

# PATROL: Privacy-Oriented Pruning for Collaborative Inference Against Model Inversion Attacks

Shiwei Ding<sup>†</sup>, Lan Zhang<sup>†</sup>, Miao Pan<sup>‡</sup>, Xiaoyong Yuan<sup>†</sup>

<sup>†</sup>Michigan Technological University

<sup>‡</sup>University of Houston

{shiweid,lanzhang}@mtu.edu, mpan2@uh.edu, xyyuan@mtu.edu

**Abstract**—Collaborative inference has been a promising solution to enable resource-constrained edge devices to perform inference using state-of-the-art deep neural networks (DNNs). In collaborative inference, the edge device first feeds the input to a partial DNN locally and then uploads the intermediate result to the cloud to complete the inference. However, recent research indicates model inversion attacks (MIAs) can reconstruct input data from intermediate results, posing serious privacy concerns for collaborative inference. Existing perturbation and cryptography techniques are inefficient and unreliable in defending against MIAs while performing accurate inference. This paper provides a viable solution, named PATROL, which develops privacy-oriented pruning to balance privacy, efficiency, and utility of collaborative inference. PATROL takes advantage of the fact that later layers in a DNN can extract more task-specific features. Given limited local resources for collaborative inference, PATROL intends to deploy more layers at the edge based on pruning techniques to enforce task-specific features for inference and reduce task-irrelevant but sensitive features for privacy preservation. To achieve privacy-oriented pruning, PATROL introduces two key components: Lipschitz regularization and adversarial reconstruction training, which increase the reconstruction errors by reducing the stability of MIAs and enhance the target inference model by adversarial training, respectively. On a real-world collaborative inference task, vehicle re-identification, we demonstrate the superior performance of PATROL in terms of against MIAs.

## I. INTRODUCTION

Collaborative inference has become a promising solution for using computationally intensive and memory-expensive state-of-the-art deep neural networks (DNNs) on resource-constrained edge devices. [1], [2], [3]. In collaborative inference, a large-size DNN is divided into two partitions and deployed at the edge and the cloud. The input data observed at the edge is fed to the first DNN partition locally; the intermediate output is then sent to the cloud and processed remotely by the second DNN partition. The cloud eventually returns the inference result to edge devices. Collaborative inference can potentially serve a wide range of applications, offering great advantages over the conventional edge or cloud-only inference [4].

Nevertheless, recent research on model inversion attacks (MIAs) has identified privacy risks during collaborative inference [5], [6], [7], [8], [9]. MIAs aim to reconstruct the confidential information of input data from intermediate results during inference [10], [11]. Due to the unknown communication environments or the untrusted cloud server, MIAs can

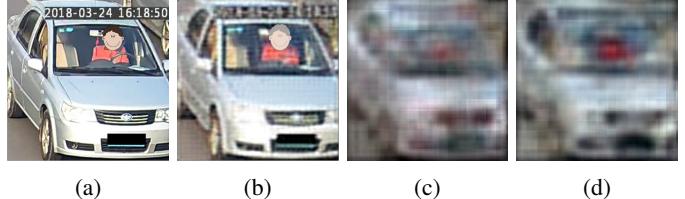


Fig. 1: Model inversion attacks (MIAs) under collaborative inference. (a) shows the original input image captured by a surveillance camera. (b) and (c) show reconstructed images by MIAs when two and four ResNet blocks are deployed at the edge, respectively. (d) shows the reconstructed image by MIA under PATROL protection, where four pruned ResNet blocks are deployed at the edge. The four pruned ResNet blocks have a comparable number of parameters to the two original ResNet blocks. Real facial information is replaced to preserve privacy.

observe intermediate outputs during collaborative inference and reconstruct raw inputs [12], [13], raising serious privacy concerns. Two mainstream defenses have been investigated against MIAs for privacy-preserving collaborative inference. First, perturbation techniques are used to modify intermediate outputs, such as by injecting noises to raw inputs, using adversarial training to add perturbations, or by randomly dropping out intermediate results [6], [7], [14], [5]. Despite the computational efficiency, perturbation-based approaches sacrifice inference performance, such as accuracy, to a large extent for sufficient privacy guarantees. Second, cryptographic techniques (e.g., secure multi-party computation [15], [16] and homomorphic encryption [17], [18]) are used to encrypt intermediate outputs before sharing. The cloud server will process the encrypted intermediate output without decryption. Although cryptographic techniques provide a strong privacy guarantee, they introduce significant delay and computational costs, such as a 14,000 $\times$  slowdown [19], rendering them ineffective for inference at resource-constrained edge devices, especially for the time-critical tasks.

To address these limitations of existing defenses, in this paper, we propose a viable solution by developing privacy-oriented pruning, named PATROL, to trade off privacy, efficiency, and utility of collaborative inference. PATROL leverages the fact that the latter layer can extract more task-specific

features from the input data than the previous layer of a DNN. When the entire DNN is deployed at the edge, the intermediate output becomes the inference result, potentially restricting MIAs' success. We illustrate this idea using the vehicle re-identification task as an example, where surveillance cameras upload intermediate results of captured images after local processing, based on which the cloud server identifies the same vehicle within these images. MIAs intend to reconstruct the task-irrelevant but sensitive information, *i.e.*, the driver's identity. We empirically evaluate MIA performance in Figure 1, where the overall inference uses a ResNet-18 [20] with four ResNet blocks, followed by fully-connected layers. When two ResNet blocks are deployed at the edge, the driver's identity can be revealed through intermediate results, as shown in (b). In comparison, it is difficult to infer the driver's identity when four ResNet blocks are deployed at the edge, as shown in (c). Motivated by these observations, PATROL intends to deploy more layers at the edge to enforce task-specific intermediate features for inference while reducing task-irrelevant but sensitive features for privacy preservation. However, given limited resource budgets at the edge, a critical question is how to store more layers cost-effectively while maintaining collaborative inference accuracy.

Neural network pruning has been well-recognized to determine the sub-network of a DNN to speed up inference without significantly sacrificing prediction performance [21], [22], [23]. However, existing research mainly focuses on balancing the accuracy and efficiency of pruned models, where redundant model parameters in terms of accuracy are pruned rather than privacy-sensitive ones. Moreover, recent research indicates pruned models potentially have more serious privacy risks [24]. Therefore, to carefully determine privacy-oriented pruning, PATROL introduces two key components: Lipschitz regularization and adversarial reconstruction training. Enlighten by the effectiveness of *Lipschitz regularization* to improve the stability of DNNs [25], [26], [27], PATROL enforces a Lipschitz constraint in an opposite manner during pruning. When selecting a pruned network structure, the Lipschitz constraint increases the reconstruction error by reducing the stability of MIAs. Besides, PATROL employs the *adversarial reconstruction training* to alternatively train the surrogate attacker and defender in a game fashion, which further strengthens the target model against strong MIAs. We illustrate the reconstructed image of the vehicle re-identification task protected by PATROL in Figure 1(c). Using a comparable number of edge-side parameters as in (b), PATROL realizes the driver's identity protection in (d), which is as effective as deploying the entire model at the edge in (c). The major contributions of this paper are summarized below.

- We develop PATROL to defend against MIAs under collaborative inference based on privacy-oriented pruning. PATROL trades off privacy, efficiency, and utility of collaborative inference, allowing resource-constrained edge devices using state-of-the-art DNNs for privacy-preserving inference.

- We introduce two key components to PATROL, the Lipschitz regularization and adversarial reconstruction training, which enable privacy-oriented pruning by enforcing task-specific features at intermediate outputs for accurate inference while reducing task-irrelevant but sensitive features for privacy preservation.
- We evaluate PATROL on a real-world collaborative inference task, vehicle re-identification. PATROL can compress the edge DNN partition by 66.7% with only 3.1% prediction accuracy loss on VeriWild datasets and compress edge DNN partition by around 92% with 12.7% prediction accuracy loss on Veri datasets. Meanwhile, PATROL successfully reduces the MIAs performance by 11.9%, and 10.9% in terms of two attack metrics, PSNR and SSIM, on the VeriWild dataset. Also, it decreases MIAs PSNR and SSIM performance by 21.5% and 20.1% on the VERI dataset. The results on both datasets demonstrate superior performance compared to the baseline defenses.

## II. RELATED WORK

### A. Model Inversion Attacks

Model inversion attacks (MIAs) reconstruct confidential information of the raw input from a target model's outputs or intermediate results during the inference phase [10], [11]. MIAs were originally proposed to recover confidential information from training data [28], but recent research has shown that they also pose a threat to raw input data during inference. For example, Yang et al. [29] proposed an inversion network for input reconstruction, where the adversary feeds the target model's output into the inversion network and trains the inversion network to predict raw input data.

Recently, MIAs have received attention under collaborative inference between the edge and the cloud platforms. Oh et al. [5] and He et al. [6] conducted MIAs by reconstructing the input data from the intermediate results transferred from edge to cloud. Salem et al. [30] and Pasquini et al. [8] recently reconstructed the input data by developing an autoencoder and a generative adversarial network (GAN), respectively. Recent research has revealed the significant threat posed by MIAs to collaborative inference.

### B. MIA Defenses under Collaborative Inference

To provide privacy-preserving collaborative inference, defenses against MIAs can be broadly categorized into cryptographic-based and perturbation-based approaches. Cryptographic technologies, such as homomorphic encryption (HE) and secure multi-party computation (SMC), have been widely used to protect inference data privacy [31], [18]. Although offering strong privacy guarantees, cryptography-based defenses impose significant computational and communication overheads [32], making them infeasible for resource-constrained edge devices.

Perturbation-based approaches have been another widely used MIA defense under collaborative inference. For example, Mireshghallah et al. [19] proposed to add noise to the features

that do not contribute to the final inference result. He [6] and Oh [5] randomly drop part of outputs and skip connection in the neural network. However, these methods significantly reduce the prediction performance. Another common way adding perturbation to defend against MIAs is the adversarial training method [33], [34], [35], [36], [37], [38], [7], [14], [39]. It tries to make the target model's outputs show less information about the inputs, which increases the difficulties for the adversary in reconstructing the user's inputs and maintaining the performance of the target model. However, these methods often rely on large perturbations to achieve satisfactory protection against MIAs, which can significantly reduce prediction performance. To address the limitations of existing defenses, this paper aims to develop a more efficient approach to protect collaborative inference privacy against MIAs while maintaining high accuracy.

### C. Neural Network Pruning

Neural network pruning aims to compress a DNN model and increase inference efficiency by removing redundant parameters [22], [40]. Pruning techniques effectively address the challenges of resource constraints on edge devices and improve inference speed. Han et al. [22] proposed to remove the model parameters with large magnitudes to increase model efficiency. Recent research has introduced the use of masks to provide a soft pruning approach to model parameters. Yang et al. [41] developed a method that adds a mask for each neuron or filter to prune according to the importance of the parameter. Li et al. [42] proposed to add a binary mask to prune the weights whose masks are 0 while training. Lin et al. [40] trained a soft mask to identify the importance of specific structures (*e.g.*, blocks, branches, or channels) in a neural network and prune the less important structures based on their soft mask values. Inspired by Lin's work, we design a privacy-oriented pruning method to prune privacy-sensitive convolution channels or blocks.

## III. METHODOLOGY

This section introduces the methodology of PATROL. We first present the threat model and then introduce the design of privacy-oriented pruning with two key components in PATROL. Figure 2 illustrates original collaborative inference under MIAs in (a) and the proposed privacy-preserving collaborative inference protected by PATROL in (b).

### A. Threat Model

We consider a collaborative inference system between the edge and the cloud. The adversary aims to reconstruct the raw input from the intermediate output of the edge-side model as shown in Figure 2. We assume the adversary has no access to the cloud and edge side model parameters (*i.e.*, black-box MIA), but the adversary has prior knowledge about the dimension of the raw input, the architecture of the cloud and edge side models, and the training dataset.

Our design focuses on a practical collaborative inference scenario where edge devices have limited computing, communication, and storage resources. Thus, it is infeasible for the

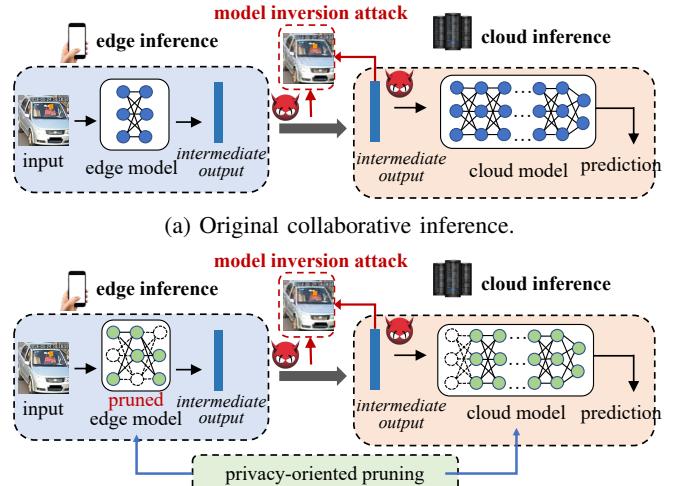


Fig. 2: Model inversion attacks against (a) original collaborative inference and (b) privacy-preserving collaborative inference protected by PATROL.

edge to store and process the entire DNN in its local memory or encrypt the inference data and upload it to the cloud in real time.

### B. Privacy-Oriented Pruning

1) *Design Overview*: PATROL aims to protect the collaborative inference system against MIAs while preserving inference utility and efficiency. Our idea comes from the fact that the latter layer can extract more task-specific features from the input data than the previous layer of a DNN and reduce the task-irrelevant but sensitive features. To accommodate more neural network layers on the edge side for privacy preservation, we introduce a privacy-oriented neural network pruning that reduces the neural network size on the edge while maintaining the utility and efficiency of the edge model. We adopt structured pruning [43] in this paper, which can remove specific structures (*e.g.*, channels or blocks) from a target model. Compared with unstructured pruning, structured pruning enables the hardware acceleration of edge devices for sparse matrix computation and accelerates the inference process.

Define the well-trained DNN for collaborative inference by  $f = f_c \circ f_e$ , named as the target model. The cloud-side partition is defined by  $f_c$  with parameters  $\theta_c$ , and the edge-side partition is defined by  $f_e$  with parameters  $\theta_e$ .  $\theta = \{\theta_e, \theta_c\}$  is trained on a training dataset  $\mathcal{D}$ . To remove structures, such as channels or blocks, from the target model  $f$ , we introduce a trainable soft mask  $m$  in PATROL training to scale the output of structures. In channel-wise pruning,  $m$  is applied to each channel's output. In block-wise pruning,  $m$  is applied to the residual mapping of each residual block. A small value in the well-trained soft mask  $m$  indicates that the output of its corresponding structure has little contribution to the final prediction. Removing these structures will not affect the prediction performance. Therefore, we formulate the following

optimization problem to train soft mask  $m$  and target model  $f$  with parameters  $\theta$ :

$$\min_m \mathcal{L}(\theta, m) + \lambda_1 \|m\|_1 + \lambda_2 \|m\|_2, \quad (1)$$

where  $\mathcal{L}(\theta, m)$  denotes the prediction loss (e.g., the cross-entropy loss in the classification tasks),  $\|m\|_1$  and  $\|m\|_2$  denote two sparsity regularizers, and  $\lambda_1$  and  $\lambda_2$  denote the hyper-parameters to balance the prediction loss and the sparsity regularizers. We introduce  $\ell_1$  sparsity regularizer ( $\|m\|_1$ ) to reduce the number of structures in the target model and achieve a high sparsity ratio. We also introduce  $\ell_2$  sparsity regularizer ( $\|m\|_2$ ) to control the magnitude of soft mask  $m$ . Therefore, incorporating  $\ell_2$  sparsity regularizer can reduce the regularization error caused by the  $\ell_1$  regularizer to compromise between the high sparsity ratio and the prediction accuracy. To achieve a high convergence rate and a high sparsity, we adopt a fast iterative shrinkage-thresholding (FISTA) algorithm [44] to update the soft mask  $m$ . Once the soft mask  $m$  and target model  $f$  have been trained, PATROL prunes the channels or blocks of the target model if their corresponding soft mask values are smaller than a threshold  $\tau$ . In other words, the structures with little contribution to the model prediction are removed.

Existing pruning techniques have primarily focused on improving accuracy and efficiency rather than addressing privacy concerns. To protect data privacy using pruning, we further incorporate Lipschitz regularization and adversarial reconstruction training into the pruning process (detailed in Section III-B3 and III-B2). Lipschitz regularization aims to increase their reconstruction errors by reducing the stability of MIAs. Adversarial reconstruction training generates a surrogate attacker during the training process to mislead the attacker in a game theoretic formulation.

2) *Lipschitz Regularization*: We introduce Lipschitz regularization in pruning to reduce the stability of model inversion attacks and increase their reconstruction errors. The idea of Lipschitz regularization is to restrict the changes of model output given a small input change, so that the output of DNN models will be stable given perturbations in the input [25], [26], [27]. We leverage Lipschitz regularization in an inverse fashion. We aim to make the model inversion attack unstable and increase the reconstruction errors by enforcing Lipschitz constraints.

Given a function  $f$ , the Lipschitz constant  $k$  of  $f$  is defined as the smallest constant in the Lipschitz condition:

$$k = \sup_{x_1 \neq x_2} \frac{\|f(x_1) - f(x_2)\|}{\|x_1 - x_2\|}. \quad (2)$$

Given a certain distance between outputs, the lower bound of the distance between inputs can be derived using Lipschitz constant  $k$ :

$$\|x_1 - x_2\| \geq \frac{1}{k} \|f(x_1) - f(x_2)\|. \quad (3)$$

The increase of  $\frac{1}{k}$  will lead to a large difference in the reconstructed data, given a small difference in the output. Thus, we can defend against model inversion attacks by maximizing  $1/k$  or minimizing the Lipschitz constant  $k$ .

---

### Algorithm 1: PATROL

---

**Input** : Training dataset  $\mathcal{D}$ , cloud-side model  $f_c$  with parameters  $\theta_c$ , edge-side model  $f_e$  with parameters  $\theta_e$ , entire DNN model  $f = f_c \circ f_e$  with parameters  $\theta = \{\theta_c, \theta_e\}$ , total number of layers  $N$ , soft mask  $m$  for pruning, max training epoch  $T$ , pruning threshold  $\tau$ , surrogate inversion model  $f_{adv}$  with parameters  $\theta_{adv}$ .

**Output**: Pruned model  $f_p$ .

- 1 Initialize the soft mask  $m$  from a Normal distribution  $m \sim \mathcal{N}(0, 1)$
- 2 Initialize the cloud-side, edge-side and surrogate inversion models  $f_c, f_e, f_{adv}$
- 3 **for** epoch  $t = 1, \dots, T$  **do**
- 4     **if**  $t \% 10 = 0$  **then**
- 5         % Train surrogate inversion model
- 6         **for** batch sample  $(x, y) \in \mathcal{D}$  **do**
- 7             Reconstruct the raw input data:  

$$x_{adv} = f_{adv}(f_e(x, \theta_e), \theta_{adv})$$
- 8             Update  $\theta_{adv}$  to minimize  $\mathcal{L}_{adv}$  in Eq. 8
- 9         **end**
- 10     **end**
- 11     **for** batch sample  $(x, y) \in \mathcal{D}$  **do**
- 12         % Perform adversarial reconstruction training
- 13         Reconstruct the raw input data:  

$$x_{adv} = f_{adv}(f_e(x, \theta_e), \theta_{adv})$$
- 14         Update parameters  $\theta$  to minimize the prediction loss  $\mathcal{L}$  and maximize the adversarial loss  $\mathcal{L}_{adv}$  in Eq. 9:  

$$\min_{\theta} \mathcal{L}(\theta, m) - \beta \mathcal{L}_{adv}(\theta_e, \theta_{adv})$$
- 15         % Perform Lipschitz regularization
- 16         **for** block  $i = 1, \dots, N$  **do**
- 17             Sample  $\delta \sim \mathcal{N}(0, 1)$
- 18             Calculate Lipschitz constraint  $k$  of the  $i$ -th block defined in Eq. 4 and 5
- 19         **end**
- 20         Update parameters  $\theta$  to minimize the prediction loss  $\mathcal{L}$  and Lipschitz loss  $\mathcal{L}_{lip}$  in Eq. 6:  

$$\min_{\theta} \mathcal{L}(\theta, m) + \mathcal{L}_{lip}(\theta_e)$$
- 21         % Train the soft mask  $m$
- 22         Update mask  $m$  to minimize the prediction loss  $\mathcal{L}$  and the size of  $m$  using Eq. 1:  

$$\min_{\theta, m} \mathcal{L}(\theta, m) + \lambda_1 \|m\|_1 + \lambda_2 \|m\|_2$$
- 23     **end**
- 24     **end**
- 25     % Perform privacy-oriented pruning
- 26     Derive pruned model  $f_p$  by removing the channels or blocks if the corresponding mask  $m_i \leq \tau$
- 27     **Return** Pruned model  $f_p$  with parameter  $\theta$ .

---

Since calculating the Lipschitz constant  $k$  is an intractable problem [25], we introduce the block-wise local Lipschitz constant  $k_i$  to approximate the Lipschitz constant. Given an edge model  $f_e$  with  $N$  blocks, we denote  $f_i$  as the  $i$ -th block of the model.  $f = f_N \circ f_{N-1} \circ \dots \circ f_1$ , ( $i = 1, 2, \dots, N$ ). For  $i \geq 2$ , we define block-wise local Lipschitz constraint  $k_i$  of the  $i$ -th block as:

$$k_i = \sup_x \frac{\|f_i f_{i-1} \dots f_1(x + \delta) - f_i f_{i-1} \dots f_1(x)\|_1}{\|f_{i-1} \dots f_1(x + \delta) - f_{i-1} \dots f_1(x)\|_1}, \quad (4)$$

where  $\delta$  denotes a random noise sampled from a Gaussian distribution. For  $i = 1$ , we define block-wise local Lipschitz constraint  $k_1$  of the first block as:

$$k_1 = \sup_x \frac{\|f_1(x + \delta) - f_1(x)\|_1}{\|\delta\|_1}. \quad (5)$$

We calculate the Lipschitz loss using the block-wise local Lipschitz constraint as follows:

$$\mathcal{L}_{lip}(\theta_e) = \sum_{i=1}^N \alpha_i k_i, \quad (6)$$

where  $\alpha_i$  is the hyper-parameter to balance the constraints. By minimizing the Lipschitz loss, we increase the accumulated errors of model inversion attacks over blocks. We include the Lipschitz loss as a regularization term in the loss function and train the model parameters  $\theta$  to minimize the prediction loss and the Lipschitz loss:

$$\min_{\theta} \mathcal{L}(\theta, m) + \mathcal{L}_{lip}(\theta_e). \quad (7)$$

3) *Adversarial Reconstruction Training*: We leverage adversarial reconstruction training to mislead the model inversion attacker and protect input data privacy. Specifically, we first generate a surrogate inversion model  $f_{adv}$  with parameters  $\theta_{adv}$ . Given an input sample  $x$ , the surrogate inversion model  $f_{adv}$  aims to extract the raw input data from the intermediate output  $f_e(x, \theta_e)$ . The parameters  $\theta_{adv}$  are trained to minimize the adversarial loss  $\mathcal{L}_{adv}$ , which measures the difference between the reconstructed data  $f_{adv}(f_e(x, \theta_e))$  and raw input sample  $x$ . The adversarial loss  $\mathcal{L}_{adv}$  can be calculated as:

$$\mathcal{L}_{adv}(\theta_e, \theta_{adv}) = \|x - f_{adv}(f_e(x, \theta_e), \theta_{adv})\|_2. \quad (8)$$

By integrating the surrogate inversion model, the target model  $f$  is trained to mislead the model inversion attackers while maintaining the prediction performance. To achieve this, we maximize the adversarial loss while minimizing the prediction loss by solving the optimization problem:

$$\min_{\theta} \mathcal{L}(\theta, m) - \beta \mathcal{L}_{adv}(\theta_e, \theta_{adv}). \quad (9)$$

We aim to identify the strongest attack given a target model and incorporate the strongest attack into the minimization problem, which can be formulated as a bi-level optimization problem:

$$\min_{\theta} \max_{\theta_{adv}} \mathcal{L}(\theta, m) - \beta \mathcal{L}_{adv}(\theta_e, \theta_{adv}), \quad (10)$$

where the inner maximization problem is to find the strongest attack for the target model, and the outer minimization problem is to train a model to mislead the strongest attack. Since it is computationally intensive to solve a bi-level optimization

Dataset	Prediction Acc. Drop	PSNR Drop	SSIM Drop	Attack Acc. Drop
VERIWild Small	2.0%	11.9%	11.7%	15.8%
VERIWild Medium	2.2%	12.0%	11.7%	25.7%
VERIWild Large	3.1%	11.9%	10.9%	28.0%
VERI	12.7%	21.5%	20.1%	14.5%

TABLE I: The defense performance of PATROL on the small, medium, and large VERIWild datasets and VERI dataset. We run five different hyper-parameter settings in PATROL and report the average values of Re-identification accuracy drop (Prediction Acc. Drop), PSNR drop, SSIM drop, and attack accuracy drop from the original target model without defenses.

problem, in this paper, we train the target model parameters  $\theta$  and the surrogate inversion model parameters  $\theta_{adv}$  iteratively, following the common practice in adversarial reconstruction training.

#### IV. EVALUATION

In this section, we experimentally evaluate the proposed PATROL on a real-world collaborative inference task, vehicle re-identification, and compare PATROL with the existing defenses against MIAs under collaborative inference.

##### A. Experimental Settings

**Vehicle Re-identification Dataset.** We consider a real-world collaborative inference task, vehicle re-identification, for evaluation. The vehicle re-identification task requires collaboration between multiple edge devices (*e.g.*, surveillance cameras) and a cloud server. Each edge device processes the captured image and uploads the intermediate results to the cloud server. The cloud server identifies if two images capture the same vehicle. Our experiments are conducted on two real-world vehicle re-identification datasets, VERIWild [45] and VERI [46], [47]. VERIWild is the most recent and largest dataset for vehicle re-identification, capturing 416,314 images of 40,671 vehicles' identities from a large CCTV system with 174 cameras during one month. The VERI dataset contains 49,357 images of 776 vehicles from 20 cameras, including 37778 for training and 11579 for testing. We evaluate PATROL on three test datasets with different sizes in VERIWild: small, medium, and large and the VERI testing dataset.

**Vehicle Re-identification Model.** We deploy ResNet-18 [20] as the backbone model for the collaborative vehicle re-identification task and train the model using the open-source framework, Fastreid<sup>1</sup> [48]. Due to resource constraints, it is infeasible to deploy the entire ResNet-18 model on edge. In our work, we consider resource-constrained edge devices, where only the first one or two ResNet blocks can be deployed on edge.

**Inversion Model and Surrogate Inversion Model.** We implement a black-box inversion model following [6]. The inversion model is designed to invert the neural network layers of the target network. For instance, given a target neural network with three  $3 \times 3$  convolution layers at the edge side, we deploy the inversion model with three  $3 \times 3$  deconvolution layers. This design ensures that the inversion model aligns with the target neural network.

<sup>1</sup> fastreid, <https://github.com/JDAI-CV/fast-reid>

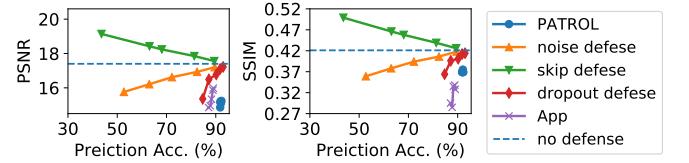
Furthermore, we introduce a surrogate inversion model in adversarial regularization training. Different from the inversion model, the surrogate inversion model uses the output of the last convolution layer of the target network as its input. We design the surrogate inversion model to invert all the neural network layers in the target network.

**Evaluation Metrics.** We consider the three types of evaluation metrics. 1) *Prediction Accuracy*: We report the target model’s top-1 accuracy on the test dataset to measure the model utility. 2) *PSNR* and *SSIM*: We use the similarity between the original and reconstructed images to measure the privacy risks. Two commonly used similarity metrics, PSNR and SSIM [49], are reported in the paper. The higher PSNR/SSIM indicates a higher reconstruction quality or worse defense performance. 3) *L2 norm distance*: We select the average  $L_2$  distance between the input images and reconstruction images as an auxiliary metric. A lower  $L_2$  norm distance proves a higher privacy risk. 4) *Attack accuracy*: We deploy a ResNet-50 model pre-trained on the ImageNet-1K dataset to predict the class of the reconstructed images. We select 13 categories related to vehicles in ImageNet-1K as target labels. If a reconstruction image has been categorized into the target labels, it means the attacker launched a successful attack. The accuracy for this classification model is called the attack accuracy.

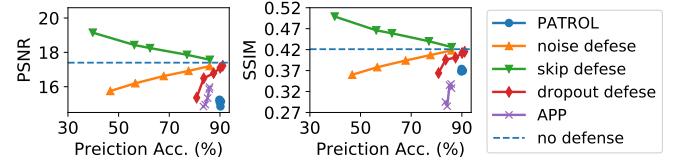
**Existing Defenses.** Since our design focuses on resource-constrained edge devices, we compare PATROL with three perturbation-based MIA defenses: (1) *noise defense*, where the intermediate output is perturbed by Gaussian noise [6], (2) *dropout defense*, where we randomly drop out the intermediate output [6], and (3) *skip defense*, where we randomly skip connections between convolution layers [5], (4) *Adversarial Privacy-Preserving* [7] (denoted by APP), which selects a reconstructed network and a discriminator to guarantee the reconstruction quality and use the reconstruction images for the adversarial training to force the output of the target network show less information about the input.

### B. Experimental Results

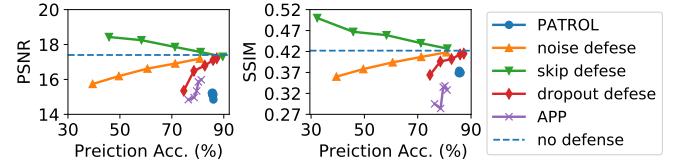
**Effectiveness of PATROL.** We evaluate the performance of PATROL on three test datasets and compare it with the original model without defenses. We assume that edge devices can only deploy 2 residual blocks (memory footprint 2.9MB) in the original model without defenses for the VERIWild dataset and can deploy 1 residual block (memory footprint 0.6MB) in the original model without defenses for the VERI dataset. By using PATROL with a high pruning ratio that removes around 66.7% parameters for the VERIWild dataset and around 92% parameters for the VERI dataset, 4 residual blocks (4.8MB memory cost for VERIWild and 0.5MB memory cost for VERI dataset) can be deployed on edge (the impact of different pruning ratios will be discussed in Section IV-C). To present the effectiveness of PATROL, we measure the prediction accuracy drop, PSNR drop, SSIM drop, and attack accuracy drop from the original model. As shown in Table I, *PATROL significantly degrades the MIA attack performance in terms of PSNR, SSIM, and attack accuracy, while incurring*



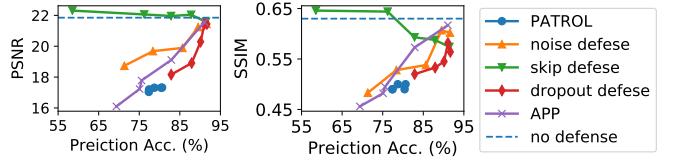
(a) The reconstructed images results on VERIWild Small dataset.



(b) The reconstructed images results on VERIWild Medium dataset.



(c) The reconstructed images results on VERIWild Large dataset.



(d) The reconstructed images results on VERI dataset.

Fig. 3: The PSNR and SSIM of reconstructed images on the VERIWild Small, Medium, Large dataset and VERI dataset. Low PSNR or SSIM indicates the low similarity between the reconstructed and original images, *i.e.*, good protection performance. PATROL outperforms the existing defenses, providing a better tradeoff between privacy and utility.



Fig. 4: PATROL performance. (a) shows the original images captured by the surveillance cameras. (b) shows the reconstructed images from the output of two ResNet blocks without defenses. Confidential information (e.g., vehicle brand, driver identity, vehicle interiors) is exposed in the images. (c) shows the reconstructed images from the output of four ResNet blocks that are protected by PATROL. The confidential information becomes invisible after PATROL. Real facial information is replaced to preserve privacy.

*an affordable prediction accuracy drop. More importantly,*

PATROL effectively preserves confidential information, such as the driver’s identity, in collaborative inference models, as shown in Figure 4.

**Comparison with Existing Defenses.** We compare PATROL with five existing defenses and the original model without defenses (baseline). Figure 3 reports the model performance (prediction accuracy) and privacy risks (PSNR and SSIM), *i.e.*, the ineffectiveness of defenses. The blue dashed line indicates the performance of the original model without any defenses.

As shown in Figure 3, the proposed PATROL achieves the lowest privacy risks compared with the noise, dropout, and skip connection defenses on two datasets. Under the same prediction accuracy, the PSNR and SSIM metrics for PATROL drop at least 11% and 10% than these three defenses on VERIWild datasets and drop at least 2% and 1% on VERI dataset. Under the same privacy level, PATROL prediction accuracy is higher at least 10% than the three defense methods on the VERIWild dataset and at least 3% on the VERI dataset. Compared with the adversarial training defense methods APP [7], the PATROL achieves a similar defense performance with higher prediction accuracy (around 4%) on the VERIWild and VERI dataset, which demonstrates PATROL capability to achieve a better trade-off for the model accuracy and privacy than the APP.

Although noise defense and dropout defense achieve a slightly better prediction accuracy when we apply little noise or a small dropout ratio, their privacy protection becomes ineffective (high PSNR and SSIM values). As we add more noise or set a high dropout ratio, their defense performance can be the same as or better than PATROL, but the prediction accuracy of the target model drops dramatically. The skip defense is ineffective in protecting the model’s privacy, although it improves the model’s efficiency when we skip more neurons in the network. The APP method performs better than noise, dropout and skip defenses on VERIWild datasets. But on the VERI dataset, the App’s advantage is not obvious compared with dropout and noise defense. The APP method achieves better prediction accuracy but less privacy-preserving when using a small trade-off parameter for adversarial training. When we apply a large trade-off parameter for the adversarial training, the APP method can keep the privacy but receives a low model accuracy.

In summary, noise defense and dropout defense methods fail to strike a balance between prediction accuracy and privacy protection performance. The APP method has a similar trade-off as PATROL, but our method takes advantage of the APP methods in two datasets. In contrast, *PATROL achieves both high model prediction accuracy and effective privacy protection*. Moreover, the pruning employed by PATROL reduces the model size and maintains the efficiency of the edge model.

### C. Ablation Study

This section presents ablation studies to show the effectiveness of the proposed designs in PATROL.

**Effectiveness of Lipschitz Regularization and Adversarial Reconstruction Training.** Lipschitz regularization and ad-

Defenses	Prediction Acc. Drop	PSNR Drop	SSIM Drop
Pruning-only	5.4%	8.3%	7.1%
Pruning with Adv	5.0%	10.3%	10.9%
Pruning with Lip	5.7%	9.5%	9.5%
<b>PATROL</b>	<b>3.2%</b>	<b>11.9%</b>	<b>10.9%</b>

TABLE II: Effectiveness of PATROL components. Both adversarial reconstruction training (Adv) and Lipschitz Regularization (Lip) reduce the attack performance compared to the pruning-only approach.

Defenses	# of blocks on edge	PSNR Drop	SSIM Drop
No defense	1	0%	0%
PATROL w/o pruning	1	0%	0%
PATROL w/ low pruning ratio	2	4.6%	8.9%
PATROL w/ high pruning ratio	3	<b>10.2%</b>	<b>15.6%</b>

TABLE III: PATROL Performance on small devices. Only one residual block can be deployed on the edge device without pruning (no defense and PATROL without pruning). PATROL with a low pruning ratio can deploy two residual blocks on the edge device, while PATROL with a high pruning ratio can deploy three residual blocks.

Defenses	# of blocks on edge	PSNR Drop	SSIM Drop
No defense	2	0%	0%
PATROL w/o pruning	2	1.8%	7.1%
PATROL w/ low pruning ratio	3	5.1%	9.5%
PATROL w/ high pruning ratio	4	<b>11.2%</b>	<b>11.9%</b>

TABLE IV: PATROL Performance on large devices. Only two residual blocks can be deployed on the edge device without pruning. PATROL with a low pruning ratio can deploy three residual blocks on the edge device, while PATROL with a high pruning ratio can deploy four residual blocks.

versarial reconstruction training are two key components to enable privacy-oriented pruning. We investigate the effectiveness of Lipschitz regularization and adversarial reconstruction training. We consider three different defense approaches for the ablation study. First, only structured pruning is used for defense (pruning-only). Second, we integrate the structure pruning with adversarial reconstruction training (pruning with Adv). Third, we integrate the structure pruning with Lipschitz regularization (pruning with Lip). We compare the performance of these three defense approaches with PATROL on VERIWild dataset. As shown in Table II, all the defense methods can degrade the attacker’s performance. Both Lipschitz regularization and adversarial reconstruction training can reduce the privacy risks (large PSNR and SSIM drop) compared with the pruning-only defense. By integrating both Lipschitz regularization and adversarial reconstruction training, the privacy-oriented pruning in PATROL achieves the best performance in both prediction accuracy (smallest accuracy drop, 3.2%) and privacy protection (largest PSNR and SSIM drop, 11.9% and 10.9%).

**Effectiveness of Pruning.** We investigate the effectiveness of the pruning ratio in defense. We consider both low and high pruning ratios and investigate two scenarios by taking edge device capabilities into account. In the small edge device sce-

nario, only one residual block in ResNet-18 can be deployed on the edge, due to the limited memory. After privacy-oriented pruning, two or three blocks can be deployed on the edge, based on the low/high pruning ratio. In the large edge device scenario, two residual blocks can be deployed on the edge side. After privacy-oriented pruning, three or four blocks can be deployed on the edge, based on the low/high pruning ratio. The testing dataset is VERIWild.

Table III and Table IV illustrate the results in the two scenarios, where four settings are considered: 1) no defense, 2) PATROL without pruning, which only uses Lipschitz regularization and Adversarial reconstruction training, 3) PATROL with low pruning ratio, and 4) PATROL with high pruning ratio. We observe that the proposed pruning method achieves lower PSNR and SSIM values, providing much better privacy protection than the model without pruning, *i.e.*, no defense and PATROL without pruning. As the high pruning ratio makes more layers to be deployed on the device, PATROL with a high pruning ratio can achieve better performance than that with a low pruning ratio.

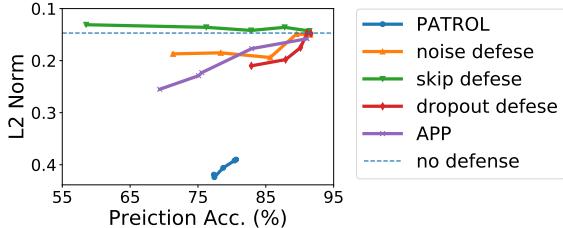


Fig. 5: The MSE (L2 norm) between reconstruction images and input images for PATROL and the Baseline methods on the VERI dataset.

**The L2 norm metric comparison between PATROL and Baseline methods.** In this section, we consider using the Mean Square Error for the reconstruction image and the input image as an auxiliary metric to present the privacy-preserving for every defense method. In this experiment, we only use the VERI dataset as the testing dataset. The results are shown in Figure 5.

The results show the same trend for each defense method as the PSNR and SSIM metrics in figure 3d. The difference is that PATROL performance is far beyond any other methods, which can be additional proof of the effectiveness of PATROL.

Defense	Attack	PSNR	SSIM
No defense	Black-box	17.18	0.43
	White-box	20.48	0.53
PATROL	Black-box	14.85	0.37
	White-box	14.39	0.30

TABLE V: Attack performance under black-box and white-box attacks with and without PATROL defense.

**Defense against White-Box Model Inversion Attacks.** The aforementioned evaluation mainly considers black-box attacks due to their popularity, while the proposed PATROL can effectively defend against white-box attacks. We consider a white-box attacker who knows the target model’s structure and

Defenses	PSNR Drop	SSIM Drop	Prediction Acc. Drop
No defense	3.0%	19.7%	22.8%
APP [7]	9.1%	33.8%	23.5%
PATROL	6.2%	24.6%	18.3%

TABLE VI: The privacy metric and Re-identification prediction accuracy drop (Prediction Acc. Drop) when we change the input size image from  $256 \times 256$  to  $128 \times 128$ . We use the Adversarial privacy-preserving method (APP) as a comparison method.

parameters and can generate the user’s input by observing the intermediate value difference. Table V compares the black-box attacks and white-box attacks, where white-box attackers know the models’ parameters (*e.g.*, via reverse engineering). We observe that without defense, the white-box attack achieves higher attack performance (higher PSNR and SSIM) compared to the black-box attack. However, *by deploying PATROL, the attack performance of the white-box attacks is comparable or even lower than that of the black-box attacks*. This is mainly because the Lipschitz regularization used in PATROL is attack-agnostic, which does not specifically target any particular attacks, which indicates that PATROL is effective against various types of attacks, including both white-box and black-box attacks.

**The impact of image size to the defense.** We try to figure out whether the input image size of vehicle identification influences the defense against MIAs. We use PATROL, APP, and no defense model as the testing method, choosing  $256 \times 256$  (default settings for all experiments) and  $128 \times 128$  as two input sizes for this experiment. Table VI presents the evaluation metrics changes when the input size changes from  $256 \times 256$  to  $128 \times 128$  for no defense, APP and PATROL methods. The results prove that the input size can influence the performance of the defense. The SSIM metrics have been more impacted by it since its metric value dropped more than the PSNR metric. But considering the prediction accuracy drop after changing the input size, the performance downgrade is caused by the downgrade of the target model performance (The no defense model prediction accuracy drops the same as the defense methods). Therefore, the input size has less impact on the defense against MIAs.

## V. CONCLUSION

This paper proposed a privacy-oriented pruning for collaborative inference, named PATROL, to defend against model inversion attacks. PATROL specially selects the sub-network of a DNN to push more layers to the edge without largely degrading prediction accuracy and efficiency. To remove the privacy-sensitive parameters, we introduced Lipschitz regularization and adversarial reconstruction training in PATROL. Defense performance of PATROL is evaluated on vehicle re-identification tasks. Experimental results show that the proposed PATROL can successfully protect collaborative inference against MIAs.

**Acknowledgments:** This work is supported in part by the National Science Foundation under Grants CCF-2106754, CCF-2221741, and CNS-2151238.

## REFERENCES

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

[2] J. H. Ko, T. Na, M. F. Amir, and S. Mukhopadhyay, “Edge-host partitioning of deep neural networks with feature space encoding for resource-constrained internet-of-things platforms,” in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2018, pp. 1–6.

[3] S. Teerapittayanon, B. McDanel, and H.-T. Kung, “Distributed deep neural networks over the cloud, the edge and end devices,” in *2017 IEEE 37th international conference on distributed computing systems (ICDCS)*. IEEE, 2017, pp. 328–339.

[4] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, and X. Chen, “Convergence of edge computing and deep learning: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.

[5] H. Oh and Y. Lee, “Exploring image reconstruction attack in deep learning computation offloading,” in *The 3rd International Workshop on Deep Learning for Mobile Systems and Applications*, 2019, pp. 19–24.

[6] 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.

[7] T. Xiao, Y.-H. Tsai, K. Sohn, M. Chandraker, and M.-H. Yang, “Adversarial learning of privacy-preserving and task-oriented representations,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12434–12441.

[8] D. Pasquini, G. Ateniese, and M. Bernaschi, “Unleashing the tiger: Inference attacks on split learning,” in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 2113–2129.

[9] H. Chen, H. Li, G. Dong, M. Hao, G. Xu, X. Huang, and Z. Liu, “Practical membership inference attack against collaborative inference in industrial iot,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 477–487, 2020.

[10] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, “Privacy in pharmacogenetics: An {End-to-End} case study of personalized warfarin dosing,” in *23rd USENIX Security Symposium (USENIX Security 14)*, 2014, pp. 17–32.

[11] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, “The secret revealer: Generative model-inversion attacks against deep neural networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 253–261.

[12] B. Hitaj, G. Ateniese, and F. Perez-Cruz, “Deep models under the gan: information leakage from collaborative deep learning,” in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 603–618.

[13] 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)*. IEEE, 2019, pp. 691–706.

[14] T. Ng, H. J. Kim, V. T. Lee, D. DeTone, T.-Y. Yang, T. Shen, E. Ilg, V. Balntas, K. Mikolajczyk, and C. Sweeney, “Ninjadesc: content-concealing visual descriptors via adversarial learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12797–12807.

[15] S. Wagh, S. Tople, F. Benhamouda, E. Kushilevitz, P. Mittal, and T. Rabin, “Falcon: Honest-majority maliciously secure framework for private deep learning,” *Proceedings on Privacy Enhancing Technologies*, vol. 1, pp. 188–208, 2021.

[16] P. Mishra, R. Lehmkuhl, A. Srinivasan, W. Zheng, and R. A. Popa, “Delphi: A cryptographic inference service for neural networks,” in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 2505–2522.

[17] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, “[GAZELLE]: A low latency framework for secure neural network inference,” in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 1651–1669.

[18] J. Liu, M. Juuti, Y. Lu, and N. Asokan, “Oblivious neural network predictions via mininonn transformations,” in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 619–631.

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

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[21] M. Mozer and P. Smolensky, “Skeletonization: A technique for trimming the fat from a network via relevance assessment,” in *Advances in Neural Information Processing Systems (NIPS)*, 1988.

[22] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding,” in *International Conference on Learning Representations (ICLR)*, 2016.

[23] D. W. Blalock, J. J. G. Ortiz, J. Frankle, and J. V. Guttag, “What is the state of neural network pruning?” in *MLSys*, 2020.

[24] X. Yuan and L. Zhang, “Membership inference attacks and defenses in neural network pruning,” in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 4561–4578. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity22/presentation/yuan-xiaoyong>

[25] A. Virmaux and K. Scaman, “Lipschitz regularity of deep neural networks: analysis and efficient estimation,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[26] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas, “Efficient and accurate estimation of lipschitz constants for deep neural networks,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[27] Z. Cranko, Z. Shi, X. Zhang, R. Nock, and S. Kornblith, “Generalised lipschitz regularisation equals distributional robustness,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 2178–2188.

[28] 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*, 2015, pp. 1322–1333.

[29] Z. Yang, J. Zhang, E.-C. Chang, and Z. Liang, “Neural network inversion in adversarial setting via background knowledge alignment,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 225–240.

[30] A. Salem, A. Bhattacharya, M. Backes, M. Fritz, and Y. Zhang, “[Updates-Leak]: Data set inference and reconstruction attacks in online learning,” in *29th USENIX security symposium (USENIX Security 20)*, 2020, pp. 1291–1308.

[31] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, “Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy,” in *International conference on machine learning*. PMLR, 2016, pp. 201–210.

[32] F. Mo, H. Haddadi, K. Katevas, E. Marin, D. Perino, and N. Kourtellis, “Ppfl: privacy-preserving federated learning with trusted execution environments,” in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, 2021, pp. 94–108.

[33] H. Edwards and A. Storkey, “Censoring representations with an adversary,” *arXiv preprint arXiv:1511.05897*, 2015.

[34] N. Raval, A. Machanavajjhala, and L. P. Cox, “Protecting visual secrets using adversarial nets,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017, pp. 1329–1332.

[35] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, “Context-aware generative adversarial privacy,” *Entropy*, vol. 19, no. 12, p. 656, 2017.

[36] 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)*, 2018, pp. 606–624.

[37] F. Pittaluga, S. Koppal, and A. Chakrabarti, “Learning privacy preserving encodings through adversarial training,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 791–799.

[38] Z. Wu, H. Wang, Z. Wang, H. Jin, and Z. Wang, “Privacy-preserving deep action recognition: An adversarial learning framework and a new dataset,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 2126–2139, 2020.

[39] I. R. Dave, C. Chen, and M. Shah, “Spact: Self-supervised privacy preservation for action recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20164–20173.

[40] S. Lin, R. Ji, C. Yan, B. Zhang, L. Cao, Q. Ye, F. Huang, and D. Doermann, “Towards optimal structured cnn pruning via generative

adversarial learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2790–2799.

[41] C. Yang, Z. Yang, A. M. Khattak, L. Yang, W. Zhang, W. Gao, and M. Wang, “Structured pruning of convolutional neural networks via  $l_1$  regularization,” *IEEE Access*, vol. 7, pp. 106385–106394, 2019.

[42] T. Li, B. Wu, Y. Yang, Y. Fan, Y. Zhang, and W. Liu, “Compressing convolutional neural networks via factorized convolutional filters,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[43] S. Anwar, K. Hwang, and W. Sung, “Structured pruning of deep convolutional neural networks,” *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 13, no. 3, pp. 1–18, 2017.

[44] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[45] Y. Lou, Y. Bai, J. Liu, S. Wang, and L.-Y. Duan, “Veri-wild: A large dataset and a new method for vehicle re-identification in the wild,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3235–3243.

[46] X. Liu, W. Liu, T. Mei, and H. Ma, “A deep learning-based approach to progressive vehicle re-identification for urban surveillance,” in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 869–884.

[47] ———, “Provid: Progressive and multimodal vehicle reidentification for large-scale urban surveillance,” *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 645–658, 2017.

[48] L. He, X. Liao, W. Liu, X. Liu, P. Cheng, and T. Mei, “Fastreid: A pytorch toolbox for general instance re-identification,” *arXiv preprint arXiv:2006.02631*, 2020.

[49] A. Hore and D. Ziou, “Image quality metrics: Psnr vs. ssim,” in *2010 20th international conference on pattern recognition*. IEEE, 2010, pp. 2366–2369.