruction defenses such as noising and blur based approaches.

## II. Contributions

This paper proposes a way to mitigate reconstruction attacks on raw data in the distributed machine learning settings of private collaborative inference. To this end, the contributions of this work can be summarized as follows:

**1**) We introduce NoPeek-Infer to prevent reconstruction attacks during activation sharing in PCI via minimization of a statistical dependency measure called distance correlation [41], [40]between raw data and any intermediary communications across the clients or server.

**2**) We evaluate the performance of our method on face datasets and share detailed results upon applying two state of the art reconstruction attacks: i) supervised decoder attacks and ii) likelihood maximization attacks in addition to some standard baselines. The likelihood maximization attack has not received attention in current works on private activation sharing while it has been widely used [43] in the computer vision community of late.

**3**) In order to promote rigorous benchmarking in the PCI domain, we introduce a dataset of privatized activations using different PCI techniques for two face datasets Fairface [20] and CelebA [27]. This dataset will act as a benchmark for the evaluation of existing and future attack and defense techniques.

### A. Benefits of NoPeek-Infer

**1**) A key benefit of the NoPeek-Infer defense over other existing defenses is that it does not require any additional adversarial network for it to be learnt unlike the rest. This reduces the number of parameters that need to be trained in NoPeek-Infer in comparison to other existing defense methods.

**2**) NoPeek-Infer does not require any modification to the client side architecture which holds the network up to an intermediate layer unlike existing methods thereby making it highly suitable for the machine learning as a service (MLaaS) mode of deployments.

## III. Related work:

### A. Attacks

Attacks in distributed machine learning can be categorized as shown in Table Ia based on time of attack (during train/test) and mode of training (distributed, peer to peer, on-premise). Other factors include the type of input (entire dataset/specific attributes) that the malicious attacker has access to and the target dataset that it aims to reconstruct. Attackers can reside in any client or server that receives communications from another client. We now enumerate various reconstruction attacks. We compare the performance of NoPeek-Infer in defending against the supervised decoder and likelihood maximization based reconstruction attacks. These two are the most relevant to our settings from this list of attacks.

**1**) **Feature space hijacking attack** is applied for distributed training of neural networks to reconstruct private data samples from the shared activations [36]. As opposed to their setup, our focus in NoPeek-Infer is to protect client's query data in distributed prediction/inference phase.

**2**) **Federated/client-side attack:** In federated learning [28], [19], [18], the untrusted party has access to the averaged weights of all the clients. Similarly in split learning [12], [44], the local weights of the client-side network need to be shared peer to peer with one other adjacent client.

**3**) **Attribute attack:** In this setting the attacker attempts to reconstruct only a subset of input data attributes that are considered to be sensitive [13], [29], [37], [38], [31], [47] as opposed to the entire input sample as in NoPeek-Infer.

**4**) **Offline supervised decoder attack:** In a worst-case reconstruction attack setting, the attacker has access to a leaked subset of samples of training data $x$ along with corresponding transformed activations $z$ at a given layer, which are always exposed to other clients/server for distributed training of the network to be possible. The attacker could reside in any untrusted client or server that is part of the distributed training setup. The attacker also has access to the rest of the activations corresponding to unleaked training data at the same layer. This is also by design, for distributed training to be possible. The attacker tries to learn an image to image translation model from the transformed activations to the leaked raw data. The attacker can then use this model to reconstruct raw data from activations corresponding to unleaked training data or unleaked test/validation data. This offline attack is also illustrated in Figure 1.

**5**) **Likelihood maximization attack:** Unlike the above scheme, this attack does not require pairs of raw images and corresponding activations, $(z, x)$ in order to reconstruct the sensitive input. Instead, the attacker uses weights $\theta_1$ of the client side network. The attacker randomly initializes a network $\hat{f}(\hat{\theta}; \cdot)$ such that it 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 activations is minimized by optimizing for the weights $\hat{\theta}$. This attack scheme is inspired by deep image prior [43] for feature inversion. One drawback with this attack is that it is only applicable to the sensitive input

| Attack name | Time of attack | Mode of training | Mode of prediction | Input for attacker | Target of attack |
|---|---|---|---|---|---|
| Feature space hijacking attack | Training | Distributed | Distributed/On-premise | Intermediate activations | Training/Test Data |
| Federated/Client-side attack | Training | P2P/Distributed | Distributed/On-premise | Client weights | Training/Test Data |
| Attribute attack | Train/Test | Distributed | Distributed/On-Premise | Intermediate activations/weights | Specific attributes |
| Decoder and Likelihood attacks | Test | On-premise | Distributed | Intermediate activations/weights | Test Data |

(a) We categorize several forms of reconstruction attacks within the context of distributed machine learning. The last row shows the attacks that are relevant to the setting of private collaborative inference considered in this paper.

| Method | Sensitive Input | Sensitive Attribute | No client arch. alteration | Adversary Free |
|---|---|---|---|---|
| Osia et al [35] | ✗ | ✓ | ✗ | ✓ |
| Min-max filters [13] | ✗ | ✓ | ✓ | ✗ |
| DeepObfuscator [23] | ✓ | ✓ | ✓ | ✗ |
| Shredder [29] | ✗ | ✓ | ✗ | ✓ |
| Mitigating information [37] | ✗ | ✓ | ✓ | ✗ |
| Kernelized ARL [38] | ✗ | ✓ | ✓ | ✗ |
| PrivacyNet [31] | ✗ | ✓ | ✗ | ✗ |
| IdentityDP [47] | ✗ | ✓ | ✗ | ✗ |
| **NoPeek-Infer (Ours)** | ✓ | ✗ | ✓ | ✓ |

(b) Different defense mechanisms for private inference. The third column *no alteration of client architecture* refers to techniques where additional operations or layers are not required for removing sensitive information from data. The last column *adversary free* refers to techniques which require a proxy adversary during training. *Sensitive input* refers to protection of entire raw data and *sensitive attribute* refers to techniques that protect only a given subset of attributes.

TABLE I

RECONSTRUCTION ATTACKS AND DEFENCES STUDIED WITHIN THE CONTEXT OF SPLIT LEARNING AND ITS VARIANTS.

protection and not sensitive attribute. This attack setting is stronger and also harder to defend against because it does not require access to the $(z, x)$ pairs.

### B. Defenses

Defenses that are relevant to our work are categorized in Table I. We categorize them based on a.) the type of sensitive data under consideration and b.) whether additional privatizing operations and/or an additional adversarial model is required. Our proposed method of NoPeek-Infer is the only method to the best of our knowledge that does not have either of the requirements stated in b.). We also note that NoPeek-Infer focuses on preventing reconstruction of input data, as opposed to specific attributes that has been the focus of the majority of the defense schemes.

1) **Noisy perturbations:** Differential privacy [10] is a popular notion of privacy for various queries to prevent membership inference attacks. In the context of model training it is implemented via noisy perturbations of gradient updates as in [1], [46], [48], [2], [5], [47]. Our proposed mechanism of NoPeek Infer is instead for the setting of private collaborative inference rather than training. In the context of split learning, [35] and [29] learn informal noisy perturbations to prevent reconstruction attacks but require altering the architecture of the client network that is being privatized. These works are also specific to preventing reconstruction of a target attribute as opposed to the input dataset itself. Typically, adding noise to the activations leads to a costly trade-off of privacy versus accuracy.

2) **Siamese defense:** In this defense, a contrastive loss is used to nudge points from same class label to be closer to each other in a learnt representation space. This loss is used in combination with an accuracy loss for prediction purposes. Such siamese embeddings have been used in various works outside the realm of privacy prior to being introduced by [34] solely for privacy purposes within the distributed setting involving intermediate activation sharing.

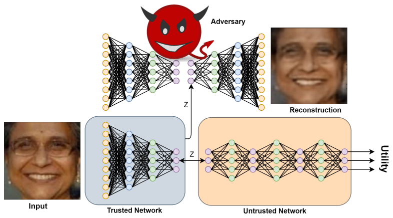

Fig. 1. **Face reconstruction attack** The attack is possible when activations are shared for distributed predictive inference if a proper defense is not in place. Information about sensitive raw input data can get leaked through intermediate activations even after input data passes through multiple layers. Upon sending these intermediate activations from a trusted network on a client to an untrusted network for computing rest of the task, an adversary on server-side can reconstruct original raw face data from the activations.

3) **Adversarial defenses:** [23], [37], [38], [24], [31], [39] attempt to learn activations of a given network at chosen layers while attempting to protect against an adversary that attempts to reconstruct raw data or partial attributes of raw data from these activations. These methods require an adversarial deep network to be trained in addition to the original deep network that is used for prediction. This is in contrast to our method which does not require any other additional network, which sharply reduces the number of parameters to be trained in our case.

## IV. METHOD

**Key idea:** The key idea of our proposed method is to reduce information leakage by adding an additional loss term to the commonly used classification loss term, categorical cross-entropy. The information leakage reduction loss term we use

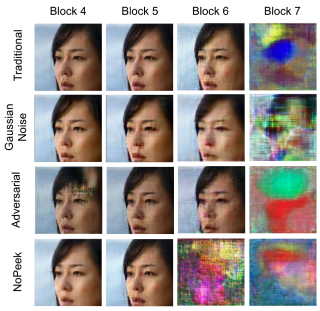

Fig. 2. **Reconstruction results on CelebA:** We apply the likelihood maximization attack on activations obtained from different blocks of ResNet-18 [14] for different mechanisms. For brevity we only show block 4-7 since blocks before 4 get full reconstruction and blocks after 8 do not obtain a reasonable reconstruction.

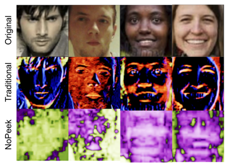

Fig. 3. Visualization of the activations of the first layer of a ResNet. In the activation maps in the second row, subtle facial features can be observed from the activations about the raw image while, in the third row, the NoPeek-Infer-Infer method forces the network to decorrelate the features with respect to raw data, hence making it hard to interpret.

is distance correlation [41]; a powerful measure of non-linear (and linear) statistical dependence between random variables. The distance correlation loss is minimized between raw input data and the output of any chosen layer whose outputs need to be communicated from the client to another untrusted client or untrusted server. This setting is crucial to some popular forms of distributed machine learning that require sharing of activations from an intermediate layer. This has been motivated under the 'activation sharing' subsection in the motivation section.

Optimization of this combination of losses helps ensure the activations resulting from the protected layer have minimal information for reconstructing raw data while still being useful enough to achieve reasonable classification accuracies upon post-processing. The quality of preventing reconstruction of raw input data while maintaining reasonable classification accuracies is qualitatively and quantitatively substantiated in the experiments section. The joint minimization of distance correlation with cross entropy leads to a specialized feature extraction or transformation such that it is imperceptible in leaking information about the raw dataset with respect to both the human visual system and more sophisticated reconstruction attacks as we show later in the experiments section.

**Loss function:** The total loss function using $n$ samples of input data $\mathbf{X}$, activations from protected layer $\mathbf{Z}$, true labels $\mathbf{Y}_{true}$, predicted labels $\mathbf{Y}$, and scalar weight $\alpha$ is given along with distance correlation being *DCOR* and categorical cross entropy being *CCE* as

$$\alpha DCOR(\mathbf{X},\mathbf{Z}) + (1-\alpha)CCE(\mathbf{Y}_{true},\mathbf{Y}) \qquad (1)$$

The following subsections introduce the definition of distance correlation while the gradient of distance correlation is provided for optimization purposes in Appendix although we optimize our loss using *Autograd*, thereby not requiring this gradient in an explicit manner. That said, useful deep learning friendly code for computing distance correlation is also provided in Appendix for reproducibility.

**Sample Distance Correlation [41]**: We first give some required notation for defining sample distance correlation which is a statistical estimator for population distance correlation. We denote i.i.d samples of data as $\mathcal{X} \times \mathcal{Y} = \{(\mathbf{x}_k,\mathbf{y}_k)|k=1,2,3,\dots,n\}$ and corresponding double centered Euclidean distance matrices as $\widehat{\mathbf{E}}_\mathbf{X}$ and $\widehat{\mathbf{E}}_\mathbf{Y}$ obtained by multiplying each of the corresponding Euclidean distance matrix from both sides with a centering matrix given by $\mathbf{I} - \frac{1}{n}ee^T$ where $I$ is the identity matrix and $e$ is a vector all 1's. Now the squared sample distance covariance is defined as,

$$\hat{\nu}^2(\mathbf{X},\mathbf{Y}) = \frac{1}{n^2}\sum_{k,l=1}^{n}[\widehat{\mathbf{E}}_\mathbf{X}]_{k,l}[\widehat{\mathbf{E}}_\mathbf{Y}]_{k,l}, \qquad (2)$$

and using this the sample distance correlation is given by

$$\hat{\rho}^2(\mathbf{X},\mathbf{Y}) = \begin{cases} \frac{\hat{\nu}^2(\mathbf{X},\mathbf{Y})}{\sqrt{\hat{\nu}^2(\mathbf{X},\mathbf{X})\hat{\nu}^2(\mathbf{Y},\mathbf{Y})}}, & \hat{\nu}^2(\mathbf{X},\mathbf{X})\hat{\nu}^2(\mathbf{Y},\mathbf{Y}) > 0. \\ 0, & \hat{\nu}^2(\mathbf{X},\mathbf{X})\hat{\nu}^2(\mathbf{Y},\mathbf{Y}) = 0. \end{cases}$$

This sample distance correlation is a statistical estimator for the following population distance correlation which is defined below.

**Distance Covariance [41]:** Distance covariance between random variables $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^m$ with finite first moments is a nonnegative number given by

$$\nu^2(\mathbf{x},\mathbf{y}) = \int_{\mathbb{R}^{d+m}} |f_{\mathbf{x},\mathbf{y}}(t,s) - f_\mathbf{x}(t)f_\mathbf{y}(s)|^2 w(t,s)dtds$$

where $f_\mathbf{x}, f_\mathbf{y}$ are characteristic functions of $\mathbf{x},\mathbf{y}$, $f_{\mathbf{x},\mathbf{y}}$ is the joint characteristic function, and $w(t,s)$ is a weight function defined as

$$w(t,s) = (C(p,\alpha)C(q,\alpha)|t|_p^{\alpha+p}|s|_q^{\alpha+q})^{-1}$$

with

$$C(d,\alpha) = \frac{2\pi^{d/2}\Gamma(1-\alpha/2)}{\alpha 2^\alpha \Gamma((\alpha+d)/2)}$$

for chosen values of $\alpha$ which refers to the choice of norm considered in obtaining the distance matrices. $\Gamma$ refers to the popular complete Gamma function, that is defined to be an extension of the concept of a factorial to complex and real numbers as opposed to just the integers. Note that for random variables that admit a density, the characteristic function is the Fourier transform of the probability density function. From above definition of distance covariance, we have the following expression for the square of distance correlation between random variables $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^m$ with finite first moments and is a non-negative number defined as

$$\rho^2(\mathbf{x},\mathbf{y}) = \begin{cases} \frac{\nu^2(\mathbf{x},\mathbf{y})}{\sqrt{\nu^2(\mathbf{x},\mathbf{x})\nu^2(\mathbf{y},\mathbf{y})}}, & \nu^2(\mathbf{x},\mathbf{x})\nu^2(\mathbf{y},\mathbf{y}) > 0. \\ 0, & \nu^2(\mathbf{x},\mathbf{x})\nu^2(\mathbf{y},\mathbf{y}) = 0. \end{cases}$$

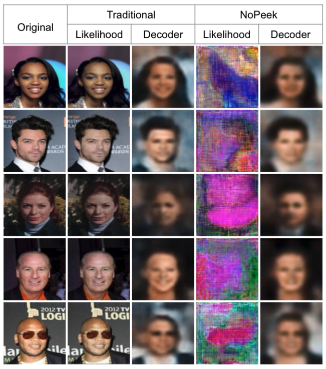

Fig. 4. **Likelihood vs. Decoder Reconstruction Attacks:** A qualitative comparison between likelihood and supervised decoder reconstruction attacks on traditional and NoPeek methods. While likelihood attack performs a visually similar reconstruction for the traditional approach, the decoder attack gets a better reconstruction result for NoPeek. However, in the case of NoPeek the attack results in a blurred and average face image across certain set of facial attributes. The purpose of this result is to illustrate the relative benefit of using different types of adversaries when evaluating NoPeek and other baselines.

This always lies within the interval $[0, 1]$ with $0$ referring to independence and $1$ referring to dependence.

### A. Advantages of using distance correlation

Estimation of classical information theoretic-measures as used in [30], [50], [49] is a known hard problem. Recent approaches to estimate it effectively like [3] are based on iterative optimization. A recent data efficient version of it requires 3 nested for loops of optimization [25]. In the context of deep learning, every epoch of learning the weights is dependent on this iterative optimization. In contrast our approach uses distance correlation. Fast estimators of distance correlation requires $\mathcal{O}(nlogn)$ [7], [16] computational complexity for univariate and $\mathcal{O}(nKlogn)$ complexity [15] for multivariate settings with $\mathcal{O}(\max(n, K))$ memory, where $K$ is the number of random projections required as part of the estimation. Distance correlation has been shown to be a simpler special case of other recent popular measures of dependence such as Hilbert-Schmidt Independence Criterion (HSIC), Maximum Mean Discrepancy (MMD) and Kernelized Mutual Information (KMI) that have been extensively studied and used in the machine learning and statistics community [40], [42]An advantage of using a simpler alternative is that in addition to being differentiable and easily computable with a closed-form, it requires no other tuning of parameters and is self-contained unlike HSIC, MMD and KMI that depend on a choice of separate kernels for features as well as labels along with their respective tuning parameters.

## V. Experiments

**Reconstruction attack testbed:** We empirically examine the privacy aspects of our method by designing a testbed that performs feature inversion [8] under different threat models for PCI. The goal of the testbed is to emulate attackers in order to examine information leakage both qualitatively and quantitatively. We use the attack testbed for both supervised decoder and likelihood maximization attacks as described in the attacks part of section III.

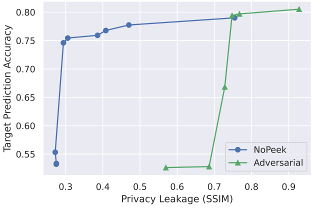

Fig. 5. **Privacy-Utility Trade-off:** We vary the value of $\alpha$ to display the relationship between privacy leakage and task utility. Leakage is measured as the SSIM score between input and reconstructed images from the likelihood attack scheme.

The decoder attack architecture consists of upsampling layers composed of transpose convolutions. Similar architectures have been used in generative models for generating images from low-dimensional latent codes. Under the threat model for decoder attack, the attacker has access to a dataset consisting of multiple samples of $(z_l, x)$. Input to the testbed is the intermediate activations, $z_l$ from any arbitrary layer $l$ of the target model, and output is the generated image $\hat{x}$. After the training of the defense component (NoPeek or baselines), we use a held-out validation set to generate intermediate activations using the client network of defense component. We thereby generate a paired dataset of activations and corresponding images. We use this paired dataset to train the reconstruction testbed to emulate the attacker. We use 90% of the original validation dataset for training the reconstruction testbed and the remaining 10% as the test-set for qualitative evaluation of reconstruction quality. The training is a standard supervised decoder training on a dataset of $z_l, x$ pairs with a loss function of the euclidean norm between $x$ and $\hat{x}$. We want to emphasize that there may potentially be a better design for architectures of the reconstruction testbed and better loss functions, but the goal of this paper is just to have a fair comparison between NoPeek-Infer based training and regular training of deep networks using a reasonable reconstruction architecture. The number of upsampling layers in the architecture of the testbed vary depending upon the difference in the dimensionality of $z_l$ and $x$. Next, we evaluate the performance on likelihood maximization attack. The threat model for likelihood maximization attack requires the adversary to have access to client network weights and the architecture. The details of the likelihood maximization attack inspired by the work on deep image priors [43] is described in section III. This attack has not

tc

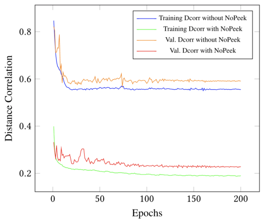

Fig. 6. We plot distance correlation during training and testing as the network gets trained on UTK faces with and without NoPeek-Infer.

been used in the privacy community looking at the feature inversion problem but used for several vision tasks like super-resolution, denoising and feature inversion.

### A. Datasets

*1) CelebA:* CelebA [27] is a large scale celebrity face dataset with 202,599 face images that are well aligned and centered. These faces span 10,177 identities each of which is associated with 40 different binary attributes.

*2) Fairface:* Fairface [20] is a dataset of 108,501 face images with three attributes – gender, race, and ethnicity. The images are centered but contain different poses and lighting. We evaluate our approach using gender as the target attribute for both datasets.

**Baselines:** Our experiments consists of four categories of activation sharing methods - traditional (no defense), adversarial defense, siamese embedding defense and noise based defense as detailed in section III-B. Traditional refers to the setup where activations are shared by the client to the server without any specific defense. Adversarial refers to the set of techniques [23], [4] that jointly trains a proxy adversary resulting in a min-max optimization between the adversary and client network. Siamese embedding based privacy is via a combiination of a contrastive loss and an accuracy loss as detailed in [34]. Noise is the category of baseline where we add Gaussian noise to the intermediate activations. While not related to activation sharing, many differentially private mechanisms add similar noise calibrated to sensitivity [9], [10]. Even though we do not calibrate the noise, we try a broad range of noise spanning across the highest and lowest attainable utility. In all of our experiments, we train a standard ResNet-18 [14] for minimizing the loss on the main task. In all of our reported experiments, we use Adam optimizer with an initial learning rate of $1 \times e^{-3}$ and exponential decay for training. In the first experiment, we study the role of intermediate layer $l$ by evaluating privacy and utility across different blocks of ResNet-18 for different methods. Figure 2 shows the qualitative results for different approaches. For the first five blocks, all techniques fail to defend against the likelihood attack. However, NoPeek-Infer provides adequate protection at the block-6. In order to prevent any selection bias for the qualitative result, we also show reconstruction for six random samples from the dataset in Figure 8. We compared the baselines and NoPeek-Infer on different metrics of image reconstruction quality and predictive utility of the model as shown in Table II. We compared defenses of NoPeek-Infer & various baselines on reconstruction of sensitive input with respect to likelihood maximization attack & observe that the defense of NoPeek-Infer performs the best by achieving a worst reconstruction when attacked which indicates that NoPeek-Infer is a better method for preventing reconstruction attacks. In terms of the broader trend we observe that NoPeek-Infer fared the best followed by DeepObfuscator and then followed by Siamese Embedding, PAN and Noise (& Blur) approaches in preventing the reconstruction attack in terms of SSIM score, PSNR and $l_1$ metrics. We also compare against a primitive baseline that is based on reduction of linear correlation as opposed to our proposed approach of nonlinear correlation minimization to show that the distance correlation (or nonlinear correlation) based approach is substantially better. While this comes at the cost of a small drop in accuracy, we note that the improvement in privacy is much higher than the corresponding reduction in utility. To further examine the privacy-utility trade-off, we vary the trade-off parameter for both adversarial and NoPeek-Infer and plot different points along the privacy-utility trade-off in Figure 5. As we reach towards higher privacy, the utility performance drops faster for adversarial in comparison to NoPeek-Infer. This makes NoPeek-Infer amenable for high privacy regimes without any significant loss in the utility. It is an accepted standard that the privacy-utility trade-offs exist in privacy preserving machine learning; and thereby the above tradeoff observed in NoPeek-Infer is competitive.

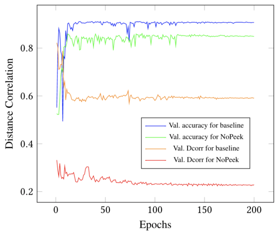

Fig. 7. By introducing NoPeek-Infer in the training of the network, we obtain a major decrease in the distance correlation from 0.6 (baseline) to 0.22 (NoPeek-Infer) while the decrease in the accuracies is relatively much lesser.

## VI. Discussion

For comparing with noise baseline, we add gaussian noise to every component of the $z_l$ vector with varying standard deviation $\sigma$ of the noise for different experiments. We empirically observe that even for $\sigma = 400$ the reconstruction happens successfully using the likelihood attack while the utility gets close to chance accuracy. This illustrates that gaussian noise mechanism is approximately same as the *traditional* category due to its inability to provide any privacy-utility trade-off despite adjusting $\sigma = 0$.

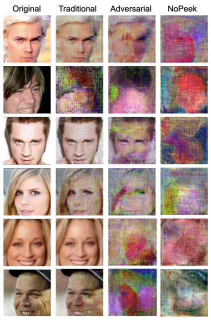

Fig. 8. **Reconstruction across different samples:** For illustrating reconstruction results, we plot images randomly sampled from CelebA's test set under 'Original'. Reconstruction was performed using likelihood attack on sixth block of ResNet-18. Results showed good reconstruction for traditional (no defense) while *adversarial* obtains certain degree of protection as very few facial attributes can be inferred. In comparison, NoPeek does not leak any facial attribute.

To show the trade-off between privacy and utility via choice of $\alpha$ we plot the distance correlation of a given intermediate activation during training a NoPeek-Infer network and a traditional network without NoPeek-Infer in Figure 6. This demonstrates that the network without NoPeek-Infer

| Dataset | Method | SSIM ↓ | PSNR ↓ | $\ell_1$ ↑ | Utility ↑ |
|---|---|---|---|---|---|
| Fairface | Traditional [12] | 0.915 ± 0.110 | 72.982 ± 6.682 | 0.066 ± 0.051 | **0.9912** |
| | PAN [26] | 0.777 ± 0.218 | 69.585 ± 7.403 | 0.097 ± 0.069 | 0.9864 |
| | NoPeek-Infer (Ours) | **0.306** ± **0.141** | **60.453** ± **2.813** | **0.206** ± **0.057** | 0.9803 |
| | Blur [33] | 0.893 ± 0.884 | 61.2864 ± 2.5906 | 0.1066 ± 0.045 | 0.9881 |
| | Gaussian Noise | 0.842 ± 0.233 | 70.235 ± 2.672 | 0.0771 ± 0.045 | 0.8857 |
| | Laplacian Noise | 0.733 ± 0.1495 | 69.488 ± 5.539 | 0.0701 ± 0.0858 | 0.8568 |
| | DeepObfuscator [23] | 0.4467 ± 0.107 | 61.19 ± 3.935 | 0.191 ± 0.0894 | 0.9811 |
| | Siamese Embedding [34] | 0.484 ± 0.117 | 61.712 ± 1.169 | 0.198 ± 0.066 | 0.9511 |
| | Linear Correlation | 0.585 ± 0.02 | 67.789 ± 3.283 | 0.0625 ± 0.01 | 0.9115 |
| CelebA | Traditional [12] | 0.563 ± 0.237 | 65.655 ± 4.968 | 0.123 ± 0.067 | **0.9759** |
| | PAN [26] | 0.646 ± 0.168 | 64.650 ± 4.485 | 0.121 ± 0.056 | 0.9513 |
| | NoPeek-Infer (Ours) | **0.239** ± **0.081** | 58.901 ± 1.835 | **0.240** ± **0.053** | 0.9488 |
| | Blur [33] | 0.524 ± 0.168 | 60.248 ± 5.15 | 0.1373 ± 0.0669 | 0.9452 |
| | Gaussian Noise | 0.656 ± 0.187 | 63.584 ± 2.896 | 0.1348 ± 0.0352 | 0.9608 |
| | Laplacian Noise | 0.6276 ± 0.168 | 61.868 ± 5.011 | 0.1487 ± 0.0572 | 0.966 |
| | DeepObfuscator [23] | 0.2874 ± 0.0436 | **56.3463** ± **1.479** | 0.2189 ± 0.032 | 0.9531 |
| | Siamese Embedding [34] | 0.539 ± 0.249 | 59.243 ± 3.5206 | 0.185 ± 0.085 | 0.9376 |
| | Linear Correlation | 0.4154 ± 0.0913 | 60.342 ± 4.17 | 0.203 ± 0.0745 | 0.944 |

TABLE II

**COMPARISON FOR SENSITIVE INPUT LEAKAGE:** WE COMPARE DEFENSES OF NOPEEK-INFER & BASELINES ON RECONSTRUCTION OF SENSITIVE INPUT WITH RESPECT TO LIKELIHOOD MAXIMIZATION ATTACK & OBSERVE THAT THE DEFENSE OF NOPEEK-INFER PERFORMS THE BEST BY ACHIEVING A WORST RECONSTRUCTION WHEN ATTACKED.

naturally reduces distance correlation during training and our proposed method can be seen as an additional regularizer which forces the network to regularize for the reduction in distance correlation at a much higher rate between raw data and activations. The consistency between training and testing distance correlation in Figure 6 also demonstrates the capability of weights learnt by NoPeek-Infer in generalizing the decorrelation phenomenon to prevent reconstruction attacks.

The first row of Figure 3 shows some raw input images and the output of the first layer of the trained network when NoPeek-Infer is not used is shown in the second row. The third row shows the output at the first layer in the case when NoPeek-Infer is used. We restrict it to only three output channels to visualize only the RGB component as part of an qualitative investigation. As seen, the second row visually leaks a lot of information about the raw image in comparison to the third row. This demonstrates semantically meaningful obfuscation performed by the layers of the client network when trained with NoPeek-Infer. In Figure 7 we observe that the accuracy dropped by a relatively small amount compared to the drop in distance correlation (or leakage of sensitive information) and this relative difference can be controlled by tuning $\alpha$. The important aspect to note from the figure is that distance correlation between the samples and activations can be reduced significantly without any significant drop in accuracy.

## VII. CONCLUSION

The proposed NoPeek-Infer schemes based on distance correlation seem to have versatile applicability in the space of privacy, computer vision and machine learning given that it does not require major changes in the model setup and architectures except for the proposed modification to the loss function. It would be great to realize on-device implementations of the NoPeek-Infer scheme. With regards to human visual perception of bias and privacy, we would also like to conduct a large-scale crowdsourced survey to compare performance of human participants in deciphering the true sensitive image upon looking at NoPeek-Infer results in comparison to a uniform random choice.

## APPENDIX

Distance correlation between centered data can be represented as $\frac{Tr(\mathbf{X^TXZ^TZ})}{\sqrt{Tr(\mathbf{X^TX})^2 Tr(\mathbf{Z^TZ})^2}}$ in a graph-theoretic dual space [45]. Distance covariance in the numerator can be written as $Tr(\mathbf{X^TZX}) = \sum_{ij}\langle z_i, z_j\rangle(\|x_i - x_j\|)^2$. This can be written in matrix form using basis vectors $e_i, e_j$ as

$$\sum_{ij}[Tr(\mathbf{Z^T e_i e_j^T Z})\mathbf{Tr}(\mathbf{X^T(e_i - e_j)(e_i - e_j)^T X})] \quad (3)$$

Simplifying the notation with $M_{ij} = e_i e_j^T$ and $A_{ij} = (e_i - e_j)(e_i - e_j)^T$ we have $\frac{\partial Tr(\mathbf{Z^T L_Z Z})}{\partial \mathbf{Z}} = \sum_{ij}(\mathbf{2M_{ij}Z})\mathbf{Tr}(\mathbf{X^T A_{ij} X})$. On the lines of 3, we have $Tr(\mathbf{Z^T L_Z Z}) = \sum_{ij}[Tr(\mathbf{Z^T M_{ij} Z})\mathbf{Tr}(\mathbf{Z^T A_{ij} Z})]$. Therefore utilizing these identities, the derivative of squared distance correlation w.r.t $\mathbf{Z}$ can be written as $\frac{c_x Tr(\mathbf{Z^T L_Z Z})\frac{\partial Tr(\mathbf{X^T L_Z X})}{\partial \mathbf{Z}} - [Tr(\mathbf{X^T L_Z X})]^2 c_x \frac{\partial Tr(\mathbf{Z^T L_Z Z})}{\partial \mathbf{Z}}}{[Tr(\mathbf{Z^T L_Z Z})]^2}$ upto a constant.

### A. Deep-learning friendly source code for sample distance correlation

```
def pairwise_dist(A):
    r = tf.reduce_sum(A*A, 1)
    r = tf.reshape(r, [-1, 1])
    D = tf.maximum(r - 2*tf.matmul(A, tf.transpose(A)) + tf.transpose(r), 1e-7)
    D = tf.sqrt(D)
    return D

def dist_corr(X, Y):
    n = tf.cast(tf.shape(X)[0], tf.float32)
    a = pairwise_dist(X)
    b = pairwise_dist(Y)
    A = a - tf.reduce_mean(a, axis=1) -\
    tf.expand_dims(tf.reduce_mean(a,axis=0),axis=1)+\
    tf.reduce_mean(a)
    B = b - tf.reduce_mean(b, axis=1) -\
    tf.expand_dims(tf.reduce_mean(b,axis=0),axis=1)+\
    tf.reduce_mean(b)
    dCovXY = tf.sqrt(tf.reduce_sum(A*B) / (n ** 2))
    dVarXX = tf.sqrt(tf.reduce_sum(A*A) / (n ** 2))
    dVarYY = tf.sqrt(tf.reduce_sum(B*B) / (n ** 2))

    dCorXY = dCovXY / tf.sqrt(dVarXX * dVarYY)
    return dCorXY
```

## REFERENCES

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pages 308–318, 2016.

[2] N. Agarwal, A. T. Suresh, F. Yu, S. Kumar, and H. B. Mcmahan. cpsgd: Communication-efficient and differentially-private distributed sgd. arXiv preprint arXiv:1805.10559, 2018.

[3] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm. Mine: mutual information neural estimation. arXiv preprint arXiv:1801.04062, 2018.

[4] M. Bertran, N. Martinez, A. Papadaki, Q. Qiu, M. Rodrigues, G. Reeves, and G. Sapiro. Adversarially learned representations for information obfuscation and inference. In International Conference on Machine Learning, pages 614–623. PMLR, 2019.

[5] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers. Protection against reconstruction and its applications in private federated learning. arXiv preprint arXiv:1812.00984, 2018.

[6] B. Bortolato, M. Ivanovska, P. Rot, J. Križaj, P. Terhörst, N. Damer, P. Peer, and V. Štruc. Learning privacy-enhancing face representations through feature disentanglement. In 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020), pages 495–502. IEEE, 2020.

[7] A. Chaudhuri and W. Hu. A fast algorithm for computing distance correlation. Computational Statistics & Data Analysis, 2019.

[8] A. Dosovitskiy and T. Brox. Inverting visual representations with convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4829–4837, 2016.

[9] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In Theory of cryptography conference, pages 265–284. Springer, 2006.

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

[11] A. T. Elthakeb, P. Pilligundla, F. Mireshghallah, A. Cloninger, and H. Esmaeilzadeh. Divide and conquer: Leveraging intermediate feature representations for quantized training of neural networks. In International Conference on Machine Learning, pages 2880–2891. PMLR, 2020.

[12] O. Gupta and R. Raskar. Distributed learning of deep neural network over multiple agents. Journal of Network and Computer Applications, 116:1–8, 2018.

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

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. CoRR, abs/1512.03385, 2015.

[15] C. Huang and X. Huo. A statistically and numerically efficient independence test based on random projections and distance covariance. arXiv preprint arXiv:1701.06054, 2017.

[16] X. Huo and G. J. Székely. Fast computing for distance covariance. Technometrics, 58(4):435–447, 2016.

[17] Z. Huo, B. Gu, and H. Huang. Training neural networks using features replay. arXiv preprint arXiv:1807.04511, 2018.

[18] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al. Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977, 2019.

[19] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492, 2016.

[20] K. Kärkkäinen and J. Joo. Fairface: Face attribute dataset for balanced race, gender, and age, 2019.

[21] M. Laskin, L. Metz, S. Nabarrao, M. Saroufim, B. Noune, C. Luschi, J. Sohl-Dickstein, and P. Abbeel. Parallel training of deep networks with local updates. arXiv preprint arXiv:2012.03837, 2020.

[22] A. Li, Y. Duan, H. Yang, Y. Chen, and J. Yang. Tiprdc: task-independent privacy-respecting data crowdsourcing framework for deep learning with anonymized intermediate representations. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 824–832, 2020.

[23] A. Li, J. Guo, H. Yang, and Y. Chen. Deepobfuscator: Adversarial training framework for privacy-preserving image classification. arXiv preprint arXiv:1909.04126, 2019.

[24] M. Lin, R. Ji, Y. Wang, Y. Zhang, B. Zhang, Y. Tian, and L. Shao. Hrank: Filter pruning using high-rank feature map. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1529–1538, 2020.

[25] X. Lin, I. Sur, S. A. Nastase, A. Divakaran, U. Hasson, and M. R. Amer. Data-efficient mutual information neural estimator. arXiv preprint arXiv:1905.03319, 2019.

[26] S. Liu, J. Du, A. Shrivastava, and L. Zhong. Privacy adversarial network: representation learning for mobile data privacy. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 3(4):1–18, 2019.

[27] Z. Liu, P. Luo, X. Wang, and X. Tang. Large-scale celebfaces attributes (celeba) dataset. Retrieved August, 15:2018, 2018.

[28] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, et al. Communication-efficient learning of deep networks from decentralized data. arXiv preprint arXiv:1602.05629, 2016.

[29] F. Mireshghallah, M. Taram, P. Ramrakhyani, D. Tullsen, and H. Esmaeilzadeh. Shredder: Learning noise distributions to protect inference privacy. 2019.

[30] V. Mirjalili, S. Raschka, and A. Ross. Flowsan: Privacy-enhancing semi-adversarial networks to confound arbitrary face-based gender classifiers. arXiv preprint arXiv:1905.01388, 2019.

[31] V. Mirjalili, S. Raschka, and A. Ross. Privacynet: semi-adversarial networks for multi-attribute face privacy. IEEE Transactions on Image Processing, 29:9400–9412, 2020.

[32] C. Neustaedter, S. Greenberg, and M. Boyle. Blur filtration fails to preserve privacy for home-based video conferencing. ACM Transactions on Computer-Human Interaction (TOCHI), 13(1):1–36, 2006.

[33] S. J. Oh, R. Benenson, M. Fritz, and B. Schiele. Faceless person recognition: Privacy implications in social media. In European Conference on Computer Vision, pages 19–35. Springer, 2016.

[34] S. A. Osia, A. S. Shamsabadi, S. Sajadmanesh, A. Taheri, K. Katevas, H. R. Rabiee, N. D. Lane, and H. Haddadi. A hybrid deep learning architecture for privacy-preserving mobile analytics. IEEE Internet of Things Journal, 7(5):4505–4518, 2020.

[35] S. A. Osia, A. Taheri, A. S. Shamsabadi, K. Katevas, H. Haddadi, and H. R. Rabiee. Deep private-feature extraction, 2018.

[36] D. Pasquini, G. Ateniese, and M. Bernaschi. Unleashing the tiger: Inference attacks on split learning. arXiv preprint arXiv:2012.02670, 2020.

[37] P. C. Roy and V. N. 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.

[38] B. Sadeghi, R. Yu, and V. Boddeti. On the global optima of kernelized adversarial representation learning. In Proceedings of the IEEE International Conference on Computer Vision, pages 7971–7979, 2019.

[39] M. Samragh, H. Hosseini, A. Triastcyn, K. Azarian, J. Soriaga, and F. Koushanfar. Unsupervised information obfuscation for split inference of neural networks. arXiv preprint arXiv:2104.11413, 2021.

[40] D. Sejdinovic, B. Sriperumbudur, A. Gretton, K. Fukumizu, et al. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. The Annals of Statistics, 41(5):2263–2291, 2013.

[41] G. J. Székely, M. L. Rizzo, N. K. Bakirov, et al. Measuring and testing dependence by correlation of distances. The annals of statistics, 35(6):2769–2794, 2007.

[42] C. J. Tonde. Supervised feature learning via dependency maximization. PhD thesis, Rutgers University-Graduate School-New Brunswick, 2016.

[43] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Deep image prior. CoRR, abs/1711.10925, 2017.

[44] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. arXiv preprint arXiv:1812.00564, 2018.

[45] P. Vepakomma, C. Tonde, A. Elgammal, et al. Supervised dimensionality reduction via distance correlation maximization. Electronic Journal of Statistics, 12(1):960–984, 2018.

[46] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor. Federated learning with differential privacy: Algorithms and performance analysis. IEEE Transactions on Information Forensics and Security, 15:3454–3469, 2020.

[47] Y. Wen, L. Song, B. Liu, M. Ding, and R. Xie. Identitydp: Differential private identification protection for face images. arXiv preprint arXiv:2103.01745, 2021.

[48] Y. Wu, F. Yang, and H. Ling. Privacy-protective-gan for face de-identification. arXiv preprint arXiv:1806.08906, 2018.

[49] Z. Wu, Z. Wang, Z. Wang, and H. Jin. Towards privacy-preserving visual recognition via adversarial training: A pilot study. In Proceedings of the European Conference on Computer Vision (ECCV), pages 606–624, 2018.

[50] B. H. Zhang, B. Lemoine, and M. Mitchell. Mitigating unwanted biases with adversarial learning. In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, pages 335–340. ACM, 2018.