Reimagining Mutual Information for Defense against Data Leakage in Collaborative Inference

Lin Duan[1]*, **Jingwei Sun**[1]*, **Jinyuan Jia**[2], **Yiran Chen**[1], **Maria Gorlatova**[1]
[1] Department of Electrical and Computer Engineering, Duke University
[2] College of Information Sciences and Technology, Pennsylvania State University
[1] {lin.duan, jingwei.sun, yiran.chen, maria.gorlatova}@duke.edu
[2] jinyuan@psu.edu

Abstract

Edge-cloud collaborative inference empowers resource-limited IoT devices to support deep learning applications without disclosing their raw data to the cloud server, thus protecting user's data. Nevertheless, prior research has shown that collaborative inference still results in the exposure of input and predictions from edge devices. To defend against such data leakage in collaborative inference, we introduce InfoScissors, a defense strategy designed to reduce the mutual information between a model's intermediate outcomes and the device's input and predictions. We evaluate our defense on several datasets in the context of diverse attacks. Besides the empirical comparison, we provide a theoretical analysis of the inadequacies of recent defense strategies that also utilize mutual information, particularly focusing on those based on the Variational Information Bottleneck (VIB) approach. We illustrate the superiority of our method and offer a theoretical analysis of it.

1 Introduction

Edge devices are becoming smarter and more versatile. These devices are expected to efficiently perform a wide range of deep learning (DL) inference tasks with remarkable performance. However, implementing DL inference applications on such edge devices is challenging due to the constraints imposed by the on-device resource availability. As we see the rise of state-of-the-art (SOTA) DL models, such as Large Language Models [1, 2], they are becoming increasingly complex, housing a colossal number of parameters. This escalation in complexity and size makes it difficult to store a DL model on an edge device, which typically has limited memory space. Furthermore, the restricted computational resources could lead to prolonged latency during inference. One potential solution to this predicament is to transmit the input data directly from the edge device to a cloud server. The server, which houses the DL model, then conducts inference and sends the prediction back to the device. However, this approach carries a risk of data leakage, particularly if the input data are sensitive in nature - such as facial images. In addition, the output data (i.e., predictions) can also contain confidential information, such as the patient's diagnostic results.

Collaborative inference [3–9] has emerged as an approach to prevent data leakage when deploying DL inference applications on commodity edge devices with constrained computing resources. Fig. 1 shows a general collaborative inference system. Suppose an edge device and a cloud server conduct collaborative inference. The deep learning model can be divided into three parts[2]. The first and last few layers of the network are deployed on the edge device, while the remaining layers are offloaded

[*] Co-first authors.

[2] Note that some applications might divide the model into two parts, and the edge devices might hold the first or the last few layers, which have different data leakage problems. This paper considers the general setting.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

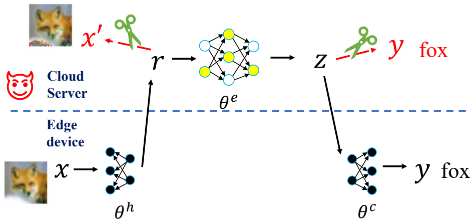


Figure 1: A general framework of collaborative inference. The malicious server can infer input and predictions on the edge device. Our method defends against data leakage by reducing the mutual information between the model's intermediate outcomes and the edge device's data and predictions.

to the cloud server. This division allows most of the computational tasks to be handled by the server, effectively mitigating the resource limitations on the device. The edge device and the cloud server communicate only the intermediate outputs of the model, ensuring that the raw input and predictions remain inaccessible to the server. However, recent works [10, 11] have revealed that sharing these intermediate outputs can still lead to data leakage from edge devices, including input data and predictions. A malicious server can, for instance, reconstruct input data from the representations (i.e., r in Fig. 1) uploaded by the device through Model Inversion (MI) attacks [12, 13, 11]. Furthermore, the high-level features (i.e., z in Fig. 1) contain rich information about the predictions, making it feasible for a malicious server to infer the device's predictions through these features [14–16]. While there have been considerable explorations into data protection in collaborative inference [10, 11, 17, 18], existing defenses tend to significantly degrade model utility. This degradation is particularly evident in scenarios where attacks are relatively strong. For example, when the head model on the device (i.e., θ^h in Fig. 1) is shallow, existing defenses [10, 11, 19, 20, 20, 18] cannot guarantee robustness against MI attacks without a significant drop in model accuracy as shown in our results.

We propose InfoScissors, a defense method designed from a mutual information perspective to protect the edge device's data in collaborative inference. This approach works by protecting both the device's input data and its predictions. The goal of our method is to preserve user privacy in collaborative inference, and privacy preservation is achieved by manipulating the training phase (i.e., collaborative training). By applying our defense, the model is normalized to filter the private information when extracting features and representations, such that privacy is preserved during the inference phase. To protect the input data, we regularize the head model on the device to extract representations that contain less mutual information with the input. To protect the prediction, we regularize the features extracted by the server's encoder to minimize the mutual information they contain with the label. We derive a variational mutual information upper bound and develop an adversarial training method to minimize this bound on the device side. There are works [21, 17, 18] that prevent input leakage from the mutual information perspective, and most of them are based on *Variational Information Bottleneck (VIB) [22].* Our work is not a simple replacement of the mutual information approximation compared with these works. We analyze the inadequacies of the VIB-based methods in protecting input data in the context of collaborative inference and illustrate the superiority of our method. We evaluate our method on CIFAR10 and CIFAR100 against input leakage using both black-box and white-box MI attacks. The results show that our method can effectively defend the attacks with less than a 3% drop in model accuracy even when the head model on the device has only one convolutional layer, where the attacks are extremely strong. We also evaluate our defense against prediction leakage using multiple Model Completion (MC) attacks [14, 15]. The results show that our defense achieves the best trade-off between the model accuracy and the defense effectiveness compared to the baselines. Our contributions are summarized as follows:

- We propose InfoScissors, a defense method against data leakage in collaborative inference from the mutual information perspective, encompassing both input leakage and prediction leakage.
- We offer a theoretical analysis of our defense against input recovery attacks and prediction inference attacks. We also analyze the superiority of our method compared with the VIB-based methods.
- We empirically evaluate InfoScissors across multiple datasets and against multiple attacks. Our method effectively defends against MI and MC attacks, outperforming the baselines.

2 Related Work

Data Leakage in Collaborative Inference Data leakage is drawing more and more attention as the rapid growth of commercial deployment of DL, especially in collaborative learning scenarios, whose primary concern is data safety. In collaborative inference, we categorize data leakage into two types, i.e., input leakage [23, 10, 24, 25] and prediction leakage [14–16]. For input leakage, [23] proposes general attack methods for complex models, such as Neural Networks, by matching the correlation between adversary features and target features, which can be seen as a variant of model inversion [26, 27]. [10, 28, 25, 29, 30, 24] also propose variants of model inversion attack. While all these attacks are in the inference phase, [25] proposes a variant of DLG [12], which can perform attacks in the training phase. For prediction leakage, [15] proposes an attack and defense method for two-party split learning on binary classification problems, a special collaborative inference setting. Additionally, [14] proposes three different label inference attack methods considering different settings in collaborative inference: direct label inference attack, passive label inference attack, and active label inference attack.

Defense in Collaborative Inference Defensive methods have been proposed against data leakage in collaborative inference. To defend against input leakage, some works apply differential privacy (DP) [10, 11, 19, 31] and compression [10, 11, 17, 32] to the representations and models. While these methods can successfully defend against input leakage from the representations, they cause substantial model performance degradation because they weaken the knowledge/information in the representations. Some recent works also try to prevent input leakage by regularizing the representations from the mutual information perspective [17, 18, 21]. However, their methods only achieve decent results when the head model on the edge device is deep, which is not practical when the computation power is constrained on the edge device. Our paper analyzes that such a disadvantage comes from the inadequacies of *VIB* in the context of input protection. Some other works [33–35] apply mutual information on input space to protect input data, but their methods are only feasible with limited input dimension in the context of collaborative inference due to computational constrain on edge devices. One recent work [36] studies inference defense under a similar setting to our paper. But they only focus on prediction protection. They add noise to the training label by randomly sampling a class label, which is intuitively inspired by DP. Our training method is theoretically derived from the perspective of mutual information, and we provide a theoretical analysis of our defense performance. To defend against prediction leakage, [16] manipulates the labels following specific rules to defend the direct label inference attack, which can be seen as a variant of label differential privacy (label DP) [37, 38] in collaborative inference. Compression and quantization of the gradients [14, 18] are also applied to prevent prediction leakage. However, similar to the defense against input leakage, these defenses cause substantial model performance degradation to achieve decent defense performance.

3 Preliminary

3.1 Collaborative Inference Setting

Suppose an edge device and a cloud server conduct collaborative inference. Following the setting in Fig. 1, the deep learning model is divided into a head model $f^h_{\theta^h}$, an encoder $f^e_{\theta^e}$ and a classifier $f^c_{\theta^c}$. The head model and classifier are deployed on the edge device, and the encoder is on the cloud server. Given an input x_i, the edge device first calculates the representation $r_i = f^h_{\theta^h}(x_i)$ and sends r_i to the server. Then the server extracts the feature from the received representation $z_i = f^e_{\theta^e}(r_i)$ and sends z_i back to the edge device. After receiving the feature, the edge device calculates the prediction $\hat{y}_i = f^c_{\theta^c}(z_i)$. In this paper, the results of $f^h_{\theta^h}$ sent from the device to the server are referred to as *representations*, and*features* refer to the results of $f^e_{\theta^e}$ sent from the server to the device. The overall inference procedure is formulated as:

$$\hat{y}_i = f^c_{\theta^c}(f^e_{\theta^e}(f^h_{\theta^h}(x_i))). \tag{1}$$

In the real world, the input x_i and prediction $\hat{y}_i$ are important intellectual properties of the edge device and may contain personal information. In the inference procedure, the edge device does not send raw input to the server, and the inference results are also inaccessible to the server.

3.2 Threat Model

Our goal is to protect the edge device's input and predictions from being inferred by the cloud server. The device only uploads the representations to the server and never leaks raw input or predictions to the server. However, the cloud server is untrusted, attempting to steal input and predictions. We assume the untrusted server strictly follows the collaborative inference protocols, and it cannot compromise the inference process conducted by the device. Nevertheless, the adversary (i.e., malicious server) is capable of training a surrogate classifier and generator to mimic the victim's data. With the received representation r_i, the server can reconstruct the input x_i on the device by conducting MI attacks [12, 13, 10]. Notably, the head model on the device is usually shallow due to the computation resource limitation, which aggravates the input leakage from the representation [11]. The encoder on the server extracts high-level features containing rich information about the prediction, which enables the server to infer predictions of the device. We conduct preliminary experiments to illustrate the data leakage in collaborative inference, which can be found in Appendix A.

4 Method

4.1 Defense Formulation

To defend against data leakage, we propose InfoScissors, a learning algorithm that regularizes the model during the training phase. Following the setup of 3.1, suppose the edge device has sample pairs $\{(x_i,y_i)\}_{i=1}^N$ drawn from a distribution $p(\mathrm{x},\mathrm{y})$. The representation is calculated as $\mathrm{r} = f^h_{\theta^h}(\mathrm{x})$ by the edge device, and the cloud server computes features $\mathrm{z} = f^e_{\theta^e}(\mathrm{r})$. We apply x,y,r,z here to represent random variables, while x_i,y_i,r_i,z_i are deterministic values. To defend against the leakage of the edge device's input data and inference results, InfoScissors is designed to achieve three goals:

- Goal 1: To preserve the performance of collaborative inference, the main objective loss should be minimized.
- Goal 2: To prevent the input leakage from the representations, θ^h should not extract representations r containing much information about the input data x.
- Goal 3: To reduce the leakage of the predictions on the edge device, θ^e on the cloud server should not be able to extract features z containing much information about the true label y.

Formally, we have three training objectives:

$$\begin{aligned}&\textbf{Prediction:} \min_{\theta^h,\theta^e,\theta^c} \mathcal{L}\big(f^c_{\theta^c}\big(f^e_{\theta^e}\big(f^h_{\theta^h}(\mathrm{x})\big)\big),\mathrm{y}\big),\\ &\textbf{Input protection:} \min_{\theta^h} \mathrm{I}(\mathrm{r};\mathrm{x}),\\ &\textbf{Prediction protection:} \min_{\theta^h,\theta^e} \mathrm{I}(\mathrm{z};\mathrm{y}),\end{aligned} \tag{2}$$

where I(r;x) is the mutual information between the representation and the input, which indicates how much information r retains about the input data x. Similarly, I(z;y) is the mutual information between the feature and the label. We minimize these mutual information terms to prevent the cloud server from inferring the input x and label y from r and z, respectively.

The prediction objective is usually easy to optimize (e.g., cross-entropy loss for classification). However, the mutual information terms are hard to calculate in practice for two reasons: 1. r and x are high-dimensional, and it is extremely computationally heavy to compute their joint distribution; 2. Calculating the mutual information requires knowing the distributions p(x|r) and p(y|z), which are both difficult to compute. We do not follow existing works [18, 21, 17] to employ VIB to derive tractable estimations of the mutual information objectives. We analyze the inadequacies of VIB in protecting input, which can be found in Sec. 4.4, and leverage CLUB [39] to formulate variational upper bounds of mutual information terms. We first formulate a variational upper bound of I(r;x):

$$\mathrm{I}(\mathrm{r};\mathrm{x}) \le \mathrm{I}_{\mathrm{vCLUB}}(\mathrm{r};\mathrm{x}) := \mathbb{E}_{p(\mathrm{r},\mathrm{x})} \log q_\psi(\mathrm{x}|\mathrm{r}) - \mathbb{E}_{p(\mathrm{r})p(\mathrm{x})} \log q_\psi(\mathrm{x}|\mathrm{r}), \tag{3}$$

where $q_\psi(\mathrm{x}|\mathrm{r})$ is a variational distribution with parameters ψ to approximate $p(\mathrm{x}|\mathrm{r})$. To guarantee the inequality of Eq. (3), $q_\psi(\mathrm{x}|\mathrm{r})$ should satisfy:

$$\mathrm{KL}(p(\mathrm{r},\mathrm{x})||q_\psi(\mathrm{r},\mathrm{x})) \le \mathrm{KL}(p(\mathrm{r})p(\mathrm{x})||q_\psi(\mathrm{r},\mathrm{x})), \tag{4}$$

which can be achieved by minimizing $\mathrm{KL}(p(\mathrm{r},\mathrm{x})||q_\psi(\mathrm{r},\mathrm{x}))$:

$$\psi = \operatorname*{argmin}_\psi \mathrm{KL}(p(\mathrm{r},\mathrm{x})||q_\psi(\mathrm{r},\mathrm{x})) = \operatorname*{argmax}_\psi \mathbb{E}_{p(\mathrm{r},\mathrm{x})} \log(q_\psi(\mathrm{x}|\mathrm{r})). \tag{5}$$

With sample pairs $\{(x_i,y_i)\}_{i=1}^N$, we apply the sampled vCLUB (vCLUB-S) mutual information estimator in [39] to reduce the computational overhead, which is an unbiased estimator of $\mathrm{I}_{\mathrm{vCLUB}}$ and is formulated as:

$$\hat{\mathrm{I}}_{\text{vCLUB-S}}(\mathrm{r;x}) = \frac{1}{N}\sum_{i=1}^{N}\left[\log q_\psi(x_i|r_i) - \log q_\psi\left(x_{k_i'}|r_i\right)\right], \tag{6}$$

where k_i' is uniformly sampled from indices $\{1,...,N\}$. With Eq. (3), Eq. (5) and Eq. (6), the objective of input protection is formulated as:

$$\min_{\theta^h}\mathrm{I}(\mathrm{r;x}) \Leftrightarrow \min_{\theta^h}\hat{\mathrm{I}}_{\text{vCLUB-S}}(\mathrm{r;x}) = \min_{\theta^h}\frac{1}{N}\sum_{i=1}^{N}\left[\max_{\psi}\log q_\psi(x_i|r_i) - \log q_\psi\left(x_{k_i'}|r_i\right)\right]. \tag{7}$$

Similarly, we can use a variational distribution $q_\phi(y|z)$ with parameter ϕ to approximate $p(y|z)$, and formulate the objective of label protection as:

$$\min_{\theta^h,\theta^e}\mathrm{I}(\mathrm{z;y}) \Leftrightarrow \min_{\theta^h,\theta^e}\hat{\mathrm{I}}_{\text{vCLUB-S}}(\mathrm{z;y}) = \min_{\theta^h,\theta^e}\frac{1}{N}\sum_{i=1}^{N}\left[\max_{\phi}\log q_\phi(y_i|z_i) - \log q_\phi\left(y_{n_i'}|z_i\right)\right]. \tag{8}$$

Suppose we use g_ψ, h_ϕ to parameterize q_ψ and q_ϕ, respectively. By combining Eq. (7), Eq. (8) and the prediction objective with weight hyper-parameters λ_d and λ_l, the overall optimizing objective is:

$$\begin{aligned}
&\min_{\theta^h,\theta^e,\theta^c}(1-\lambda_d-\lambda_l)\underbrace{\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}\left(f_{\theta^c}^c\left(f_{\theta^e}^e\left(f_{\theta^h}^h(x_i)\right)\right),y_i\right)}_{\mathcal{L}_c}\\
&+\min_{\theta^h}\max_{\psi}\lambda_d\underbrace{\frac{1}{N}\sum_{i=1}^{N}\log g_\psi\left(x_i|f_{\theta^h}^h(x_i)\right)}_{\mathcal{L}_{d\_a}}+\min_{\theta^h}\lambda_d\underbrace{\frac{1}{N}\sum_{i=1}^{N}-\log g_\psi\left(x_{k_i'}|f_{\theta^h}^h(x_i)\right)}_{\mathcal{L}_{d\_r}}\\
&+\min_{\theta^h,\theta^e}\max_{\phi}\lambda_l\underbrace{\frac{1}{N}\sum_{i=1}^{N}\log h_\phi\left(y_i|f_{\theta^e}^e\left(f_{\theta^h}^h(x_i)\right)\right)}_{\mathcal{L}_{l\_a}}+\min_{\theta^h,\theta^e}\lambda_l\underbrace{\frac{1}{N}\sum_{i=1}^{N}-\log h_\phi\left(y_{n_i'}|f_{\theta^e}^e\left(f_{\theta^h}^h(x_i)\right)\right)}_{\mathcal{L}_{l\_r}}.
\end{aligned} \tag{9}$$

h_ϕ can be easily constructed to estimate $p\,(\mathrm{y|z})$ given the task of inference (e.g., classifier for classification task). To estimate $p(\mathrm{x|r})$, we assume that x follows the Gaussian distribution of which the mean vector is determined by r and the variance is 1. Under this assumption, we apply a generator g_ψ to estimate the mean vector of x given r.

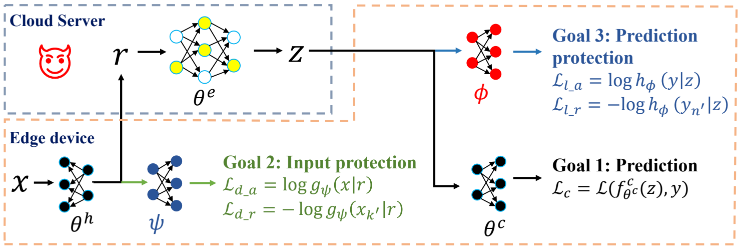

Cloud Server
r
θ^e
z
ϕ
Goal 3: Prediction protection
$\mathcal{L}_{l\_a} = \log h_\phi\,(y|z)$
$\mathcal{L}_{l\_r} = -\log h_\phi\,(y_{n'}|z)$
Edge device
x
θ^h
ψ
Goal 2: Input protection
$\mathcal{L}_{d\_a} = \log g_\psi(x|r)$
$\mathcal{L}_{d\_r} = -\log g_\psi(x_{k'}|r)$
θ^c
Goal 1: Prediction
$\mathcal{L}_c = \mathcal{L}(f_{\theta^c}^c(z), y)$

Figure 2: An overview of InfoScissors. Training step 1: Optimize the classifiers θ^c and ϕ by minimizing $\mathcal{L}_c$ and maximizing $\mathcal{L}_{l\_a}$, respectively. Step 2: Optimize the generator ψ by maximizing $\mathcal{L}_{d\_a}$. Step 3: Optimize θ^h and θ^e by minimizing $(1-\lambda_d-\lambda_l)\mathcal{L}_c+\lambda_l\mathcal{L}_{l\_a}+\lambda_l\mathcal{L}_{l\_r}+\lambda_d\mathcal{L}_{d\_a}+\lambda_d\mathcal{L}_{d\_r}$.

4.2 Learning Algorithm

The overall objective has five terms. For simplicity, we denote these five objective terms as $\mathcal{L}_c$, $\mathcal{L}_{d\_a}$, $\mathcal{L}_{d\_r}$, $\mathcal{L}_{l\_a}$ and $\mathcal{L}_{l\_r}$, respectively, as shown in Eq. (9). $\mathcal{L}_c$ is the prediction objective. $\mathcal{L}_{d\_a}$ and $\mathcal{L}_{d\_r}$ comprise the input data protection objective. $\mathcal{L}_{d\_a}$ is an adversarial training objective where an auxiliary generator g_ψ is trained to capture input information while the head layers $f_{\theta^h}^h$ are trained to extract as little input information as possible. $\mathcal{L}_{d\_r}$ regularizes $f_{\theta^h}^h$ to extract representations that can be used to generate randomly picked samples. $\mathcal{L}_{l\_a}$ and $\mathcal{L}_{l\_r}$ have similar effect with $\mathcal{L}_{d\_a}$ and

$\mathcal{L}_{d\_r}$, respectively. We can reorganize the overall training objective as:

$$
\begin{aligned}
&\theta^h,\theta^e,\theta^c,\psi,\phi \\
&= \arg\min_{\theta^h,\theta^e}[(1-\lambda_d-\lambda_l)\min_{\theta^c}\mathcal{L}_c+\lambda_l\max_{\phi}\mathcal{L}_{l\_a}+\lambda_l\mathcal{L}_{l\_r}+\lambda_d\max_{\psi}\mathcal{L}_{d\_a}+\lambda_d\mathcal{L}_{d\_r}].
\end{aligned}
\tag{10}
$$

Based on Eq. (10), we develop a collaborative learning algorithm. For each batch of data, the device first optimizes the classifiers θ^c and ϕ by minimizing $\mathcal{L}_c$ and maximizing $\mathcal{L}_{l\_a}$, respectively. Then, the device optimizes the generator ψ by maximizing $\mathcal{L}_{d\_a}$. Finally, θ^h and θ^e are optimized by minimizing $(1-\lambda_d-\lambda_l)\mathcal{L}_c+\lambda_l\mathcal{L}_{l\_a}+\lambda_l\mathcal{L}_{l\_r}+\lambda_d\mathcal{L}_{d\_a}+\lambda_d\mathcal{L}_{d\_r}$. The detailed algorithm can be found in Appendix B. Note that θ^h,θ^c,ψ, and ϕ are deployed on devices, and their training does not need additional information from the cloud server compared with training without our defense. The training procedure of θ^e does not change, which makes our defense concealed from the cloud server.

4.3 Theoretical Analysis

We provide a theoretical analysis of our defenses against prediction and input leakage. Following the notations in Sec. 4.1, we have the following theorem of defense performance for prediction leakage after applying InfoScissors. All the proofs can be found in Appendix C.

Theorem 1. *Let h_ϕ parameterize q_ϕ in Eq.* (8)*. Suppose the malicious server optimizes an auxiliary model $h^m(\mathrm{y}|\mathrm{z})$ to estimate $p(\mathrm{y}|\mathrm{z})$. For any $h^m(\mathrm{y}|\mathrm{z})$, we always have:*

$$
\frac{1}{N}\sum_{i=1}^{N}\log h^m(y_i|z_i)<\frac{1}{N}\sum_{i=1}^{N}\log p(y_i)+\epsilon,
\tag{11}
$$

where

$$
\epsilon=I_{vCLUB_{h_\phi}}(\mathrm{z};\mathrm{y})+KL(p(\mathrm{y}|\mathrm{z})||h_\phi(\mathrm{y}|\mathrm{z})).
\tag{12}
$$

Specifically, if the task of collaborative inference is classification, we have the following corollary:

Corollary 1. *Suppose the task of collaborative inference is classification. Following the notations in Theorem 1 and let $epsilon$ be defined therein, we have:*

$$
\frac{1}{N}\sum_{i=1}^{N}CE[h^m(z_i),y_i]>CE_{random}-\epsilon,
\tag{13}
$$

where CE denotes the cross-entropy loss, and CE_{random} is the cross-entropy loss of random guessing.

For input leakage, we have the following theorem.

Theorem 2. *Let the assumption of $p(\mathrm{x}|\mathrm{r})$ in Sec. 4.1 hold and g_ψ parameterize the mean of q_ψ in Eq.* (7)*. Q denotes the dimension of* x*. Suppose the malicious server optimizes an auxiliary model $g^m(\mathrm{x}|\mathrm{r})$ to estimate the mean of $p(\mathrm{x}|\mathrm{r})$. For any $g^m(\mathrm{x}|\mathrm{r})$, we always have:*

$$
\frac{1}{N}\sum_{i=1}^{N}MSE[g^m(r_i),x_i]>\frac{2(\kappa-\epsilon)}{Q},
\tag{14}
$$

where MSE denotes the ***mean square error****, and*

$$
\kappa=-\frac{1}{N}\sum_{i=1}^{N}\log\frac{\sqrt{2\pi}}{p(x_i)},\ \epsilon=I_{vCLUB_{g_\psi}}(\mathrm{r};\mathrm{x})+KL(p(\mathrm{x}|\mathrm{r})||g_\psi(\mathrm{x}|\mathrm{r})).
\tag{15}
$$

4.4 Superiority over VIB

Recent works also try to prevent input leakage by regularizing the representations from the mutual information perspective [21, 17, 18]. The ultimate goal of these works is the same as our method, which is to minimize $I(r;x)$. However, most of these works apply *Variational Information Bottleneck (VIB) [22]* to derive the upper bound of $I(\mathrm{r};\mathrm{x})$. Even though VIB is commonly applied in DNN feature regularization, we analyze that it is not optimal in defending against input leakage in collaborative inference.

The variational upper bound of $I(\mathrm{r};\mathrm{x})$, which is also an objective to minimize, derived through VIB is formulated as

$$I(\mathrm{r};\mathrm{x}) \leq \mathbb{E}_{\epsilon\sim p(\epsilon),\mathrm{x}\sim p(\mathrm{x})}\mathrm{KL}[p_{\theta^h}(\mathrm{r}|\mathrm{x},\epsilon), r(\mathrm{r})], \tag{16}$$

where ϵ is the Gaussian random variable used to reparameterize the randomness of representation extractor θ^h. $r(\mathrm{r})$ is the variational approximation of the marginal distribution $p(\mathrm{r})$. In practice, without any prior information, $r(\mathrm{r})$ is set to be a fixed spherical Gaussian, $r(\mathrm{r}) = \mathcal{N}(\mathrm{r}|0, I)$. To minimize $I(\mathrm{r};\mathrm{x})$, the VIB-based works [21, 17, 18] regularize the representation distribution to be close to a fixed Gaussian. This will cause the **complementary loss of information** contained by r, including the mutual information with y, which is crucial for the primary inference task. In contrast, the training objective derived in our method is to **only filter out the information in** r **that is crucial for reconstructing** x, which will cause less information loss and inference performance drop.

We can also analyze the difference between VIB-based methods and our method from a high level, which also motivates us to choose CLUB approximation. The mutual information $I(\mathrm{r};\mathrm{x})$ can be expanded as

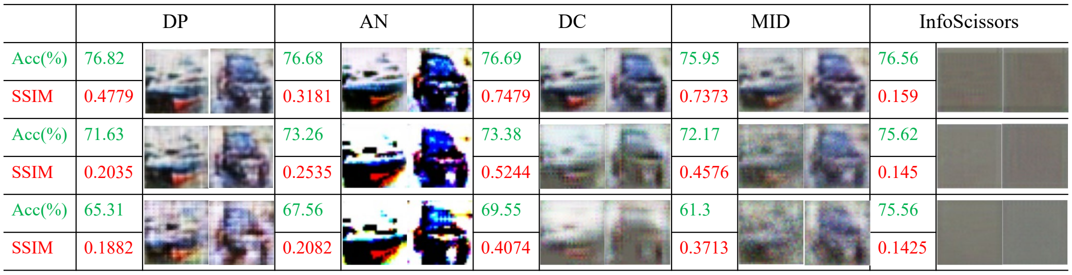
$$\begin{aligned} I(\mathrm{r};\mathrm{x}) &= \int dr dx p(r,z) \log \frac{p(r|x)}{p(r)} \\ &= \int dr dx p(r,z) \log \frac{p(x|r)}{p(x)}. \end{aligned} \tag{17}$$

VIB follows the first equality to derive a tractable upper bound of $I(\mathrm{r};\mathrm{x})$ based on a parameterized variational $\hat{p}(r|x)$. Subsequently, the training objective is focused on minimizing this variational $\hat{p}(r|x)$. We follow the second equality to parameterize $p(x|r)$ and derive an upper bound based on CLUB. Our training goal is centered on minimizing the variational $\hat{p}(x|r)$. Importantly, in the context of defending against model inversion attacks, the primary aim is to attain a low $p(x|r)$. This aim is more closely aligned with our training objective compared to VIB-based methods. This alignment, alongside the empirical results presented in the subsequent section, highlights the superiority of our method.

5 Experiments

We first evaluate our method against input leakage and prediction leakage, both separately and in an integrated manner. The experiments are conducted on a server with 4 RTX TITAN GPUs.

5.1 Experimental Setup



	DP	AN	DC	MID	InfoScissors
Acc(%)	76.82	76.68	76.69	75.95	76.56
SSIM	0.4779	0.3181	0.7479	0.7373	0.159
Acc(%)	71.63	73.26	73.38	72.17	75.62
SSIM	0.2035	0.2535	0.5244	0.4576	0.145
Acc(%)	65.31	67.56	69.55	61.3	75.56
SSIM	0.1882	0.2082	0.4074	0.3713	0.1425

Figure 3: Images reconstructed by the KA attack on CIFAR10 under different defenses. Each row represents a different defense level. The bottom row applies the strongest defense, resulting in lower-quality reconstructed images and a sacrifice in accuracy.

Attack methods For input leakage, we evaluate InfoScissors against two Model Inversion (MI) attacks: (1) **Knowledge Alignment (KA)** [17] is a black-box MI attack, in which the malicious server trains an inversion model that swaps the input and output of the target model using an auxiliary dataset. The inversion model is then used to reconstruct the input given any representation. (2) **Regularized Maximum Likelihood Estimation (rMLE)** [11] is a white-box MI attack that the malicious server has access to the device's extractor model θ^h. The server trains input to minimize the distance between the fake representations and the received ground-truth representations. It is an unrealistic assumption that the server can access the model on the device, and we apply this white-box attack to evaluate our defense against extremely strong attacks. For prediction leakage, we evaluate our defense against two Model Completion (MC) attacks: (1) **Passive Model Completion (PMC)** [14] attack assumes that the malicious server has access to an auxiliary labeled dataset and utilizes this auxiliary dataset to fine-tune a classifier that can be applied to its encoder. (2) **Active Model Completion (AMC)** [14] attack is included as an *adaptive attack* against our defense. The primary goal of our defense is to reduce the collaborative model's reliance on the server-side encoder for specific tasks, allowing data and prediction information to be filtered during inference. Under the adaptive attack setting, the adversary is allowed to modify the training profiling such that it can trick the collaborative model into relying more on its encoder, thereby extracting more private data information from the encoder's features. In this setting, the adversary is directly confronting the fundamental principles of our defense method, constituting a highly potent form of adaptive attack.

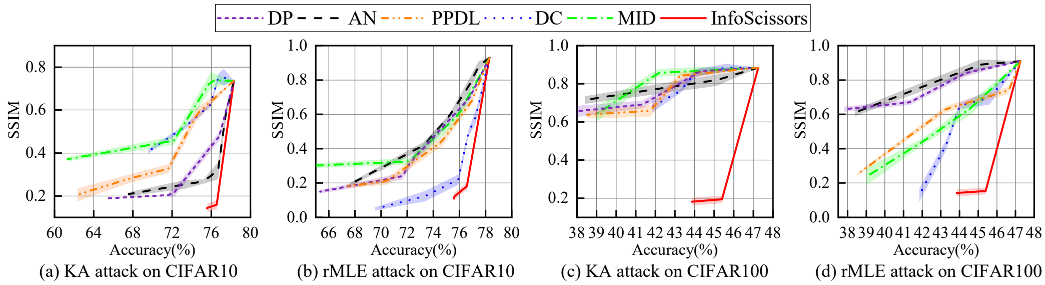
Baselines We compare InfoScissors with five existing defense baselines: (1) **Differential Privacy (DP)** [10, 11, 19], (2) **Adding Noise (AN)** [40], (3) **Data Compression (DC)** [20], (4) **Privacy-preserving Deep Learning (PPDL)** [41], and (5) **Mutual Information Regularization Defense (MID)** [18], which is the SOTA defense against data leakage in collaborative inference based on *Variational Information Bottleneck (VIB)*. The details of the baselines can be found in Appendix D.

Dataset & Hyperparameter configurations We evaluate on CIFAR10 and CIFAR100. For both datasets, we apply ResNet18 as the backbone model. The first convolutional layer and the last basic block are deployed on the device as the representation extractor and the classifier, respectively. We set batch size B as 32 for both datasets. We apply SGD as the optimizer with the learning rate η set to be 0.01. The server has 40 and 400 labeled samples to conduct KA and MC attacks for CIFAR10 and CIFAR100, respectively. For InfoScissors, we apply a 1-layer decoder and a 3-layer MLP to parameterize ψ and ϕ. For AN, we apply Laplacian noise with a mean of zero and a scale between 0.0001-0.01. For DC, we set the compression rate from 90% to 100%. For PPDL, we set the Laplacian noise with a scale of 0.0001-0.01, $\tau = 0.001$ and θ between 0 and 0.01. For MID, we set the weight of mutual information regularization between 0-0.1.

Evaluation metrics (1) **Utility metric (Model accuracy)**: We use the test data accuracy of the classifier on the device to measure the performance of the collaborative model. (2) **Robustness metric (SSIM)**: We use SSIM (structural similarity) between the reconstructed images and the raw images to evaluate the effectiveness of the defense against input leakage. The lower the SSIM, the better the defense performance. (3) **Robustness metric (Attack accuracy)**: We use the test accuracy of the server's classifier after conducting MC attacks to evaluate the defense against prediction leakage. The lower the attack accuracy, the higher the robustness against prediction leakage.

5.2 Results of Input Protection

We conduct experiments on CIFAR10 and CIFAR100 to evaluate our defense against the KA attack and the rMLE attack. We set different defense levels for our methods (i.e., different λ_d values in Eq. (9)) and baselines to conduct multiple experiments to show the trade-off between the model accuracy and SSIM of reconstruction. The results are shown in Fig. 4.



(a) KA attack on CIFAR10 (b) rMLE attack on CIFAR10 (c) KA attack on CIFAR100 (d) rMLE attack on CIFAR100

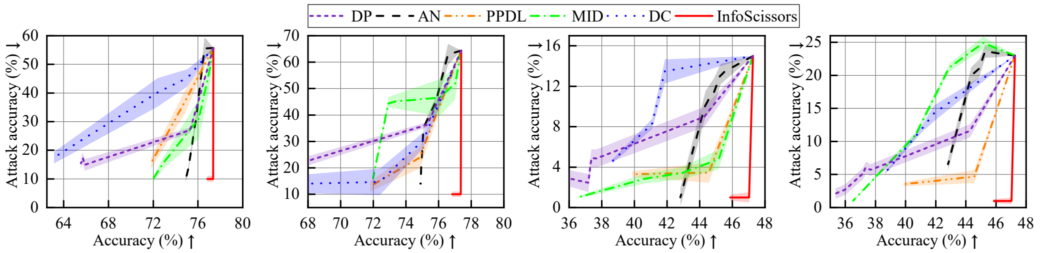
Figure 4: Model accuracy v.s. SSIM on CIFAR10 and CIFAR100 against MI attacks.

For defense against KA attack, our InfoScissors can reduce the SSIM of reconstruction to lower than 0.2 with a model accuracy drop of less than 2% for CIFAR10. In contrast, the other baselines reduce model accuracy by more than 10% and cannot achieve the same defense effect even with an accuracy drop of more than 10%. Notably, the malicious server has more auxiliary data on CIFAR100 than CIFAR10, making the defense harder on CIFAR100. However, InfoScissors can still achieve an SSIM of lower than 0.2 with a model accuracy drop of less than 2%. We also evaluate our defense against the KA attack with a larger auxiliary dataset on the malicious server, and the results, which can be found in Appendix D, show that our defense can effectively defend against the KA attack when the server has more auxiliary samples. For defense against rMLE attacks, InfoScissors achieves similar results of reducing the SSIM to lower than 0.2 with a model accuracy drop of less than 2% for CIFAR10 and 1% for CIFAR100, respectively, which outperforms the other baselines significantly.

To perceptually demonstrate the effectiveness of our defense, we show the reconstructed images by the KA attack on CIFAR10 after applying baseline defenses and our defense in Fig. 3. It is shown that by applying the baseline defenses, the reconstructed images still contain enough information to be recognizable with the model accuracy of lower than 70%. For our method, the reconstructed images do not contain much information about the raw images, with the model accuracy higher than 76%.

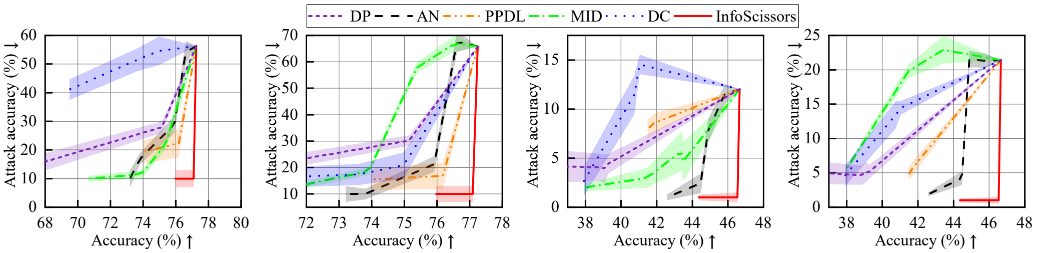
5.3 Results of Prediction Protection

We evaluate InfoScissors on two datasets against the PMC attack and the AMC attack. We set different defense levels for our methods (i.e., different λ_l values in Eq. (9)) and baselines to conduct multiple experiments to show the trade-off between the model accuracy and attack accuracy. The defense results against PMC and AMC attacks are shown in Fig. 5 and Fig. 6, respectively. To simulate the realistic settings where the malicious server uses different model architectures to conduct MC attacks, we apply different model architectures (MLP & MLP_sim) for MC attacks. The detailed model architectures can be found in Appendix D.



(a) PMC-CIFAR10-MLP (b) PMC-CIFAR10-MLP_sim (c) PMC-CIFAR100-MLP (d) PMC-CIFAR100-MLP_sim

Figure 5: Model accuracy v.s. attack accuracy on CIFAR10 and CIFAR100 against PMC attack.



(a) AMC-CIFAR10-MLP (b) AMC-CIFAR10-MLP_sim (c) AMC-CIFAR100-MLP (d) AMC-CIFAR100-MLP_sim

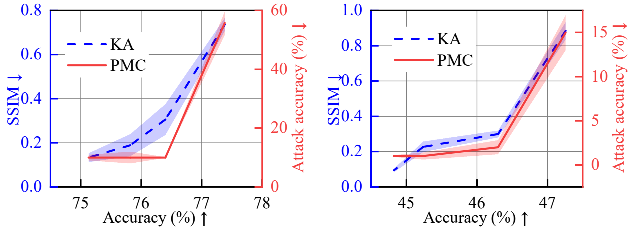
Figure 6: Model accuracy v.s. attack accuracy on CIFAR10 and CIFAR100 against AMC attack.

For defense against PMC on CIFAR10, InfoScissors achieves 10% attack accuracy (equal to random guess) by sacrificing less than 0.5% model accuracy, while the other baselines suffer a model accuracy drop by more than 4% to achieve the same defense effect. Similarly, InfoScissors achieves 1% attack accuracy on CIFAR100 by sacrificing less than 1% model accuracy, while the other baselines achieve the same defense effect by sacrificing more than 6% model accuracy.

InfoScissors also shows robustness against AMC attack. InfoScissors achieves attack accuracy of the rate of random guess by sacrificing less than 1% and 0.5% model accuracy on CIFAR10 and CIFAR100, respectively. The other baselines achieve the same defense performance by sacrificing more than 5% and 4% model accuracy, respectively.

5.4 Integration of Input and Prediction Protection

We evaluate the integration of input and prediction protection of InfoScissors. We set λ_d and λ_l between 0.05-0.4 and evaluate the defenses. The results of defense against the KA and PMC attacks on CIFAR10 and CIFAR100 are shown in Fig. 7. It is shown that InfoScissors can effectively protect input data and predictions simultaneously with less than a 2% accuracy drop for both datasets.



(a) CIFAR10

(b) CIFAR100

Figure 7: InfoScissors against KA and PMC.

6 Conclusion and Limitation

We propose a defense method (InfoScissors) to defend against data leakage in collaborative inference by reducing the mutual information between the model's intermediate outcomes and the device's input data and predictions. The experimental results show that our method can defend against input leakage and prediction leakage effectively. Our work can make the public aware of the risk of privacy leakage posed by collaborative inference, which is considered a positive societal impact. One limitation of this paper is that we only focus on the scenario where there is only one edge device, even though our defense can be easily applied to the collaborative inference scenario with multiple edge devices.

Acknowledgments and Disclosure of Funding

This material is based upon work supported by the U.S. National Science Foundation under award CNS-2112562 and NAIAD Award 2332744. This work is also supported by ARO W911NF-23-2-0224. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. National Science Foundation, ARO, and their contractors.

References

[1] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, *et al.*, "Emergent abilities of large language models," *arXiv preprint arXiv:2206.07682*, 2022.

[2] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier, *et al.*, "Chatgpt for good? on opportunities and challenges of large language models for education," *Learning and individual differences*, vol. 103, p. 102274, 2023.

[3] G. Li, L. Liu, X. Wang, X. Dong, P. Zhao, and X. Feng, "Auto-tuning neural network quantization framework for collaborative inference between the cloud and edge," in *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*, pp. 402–411, Springer, 2018.

[4] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," *ACM SIGARCH Computer Architecture News*, vol. 45, no. 1, pp. 615–629, 2017.

[5] A. E. Eshratifar, M. S. Abrishami, and M. Pedram, "Jointdnn: An efficient training and inference engine for intelligent mobile cloud computing services," *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 565–576, 2019.

[6] A. Banitalebi-Dehkordi, N. Vedula, J. Pei, F. Xia, L. Wang, and Y. Zhang, "Auto-split: A general framework of collaborative edge-cloud ai," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2543–2553, 2021.

[7] M. Li, Y. Li, Y. Tian, L. Jiang, and Q. Xu, "Appealnet: An efficient and highly-accurate edge/cloud collaborative architecture for dnn inference," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pp. 409–414, IEEE, 2021.

[8] N. Shlezinger, E. Farhan, H. Morgenstern, and Y. C. Eldar, "Collaborative inference via ensembles on the edge," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8478–8482, IEEE, 2021.

[9] H. Zhou, W. Zhang, C. Wang, X. Ma, and H. Yu, "Bbnet: a novel convolutional neural network structure in edge-cloud collaborative inference," *Sensors*, vol. 21, no. 13, p. 4494, 2021.

[10] Z. He, T. Zhang, and R. B. Lee, "Model inversion attacks against collaborative inference," in *Proceedings of the 35th Annual Computer Security Applications Conference*, pp. 148–162, 2019.

[11] Z. He, T. Zhang, and R. B. Lee, "Attacking and protecting data privacy in edge–cloud collaborative inference systems," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9706–9716, 2020.

[12] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[13] B. Zhao, K. R. Mopuri, and H. Bilen, "idlg: Improved deep leakage from gradients," *arXiv preprint arXiv:2001.02610*, 2020.

[14] C. Fu, X. Zhang, S. Ji, J. Chen, J. Wu, S. Guo, J. Zhou, A. X. Liu, and T. Wang, "Label inference attacks against vertical federated learning," in *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1397–1414, 2022.

[15] O. Li, J. Sun, X. Yang, W. Gao, H. Zhang, J. Xie, V. Smith, and C. Wang, "Label leakage and protection in two-party split learning," *arXiv preprint arXiv:2102.08504*, 2021.

[16] Y. Liu, Z. Yi, Y. Kang, Y. He, W. Liu, T. Zou, and Q. Yang, "Defending label inference and backdoor attacks in vertical federated learning," *arXiv preprint arXiv:2112.05409*, 2021.

[17] T. Wang, Y. Zhang, and R. Jia, "Improving robustness to model inversion attacks via mutual information regularization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 11666–11673, 2021.

[18] T. Zou, Y. Liu, and Y.-Q. Zhang, "Mutual information regularization for vertical federated learning," *arXiv preprint arXiv:2301.01142*, 2023.

[19] S. Oh, J. Park, S. Baek, H. Nam, P. Vepakomma, R. Raskar, M. Bennis, and S.-L. Kim, "Differentially private cutmix for split learning with vision transformer," *arXiv preprint arXiv:2210.15986*, 2022.

[20] C. Fu, X. Zhang, S. Ji, J. Chen, J. Wu, S. Guo, J. Zhou, A. X. Liu, and T. Wang, "Label inference attacks against vertical federated learning," in *31st USENIX Security Symposium (USENIX Security 22)*, (Boston, MA), pp. 1397–1414, USENIX Association, Aug. 2022.

[21] A. Makhdoumi, S. Salamatian, N. Fawaz, and M. Médard, "From the information bottleneck to the privacy funnel," in *2014 IEEE Information Theory Workshop (ITW 2014)*, pp. 501–505, IEEE, 2014.

[22] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," *arXiv preprint arXiv:1612.00410*, 2016.

[23] X. Luo, Y. Wu, X. Xiao, and B. C. Ooi, "Feature inference attack on model predictions in vertical federated learning," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 181–192, IEEE, 2021.

[24] X. Jiang, X. Zhou, and J. Grossklags, "Comprehensive analysis of privacy leakage in vertical federated learning during prediction," *Proceedings on Privacy Enhancing Technologies*, vol. 2022, no. 2, pp. 263–281, 2022.

[25] X. Jin, P.-Y. Chen, C.-Y. Hsu, C.-M. Yu, and T. Chen, "Cafe: Catastrophic data leakage in vertical federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 994–1006, 2021.

[26] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1322–1333, 2015.

[27] J. Sun, A. Li, B. Wang, H. Yang, H. Li, and Y. Chen, "Soteria: Provable defense against privacy leakage in federated learning from representation perspective," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9311–9319, 2021.

[28] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16937–16947, 2020.

[29] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via gradinversion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16337–16346, 2021.

[30] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE symposium on security and privacy (SP)*, pp. 691–706, IEEE, 2019.

[31] F. Mireshghallah, M. Taram, P. Ramrakhyani, A. Jalali, D. Tullsen, and H. Esmaeilzadeh, "Shredder: Learning noise distributions to protect inference privacy," in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 3–18, 2020.

[32] A. Singh, A. Chopra, E. Garza, E. Zhang, P. Vepakomma, V. Sharma, and R. Raskar, "Disco: Dynamic and invariant sensitive channel obfuscation for deep neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12125–12135, 2021.

[33] B. Rassouli and D. Gündüz, "Optimal utility-privacy trade-off with total variation distance as a privacy measure," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 594–603, 2019.

[34] Q. Wu, J. Tang, S. Dang, and G. Chen, "Data privacy and utility trade-off based on mutual information neural estimator," *Expert Systems with Applications*, vol. 207, p. 118012, 2022.

[35] F. Mireshghallah, M. Taram, A. Jalali, A. T. T. Elthakeb, D. Tullsen, and H. Esmaeilzadeh, "Not all features are equal: Discovering essential features for preserving prediction privacy," in *Proceedings of the Web Conference 2021*, pp. 669–680, 2021.

[36] M. Malekzadeh and F. Kawsar, "Salted inference: Enhancing privacy while maintaining efficiency of split inference in mobile computing," in *Proceedings of the 25th International Workshop on Mobile Computing Systems and Applications*, pp. 14–20, 2024.

[37] K. Chaudhuri and D. Hsu, "Sample complexity bounds for differentially private learning," in *Proceedings of the 24th Annual Conference on Learning Theory*, pp. 155–186, JMLR Workshop and Conference Proceedings, 2011.

[38] B. Ghazi, N. Golowich, R. Kumar, P. Manurangsi, and C. Zhang, "Deep learning with label differential privacy," *Advances in neural information processing systems*, vol. 34, pp. 27131–27145, 2021.

[39] P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin, "Club: A contrastive log-ratio upper bound of mutual information," in *International conference on machine learning*, pp. 1779–1788, PMLR, 2020.

[40] M. Yang, Z. Li, J. Wang, H. Hu, A. Ren, X. Xu, and W. Yi, "Measuring data reconstruction defenses in collaborative inference systems," *Advances in Neural Information Processing Systems*, vol. 35, pp. 12855–12867, 2022.

[41] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, (New York, NY, USA), p. 1310–1321, Association for Computing Machinery, 2015.

A Data Leakage in Collaborative Inference

A.1 Input Leakage from the Representation

With the received representation r_i, the server can reconstruct the input data x_i on the edge device by conducting model inversion (MI) attacks [17]. Notably, the head model on the edge device is usually shallow due to the computation resource limitation, which aggravates input leakage from the representation [11]. We conduct experiments on CIFAR10 with ResNet18 to demonstrate the input leakage problem. One convolutional layer is deployed on the device as the head model, and one basic block is deployed as the classifier. The malicious server conducts Knowledge Alignment (KA) attack [17] to recover the input image from the received representation through a generator. The detailed experimental settings can be found in Sec. 5. The reconstructed images are shown in Fig. 8. The high-quality reconstructed images illustrate the collaborative inference's vulnerability to the device's input leakage from the representation.

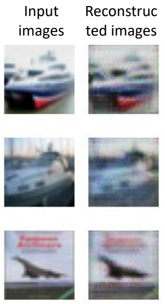


Figure 8: Reconstructed images of KA attack.

A.2 Prediction Leakage from the Feature

The training process enables the cloud server to extract high-level features useful for the collaborative inference task. These high-level features allow the malicious server to fine-tune a classifier head with very few labeled data and accurately conduct inference. The leakage of the prediction makes the edge device holder's privacy, including behavior and preference, exposed to the cloud server. For example, prediction leakage of a collaborative inference-based navigation mobile app allows the cloud server to infer the positions and destinations of the app users. To demonstrate the extent of prediction leakage by the features, we follow the experimental setup in Appendix A.1 and let the cloud server conduct model completion (MC) attack [14] to train a classifier using a small number of auxiliary labeled samples. We also let the cloud server train an entire model with the auxiliary dataset from scratch for comparison. The results are shown in Tab. 1.

Table 1: Compared accuracy of the classifier on the device and the models on the cloud server by conducting MC attack and training from scratch.

	Accuracy(%)
Classifier on the device (clean accuracy)	77.20
MC attack on the server(40 labels)	69.31
Train from scratch on the server(40 labels)	15.34

It is shown that by fine-tuning a classifier with the collaboratively trained encoder, the cloud server can achieve an accuracy of nearly 70% using an auxiliary dataset with only 40 labeled samples. However, training from scratch cannot achieve decent accuracy using the same auxiliary dataset, which shows that the high-level features extracted by the encoder on the server cause prediction leakage.

B Algorithm

Algorithm 1 Training algorithm of InfoScissors. ← means information is sent to the server; ← means information is sent to the device; red steps are conducted on the cloud server.

```
Input: Dataset $\{(x_i,y_i)\}_{i=1}^{N}$; Learning rate $\eta$.
Output: $\theta^h$;$\theta^e$;$\theta^c$;$\psi$;$\phi$.
Initialize $\theta^h$,$\theta^e$,$\theta^c$,$\psi$,$\phi$;
for a batch of data $\{(x_i,y_i)\}_{i\in\mathbb{B}}$ do
    $\{r_i\}_{i\in\mathbb{B}} \leftarrow \{f^h_{\theta^h}(x_i)\}_{i\in\mathbb{B}}$;
    $\mathcal{L}_{d\_a} \leftarrow \frac{1}{|\mathbb{B}|}\sum_{i\in\mathbb{B}} \log g_\psi(x_i|r_i)$;
    $\psi \leftarrow \psi + \eta\nabla_\psi \mathcal{L}_{d\_a}$;
    $\{z_i\}_{i\in\mathbb{B}} \leftarrow \{f^e_{\theta^e}(r_i)\}_{i\in\mathbb{B}}$;
    $\mathcal{L}_c \leftarrow \frac{1}{|\mathbb{B}|}\sum_{i\in\mathbb{B}} \mathcal{L}(f^c_{\theta^c}(z_i),y_i)$;
    $\mathcal{L}_{l\_a} \leftarrow \frac{1}{|\mathbb{B}|}\sum_{i\in\mathbb{B}} \log h_\phi(y_i|z_i)$;
    $\theta^c \leftarrow \theta^c - \eta\nabla_{\theta^c}\mathcal{L}_c$;
    $\phi \leftarrow \phi + \eta\nabla_\phi \mathcal{L}_{l\_a}$;
    $\{y_{n'_i}\}_{i\in\mathbb{B}} \leftarrow$ randomly sample $\{y_{n'_i}\}_{i\in\mathbb{B}}$ from $\{y_i\}_{i\in[N]}$;
    $\{x_{k'_i}\}_{i\in\mathbb{B}} \leftarrow$ randomly sample $\{x_{k'_i}\}_{i\in\mathbb{B}}$ from $\{x_i\}_{i\in[N]}$;
    $\mathcal{L}_{d\_r} \leftarrow \frac{1}{|\mathbb{B}|}\sum_{i\in\mathbb{B}} -\log g_\psi\left(x_{k'_i}|r_i^2\right)$;
    $\mathcal{L}_{l\_r} \leftarrow \frac{1}{|\mathbb{B}|}\sum_{i\in\mathbb{B}} -\log h_\phi\left(y_{n'_i}|z_i^2\right)$;
    $\{\nabla_{z_i}\mathcal{L}\}_{i\in\mathbb{B}} \leftarrow \{\nabla_{z_i}[(1-\lambda_d-\lambda_l)\mathcal{L}_c + \lambda_l\mathcal{L}_{l\_a} + \lambda_l\mathcal{L}_{l\_r}$;
  $+\lambda_d\mathcal{L}_{d\_a} + \lambda_d\mathcal{L}_{d\_r}]\}_{i\in\mathbb{B}}$;
    $\nabla_{\theta^e}\mathcal{L} \leftarrow \frac{1}{|\mathbb{B}|}\sum_{i\in\mathbb{B}} \nabla_{z_i}\mathcal{L}\nabla_{\theta^e}z_i$;
    $\theta^e \leftarrow \theta^e - \eta\nabla_{\theta^e}\mathcal{L}$;
    $\{\nabla_{r_i}\mathcal{L}\}_{i\in\mathbb{B}} \leftarrow \{\nabla_{z_i}\mathcal{L}\nabla_{r_i}z_i\}_{i\in\mathbb{B}}$;
    $\nabla_{\theta^h}\mathcal{L} \leftarrow \frac{1}{|\mathbb{B}|}\sum_{i\in\mathbb{B}} \nabla_{r_i}\mathcal{L}\nabla_{\theta^h}r_i$;
    $\theta^h \leftarrow \theta^h - \eta\nabla_{\theta^h}\mathcal{L}$;
end for
```

C Proofs of theorems

Proof. According to Corollary 3.3 in [39], we have:

$$\mathrm{I(z;y)} < \mathrm{I_{vCLUB}(z;y)} + \mathrm{KL}(p(y|z)||h_\phi(y|z)). \tag{18}$$

Then we have

$$\mathrm{I(z;y)} = \mathbb{E}_{p(\mathrm{z,y})}\log p(\mathrm{y|z}) - \mathbb{E}_{p(\mathrm{y})}\log p(\mathrm{y}) < \epsilon, \tag{19}$$

where $\epsilon = \mathrm{I_{vCLUB}(z;y)} + \mathrm{KL}(p(y|z)||h_\phi(y|z))$. With the samples $\{x_i, y_i\}$, $\mathrm{I(z;y)}$ has an unbiased estimation as:

$$\frac{1}{N}\sum_{i=1}^{N}\log p(y_i|z_i) - \frac{1}{N}\sum_{i=1}^{N}\log p(y_i) < \epsilon. \tag{20}$$

Suppose the adversary has an optimal model h^m to estimate $p(y_i|z_i)$ such that $h^m(y_i|z_i) = p(y_i|z_i)$ for any i, then

$$\frac{1}{N}\sum_{i=1}^{N}\log h^m(y_i|z_i) - \frac{1}{N}\sum_{i=1}^{N}\log p(y_i) < \epsilon. \tag{21}$$

For **classification tasks**, we have

$$\frac{1}{N}\sum_{i=1}^{N}\mathrm{CE}[h^m(z_i), y_i] > \mathrm{CE}_{random} - \epsilon. \tag{22}$$

□

Proof. Similar with Eq. (20), we derive the following for data protection:

$$\frac{1}{N}\sum_{i=1}^{N}\log p(x_i|r_i) - \frac{1}{N}\sum_{i=1}^{N}\log p(x_i) < \epsilon, \tag{23}$$

where $\epsilon = \mathrm{I_{vCLUB}(r;x)} + \mathrm{KL}(p(x|r)||g_\psi(x|r))$. Following the assumption that $p(x|r)$ follows a Gaussian distribution of variance 1, suppose the adversary obtains an optimal estimator g_m of the mean of $p(x|r)$ such that $g^m(x_i|r_i) = p(x_i|r_i)$ for any i. Then we have

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N}\log g^m(x_i|r_i) &< \frac{1}{N}\sum_{i=1}^{N}\log p(x_i) + \epsilon \\
\frac{1}{N}\sum_{i=1}^{N}\log\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}[x_i - g^m(r_i)]^T[x_i - g^m(r_i)]} &< \frac{1}{N}\sum_{i=1}^{N}\log p(x_i) + \epsilon \\
-\frac{1}{N}\sum_{i=1}^{N}\log\sqrt{2\pi} - \frac{1}{2N}\sum_{i=1}^{N}[x_i - g^m(r_i)]^T[x_i - g^m(r_i)] &< \frac{1}{N}\sum_{i=1}^{N}\log p(x_i) + \epsilon \\
\frac{1}{2N}\sum_{i=1}^{N}[x_i - g^m(r_i)]^T[x_i - g^m(r_i)] &> \frac{1}{N}\sum_{i=1}^{N}\log\frac{\sqrt{2\pi}}{p(x_i)} - \epsilon.
\end{aligned} \tag{24}$$

We denote the dimension of x as Q and $\frac{1}{N}\sum_{i=1}^{N}\log\frac{\sqrt{2\pi}}{p(x_i)}$ as κ. Then we have

$$\frac{1}{N}\sum_{i=1}^{N}\mathrm{MSE}[g^m(r_i), x_i] > \frac{2(\kappa - \epsilon)}{Q}. \tag{25}$$

□

D Baselines and Additional Experimental Results

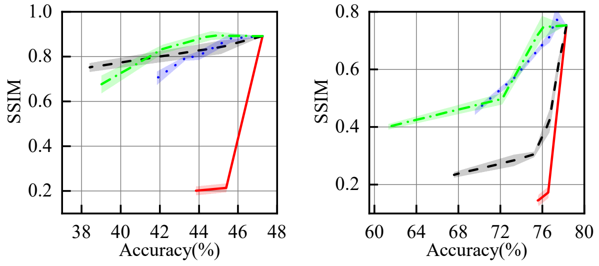
D.1 Baselines

We compare InfoScissors with five existing defense baselines: (1) **Differential Privacy (DP)** [10, 11, 19] protects the data with a theoretical guarantee by clipping the representation and gradients norm and injecting perturbations to the representations and gradients. (2) **Adding Noise (AN)** [40] is proven effective against data leakage in collaborative learning by adding Laplacian noise to the representations and gradients. (3) **Data Compression (DC)** [20] prunes representations and gradients that are below a threshold magnitude, such that only a part of the representations and gradients are sent to the server. (4) **Privacy-preserving Deep Learning (PPDL)** [41] is a comprehensive privacy-enhancing method including three defense strategies: differential privacy, data compression, and random selection. (5) **Mutual Information Regularization Defense (MID)** [18] is the SOTA defense against data leakage in split learning and collaborative inference. MID is also based on mutual information regularization by applying *Variational Information Bottleneck (VIB)*.

D.2 Additional results

The malicious server uses models with different architectures (MLP and MLP_sim) to conduct MC attacks. MLP_sim has one FC layer. MLP has three FC layers with a hidden layer of size 512×256.

Besides the experiments in Sec. 5, we also evaluate our defense against the KA attack with a larger auxiliary dataset on the malicious server. The server has 80 and 800 labeled samples to conduct KA and MC attacks for CIFAR10 and CIFAR100, respectively, and the results are shown in Fig. 9.



(a) CIFAR10

(b) CIFAR100

Figure 9: Model accuracy v.s. SSIM on CIFAR10 and CIFAR100 against KA attack with double numbers of auxiliary samples in Fig. 3.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contributions in the introduction conclude the main contributions of this paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please see Sec. 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Please check Sec. 4.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please check Sec. 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Please check our supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please check Sec. 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Please check Sec. 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please check Sec. 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please check Sec. 6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We do not release a model or scraped datasets.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the corresponding papers when using the pre-trained models and existing datasets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We do not have new datasets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.