

# CompLeak: Deep Learning Model Compression Exacerbates Privacy Leakage

Na Li<sup>1\*</sup> Yansong Gao<sup>2\*</sup> Hongsheng Hu<sup>3</sup> Boyu Kuang<sup>1 †</sup> Anmin Fu<sup>1 4 †</sup>

<sup>1</sup>*School of Cyber Science and Engineering, Nanjing University of Science and Technology, China.*

<sup>2</sup>*School of Cyber Science and Engineering, Southeast University, China.*

<sup>3</sup>*School of Computer Science, Shanghai Jiao Tong University, China.*

<sup>4</sup>*School of Computer Science and Engineering, Nanjing University of Science and Technology, China.*

Email: li\_na, kuang, fuam@njjust.edu.cn, gao.yansong@hotmail.com, hongsheng.hu@sjtu.edu.cn

## Abstract

Model compression is crucial for minimizing memory storage and accelerating inference in deep learning (DL) models. Users can access different compressed model versions according to their resources and budget. However, while existing compression operations primarily focus on optimizing the trade-off between resource efficiency and model performance, the privacy risks introduced by compression remain overlooked and insufficiently understood.

In this work that focuses on typical classification tasks, through the lens of membership inference attack (MIA), we propose *CompLeak*, the first privacy risk evaluation framework examining three widely used compression configurations that are pruning, quantization, and weight clustering all supported by the commercial model compression framework of Google’s TensorFlow-Lite (TF-Lite), and first two supported by Facebook’s PyTorch Mobile and the open-source toolkit of Microsoft NNI. *CompLeak* has three variants, given access to the available number of compressed models and/or the original model. *CompLeak<sub>NR</sub>* starts by adopting existing MIA methods to attack each individual compressed model, and identifies that different compressed models influence members and non-members differently. When the original model and one compressed model are available, *CompLeak<sub>SR</sub>* leverages the compressed model as a reference to the original model and uncovers more privacy by combining meta information (e.g., confidence vector) from both models. When multiple compressed models are available with/without accessing the original model, *CompLeak<sub>MR</sub>* innovatively exploits privacy leakage info from multiple compressed versions to substantially signify the overall privacy leakage. We conduct extensive experiments on six diverse model architectures (from ResNet to BERT and GPT-2), and five image and textual benchmark datasets. Our experimental results show that *CompLeak<sub>MR</sub>* achieves the best MIA performance on all evaluation metrics, including TPR @ 0.1% FPR, proving that model compression exacerbates privacy leakage.

\*The authors contribute equally to this paper.

†Corresponding authors.

## 1 Introduction

Driven by the large-scale data, DL has witnessed remarkable advancements, excelling in applications such as identity recognition [17], disease diagnosis [79], self-driving [52] and protein structure prediction [35]. However, these outstanding state-of-the-art models are scaled with an increased number of parameters, which require considerable computational resources and memory footprint, challenging their deployment on resource-constrained devices.

Model compression [15, 16, 24, 29, 30, 50] has been a mainstream technique to resolve such challenges, which are already widely used by industries. Commercially available compression frameworks, such as Google’s TF-Lite and Facebook’s PyTorch Mobile as well as open-source toolkits such as Microsoft NNI, facilitate the deployment of DL models on Internet of Things (IoT) and mobile devices [47]. These frameworks enable model providers to easily generate compressed models using operations like weight clustering, pruning, and quantization, either during training or post-training. Furthermore, model providers can also employ compression operations to provide interfaces for models with varying model sizes—larger models generally deliver superior performance but are more expensive—enabling users to selectively access customized models according to their needs and budget.

### 1.1 Limitation

While model compression greatly accelerates inference and reduces memory storage, it has been delicately studied and shown to be vulnerable to security attacks such as backdoor attacks [21, 47]. However, the *privacy risks* imposed by *model compression* are overlooked and poorly understood, although the privacy risks of DL models have been widely studied [9, 11, 28, 40–42, 60, 72].

DL models inherently memorize sensitive information from their training datasets, with MIA emerging as a commonly used auditing technique for evaluating such privacy risks [12, 60, 73]. MIA exploits the model’s memorization of

its training data, which manifests as significant differences in outputs between members and non-members. This vulnerability enables attackers to infer whether a given data sample was part of the training dataset, posing a significant threat to individual privacy. For instance, an attacker could deduce a person’s participation in a confidential clinical trial by determining that his medical record was used to train a predictive model for an experimental drug. On the other hand, MIA can serve as a valuable tool for auditing privacy leakage, particularly in light of stringent privacy regulations such as the General Data Protection Regulation (GDPR) [1], which mandates strong protections for user data.

Notably, numerous mainstream compression operations (e.g., pruning) require a fine-tuning phase with the original training data to restore model performance, which not only transforms the model but also re-optimizes it, thereby altering its memory of the training data and raising potential privacy concerns. However, the privacy risks stemming from such compression operations remain unexplored—especially in scenarios where multiple compressed models are published and accessible. To our knowledge, the most relevant study is by Yuan *et al.* [74], which assesses the privacy leakage of a pruned model via MIA. However, their evaluation is fundamentally different from our work, which is limited to reliance on information from a single-pruned model and does not account for the unique privacy risks introduced by model compression, where new insights could be derived by correlating information across different compressed model versions and the original model.

Similar to [11, 28, 40, 75], by focusing on typical classification tasks, we ask the following research questions to underscore the urgent need for a comprehensive investigation into the privacy risks of mainstream compression technologies.

Does model compression exacerbate privacy leakage with increasing access to multiple compressed models? If so, to what extent does it amplify privacy risks?

## 1.2 Our Work

This work, for the first time, unveils and confirms that model compression exacerbates privacy leakage through the lens of MIA as a privacy auditing approach. The primary reason is that differing compression operations affect members and non-members differently, and different compressed model versions leak privacy in slightly different ways due to variations in their memorization capacity and the inherent randomness of compression operations (e.g., pruning at different or even identical rates, weight clustering with varying or identical numbers of clusters). Consequently, aggregating leakage from multiple sources amplifies the overall privacy risk. To quantify the extent of this additional leakage, we design various MIA methods tailored for a wide range of compression scenarios,

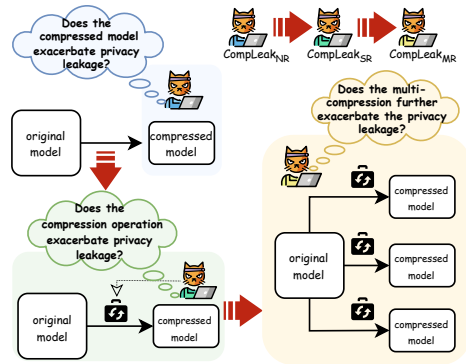


Figure 1: Compression scenarios targeted by the three  $\text{CompLeak}$  variants: blue/green/yellow attacker icon stands for  $\text{CompLeak}_{\text{NR}}$ / $\text{CompLeak}_{\text{SR}}$ / $\text{CompLeak}_{\text{MR}}$  considering number of accessible compressed model versions.

as illustrated in Figure 1, primarily considering the number of accessible compressed model versions.  $\text{CompLeak}_{\text{NR}}$ , which directly adopts existing MIA techniques, evaluates privacy leakage per compressed model without relying on any reference or paired model.  $\text{CompLeak}_{\text{SR}}$  introduces new MIA techniques to capture additional privacy leakage from a compressed model version (i.e., functioning as a reference) paired with the original model. Furthermore,  $\text{CompLeak}_{\text{MR}}$  enhances MIA techniques to assess privacy risks when multiple compressed model versions are accessible, even in real-world scenarios where the original model is absent.

Notably, this work evaluates privacy risks on three compression techniques—quantization, pruning, and weight clustering—all of which retain the same underlying architecture, are widely supported by commercial (e.g., TensorFlow-Lite) or open-source compression frameworks, and are commonly deployed on end, mobile, and IoT devices. In contrast, other compression technologies, such as knowledge distillation, rely on different architectures and are not primarily targeted at mobile or IoT scenarios. Below, we highlight the key findings of each variant under the  $\text{CompLeak}$  framework and brief its core attack design.

**$\text{CompLeak}_{\text{NR}}$ .** We refer to existing MIAs [51, 60, 63, 73–75], which are conducted solely based on the leaked information from an underlying model itself—where no reference model is used—as  $\text{CompLeak}_{\text{NR}}$ . Then, we utilize  $\text{CompLeak}_{\text{NR}}$  to audit the vulnerabilities of the original model and per compressed model with varying compression degrees. Each compressed model obtained through pruning, quantization, and weight clustering compression operations is evaluated in our experiments upon  $\text{CompLeak}_{\text{NR}}$ . Notably, consistent with the results in [74], we observe that highly compressed models are generally less vulnerable than the original model. For instance, when an MLP-based attack meta-classifier [51] targets a 90%-pruned VGG16 model on Mini-ImageNet, the attack accuracy drops by 5% compared to the original model. This reduction is

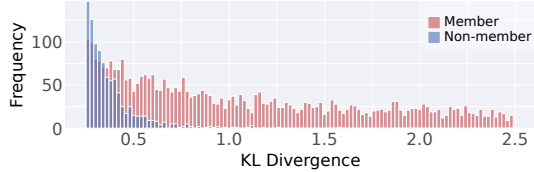


Figure 2: The KL divergence between the two posteriors on the same samples, obtained from the original MobileNetV2 (trained on Tiny-ImageNet) and the 40% pruned MobileNetV2.

potentially because high-level compression significantly limits model capacity and suppresses overfitting [29, 30, 50, 67]. On the contrary, pruning with lower sparsity, quantization into 8-bit integers, and weight clustering with more centroids exhibit comparable privacy leakage to the uncompressed model.

Despite the overall MIA accuracy remaining relatively consistent across different compressed models, the way each compressed model affects members and non-members varies. This variation serves as the foundation for additional privacy leakage, where leaked information is newly captured through  $\text{CompLeak}_{\text{SR}}$  and  $\text{CompLeak}_{\text{MR}}$ .

**CompLeak<sub>SR</sub>.** We note that the impact induced by compression operations (i.e., on the posterior probability distribution) varies substantially between members and non-members, as shown in Figure 2. Therefore, our insight is that capturing the subtle alterations imposed by compression operations will amplify privacy. Based on this intuition, instead of using information from a single model only, we incorporate leaked information from a compressed model and pair it with the information from the original model to improve the MIA performance. We refer to this  $\text{CompLeak}$  variant as  $\text{CompLeak}_{\text{SR}}$  because a single reference model is utilized.

Generally,  $\text{CompLeak}_{\text{SR}}$  utilizes *meta-data construction* to combine a pair of posteriors, one from the original model and the other from a corresponding compressed version, to form meta-data that is used to train a binary meta-classifier for membership inference. From the experimental results, regardless of the compression operation’s type and degree,  $\text{CompLeak}_{\text{SR}}$  exhibits strong capabilities, providing evidence that model compression indeed threatens privacy. Specifically,  $\text{CompLeak}_{\text{NR}}$  [73] achieves 56.7% MIA accuracy on the original VGG16 trained on Mini-ImageNet, which drops to 52.9% on the pruned model with 90% parameter removal. In contrast, when exploiting the two models together, our  $\text{CompLeak}_{\text{SR}}$  significantly improves the accuracy to 79.9%, a 27% gain.

**CompLeak<sub>MR</sub>.** Model service providers typically release a set of compressed models (i.e., more than one) with varying capacities via different interfaces [2, 3], so we devise  $\text{CompLeak}_{\text{MR}}$  to exploit multiple compressed models as references to further amplify the privacy leakage. When we intuitively adopt the same methodology as  $\text{CompLeak}_{\text{SR}}$  to combine the posteriors generated by multiple compressed models

for the same target sample as meta-data, no improvement is observed in the attack performance compared to  $\text{CompLeak}_{\text{SR}}$ . We argue that this occurs because  $\text{CompLeak}_{\text{SR}}$  utilizes the variation in posteriors between the original model and a compressed model, whereas the variation in posteriors across multiple versions is minimal and more challenging to interpret.

Encouragingly, we discover two remarkable phenomena that support our intuition that different compressed models leak privacy in slightly different ways. First, building on  $\text{CompLeak}_{\text{SR}}$ , although it becomes complicated to directly capture subtle varieties among the posteriors generated by multiple compressed models, we observe that membership inference results from  $\text{CompLeak}_{\text{SR}}$  attack meta-classifiers—each trained to target a certain level compressed model—on the same target sample exhibit notable differences. For example, the membership status predicted by  $\text{CompLeak}_{\text{SR}}$  shows approximately 22% discrepancy when targeting an 80%-pruned VGG16 versus a 90%-pruned VGG16 on the Mini-ImageNet. Second, we identify that as the compression degree increases, the evolution of loss calculated from compressed models using ground truth labels and the cross-entropy reveals disparities between members and non-members, both in direction and magnitude. Specifically, the loss for members increases with higher sparsity levels, while for non-members fluctuates.

The above findings provide us with new insights into the design of  $\text{CompLeak}_{\text{MR}}$ , aggregating leaks from multiple sources to further amplify the overall privacy risk. More concretely, an adversary can first utilize *posterior concatenation* by querying each  $\text{CompLeak}_{\text{SR}}$  attack meta-classifier for the target sample to obtain a set of  $\text{CompLeak}_{\text{SR}}$  attack meta-posteriors, which are then concatenated. Then, in the *loss concatenation* step, the target sample is fed into each compressed model in ascending order of compression degree to compute a set of losses, which are also concatenated. Finally, the concatenated posteriors and losses are stacked to form meta-data, which is used to train a  $\text{CompLeak}_{\text{MR}}$  meta-classifier for membership inference. Extensive experiments show that multiple compressed models further exacerbate membership leakage compared to  $\text{CompLeak}_{\text{SR}}$  using a single compressed model, especially in TPR @ 0.1% FPR. Additionally, we relax the adversary’s knowledge, following [74], by assuming they can only access multiple compressed versions, but not the original model. In this more realistic and challenging setting, the adversary cannot obtain  $\text{CompLeak}_{\text{SR}}$  attack meta-posteriors, instead, *posterior concatenation* refers to concatenating the posteriors obtained by querying each compressed model for the target sample. Notably, even without access to the original model, experimental results show that  $\text{CompLeak}_{\text{MR}}$  achieves a 9.7% improvement in balanced accuracy and a 14.2% gain in AUC over  $\text{CompLeak}_{\text{NR}}$  [74], demonstrating that access to multiple compressed variants alone is sufficient to achieve strong performance.

**Contribution.** Our main contributions can be summarized as:

- We propose  $\text{CompLeak}$ , the first systematic privacy risk

evaluation framework that examines three widely used compression operations—pruning, quantization, and weight clustering—through the lens of membership inference attacks.

- We employ the existing MIA as  $\text{CompLeak}_{\text{NR}}$ , relying solely on information from the underlying model, to comprehensively assess the privacy leakage per compressed model (**Section 4**).
- We present  $\text{CompLeak}_{\text{SR}}$  for a single compression scenario, unveiling that regardless of the compression degree, the compression operations indeed jeopardize privacy (**Section 5**).
- We propose  $\text{CompLeak}_{\text{MR}}$  aggregating leaks from multiple compressed models to further amplify the privacy leakage caused by model compression (**Section 6**).
- We conduct extensive experiments on typical classification tasks to demonstrate the effectiveness of  $\text{CompLeak}$ .

## 2 Background and Related Work

### 2.1 Compression

Among the various compression operations, the three that are currently most widely employed and supported by commercial compression frameworks [24, 33], namely *pruning* [8, 29, 30, 50], *quantization* [23, 71], and *weight clustering* [24]. Upon each operation, we illustrate the weight matrix results in Figure 10. Additionally, querying multiple model variants compressed from the same model is feasible in real-world serving scenarios. For instance, HuggingFace’s public APIs provide multiple model versions of MobilenetV2 (fp32, int8) [4], while TCL Research Europe offers ResNet50-D with varying pruning levels (original, 25%, 37%, 56%) [2], demonstrating the practical impacts of  $\text{CompLeak}$ .

**Pruning.** Pruning involves removing relatively unimportant parameters, typically following the “train-prune-finetune” workflow [74]. It is primarily categorized into unstructured and structured pruning, with the former delivering higher compression rates and prediction accuracy, while the latter offers superior hardware acceleration. Unstructured pruning ignores the model’s architecture and focuses on removing individual parameters. For instance, Han *et al.* [24] set the parameters with the lowest magnitudes to zero. In contrast, structured pruning leverages the model’s structure to remove parameters in a more organized manner, i.e., removing entire groups of parameters. For example, Li *et al.* [39] removed entire filters with the lowest absolute values from the convolution layers. Men *et al.* [49] observed certain layers that contribute little to the overall performance. Building on this, they used Block Influence (BI) to measure the importance of each layer based on the similarity between its input and output and performed layer pruning by removing layers with low BI values.

**Quantization.** Quantization typically converts 32-bit floating-point weight formats to the most compact and popular 8-bit integers, broadly divided into training-aware quantization (QAT) and post-training quantization. The former maintains

the model’s performance by learning the quantized parameters, but it follows a time-consuming “train-QAT-finetuning” workflow [33]. The latter avoids the training process by using a small calibration dataset to guide the quantization, allowing it to be directly applied to pre-trained full-precision models [31]. In addition, dynamic quantization belongs to the category of post-training quantization and provides support for LLMs. The system automatically selects the scale factor for activations based on the data range observed at runtime, eliminating the need for additional fine-tuning.

**Weight Clustering.** It is also known as weight sharing, which groups the weights of each layer into multiple clusters based on their similarity and assigns the centroid value of each cluster to all the weights within it. In this case, each group only needs to store the centroid value and the corresponding cluster index, as illustrated in Figure 10. During weight updates, gradients for each weight are computed and aggregated within the same cluster to perform the update [24].

### 2.2 Membership Inference Attack

MIA aims to determine whether a target sample is part of the training dataset for a given model [11, 25, 28, 40, 41, 43, 74]. Because of the simplicity of the definition, MIA has been widely used for evaluating the privacy risks of DL models [9, 32, 42, 59, 72]. Formally, considering a target sample  $x$ , a trained victim model  $\mathcal{M}$ , the process of membership inference can be defined as:

$$\mathcal{A} : x, \mathcal{M} \rightarrow \{0, 1\} \quad (1)$$

The attack meta-model  $\mathcal{A}$  is essentially a binary classifier and can be constructed in various ways. If a target sample  $x$  has been used to train  $\mathcal{M}$ ,  $\mathcal{A}$  outputs 1 (i.e., member) and 0 otherwise (i.e., non-member).

In practical scenarios, most MIAs are conducted in black-box settings, where adversaries have access only to the posterior probability distribution of the victim model’s output. A common strategy proposed by Shokri *et al.* [60] is to train shadow models that mimic the behavior of the victim model. These shadow models output posterior as meta-data to train a binary meta-classifier for membership inference. To improve attack performance, Nasr *et al.* [51] incorporated the true labels of data samples as features into the meta-data. More recently, Du *et al.* [18] emphasized the significance of exploiting membership dependencies between instances and improved MIA performance through conditional shadow training. Additionally, several studies [56, 63, 73] introduced metric-based MIAs that use metric values, e.g., posterior, entropy, or modified entropy, calculated from the victim model’s output to distinguish membership status without the need to train the attack meta-classifier. Furthermore, some works utilize MIA to investigate the privacy impact of specific technologies or systems, e.g., explainable machine learning methods [44], synthetic data [77], machine unlearning [14], visual

encoders [45, 80], speaker recognition systems [13], query-based systems [64] and multi-exit networks [42]. Recently, MIA has been extended to the foundation models, e.g., diffusion models [53, 54] and LLMs [48, 69, 76].

To the best of our knowledge, the work from Yuan *et al.* [74] is the only one that targets pruning in the context of a compressed model, namely SAMIA. However, SAMIA fundamentally differs from our work, as it targets only a single-pruned model. Because it overlooks the unique privacy risks of accessing multiple compressed model versions rooted in model compression, making it essentially the same as other works attacking a full-precision model.

### 3 Threat Model

This section defines the threat model by providing a detailed description of the adversary’s knowledge, capability, and objective. Notably, the three  $\text{CompLeak}_{\text{NR}}$  variants, which will be discussed in the following three sections, are all conducted under this threat model.

**Adversary Knowledge.** As mentioned earlier, service providers generally release interfaces to models in various sizes allowing users to selectively access. In this paper, we focus on the most commonly adopted black-box setting as in existing works [13, 28, 40, 48, 53], where the attacker only has access to the posterior probability distribution of the victim’s outputs from the original model and its associated compressed models. Moreover, we relax this assumption in Section 6.1, where the original model is inaccessible. We also align with SOTAs [28, 40, 74], assuming that the adversary has a local shadow dataset  $\mathcal{D}_s$ , which has the same distribution as the victim’s training dataset but without any overlap. The adversary also knows the architecture of the victim (original/compressed) model—accurate model architecture can be stolen in a black-box manner via side-channel information [22, 70], as well as the compression configuration—this information is often provided by the model provider, and the ground truth of the target samples whose membership status needs to be inferred.

**Adversary Capability.** The adversary can utilize  $\mathcal{D}_s$  to train a set of shadow (original/compressed) models that share the same architecture as the victim (original/compressed) models, to mimic the victim’s behavior.

**Adversary Goal.** Given a target sample, the adversary aims to determine whether it belongs to the training dataset of the victim model according to the above knowledge and ability.

### 4 $\text{CompLeak}_{\text{NR}}$

In this section, we directly utilize existing membership inference attacks [51, 60, 63, 73–75], which are based solely on leaked information from the target model—without using a reference model—denoted as  $\text{CompLeak}_{\text{NR}}$ , to quantify and

compare the privacy risks between the original model and each compressed model with varying compression degrees and configurations. Since the only difference among the MIA adopted in  $\text{CompLeak}_{\text{NR}}$  is how they leverage the underlying model’s leaked information, we begin by categorizing and describing their attack methodologies. Then, we present the evaluation results. It is crucial to emphasize that our goal here is to assess the privacy leakage of a given compressed model through  $\text{CompLeak}_{\text{NR}}$ , rather than designing the novel MIA specific to compression scenarios, which will be advanced in Section 5 and Section 6.

#### 4.1 Attack Methodology

It is important to emphasize that, although the threat model assumes the adversary can access models of varying sizes simultaneously,  $\text{CompLeak}_{\text{NR}}$  only uses the information from a single (whether compressed or not) model to quantify the extent of its membership leakage. It serves as a baseline to understand how compression exacerbates privacy leakage per compressed model being available. Here, the  $\text{CompLeak}_{\text{NR}}$  employs six different MIAs, three of which are training-based, two are metric-based, and one state-of-the-art attack.

**Training-based.** In general, training-based attacks require the adversary to train shadow models and leverage the information from these shadow models to construct meta-data. Based on this, the adversary can then train a meta-classifier for membership inference. Various approaches exist for constructing meta-data, as detailed below: Shokri *et al.* [60] utilize posteriors as meta-data, whereas Nasr *et al.* [51] concatenate ground-truth labels and posteriors for the same purpose. Furthermore, Yuan *et al.* [74] design a SAMIA specifically for a single-pruned neural network, combining the posterior, sensitivity, and ground-truth label as meta-data to train a transformer-based meta-classifier.

**Metric-based.** Metric-based attacks use a threshold to infer membership based on the metric values calculated from the victim model’s output, without the need for training the attack meta-classifier. In this work, we consider two metrics: entropy loss [73] and modified entropy [63].

**State-of-the-art MIA.** We apply  $\text{CompLeak}_{\text{NR}}$  with RMIA [75], a low-cost and high-power variant of LiRA [11], which is primarily based on the pairwise likelihood ratio test. This test compares the victim model’s output (typically true-class confidence) on a target sample against its outputs on random population samples. A sample is inferred as a member if it dominates a sufficiently large fraction of population samples under this test.

#### 4.2 Experiment Setup

**Datasets.** Following previous pioneering work [28, 40, 51, 60, 74], we consider four benchmark datasets: CIFAR-10 [5], CIFAR-100 [5], Mini-ImageNet [66], and Tiny-

Table 1: Classification accuracy under original and three compression operations across different datasets and model architectures.

Dataset	Original		Pruning				Quantization		Clustering	
	Train	Test	60%	70%	80%	90%	int-8	16	8	4
CIFAR-10	99.9%	70.4%	71.6%	71.3%	71.0%	68.6%	70.4%	71.1%	70.0%	67.9%
CIFAR-100	100%	69.3%	69.3%	69.4%	69.1%	68.5%	69.3%	69.2%	68.4%	66.2%
Mini-ImageNet	91.7%	74.1%	73.8%	73.7%	73.6%	73.2%	74.1%	73.7%	72.7%	70.8%
Tiny-ImageNet	78.9%	53.9%	53.1%	53.0%	53.0%	52.9%	53.5%	52.8%	51.3%	44.2%

Model architectures: CIFAR-10 (ResNet18), CIFAR-100 (ResNet50), Mini-ImageNet (VGG16), Tiny-ImageNet (MobileNetV2). For Tiny-ImageNet, the pruning levels are set to  $L = \{40\%, 50\%, 60\%, 70\%\}$ , as higher ratio degraded model usability.

Table 2: Attack performance of different attacks on varying pruned rate (VGG16+Mini-ImageNet).

Attack Method	TPR @ 0.1% FPR (%)					Balanced Accuracy (%)					AUC (%)				
	original	60%	70%	80%	90%	original	60%	70%	80%	90%	original	60%	70%	80%	90%
CompLeak <sub>NR</sub> [60] (LR)	0.0	0.0	0.0	0.0	0.0	48.3	48.9	47.1	48.5	50.0	47.3	47.9	45.8	48.0	48.0
CompLeak <sub>NR</sub> [51] (LR)	0.0	0.0	0.0	0.0	0.0	48.3	50.8	49.6	50.0	51.6	50.1	50.7	50.2	51.0	51.4
CompLeak <sub>NR</sub> [60] (RF)	1.6	1.6	1.5	1.4	1.1	59.3	59.4	59.1	58.6	57.8	63.2	63.1	62.7	62.1	60.7
CompLeak <sub>NR</sub> [51] (RF)	1.5	1.4	1.4	1.4	1.0	59.2	59.3	59.2	58.7	57.5	63.3	63.3	62.8	62.3	60.5
CompLeak <sub>NR</sub> [73]	0.1	0.0	0.0	0.0	0.0	56.7	57.0	56.2	54.2	52.9	54.8	54.9	54.2	52.9	50.4
CompLeak <sub>NR</sub> [53]	0.1	0.1	0.1	0.1	0.1	59.6	59.8	59.1	58.7	57.3	58.7	58.8	58.2	57.1	54.7
CompLeak <sub>NR</sub> [74]	0.9	0.7	0.7	0.8	0.7	61.9	61.7	61.4	60.5	59.7	66.0	66.4	64.7	64.8	63.3
CompLeak <sub>NR</sub> (RMIA) [75]	<b>3.4</b>	3.4	3.1	2.6	2.3	<b>66.9</b>	67.4	66.8	66.3	64.6	<b>73.7</b>	73.8	73.3	72.5	70.1
CompLeak <sub>2</sub> 1 (LR)	-	11.3	11.1	10.6	8.7	-	59.5	59.7	59.3	59.4	-	67.1	67.6	66.5	66.8
CompLeak <sub>2</sub> 2 (LR)	-	8.8	8.6	8.5	8.0	-	59.7	59.9	59.9	60.3	-	68.4	68.5	68.8	69.4
CompLeak <sub>2</sub> 1 (RF)	-	<b>39.0</b>	<b>42.1</b>	<b>34.5</b>	<b>25.5</b>	-	<b>84.1</b>	<b>84.2</b>	<b>83.4</b>	<b>79.8</b>	-	93.0	92.8	91.8	88.0
CompLeak <sub>2</sub> 2 (RF)	-	36.4	41.7	34.2	24.2	-	83.9	83.9	83.2	<b>79.9</b>	-	<b>93.2</b>	<b>93.0</b>	<b>92.1</b>	<b>88.6</b>

CompLeak<sub>1</sub> is based on the first meta-data construction method, while CompLeak<sub>2</sub> is based on the second. Except RMIA, the parentheses represent the structure of the attack meta-model used.

ImageNet [38]. A detailed description of all datasets can be found in Appendix A.

**Victim Model.** We utilize four broadly adopted architectures to simulate the victim (original or compressed) model: ResNet18 [27], ResNet50 [27], VGG16 [61], and MobileNetV2 [57].

**Meta-classifier.** Following prior work [14], we adopt four widely employed binary classifiers as the attack meta-classifier: logistic regression (LR), decision tree (DT), multi-layer perceptron (MLP), and random forest (RF), to examine how the meta-classifier with varying capabilities affects MIA performance. Due to space limitations, we present only the results for LR and RF, representing the weakest and strongest meta-classifiers.

**Metric.** We consider two average-case metrics of balanced accuracy and AUC, along with the TPR @ low FPR proposed by Carlini *et al.* [11], all widely used in existing studies [28, 40, 46, 72].

• **Balanced Accuracy.** Balanced accuracy measures the probability that an MIA correctly predicts the membership status of samples in a balanced set of members and non-members.

• **AUC.** AUC is the area under the receiver operating characteristic (ROC) curve [58] indicates the average success of membership inference.

• **TPR @ low FPR.** Carlini *et al.* [11] note that high balanced accuracy/AUC are mainly due to identifying non-members. They recommend reporting TPR @ low FPR, which evaluates the true-positive rate at a low false-positive rate (e.g., 0.1% FPR), providing a more reliable measure of privacy leakage.

**Original Model Training Settings.** To mitigate model overfitting, we employed mechanisms including  $L_2$  regularization [36] and early stopping [55]. We present the training and test accuracy of the original model in Table 1. Specifically,

for training on CIFAR-10, we follow the experimental setup in [74].

**Model Compression Settings.** Pruning, quantization, and weight clustering are three compression operations widely supported by the commercial framework, which we consider. Instead of performing the three compression operations via a single toolkit only, we consider diverse toolkits and also configurations to show that the additional privacy leakage does not adhere to a specific compression toolkit.

• **Pruning.** We apply *L1 unstructured pruning* provided by Microsoft’s NNI [6]<sup>1</sup> toolkit on the original model at four sparsity levels:  $L = \{60\%, 70\%, 80\%, 90\%\}$ , which represent the removal of 60%, 70%, 80%, and 90% of the parameters with the lowest absolute values from the model [24]. In addition, we follow the standard pruning workflow, which consists of the stages: “train-prune-finetune” [8, 29, 74]. Table 1 depicts the classification accuracy of the pruned models at these different sparsity degrees. We observe that the pruned versions maintain performance comparable to the original model, with only a minimal drop in accuracy as sparsity increases. Notably, in some cases, the accuracy even shows a slight improvement [10].

• **Quantization.** We choose *QAT* supported by Facebook’s PyTorch Mobile as the typical quantization operation due to its superior performance in maintaining model accuracy. In practice, converting a float 32-bit original model to an int 8-bit quantized model is the most common setting, as int 4-bit conversion often results in significant performance degradation. For example, both PyTorch Mobile and TF-Lite offer QAT support limited to int 8-bit but do not support int 4-bit. Therefore, we limit our evaluation to converting an original model to an int 8-bit quantized version. As shown in Table 1, the accuracy of the model after QAT remains virtually unchanged.

• **Weight Clustering.** We apply weight clustering to the original model’s convolutional and linear layers. Here, TF-Lite and PyTorch are used. Specifically, we leverage the K-means algorithm to partition the weights of each layer into  $N = \{4, 8, 16\}$  clusters. Table 1 shows the clustered model’s prediction accuracy at different cluster counts, with accuracy dropping as expected as the number of clusters decreases.

### 4.3 Evaluation Results

Due to space constraints, unless otherwise specified, the results in the main text are based on the VGG16 trained on Mini-ImageNet. Evaluation results on other datasets can be found in Appendix C.

**Pruning Results.** As shown in Table 2, the performance of CompLeak<sub>NR</sub> (except for using the low-capability LR as the meta-classifier) on pruned models exhibits a declining trend as the sparsity increases. Furthermore, most CompLeak<sub>NR</sub> achieve comparable performance against pruned models with

<sup>1</sup>It has 14300 stars in GitHub.

Table 3: Attack performance of different attacks on original and quantized model (VGG16+Mini-ImageNet).

Attack Method	TPR @ 0.1% FPR (%)		Balanced Accuracy (%)		AUC (%)	
	original	int-8	original	int-8	original	int-8
CompLeak <sub>NR</sub> [60] (LR)	0.0	0.0	48.3	48.3	47.3	47.4
CompLeak <sub>NR</sub> [51] (LR)	0.0	0.0	48.3	51.1	50.1	50.5
CompLeak <sub>NR</sub> [60] (RF)	1.6	1.3	59.3	59.4	63.2	63.4
CompLeak <sub>NR</sub> [51] (RF)	1.5	1.2	59.2	59.3	63.3	63.5
CompLeak <sub>NR</sub> [73]	0.1	0.1	56.7	58.2	54.8	55.3
CompLeak <sub>NR</sub> [63]	0.1	0.1	59.6	59.6	58.7	59.1
CompLeak <sub>NR</sub> [74]	0.9	0.5	61.9	61.1	66.0	64.5
CompLeak <sub>NR</sub> (RMIA) [75]	<b>3.4</b>	3.2	<b>66.9</b>	66.4	<b>73.7</b>	72.9
CompLeak <sub>SR</sub> 1 (LR)	-	23.1	-	60.8	-	71.1
CompLeak <sub>SR</sub> 2 (LR)	-	10.0	-	59.6	-	70.8
CompLeak <sub>SR</sub> 1 (RF)	-	<b>81.0</b>	-	<b>91.1</b>	-	<b>98.3</b>
CompLeak <sub>SR</sub> 2 (RF)	-	80.7	-	90.3	-	<b>98.3</b>

Table 4: Attack performance of different attacks on varying number of clusters (VGG16+Mini-ImageNet).

Attack Method	TPR @ 0.1% FPR (%)				Balanced Accuracy (%)				AUC (%)			
	original	16	8	4	original	16	8	4	original	16	8	4
CompLeak <sub>NR</sub> [60] (LR)	0.0	0.0	0.0	0.0	48.3	49.3	47.4	51.2	47.3	47.8	48.0	48.7
CompLeak <sub>NR</sub> [51] (LR)	0.0	0.0	0.0	0.0	48.3	50.6	49.7	50.8	50.1	50.8	50.8	50.4
CompLeak <sub>NR</sub> [60] (RF)	1.6	1.4	1.2	1.4	59.3	59.1	58.5	57.9	63.2	63.2	62.4	61.7
CompLeak <sub>NR</sub> [51] (RF)	1.5	1.3	1.2	1.3	59.2	58.6	57.9	57.4	63.3	62.8	62.2	61.6
CompLeak <sub>NR</sub> [73]	0.1	0.1	0.1	0.0	56.7	56.5	56.0	54.4	54.8	54.8	54.7	53.7
CompLeak <sub>NR</sub> [63]	0.1	0.1	0.1	0.1	59.6	60.1	59.6	59.3	58.7	58.9	58.6	58.3
CompLeak <sub>NR</sub> [74]	0.9	1.3	0.6	0.7	61.9	61.7	61.5	61.2	66.0	64.3	64.1	62.9
CompLeak <sub>NR</sub> (RMIA) [75]	<b>3.4</b>	3.2	3.1	3.1	<b>66.9</b>	66.0	65.5	65.4	<b>73.7</b>	72.5	72.6	71.5
CompLeak <sub>SR</sub> 1 (LR)	-	16.4	11.3	8.1	-	60.4	60.1	59.3	-	69.6	68.3	65.0
CompLeak <sub>SR</sub> 2 (LR)	-	10.0	8.1	6.2	-	59.9	59.9	60.2	-	70.8	68.4	67.3
CompLeak <sub>SR</sub> 1 (RF)	-	<b>67.2</b>	<b>47.3</b>	<b>28.5</b>	-	<b>93.2</b>	89.7	84.1	-	<b>98.7</b>	96.8	92.4
CompLeak <sub>SR</sub> 2 (RF)	-	66.3	46.8	28.5	-	<b>93.2</b>	<b>90.1</b>	<b>84.5</b>	-	<b>98.7</b>	<b>97.0</b>	<b>92.8</b>

lower sparsity levels (e.g., 60%, 70%) to their performance on the original model, but their effectiveness reveals a marked decrease when targeting the highly pruned model (e.g., 90%). We attribute these to the reduced model capacity, which limits its ability to capture members’ details, while the generalization on non-members remains largely unchanged, making the behavior of members more similar to non-members.

**Quantization Results.** As detailed in Table 3, similar to the attack results on low sparsity pruned models, most CompLeak<sub>NR</sub> exhibit nearly identical attack performance between the 8-bit quantized model and the original model, with the balanced accuracy difference predominantly below 1.5%.

**Weight Clustering Results.** Similar to pruning, we observe that the attack performance of CompLeak<sub>NR</sub> declines with fewer clusters due to the increasing similarity between member and non-member behavior as model capacity reduces. Specifically, when the number of clusters is 4, CompLeak<sub>NR</sub> shows lower privacy leakage on the clustered model compared to the original one, and when there are 16 clusters, the leakage is nearly identical.

**Summary.** The above results indicate that solely relying on the relationship between members and non-members of the underlying model, without any reference information, the privacy leakage in the compressed model is, in most cases, comparable to that in the original model. Surprisingly, the highly compressed version is less vulnerable to CompLeak<sub>NR</sub>.

## 5 CompLeak<sub>SR</sub>

After quantifying the membership leakage of the compressed model through CompLeak<sub>NR</sub>, which utilizes information from only a target compressed version, in this section, we focus on

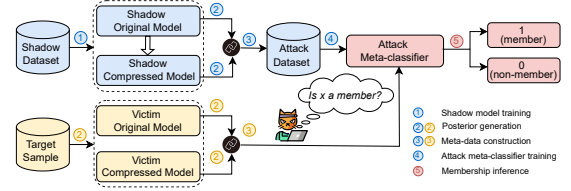


Figure 3: Attack pipeline overview of CompLeak<sub>SR</sub>.

the privacy leakage due to the compression operation. This is achieved through CompLeak<sub>SR</sub>, in which the core idea is to treat the single compressed model as a reference, consistently pairing it with the corresponding original model, to capture the variations caused by the compression operation, i.e., the impact of the compression operation on posteriors. Next, we present the design insight, the detailed pipeline of CompLeak<sub>SR</sub>, evaluation results, and the discussion.

### 5.1 Design Insight

We hypothesize that the compression operation affects members and non-members differently, although it is not obvious if we only look at the overall MIA accuracy given a specific compressed model version, as we showed in the CompLeak<sub>NR</sub>. To prove our hypothesis, we utilize the KL divergence [37] to visualize the distance between the two posteriors on the same target samples, one obtained from the original model and the other from one of its paired compressed models. As shown in Figure 2, when 40% of the parameters are pruned from MobileNetV2 on the Tiny-ImageNet, the influence of the compression operation on posteriors (i.e., the changes after compression) is more noticeable for members than non-members. Essentially, CompLeak<sub>NR</sub> [51] on the pruned MobileNetV2 exhibits a 1.3% lower balanced accuracy compared to the original model. This is because the reduced model capacity, compared to the uncompressed model, is unable to capture the fine-grained features of members. However, the change in generalization is relatively minor, leading to a smaller impact on non-members.

### 5.2 Attack Methodology

Building on the above findings, we propose CompLeak<sub>SR</sub> specifically for the single-compression scenario, where the key principle is to take a single compressed model as the reference, with the paired model always being the original model. More specifically, we combine a pair of posteriors from a single compressed model and the original model as meta-data through *meta-data construction* to train the attack meta-classifier for membership inference, capturing the different influences on members and non-members in two aspects: the inherent discrepancy in each posterior, as reflected by CompLeak<sub>NR</sub> [51, 60, 63, 73–75], and the differences be-

tween the two posteriors, which reflect the differential changes caused by the compression operation.

CompLeak<sub>SR</sub> attack consists of five stages as shown in Figure 3: shadow model training, posterior generation, meta-data construction, attack meta-classifier training, and attack meta-classifier membership inference. The first five stages are performed once during offline, while the last stage corresponds to the online phase.

**Shadow Model Training.** As mentioned in Section 3, the adversary possesses a shadow dataset  $\mathcal{D}_s$ , which is divided into two disjoint subsets: the shadow train set  $\mathcal{D}_{\text{train}}^s$ , and the shadow test set  $\mathcal{D}_{\text{test}}^s$ . The adversary trains a shadow original model  $\mathcal{M}_o^s$  on  $\mathcal{D}_{\text{train}}^s$ , then subjects it to a compression algorithm to produce a shadow compressed model  $\mathcal{M}_c^s$  as the shadow reference model.

**Posterior Generation.** The adversary queries both  $\mathcal{M}_o^s$  and  $\mathcal{M}_c^s$  for each data sample from  $\mathcal{D}_s$ , obtaining one pair of posteriors as  $\mathcal{P}_o^s$  and  $\mathcal{P}_c^s$ , respectively.

**Meta-data Construction.** Inspired by and following [14], we use *meta-data construction* to obtain meta-data and provide two construction methods here. In order to better understand posteriors, we first need to sort  $\mathcal{P}_o^s$  in descending order and apply this order to  $\mathcal{P}_c^s$ , obtaining  $\mathcal{P}_o^{s'}$  and  $\mathcal{P}_c^{s'}$ , respectively [14].

- The first method is to directly concatenate  $\mathcal{P}_o^{s'}$  and  $\mathcal{P}_c^{s'}$  as meta-data, i.e.,  $\mathcal{P}_o^{s'} \parallel \mathcal{P}_c^{s'}$ , where  $\parallel$  denotes the concatenation operation.
- Since the adversary can access the ground truth label of the audited sample, it has been shown in previous works that the divergence between members and non-members varies in a fine-grained manner across different classes [51, 74]. Thus, the second method is to apply one-hot encoding on the ground truth label to generate  $y$ , and then concatenate it with  $\mathcal{P}_o^{s'}$  and  $\mathcal{P}_c^{s'}$  as meta-data, i.e.,  $\mathcal{P}_o^{s'} \parallel \mathcal{P}_c^{s'} \parallel y$ .

Note that additional meta-data construction methods (i.e., direct concatenation) have been considered. Their details and corresponding attack performance can be found in Appendix B.

**Attack Meta-classifier Training.** The adversary labels the meta-data  $X_a$  as 1 if it originates from  $\mathcal{D}_{\text{train}}^s$ , and as 0 from  $\mathcal{D}_{\text{test}}^s$ . These labels, denoted as  $Y_a$ , along with  $X_a$ , constitute the attack training dataset used to train the attack meta-classifier  $\mathcal{M}_{\text{SR}}$ , a binary classifier for membership inference. During training, binary cross-entropy loss is applied to compute the loss, with the objective of minimizing  $\mathcal{L}(\mathcal{M}_{\text{SR}}(X_a, Y_a))$ .

**Attack Meta-classifier Membership Inference.** Once  $\mathcal{M}_{\text{SR}}$  is trained, the adversary can determine the membership of a given target sample. To achieve this, the adversary queries both the victim’s original model and a single compressed model to obtain paired posteriors, denoted as  $\mathcal{P}_o^v$  and  $\mathcal{P}_c^v$ . These paired posteriors are then processed through a meta-data construction step and fed into  $\mathcal{M}_{\text{SR}}$ . If  $\mathcal{M}_{\text{SR}}$  outputs 1, the target sample is regarded as a member; otherwise, it is classified as a non-member.

Finally, it is noteworthy that other MIA methods that utilize

the full posterior distribution can be adapted in a manner akin to CompLeak<sub>SR</sub>, those relying on a single predicted probability for the ground truth class (e.g., LiRA [11] or RMIA [75]) cannot be modified in the same way.

### 5.3 Evaluation Results

**Pruning Results.** As shown in Table 2, regardless of the pruning degree, CompLeak<sub>SR</sub> consistently surpasses all CompLeak<sub>NR</sub> attacks (including CompLeak<sub>NR</sub> (RMIA)) on the original model, providing compelling evidence that pruning operations leak additional privacy. For instance, the AUC of CompLeak<sub>SR</sub> on the 60%-pruned VGG16 is 93.2%, while the best CompLeak<sub>NR</sub> (with RMIA) achieves an AUC of 73.7% on the original model. Notably, CompLeak<sub>SR</sub> also exhibits pronounced effectiveness in pruning scenarios. For example, when targeting the 90%-pruned VGG16 on Mini-ImageNet, CompLeak<sub>SR</sub> improves the best TPR @0.1% FPR from 2.3% to 25.5%, best AUC from 70.1% to 88.6%, and best balanced accuracy from 64.6% to 79.9%.

**Quantization Results.** As detailed in Table 3, CompLeak<sub>SR</sub> leverages information from both original and quantized models, outperforming all CompLeak<sub>NR</sub>’s attacks (including CompLeak<sub>NR</sub> (RMIA)) performance on the original or quantized model, thereby highlighting that quantization operations indeed amplify privacy risks and demonstrating the effectiveness of our CompLeak<sub>SR</sub>. For instance, on the Mini-ImageNet, compared to the best-performing CompLeak<sub>NR</sub> (with RMIA) on the original model (quantized model), CompLeak<sub>SR</sub> achieves a significant improvement of 24.6% (25.4%) in AUC and 24.2% (24.7%) in balanced accuracy. In addition, we also observe that the privacy leakage induced by int 8-bit QAT is substantially higher than  $L1$  unstructured pruning, i.e., at TPR @ 0.1% FPR, the leakage is 38.9% more than 70%-pruned VGG16.

**Weight Clustering Results.** Similar to pruning, although the attack performance of CompLeak<sub>SR</sub> decreases with fewer clusters due to the proximity of member and non-member behavior as model capacity diminishes, it still outperforms all CompLeak<sub>NR</sub> attacks (including CompLeak<sub>NR</sub> (RMIA)) on the original model (or clustered model), regardless of the number of clusters, highlighting that weight clustering imposes additional privacy leakage and the effectiveness of CompLeak<sub>SR</sub>. Specifically, for the TPR @ 0.1% FPR shown in Table 4, CompLeak<sub>SR</sub> presents an order of magnitude improvement compared to CompLeak<sub>NR</sub> on the original or clustered model.

### 5.4 Discussion

In this subsection, we start by analyzing the characteristics of samples that become vulnerable to MIA after compression, and then provide the results of CompLeak<sub>SR</sub> against MIA defense using DP-SGD [7].



Figure 4: Number of members and non-members that transitioned from attack failure to success after compression. For Mini-ImageNet, the number of members and non-members is 8000, and for CIFAR-10, it is 13500. The pruning level is 0.9, the number of clusters is 8.

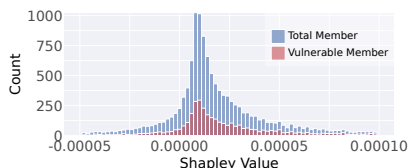


Figure 5: KNN-Shapley values of total member and vulnerable member in pruning scenario of CIFAR-10.

**Which Data Samples are Vulnerable?** In the above experiments, we have confirmed that pruning, quantization, and weight clustering all lead to an increased risk of privacy leakage. Here, we further identify which samples amplify this leakage post-compression and analyze their characteristics.

To begin with, we focus on samples that transitioned from attack inference failure using  $\text{CompLeak}_{\text{NR}}$  on the original model (measured by [51]), to success after compression when employing  $\text{CompLeak}_{\text{SR}}$ . Firstly, as revealed in Figure 4, most of these samples are members. Since members are more private than non-members, this exposes a critical vulnerability in model compression that enables adversaries to extract more sensitive information. Secondly, to assess the importance of these vulnerable member samples within the entire member set, we followed the approach outlined in [68], utilizing KNN-Shapley [34] to compute the importance of both all members and vulnerable members. As larger Shapley values represent higher importance, Figure 5 illustrates that these vulnerable members are relatively more important within the overall member set. These critical findings highlight that compression not only increases the number of member samples effectively inferred but also amplifies the exposure of high-value members.

**Attack Against DP-SGD.** Differential privacy [20] is a widely adopted defense mechanism for mitigating privacy leakage risks. Following [40], we implement DP-SGD via the Opacus toolkit with privacy parameters ( $\delta = 1e-5$ ,  $C = 1$ ). However, stronger defenses typically lead to a significant drop in model utility, which could be unacceptable in practice. To carefully consider the trade-off between the defense level and model accuracy, we choose two noise multipliers

Table 5: Attack performance on  $\mathcal{D}_f/\mathcal{D}_{n_f}$  with different  $N_f$  (80%-pruned ResNet18+CIFAR-10).

Attack	Balanced Accuracy (%)			AUC (%)		
	90%	50%	10%	90%	50%	10%
$\text{CompLeak}_{\text{NR}}$ [60]	59.2/57.2	59.6/55.6	61.8/56.6	62.3/60.0	62.9/58.2	65.8/58.9
$\text{CompLeak}_{\text{NR}}$ [51]	59.6/57.4	60.2/56.0	62.1/56.8	62.9/60.5	63.3/59.0	66.2/59.6
$\text{CompLeak}_{\text{SR}}$ 2	61.8/61.4	62.2/60.9	64.2/61.0	65.4/65.0	65.7/64.6	67.9/64.8

$\sigma \in \{0.2, 0.5\}$ , where a larger  $\sigma$  provides stronger protection. Table 16 presents the attack performance under DP-SGD, evaluated on 70%-pruned ResNet18 on Mini-ImageNet.

We observe that DP-SGD offers an effective defense against all MIAs, with attack performance gradually decreasing as defense strength increases. For example, for  $\text{CompLeak}_{\text{SR}}$ , the balanced accuracy drops from 86.5% (no defense) to 73.4% ( $\sigma=0.2$ ) and 71.6% ( $\sigma=0.5$ ). Notably,  $\text{CompLeak}_{\text{SR}}$  consistently achieves the best attack performance, compared to  $\text{CompLeak}_{\text{NR}}$  targeting either the original or compressed model. This highlights that the compression operation still leads to additional leakage even after deploying the DP-SGD.

## 5.5 Ablation Study

We conduct experiments to examine the influence of several key factors on  $\text{CompLeak}_{\text{SR}}$  performance. Because of limited space, an analysis of dataset effects and the overfitting level is provided in Appendix E.

**Amount of Data Used for Fine-tuning.** Fine-tuning the compressed model using the original training dataset is a crucial step in many compression operations. In the experiments above, we utilized the entire training dataset for fine-tuning. However, it is more practical to fine-tune the model using a subset of the original training dataset, as this can considerably reduce fine-tuning time. This subset, denoted as  $\mathcal{D}_f$  ( $|\mathcal{D}_f| = N_f$ ), typically contains more valuable and sensitive data, while the remainder of the original training dataset, not used for fine-tuning, is denoted as  $\mathcal{D}_{n_f}$ . Then, we explore the impact of  $N_f$  by fine-tuning the compressed model using three portions (90%, 50%, and 10%) of the original training dataset and measuring the attack performance on both  $\mathcal{D}_f$  and  $\mathcal{D}_{n_f}$ .

Table 5 reports that the privacy leakage for  $\mathcal{D}_f$  is higher than for  $\mathcal{D}_{n_f}$ , and this gap becomes more pronounced as  $N_f$  decreases. This stems from the exclusion of  $\mathcal{D}_{n_f}$  during fine-tuning, which reduces model accuracy on this dataset and causes its behavior to resemble that of non-members. Moreover, the MIA performance on  $\mathcal{D}_f$  gradually enhances as  $N_f$  decreases. We hypothesize that this occurs because, as  $N_f$  decreases, the model learns more refined features from  $\mathcal{D}_f$ , while its generalization capability to non-members deteriorates, leading to an increasing disparity between  $\mathcal{D}_f$  and non-members. Thus, fine-tuning using a subset of the training dataset exacerbates privacy leakage risks for the more valuable data used during fine-tuning.

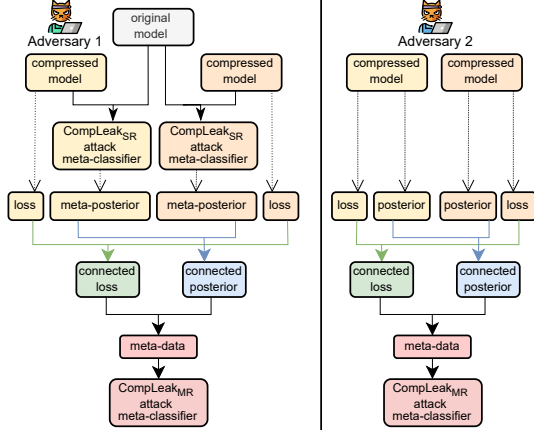


Figure 6: General attack pipeline of  $\text{CompLeak}_{\text{MR}}$ , using two compressed models for example.

## 6 $\text{CompLeak}_{\text{MR}}$

Through  $\text{CompLeak}_{\text{SR}}$ , which leverages a single compressed model as a reference, we conclude that compression operations indeed lead to additional privacy leakage. More interestingly, in this section, based on the intuition that different compressed models leak privacy in slightly different ways, we demonstrate that when multiple compressed models are utilized as references—referred to as  $\text{CompLeak}_{\text{MR}}$ —the privacy leakage induced by compression is further exacerbated. We begin by considering two distinct adversarial settings, whether the original model is accessible or not. Then, we present the attack methodology, evaluation results, and conclude with ablation studies.

### 6.1 Adversarial Knowledge

Here, we assume two different threat models, gradually limiting the adversary’s knowledge to show the broader attack scenarios.

**Adversary 1:** Consistent with the threat model described in Section 3, the adversary is assumed to have black-box access to the original model as well as its multiple compressed versions. Moreover, they also know the sparsity level or model size associated with each compressed model.

**Adversary 2:** Compared to Adversary 1, Adversary 2 adopts a stricter threat model. Following the setup in [74], we assume that the adversary has black-box access only to multiple compressed models, with *no access* to the original model. All other conditions are the same as Adversary 1.

### 6.2 Attack Methodology

When the reference model is expanded from a single compressed model to multiple compressed models, we observe that different compressed versions leak privacy differently,

revealed by both the loss and the membership inference results from the  $\text{CompLeak}_{\text{SR}}$  attack meta-classifier. Therefore, to implement  $\text{CompLeak}_{\text{MR}}$ , the adversary aggregates the leaked information by extracting and concatenating the losses and  $\text{CompLeak}_{\text{SR}}$  meta-posteriors through each compressed model and  $\text{CompLeak}_{\text{SR}}$  attack meta-classifiers separately, then stacks them to train an MLP-based  $\text{CompLeak}_{\text{MR}}$  attack meta-classifier for membership inference.

Similar to  $\text{CompLeak}_{\text{SR}}$ , performing  $\text{CompLeak}_{\text{MR}}$  involves five key stages: shadow model training, loss concatenation, posterior concatenation, attack meta-classifier training, and attack meta-classifier membership inference. We present the detailed attack pipeline of  $\text{CompLeak}_{\text{MR}}$  in Figure 6. The following description is based on the scenario of Adversary 1, any differences for Adversary 2 will be specified separately.

**Shadow Model Training.** In the context of a multi-compressed model scenario, there are three types of shadow models. More concretely, the adversary first trains a shadow original model  $\mathcal{M}_o^s$  on the shadow training set  $\mathcal{D}_{\text{train}}^s$ , then applies the compression algorithm to generate the corresponding shadow compressed models  $\mathcal{M}_{c_1}^s, \mathcal{M}_{c_2}^s, \dots, \mathcal{M}_{c_n}^s$ . Finally, the adversary conducts  $\text{CompLeak}_{\text{SR}}$  on each shadow compressed model, training a set of shadow  $\text{CompLeak}_{\text{SR}}$  attack meta-classifiers  $\mathcal{M}_{\text{SR}_1}^s, \mathcal{M}_{\text{SR}_2}^s, \dots, \mathcal{M}_{\text{SR}_n}^s$ . Notably, Adversary 2 is unable to get  $\mathcal{M}_{\text{SR}_i}^s$ .

**Posterior Concatenation.** We first present the design rationale, followed by the detailed implementation.

*Design Rationale.* Building on  $\text{CompLeak}_{\text{SR}}$ , we observe that the membership inference results from  $\mathcal{M}_{\text{SR}_1}^s, \mathcal{M}_{\text{SR}_2}^s, \dots, \mathcal{M}_{\text{SR}_n}^s$  on the same target sample exhibit striking differences. For example, when the  $\mathcal{M}_{\text{SR}_i}^s$  are all based on a random forest structure, the inference results targeting an 80%-pruned VGG16 on Mini-ImageNet differ by approximately 22% from targeting a 90%-pruned VGG16. This can be explained by different compression degrees exerting varying influences on the same sample, and through  $\mathcal{M}_{\text{SR}_1}^s, \mathcal{M}_{\text{SR}_2}^s, \dots, \mathcal{M}_{\text{SR}_n}^s$ , we can effectively capture these subtle differences.

*Implementation.* Due to the differing knowledge of two adversaries, the methods for obtaining the posteriors also differ.

- Adversary 1. Because the original model is accessible under this assumption, the adversary can train  $\mathcal{M}_{\text{SR}_1}^s, \mathcal{M}_{\text{SR}_2}^s, \dots, \mathcal{M}_{\text{SR}_n}^s$  through  $\text{CompLeak}_{\text{SR}}$  during the Shadow Model Training step. Based on design rationale, the adversary queries  $\text{CompLeak}_{\text{SR}}$  meta-classifiers  $\mathcal{M}_{\text{SR}_1}^s, \mathcal{M}_{\text{SR}_2}^s, \dots, \mathcal{M}_{\text{SR}_n}^s$  for each sample from  $\mathcal{D}_s$  to get  $\text{CompLeak}_{\text{SR}}$  meta-posteriors  $\mathcal{P}_{\text{SR}_1}^s, \mathcal{P}_{\text{SR}_2}^s, \dots, \mathcal{P}_{\text{SR}_n}^s$ .

- Adversary 2. Due to the lack of access to the original model, the adversary is unable to acquire  $\mathcal{M}_{\text{SR}_1}^s, \mathcal{M}_{\text{SR}_2}^s, \dots, \mathcal{M}_{\text{SR}_n}^s$ . Instead, the posteriors here are derived by directly querying  $\mathcal{M}_{c_1}^s, \mathcal{M}_{c_2}^s, \dots, \mathcal{M}_{c_n}^s$  for each sample from  $\mathcal{D}_s$ , denoted as  $\mathcal{P}_{c_1}^s, \mathcal{P}_{c_2}^s, \dots, \mathcal{P}_{c_n}^s$ .

Finally, the adversary performs the concatenation oper-

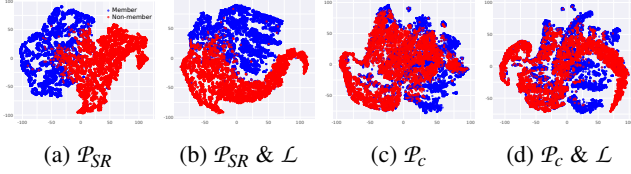


Figure 7: Using t-SNE to visualize the four distributions.  $\mathcal{P}_{SR}$ ,  $\mathcal{P}_c$  refers to posterior concatenation for Adversary 1 and Adversary 2, and  $\mathcal{L}$  refers to loss concatenation.

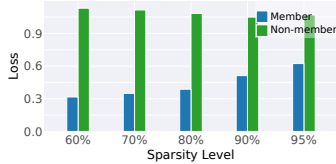


Figure 8: Average loss of members and non-members under varying sparsity levels for the VGG16 pruned on the Mini-ImageNet.

ation on  $\mathcal{P}_{SR_1}^s, \mathcal{P}_{SR_2}^s, \dots, \mathcal{P}_{SR_n}^s$  ( $\mathcal{P}_{c_1}^s, \mathcal{P}_{c_2}^s, \dots, \mathcal{P}_{c_n}^s$ ) to generate  $\mathcal{P}_{SR}^s$  ( $\mathcal{P}_c^s$ ), i.e.,  $\mathcal{P}_{SR}^s = \mathcal{P}_{SR_1}^s \parallel \mathcal{P}_{SR_2}^s \parallel \dots \parallel \mathcal{P}_{SR_n}^s$ . We visualize  $\mathcal{P}_{SR}^s$  using t-SNE [65] in Figure 7a, a clear boundary is observed between members and non-members compared to  $\mathcal{P}_c^s$  in Figure 7c.

**Loss Concatenation.** We begin with the design rationale and then describe the implementation.

*Design Rationale.* We find that as the compression degree increases, the evolution of loss calculated from compressed models on the same target sample reveals disparities between members and non-members, both in direction and magnitude. For clarity, we use pruning as an example to explain. Specifically, as illustrated in Figure 8, the loss for members rises with higher sparsity levels, while the loss for non-members fluctuates. This can be attributed to the sparse double descent found in [30], as model capacity decreases, non-member accuracy declines while member accuracy remains stable, however, after a critical sparsity point, member accuracy drops and non-member accuracy rises before declining again.

*Implementation.* The adversary feeds each data sample from  $\mathcal{D}_s$  to  $\mathcal{M}_{c_1}^s, \mathcal{M}_{c_2}^s, \dots, \mathcal{M}_{c_n}^s$ , calculating a series of loss values  $\mathcal{L}_1^s, \mathcal{L}_2^s, \dots, \mathcal{L}_n^s$ . The loss  $\mathcal{L}_i^s$  for each shadow compressed model  $\mathcal{M}_{c_i}^s$  is calculated through the cross-entropy:

$$\mathcal{L}_i^s = -\sum_{k=1}^C y_k \log(p_k^{(i)}).$$
 Where  $C$  is the number of classes,  $y_k$  is the true label for class  $k$  (one-hot encoded), and  $p_k^{(i)}$  is the predicted posterior probability for class  $k$  generated from the model  $\mathcal{M}_{c_i}^s$ . Finally, the adversary concatenates each loss to obtain  $\mathcal{L}_s$ , i.e.,  $\mathcal{L}_s = \mathcal{L}_1^s \parallel \mathcal{L}_2^s \parallel \dots \parallel \mathcal{L}_n^s$ . As illustrated in Figures 7b and 7d, incorporating the  $\mathcal{L}_s$ —whether for  $\mathcal{P}_{SR}^s$  or  $\mathcal{P}_c^s$ —further enhances the ability to distinguish between members and non-members.

**Attack Meta-classifier Training.** The adversary constructs

Table 6: Average performance of  $\text{CompLeak}_{MR}$  on Mini-ImageNet under Adversary 1 (five repetitions).

Operation	TPR @ 0.1% FPR (%)	Balanced Accuracy (%)	AUC (%)
Pruning	61.3 (19.2% $\uparrow$ )	94.8 (10.6% $\uparrow$ )	98.9 (5.7% $\uparrow$ )
Clustering	72.7 (5.5% $\uparrow$ )	95.1 (1.9% $\uparrow$ )	99.0 (0.3% $\uparrow$ )
Three Operations	95.7 (14.7% $\uparrow$ )	98.8 (5.6% $\uparrow$ )	99.9 (1.2% $\uparrow$ )

In parentheses is the improvement over the optimal result when  $\text{CompLeak}_{SR}$  attacking a single compressed model in the respective compression scenario.

Table 7: Average performance of  $\text{CompLeak}_{MR}$  on Mini-ImageNet under Adversary 2 (five repetitions).

Operation	TPR @ 0.1% FPR (%)	Balanced Accuracy (%)	AUC (%)
Pruning	2.6 (1.8% $\uparrow$ )	71.3 (9.6% $\uparrow$ )	77.6 (11.2% $\uparrow$ )
Clustering	1.1 (0.2% $\downarrow$ )	65.6 (3.9% $\uparrow$ )	70.6 (6.3% $\uparrow$ )
Three Operations	8.8 (7.5% $\uparrow$ )	71.4 (9.7% $\uparrow$ )	80.6 (14.2% $\uparrow$ )

In parentheses is the improvement over the optimal result when SAMIA [74] attacking a single compressed model in the respective compression scenario.

the attack binary training dataset  $\mathcal{D}_{train}^{attack}$  by stacking  $\mathcal{L}_s$  and  $\mathcal{P}_{SR}^s$  (or  $\mathcal{P}_c^s$  for Adversary 2), labeling the stacked data as 1 (member) if it comes from  $\mathcal{D}_{train}^s$ , and 0 (non-member) otherwise. Subsequently, the adversary trains an MLP-based  $\text{CompLeak}_{MR}$  meta-classifier  $\mathcal{M}_{MR}$  on  $\mathcal{D}_{train}^{attack}$  to perform membership inference.

**Attack Meta-classifier Membership Inference.** The attacker can ultimately conduct MIA on each given target sample following these steps: First, the adversary performs  $\text{CompLeak}_{SR}$  on each victim’s compressed models to train the victim  $\text{CompLeak}_{SR}$  attack meta-classifiers  $\mathcal{M}_{SR_1}^v, \mathcal{M}_{SR_2}^v, \dots, \mathcal{M}_{SR_n}^v$  (for Adversary 2, this step is not required). Then, the target sample is subjected to loss concatenation and posterior concatenation to obtain  $\mathcal{L}_v$  and  $\mathcal{P}_{SR}^v$  (or  $\mathcal{P}_c^v$  for Adversary 2), respectively. Finally, the adversary stacks  $\mathcal{L}_v$  and  $\mathcal{P}_{SR}^v$  (or  $\mathcal{P}_c^v$ ), feeding them into  $\mathcal{M}_{MR}$  to predict the sample’s membership status, i.e., 1 (member) or 0 (non-member).

## 6.3 Evaluation

**Experimental Setup.** For the selection of compression degrees in multiple compressed models, we choose pruned models with sparsity levels  $L = \{60\%, 70\%, 80\%, 90\%\}$ , clustered models with cluster centers  $N = \{4, 8, 16\}$ .

**Results for Adversary 1.** Table 6 depicts the performance of  $\text{CompLeak}_{MR}$  (Adversary 1) and highlights the improvements achieved compared to the best performance of  $\text{CompLeak}_{SR}$  on a single compressed model. Encouragingly, we observe that the  $\text{CompLeak}_{MR}$  clearly demonstrates superior performance compared to  $\text{CompLeak}_{SR}$  in TPR @ 0.1% FPR. Notably, when multiple compressed models are derived from three compression operations (eight models in total), all evaluation metrics show exceptionally high values, with the AUC approaching 100%. These results provide strong evidence that multiple compressed models leak significantly more information compared to a single compressed model.

**Results for Adversary 2.** As previously mentioned, Adver-

Table 8: Attack performance of  $\text{CompLeak}_{\text{MR}}$  with different numbers of pruned models on the Mini-ImageNet.

Number	TPR @ 0.1% FPR (%)	Balanced Accuracy (%)	AUC (%)
$L = \{60\%$	40.2	85.8	93.9
$L = \{60\%, 70\%$	44.8	92.9	98.0
$L = \{60\%, 70\%, 80\%$	54.0	93.6	98.3
$L = \{60\%, 70\%, 80\%, 90\%$	61.3	94.8	98.9

sary 2, without access to the original model, inherently leads to lower  $\text{CompLeak}_{\text{MR}}$  attack performance compared to Adversary 1. Here, we compare with SAMIA [74]—the most relevant prior work to ours, which specifically focuses on pruning—in  $\text{CompLeak}_{\text{NR}}$ , as  $\text{CompLeak}_{\text{NR}}$  also avoids using the information from the original model. Table 7 presents the performance improvement of  $\text{CompLeak}_{\text{MR}}$  relative to SAMIA’s optimal performance on a single compressed model. Notably,  $\text{CompLeak}_{\text{MR}}$  still maintains significant advantages by leveraging multiple compressed models. For example, when all three compression operations are applied, there is a 9.7% increase in balanced accuracy, and a 14.2% enhancement in AUC. This result further emphasizes the superior capability of  $\text{CompLeak}_{\text{MR}}$  in effectively exploiting multiple compressed models to amplify privacy leakage.

## 6.4 Ablation Study

**Impact of the Number of Compressed Models.** We conduct  $\text{CompLeak}_{\text{MR}}$  (Adversary 1) utilizing 1, 2, 3, and 4 pruned models, respectively, to systematically assess the influence of the number of pruned models on overall attack performance. As shown in Table 8, the attack performance improves with more pruned models, aligning with the expectation that additional pruned models provide more exploitable leakage information. For example, using two pruned models increases balanced accuracy by 7.1% compared to one pruned model.

Given space limitations, the detailed performance of individual components is presented in Appendix F.

## 7 Discussion

**Compression vs Duplication.** To further illustrate that the superiority of  $\text{CompLeak}_{\text{SR}}$  and  $\text{CompLeak}_{\text{MR}}$  primarily stems from privacy leakage caused by the model compression, rather than simply relying on aggregating leaked information from duplicated models to enhance MIA performance, we conducted a baseline experiment. Specifically, we use the uncompressed model from previous experiments as the original target model. To simulate a duplicated multi-model setting, we instantiate four different versions of the model with the same architecture but different training hyperparameter settings (e.g., learning rate, training epochs, and random seed). Each of the different settings creates a duplicated model with almost the same testing accuracy as the original model.

As shown in Table 17, although  $\text{CompLeak}_{\text{SR}}$  outperforms

Table 9: The attack results of BERT-base/GPT-2 finetuned on SST-5.

Attack Method	TPR @ 0.1% FPR (%)			Balanced Accuracy (%)			AUC (%)		
	original	pruned	quantized	original	pruned	quantized	original	pruned	quantized
$\text{CompLeak}_{\text{NR}}$ [51] (RF)	1.1/1.1	0.6/1.0	0.4/0.8	83.7/77.0	80.5/74.7	62.4/72.6	88.7/85.0	87.4/82.5	66.7/79.9
$\text{CompLeak}_{\text{NR}}$ [73]	0.3/1.8	0.6/1.1	0.2/1.0	69.1/67.4	66.8/71.0	50.3/61.4	74.8/75.4	84.8/80.1	50.9/73.0
$\text{CompLeak}_{\text{NR}}$ [63]	0.3/2.9	1.0/1.6	0.3/0.8	81.8/78.6	77.6/74.5	63.4/71.4	85.7/85.7	91.5/87.5	68.5/84.5
$\text{CompLeak}_{\text{NR}}$ [74]	0.9/1.6	0.5/0.6	0.1/0.4	80.3/79.6	78.4/78.1	63.8/73.7	85.0/85.6	81.0/81.7	68.0/81.7
$\text{CompLeak}_{\text{MR}}$ 2 (LR)	-	0.6/2.4	0.9/1.9	-	85.6/80.7	75.6/76.7	-	91.8/87.7	81.9/84.5
$\text{CompLeak}_{\text{MR}}$ 2 (RF)	-	1.6/0.9	1.1/1.5	-	84.7/76.6	78.8/71.7	-	91.0/84.4	85.2/82.9

$\text{CompLeak}_{\text{NR}}$ , there remains a significant gap compared to its performance on compression setting. For example, Table 3 exhibits that  $\text{CompLeak}_{\text{SR}}$  achieves up to 91.1% attack accuracy on a quantized model, whereas its best performance under the baseline reaches only 72.7%. This is because, while the baseline introduces output variations between the original target model and reference models due to training hyperparameter differences, such differences are relatively small and lack consistent patterns. In contrast, model compression substantially alters the model’s capacity and representational behavior, leading to more pronounced and structured differences in outputs, thereby providing stronger discriminative signals for MIA. Furthermore, we construct  $\text{CompLeak}_{\text{MR}}$  using four different reference models and observe only marginal improvement over  $\text{CompLeak}_{\text{SR}}$ . Thus, these results indicate that the advantage of  $\text{CompLeak}$  primarily stems from compression-induced privacy leakage, rather than the accumulation of minor variations across multiple duplicated models.

**Reasons for Less Effectiveness of RMIA.** We extend RMIA to the  $\text{CompLeak}_{\text{SR}}$  to examine its effectiveness by multiplying the pairwise likelihood ratios obtained from the original model and the compressed model. This is based on the intuition that members tend to exhibit high fit in the original model and maintain high confidence after compression due to retraining-induced memorization, whereas non-members’ spurious confidence may be suppressed by compression. When attacking the 70%-pruned VGG16 on Mini-ImageNet,  $\text{CompLeak}_{\text{SR}}$  (RMIA) achieves an accuracy of 67.1%, substantially lower than the 84.2% attained by our method using full softmax information. This gap arises because RMIA relies solely on true-class confidence and therefore fails to capture more subtle compression-induced membership information. Consequently, RMIA is less effective and thus was not selected for comprehensive evaluation in our work.

**Attack Against Mid-sized Foundation Models for Classification Tasks.** We quantify the membership leakage during the high-risk fine-tuning phase of mid-sized foundation models [78], which raises unique concerns due to the processing of sensitive proprietary data, in contrast to the pre-training phase operates on public corpora. Furthermore, we focus on the two most commonly studied compression operations for foundation models: pruning and quantization. The detailed settings are as follows:

*Pruning.* We utilize a simple and effective *layer pruning* [49] tailored for foundation models. As discussed in Section 2, we

assess the importance of each layer using block influence and discard the less important ones [49].

*Quantization.* High-performing QAT typically requires additional fine-tuning, which can be computationally expensive for foundation models because of their massive parameters. Consequently, in this work, we evaluate based on PyTorch’s *dynamic quantization*—a form of post-training quantization—eliminating fine-tuning and directly quantizing the foundation models into int 8-bit format.

In our experiments, we fine-tune BERT-base and GPT-2 for 30 epochs on downstream classification tasks SST-5 [62], treating them as boundary cases between classical classification models – where MIAs are known to succeed – and large-scale LLMs – where models are less vulnerable to MIA. To consider a balance between model accuracy and compression, we remove six unimportant layers from BERT-base and one from GPT-2, and apply dynamic quantization to convert the fine-tuned model to 8-bit integer precision. Figure 9 presents the prediction accuracy of pruned and quantized BERT/GPT-2. While BERT exhibits a notable accuracy drop after quantization, other configurations maintain comparable performance to the original model, with only marginal declines. Importantly, as shown in Table 9, pruning sometimes weakens  $\text{CompLeak}_{\text{NR}}$ ’s effectiveness, whereas our  $\text{CompLeak}_{\text{SR}}$  still achieves the highest performance. This suggests that removing redundant layers in the foundation model inadvertently leaks privacy. In addition, quantization degrades all MIA capabilities, particularly for  $\text{CompLeak}_{\text{NR}}$ . We attribute this to the absence of retraining after dynamic quantization, which prevents the model from recovering the memory of member-specific features, especially in BERT, where we observe a notable 33% drop in prediction accuracy on members.

**Other Privacy Threats.** MIA not only serves as a standard tool for auditing privacy leakage but also provides a strong foundation for other privacy attacks. That is, the compression-induced leakage revealed by  $\text{CompLeak}$  may extend to other privacy attacks. More specifically, white-box MIA enables access to internal signals, e.g., gradients and activations, whose variations after compression can further amplify leakage. Similarly, property inference attacks can be strengthened as properties correlated with memorization become more distinguishable due to the fact that retraining reinforces these patterns. Model inversion attack is also affected, as compression alters decision boundaries, which may guide reconstructions closer to the original training data.

**Potential Mitigation.** To mitigate the privacy risks introduced by model compression, beyond standard MIA defenses such as differential privacy—which we have evaluated in Section 5.4—we discuss three other compression-aware potential defenses from different aspects.

*Fine-tune compressed models on an auxiliary dataset.* Compression-induced leakage arises from the reduced model capacity and re-training on the original training set, which intensifies memorization. To mitigate this, fine-tuning the

compressed model exclusively on an auxiliary dataset that is disjoint from the training set but sampled from the same distribution can minimize additional memorization of training samples and effectively reduce the membership gap.

*Adopt post-training quantization.* Results in Section 7 show that post-training quantization exposes the least additional privacy because it requires no fine-tuning after compression, and its architectural and numerical difference from the full model is minimal. Therefore, choosing the proper compression technique of post-training quantization can be a simple and effective mitigation.

*Restricting information of model output.* Restricting the model’s outputs to the hard label only reduces the amount of observable information to an attacker, preventing the exploitation of compression-induced shifts and thereby lowering the exploitable signal for membership inference. This has been partially validated through the RMIA attack results in Section 7, which has a notably reduced MIA accuracy due to the usage of true-label confidence instead of the full softmax information.

## 8 Limitation

**Large-scale Generative Models.** As evidenced in classification tasks,  $\text{CompLeak}$  exposes more privacy information if the underlying model is vulnerable to MIA. However, it will not function if the underlying model task is resistant, e.g., large-scale LLMs for generative tasks. State-of-the-art studies [19, 26] indicate that MIA on large-scale LLMs typically performs no better than random guessing, primarily due to the substantially reduced overfitting from the combination of a massive training corpora and very few training iterations. Moreover, compression for LLMs is typically via post-training quantization without additional fine-tuning, which constrains further memorization. These factors limit  $\text{CompLeak}$ ’s generalization to generative tasks enabled by large-scale generative models.

## 9 Conclusion

This work presents the first in-depth privacy evaluation framework  $\text{CompLeak}$  for three widely used and commercially supported compression operations—pruning, quantization, and weight clustering—through the lens of membership inference. Specifically,  $\text{CompLeak}_{\text{SR}}$  reveals that these compression operations indeed increase privacy leakage. Notably, this leakage is further exacerbated when leveraging information from multiple compressed models. Building on this, we propose  $\text{CompLeak}_{\text{MR}}$  that stacks the loss of multiple compressed models and the meta-posterior from the  $\text{CompLeak}_{\text{SR}}$  attack meta-classifier. Extensive experiments have validated the  $\text{CompLeak}$  superior performance under diverse model architectures, datasets, and various settings.

## Ethical Considerations

We discuss the ethical considerations of CompLeak from the following perspectives.

**Stakeholders and Potential Impact.** The key stakeholders involved in our research include model providers who employ compression for efficient deployment, users whose data might be used for training, and malicious actors. Although commercial toolkits (e.g., TensorFlow Lite) are used in our work, the observed privacy leakage arises from general compression techniques, not from any specific tool implementation. This scenario is the same as [47], where the Google TensorFlow-Lite team acknowledged that there are no ethical conflicts, as the compression technique is general. CompLeak is intended to assist model providers in auditing and understanding the privacy risks of model compression, thereby motivating the development of more robust, privacy-preserving compression techniques and contributing to more secure AI systems. However, we acknowledge the risk that adversaries could use CompLeak to launch more potent MIAs. To mitigate these risks, we emphasize the need for responsible use and explicitly discourage any unethical applications.

**Mitigations.** To mitigate such risks, our findings strongly suggest that model providers adopt a range of strategies—such as incorporating differential privacy, restricting exposure to the full posterior vector, fine-tuning compressed models on an auxiliary dataset, and reducing model overfitting—before releasing models through query APIs. These practices help safeguard user data and ensure ethical standards in the deployment of model compression techniques.

**Decision.** After careful consideration, we decided to proceed with this research and publish our findings, as adhering to these principles ensures our investigation on privacy risks in model compression remains within ethical boundaries. We hope this work not only exposes a critical vulnerability but also encourages the development of more robust privacy defenses, thereby promoting the safer deployment of model compression techniques.

## Open Science

For the open science policy, all datasets (CIFAR10, CIFAR100, Mini-ImageNet, Tiny-ImageNet) used in this research are open source and can be accessed from official websites. The source code for CompLeak is accessible at: <https://doi.org/10.6084/m9.figshare.30904289>.

## Acknowledgment

We would like to thank the anonymous reviewers and our shepherd for their detailed and valuable comments. This work is supported by National Natural Science Foundation of China (62372236, 62402223).

## References

- [1] <https://gdpr-info.eu/>.
- [2] <https://huggingface.co/TCLResearchEurope>.
- [3] <https://openai.com/research/>.
- [4] <https://huggingface.co/onnxmodelzoo/mobilenetv2-12-int8>.
- [5] <https://www.cs.toronto.edu/~kriz/cifar.html>.
- [6] <https://github.com/microsoft/nni>.
- [7] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pages 308–318, 2016.
- [8] Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Structured pruning of deep convolutional neural networks. ACM Journal on Emerging Technologies in Computing Systems (JETC), 13(3):1–18, 2017.
- [9] Teodora Baluta, Shiqi Shen, S Hitharth, Shruti Tople, and Prateek Saxena. Membership inference attacks and generalization: A causal perspective. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pages 249–262, 2022.
- [10] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Gutttag. What is the state of neural network pruning? Proceedings of Machine Learning and Systems, 2:129–146, 2020.
- [11] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In 2022 IEEE Symposium on Security and Privacy (SP), pages 1897–1914. IEEE, 2022.
- [12] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, pages 343–362, 2020.
- [13] Guangke Chen, Yedi Zhang, and Fu Song. SLMIA-SR: speaker-level membership inference attacks against speaker recognition systems. In 31st Annual Network and Distributed System Security Symposium, NDSS 2024, San Diego, California, USA, February 26 - March 1, 2024. The Internet Society, 2024.

- [14] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. When machine unlearning jeopardizes privacy. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pages 896–911, 2021.
- [15] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. Model compression and acceleration for deep neural networks: The principles, progress, and challenges. IEEE Signal Processing Magazine, 35(1):126–136, 2018.
- [16] Tejalal Choudhary, Vipul Mishra, Anurag Goswami, and Jagannathan Sarangapani. A comprehensive survey on model compression and acceleration. Artificial Intelligence Review, 53:5113–5155, 2020.
- [17] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4690–4699, 2019.
- [18] Yuntao Du, Jiacheng Li, Yuetian Chen, Kaiyuan Zhang, Zhizhen Yuan, Hanshen Xiao, Bruno Ribeiro, and Ninghui Li. Cascading and proxy membership inference attacks. In 33rd Annual Network and Distributed System Security Symposium, NDSS 2026, 2026.
- [19] Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models? In First Conference on Language Modeling, 2024.
- [20] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3, pages 265–284. Springer, 2006.
- [21] Kazuki Egashira, Mark Vero, Robin Staab, Jingxuan He, and Martin Vechev. Exploiting llm quantization. In NeurIPS 2024, 2024.
- [22] Yansong Gao, Huming Qiu, Zhi Zhang, Binghui Wang, Hua Ma, Alsharif Abuadbba, Minhui Xue, Anmin Fu, and Surya Nepal. Deeptheft: Stealing dnn model architectures through power side channel. In 2024 IEEE Symposium on Security and Privacy (SP), pages 3311–3326. IEEE, 2024.
- [23] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In Low-Power Computer Vision, pages 291–326. Chapman and Hall/CRC, 2022.
- [24] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149, 2015.
- [25] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. Proceedings on Privacy Enhancing Technologies, 2019:133 – 152, 2017.
- [26] Jamie Hayes, Iliia Shumailov, Christopher A Choquette-Choo, Matthew Jagielski, Georgios Kaissis, Milad Nasr, Meenatchi Sundaram Muthu Selva Annamalai, Niloofer Mireshghallah, Igor Shilov, Matthieu Meeus, et al. Exploring the limits of strong membership inference attacks on large language models. In The Thirty-ninth Annual Conference on Neural Information Processing Systems.
- [27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.
- [28] Yu He, Boheng Li, Yao Wang, Mengda Yang, Juan Wang, Hongxin Hu, and Xingyu Zhao. Is difficulty calibration all we need? towards more practical membership inference attacks. In Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, pages 1226–1240, 2024.
- [29] Zheng He, Zeke Xie, Quanzhi Zhu, and Zengchang Qin. Sparse double descent: Where network pruning aggravates overfitting. In International Conference on Machine Learning, pages 8635–8659. PMLR, 2022.
- [30] Torsten Hoefer, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. Journal of Machine Learning Research, 22(241):1–124, 2021.
- [31] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Accurate post training quantization with small calibration sets. In International Conference on Machine Learning, pages 4466–4475. PMLR, 2021.
- [32] Bo Hui, Yuchen Yang, Haolin Yuan, Philippe Burlina, Neil Zhenqiang Gong, and Yinzi Cao. Practical blind membership inference attack via differential comparisons. In 28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021, 2021.
- [33] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training

- of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2704–2713, 2018.
- [34] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas J Spanos, and Dawn Song. Efficient task-specific data valuation for nearest neighbor algorithms. arXiv preprint arXiv:1908.08619, 2019.
- [35] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Židek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. Nature, 596(7873):583–589, 2021.
- [36] Anders Krogh and John Hertz. A simple weight decay can improve generalization. Advances in Neural Information Processing Systems, 4, 1991.
- [37] Solomon Kullback and Richard A Leibler. On information and sufficiency. The Annals of Mathematical Statistics, 22(1):79–86, 1951.
- [38] Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. CS 231N, 7(7):3, 2015.
- [39] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. ArXiv, abs/1608.08710, 2016.
- [40] Hao Li, Zheng Li, Siyuan Wu, Chengrui Hu, Yutong Ye, Min Zhang, Dengguo Feng, and Yang Zhang. Seqmia: Sequential-metric based membership inference attack. In Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, pages 3496–3510, 2024.
- [41] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. Membership inference attacks and defenses in classification models. In Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy, pages 5–16, 2021.
- [42] Zheng Li, Yiyong Liu, Xinlei He, Ning Yu, Michael Backes, and Yang Zhang. Auditing membership leakages of multi-exit networks. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pages 1917–1931, 2022.
- [43] Zhuohang Li, Andrew Lowy, Jing Liu, Toshiaki Koike-Akino, Kieran Parsons, Bradley Malin, and Ye Wang. Analyzing inference privacy risks through gradients in machine learning. In Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, pages 3466–3480, 2024.
- [44] Han Liu, Yuhao Wu, Zhiyuan Yu, and Ning Zhang. Please tell me more: Privacy impact of explainability through the lens of membership inference attack. In 2024 IEEE Symposium on Security and Privacy (SP), pages 4791–4809. IEEE, 2024.
- [45] Hongbin Liu, Jinyuan Jia, Wenjie Qu, and Neil Zhenqiang Gong. Encodermi: Membership inference against pre-trained encoders in contrastive learning. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pages 2081–2095, 2021.
- [46] Yiyong Liu, Zhengyu Zhao, Michael Backes, and Yang Zhang. Membership inference attacks by exploiting loss trajectory. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pages 2085–2098, 2022.
- [47] Hua Ma, Huming Qiu, Yansong Gao, Zhi Zhang, Alsharif Abuadbba, Minhui Xue, Anmin Fu, Jiliang Zhang, Said F Al-Sarawi, and Derek Abbott. Quantization backdoors to deep learning commercial frameworks. IEEE Transactions on Dependable and Secure Computing, 2023.
- [48] Matthieu Meeus, Shubham Jain, Marek Rei, and Yves-Alexandre de Montjoye. Did the neurons read your book? document-level membership inference for large language models. In 33rd USENIX Security Symposium (USENIX Security 24), pages 2369–2385, 2024.
- [49] Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in large language models are more redundant than you expect. arXiv preprint arXiv:2403.03853, 2024.
- [50] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In International Conference on Machine Learning, pages 2498–2507. PMLR, 2017.
- [51] Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pages 634–646, 2018.
- [52] Anselme Ndikumana, Nguyen H Tran, Ki Tae Kim, Choong Seon Hong, et al. Deep learning based caching for self-driving cars in multi-access edge computing. IEEE Transactions on Intelligent Transportation Systems, 22(5):2862–2877, 2020.

- [53] Yan Pang and Tianhao Wang. Black-box membership inference attacks against fine-tuned diffusion models. In 32nd Annual Network and Distributed System Security Symposium, NDSS 2025, San Diego, California, USA, February 24-28, 2025. The Internet Society, 2025.
- [54] Yuefeng Peng, Ali Naseh, and Amir Houmansadr. Diffe: Fencing membership privacy with diffusion models. In 32nd Annual Network and Distributed System Security Symposium, NDSS 2025, San Diego, California, USA, February 24-28, 2025. The Internet Society, 2025.
- [55] Lutz Prechelt. Early stopping-but when? In Neural Networks: Tricks of the Trade, pages 55–69. Springer, 2002.
- [56] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. MI-leaks: Model and data independent membership inference attacks and defenses on machine learning models. arXiv preprint arXiv:1806.01246, 2018.
- [57] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4510–4520, 2018.
- [58] Sriram Sankararaman, Guillaume Obozinski, Michael I Jordan, and Eran Halperin. Genomic privacy and limits of individual detection in a pool. Nature Genetics, 41(9):965–967, 2009.
- [59] Jing Shang, Jian Wang, Kailun Wang, Jiqiang Liu, Nan Jiang, Md. Armanuzzaman, and Ziming Zhao. Defending against membership inference attacks on iteratively pruned deep neural networks. In 32nd Annual Network and Distributed System Security Symposium, NDSS 2025, San Diego, California, USA, February 24-28, 2025. The Internet Society, 2025.
- [60] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In 2017 IEEE Symposium on Security and Privacy (SP), pages 3–18. IEEE, 2017.
- [61] Karen Simonyan. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
- [62] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1631–1642, 2013.
- [63] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In 30th USENIX Security Symposium (USENIX Security 21), pages 2615–2632, 2021.
- [64] Bozhidar Stevanoski, Ana-Maria Cretu, and Yves-Alexandre de Montjoye. Querycheetah: Fast automated discovery of attribute inference attacks against query-based systems. In Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, pages 3451–3465, 2024.
- [65] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of Machine Learning Research, 9(11), 2008.
- [66] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. Advances in Neural Information Processing Systems, 29, 2016.
- [67] Yijue Wang, Chenghong Wang, Zigeng Wang, Shanglin Zhou, Hang Liu, Jinbo Bi, Caiwen Ding, and Sanguthevar Rajasekaran. Against membership inference attack: Pruning is all you need. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, pages 3141–3147. ijcai.org, 2021.
- [68] Rui Wen, Michael Backes, and Yang Zhang. Understanding data importance in machine learning attacks: Does valuable data pose greater harm? arXiv preprint arXiv:2409.03741, 2024.
- [69] Rui Wen, Zheng Li, Michael Backes, and Yang Zhang. Membership inference attacks against in-context learning. In Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, pages 3481–3495, 2024.
- [70] Mengjia Yan, Christopher W Fletcher, and Josep Torrellas. Cache telepathy: Leveraging shared resource attacks to learn {DNN} architectures. In 29th USENIX Security Symposium (USENIX Security 20), pages 2003–2020, 2020.
- [71] Jiwei Yang, Xu Shen, Jun Xing, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-sheng Hua. Quantization networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7308–7316, 2019.
- [72] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In Proceedings of the

2022 ACM SIGSAC Conference on Computer and Communications Security, pages 3093–3106, 2022.

- [73] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In 2018 IEEE 31st Computer Security Foundations Symposium (CSF), pages 268–282. IEEE, 2018.
- [74] Xiaoyong Yuan and Lan Zhang. Membership inference attacks and defenses in neural network pruning. In 31st USENIX Security Symposium (USENIX Security 22), pages 4561–4578, 2022.
- [75] Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024, 2024.
- [76] Kaiyuan Zhang, Siyuan Cheng, Hanxi Guo, Yuetian Chen, Zian Su, Shengwei An, Yuntao Du, Charles Fleming, Ashish Kundu, Xiangyu Zhang, and Ninghui Li. SOFT: selective data obfuscation for protecting LLM fine-tuning against membership inference attacks. In 34st USENIX security symposium (USENIX security 25), 2025.
- [77] Yunpeng Zhao and Jie Zhang. Does training with synthetic data truly protect privacy? In The Thirteenth International Conference on Learning Representations, 2025.
- [78] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. A comprehensive survey on pre-trained foundation models: A history from bert to chat-gpt. International Journal of Machine Learning and Cybernetics, pages 1–65, 2024.
- [79] Yukun Zhou, Mark A Chia, Siegfried K Wagner, Murat S Ayhan, Dominic J Williamson, Robbert R Struyven, Timing Liu, Moucheng Xu, Mateo G Lozano, Peter Woodward-Court, et al. A foundation model for generalizable disease detection from retinal images. Nature, 622(7981):156–163, 2023.
- [80] Jie Zhu, Jirong Zha, Ding Li, and Leye Wang. A unified membership inference method for visual self-supervised encoder via part-aware capability. In Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, pages 1241–1255, 2024.

## A Datasets Description

Below is a brief description of each dataset.

**CIFAR-10.** CIFAR-10 [5] is a widely used benchmark dataset in image classification, consisting of 60,000  $32 \times 32$  color images across 10 distinct classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. It contains 50,000 training and 10,000 test images, with an equal number of images in each class.

**CIFAR-100.** CIFAR-100 [5] is similar to CIFAR-10 but includes 100 classes instead of 10, with each class containing 600 images. It consists of 60,000  $32 \times 32$  color images, divided into 50,000 training and 10,000 test images, offering a more granular challenge for image classification models.

**Mini-ImageNet.** Mini-ImageNet [66] is a widely used benchmark dataset for evaluating image recognition algorithms. A subset of the ImageNet, it consists of 100 classes, each containing 600  $84 \times 84$  color images.

**Tiny-ImageNet.** Tiny-ImageNet [38] is a widely used benchmark dataset for evaluating image recognition algorithms. It is a subset of the ImageNet, consisting of 100,000 images of 200 categories (500 for each category) downsized to  $64 \times 64$  color images.

## B Other Meta-data Construction Methods and Performance

We present two additional meta-data construction methods, similar to [14]. One method is based on direct concatenation, i.e.,  $\mathcal{P}_o^s \parallel \mathcal{P}_c^s \parallel y$ , and the other is based on calculating the  $L_2$  distance, i.e.,  $\|\mathcal{P}_o^s - \mathcal{P}_c^s\|_2 \parallel y$ . Table 10 provides the attack results for these two construction methods on the clustered MobileNetV2 with 8 clusters on Tiny-ImageNet.

Table 10: Attack performance of two construction methods on the clustered MobileNetV2 with 8 clusters on Tiny-ImageNet.

Attack Method	TPR @ 0.1% FPR (%)		Balanced Accuracy (%)		AUC (%)	
	original	clustered	original	clustered	original	clustered
CompLeak <sub>NR</sub> [60] (LR)	0.1	0.2	51.3	51.7	51.3	52.3
CompLeak <sub>NR</sub> [51] (LR)	0.0	0.0	51.5	51.9	51.6	52.2
CompLeak <sub>NR</sub> [60] (RF)	<b>5.0</b>	5.8	62.5	61.3	67.8	66.2
CompLeak <sub>NR</sub> [51] (RF)	4.7	5.5	62.2	61.1	67.7	66.0
CompLeak <sub>NR</sub> [73]	0.0	0.1	55.3	49.4	53.9	45.6
CompLeak <sub>NR</sub> [63]	0.0	0.0	61.5	57.1	63.4	57.8
CompLeak <sub>NR</sub> [74]	4.9	5.2	<b>67.2</b>	61.0	<b>72.6</b>	67.1
Direct concatenation (LR)	-	0.1	-	51.9	-	52.4
$L_2$ distance (LR)	-	21.5	-	67.8	-	69.6
Direct concatenation (RF)	-	18.1	-	<b>80.5</b>	-	<b>89.5</b>
$L_2$ distance (RF)	-	<b>36.7</b>	-	69.5	-	76.1

## C Evaluation of CompLeak<sub>NR</sub> and CompLeak<sub>SR</sub> for other datasets

### C.1 Evaluation of Pruning

The attack results for pruning at different sparsities are presented for Tiny-ImageNet in Table 11, and CIFAR-10 in Table 12.

Table 11: Attack performance of different attacks on varying pruned rate (MobileNetV2+Tiny-ImageNet).

Attack Method	TPR @ 0.1% FPR (%)				Balanced Accuracy (%)				AUC (%)						
	original	40%	50%	60%	70%	original	40%	50%	60%	70%	original	40%	50%	60%	70%
CompLeak <sub>NR</sub> [60] (LR)	0.2	0.2	0.3	0.1	0.2	51.3	50.8	51.0	51.1	50.4	51.3	51.6	51.0	51.5	50.9
CompLeak <sub>NR</sub> [51] (LR)	0.0	0.0	0.1	0.0	0.1	51.5	51.1	51.3	51.0	50.6	51.6	51.8	51.9	51.7	51.4
CompLeak <sub>NR</sub> [60] (RF)	<b>5.1</b>	5.3	5.1	3.9	5.2	62.5	61.1	61.9	60.9	60.0	67.8	66.2	67.3	65.6	64.6
CompLeak <sub>NR</sub> [51] (RF)	4.8	5.4	5.0	4.1	5.1	62.2	60.9	61.7	60.5	60.1	67.7	65.9	67.1	65.3	64.4
CompLeak <sub>NR</sub> [73]	0.1	0.1	0.1	0.0	0.0	55.3	53.5	55.6	52.3	49.9	53.9	52.0	54.5	50.9	48.0
CompLeak <sub>NR</sub> [63]	0.1	0.1	0.1	0.1	0.1	61.5	60.3	62.2	59.9	57.7	63.4	61.2	64.4	61.5	58.8
CompLeak <sub>NR</sub> [74]	4.9	3.9	1.9	0.1	0.2	<b>67.2</b>	66.3	67.1	51.1	50.1	<b>72.6</b>	72.4	71.8	55.1	49.3
CompLeak <sub>SR</sub> 1 (LR)	-	17.9	14.4	11.2	9.8	-	68.0	67.0	67.4	67.8	-	76.6	74.6	75.6	76.2
CompLeak <sub>SR</sub> 2 (LR)	-	14.5	11.5	11.8	10.5	-	68.7	68.1	68.8	69.5	-	80.2	78.7	79.7	80.4
CompLeak <sub>SR</sub> 1 (RF)	-	<b>54.4</b>	<b>33.8</b>	<b>29.5</b>	<b>24.2</b>	-	91.9	89.0	85.2	81.8	-	97.7	95.9	92.9	90.1
CompLeak <sub>SR</sub> 2 (RF)	-	54.0	<b>33.8</b>	28.8	<b>24.2</b>	-	<b>92.1</b>	<b>89.3</b>	<b>85.7</b>	<b>82.7</b>	-	<b>97.9</b>	<b>96.2</b>	<b>93.6</b>	<b>91.1</b>

Table 12: Attack performance of different attacks on varying pruned rate (ResNet18+CIFAR-10).

Attack Method	TPR @ 0.1% FPR (%)				Balanced Accuracy (%)				AUC (%)						
	original	60%	70%	80%	90%	original	60%	70%	80%	90%	original	60%	70%	80%	90%
CompLeak <sub>NR</sub> [60] (LR)	0.4	0.2	0.2	0.2	0.2	53.0	49.5	47.9	48.1	49.3	53.7	50.0	48.6	46.9	49.9
CompLeak <sub>NR</sub> [51] (LR)	0.0	0.0	0.0	0.0	0.0	55.2	49.5	47.5	48.0	46.9	55.4	50.0	50.3	47.5	51.5
CompLeak <sub>NR</sub> [60] (RF)	0.8	0.9	0.6	0.5	0.3	65.6	68.6	68.9	68.5	61.9	71.4	74.3	74.8	74.5	68.2
CompLeak <sub>NR</sub> [51] (RF)	<b>1.2</b>	<b>1.1</b>	<b>0.7</b>	<b>0.9</b>	0.3	68.7	70.9	71.4	71.3	64.9	74.6	76.4	77.1	77.1	71.6
CompLeak <sub>NR</sub> [73]	0.1	0.2	0.2	0.1	0.1	65.5	68.3	68.6	68.8	61.7	69.1	71.7	71.9	71.9	67.9
CompLeak <sub>NR</sub> [63]	0.2	0.2	0.2	0.2	0.1	68.4	70.5	71.0	71.2	68.3	72.2	74.0	74.3	74.3	71.5
CompLeak <sub>NR</sub> [74]	0.7	0.4	0.2	0.4	0.2	<b>68.9</b>	70.9	71.5	71.8	70.2	<b>74.7</b>	76.3	76.4	76.5	74.9
CompLeak <sub>SR</sub> 1 (LR)	-	0.1	0.1	0.1	0.1	-	68.2	67.5	67.8	69.0	-	72.0	72.1	72.1	70.9
CompLeak <sub>SR</sub> 2 (LR)	-	0.1	0.2	0.1	0.1	-	70.3	70.1	70.5	71.2	-	73.2	73.5	73.4	72.4
CompLeak <sub>SR</sub> 1 (RF)	-	0.6	0.4	0.5	0.3	-	71.1	70.6	70.2	70.1	-	77.2	75.3	75.6	74.8
CompLeak <sub>SR</sub> 2 (RF)	-	0.6	0.5	0.5	<b>0.3</b>	-	<b>72.6</b>	<b>72.5</b>	<b>72.2</b>	<b>72.1</b>	-	<b>78.4</b>	<b>77.1</b>	<b>77.2</b>	<b>76.8</b>

## C.2 Evaluation of Quantization

Here, we provide attack results for other datasets see Table 13.

Table 13: Attack performance of quantization.

Attack Method	TPR @ 0.1% FPR (%)		Balanced Accuracy (%)		AUC (%)	
	CIFAR-100 ResNet50	Tiny-ImageNet MobileNetV2	CIFAR-100 ResNet50	Tiny-ImageNet MobileNetV2	CIFAR-100 ResNet50	Tiny-ImageNet MobileNetV2
CompLeak <sub>NR</sub> [60] (LR)	0.3	0.1	49.7	51.4	51.4	51.6
CompLeak <sub>NR</sub> [51] (LR)	0.0	0.1	53.4	51.6	54.4	52.3
CompLeak <sub>NR</sub> [60] (RF)	1.9	3.2	73.4	61.3	81.0	66.5
CompLeak <sub>NR</sub> [51] (RF)	<b>2.2</b>	3.2	73.3	60.9	80.9	66.3
CompLeak <sub>SR</sub> 1 (LR)	0.1	28.3	77.3	68.6	78.8	77.0
CompLeak <sub>SR</sub> 2 (LR)	0.1	15.5	78.1	67.7	79.4	80.3
CompLeak <sub>SR</sub> 1 (RF)	0.5	60.6	77.6	96.7	82.6	99.4
CompLeak <sub>SR</sub> 2 (RF)	0.8	<b>62.9</b>	<b>78.4</b>	<b>97.0</b>	<b>83.2</b>	<b>99.5</b>

## C.3 Evaluation of Weight Clustering

We employ TF-Lite to perform weight clustering on the original ResNet18 model trained on CIFAR-10, with cluster numbers set to 8 and 14. Note that the original ResNet18 exhibits relatively low overfitting, achieving a train accuracy of 96% and test accuracy of 85%, which poses a challenge for MIA. As shown in Table 14, when the meta-classifier structure is LR, CompLeak<sub>SR</sub> significantly enhances attack performance, surpassing CompLeak<sub>NR</sub> on either the original or clustered models. Specifically, CompLeak<sub>SR</sub> improves the balanced accuracy by 6.5% over CompLeak<sub>NR</sub> [51] when the cluster number is 8. Similarly, when the meta-classifier structure is RF, CompLeak<sub>SR</sub> still outperforms CompLeak<sub>NR</sub> across all model variants. Additionally, we evaluate weight clustering on MobileNetV2 trained on Tiny-ImageNet (implement on PyTorch) with cluster sizes of 8, 16, and 32. The corresponding MIA results are reported in Table 15.

Table 14: Attack performance of different attacks against ResNet18 trained and clustered at various levels on CIFAR-10.

Attack Method	Balanced Accuracy (%)			AUC (%)		
	original	14	8	original	14	8
CompLeak <sub>NR</sub> [60] (LR)	49.7	49.9	49.8	49.2	50.0	50.0
CompLeak <sub>NR</sub> [51] (LR)	49.8	49.6	50.1	49.5	48.9	49.5
CompLeak <sub>NR</sub> [60] (RF)	54.1	54.7	53.8	55.9	56.8	55.7
CompLeak <sub>NR</sub> [51] (RF)	<b>56.0</b>	<b>56.3</b>	<b>55.8</b>	<b>58.2</b>	<b>59.0</b>	<b>58.3</b>
CompLeak <sub>SR</sub> 1 (LR)	-	54.9	54.8	-	54.4	54.1
CompLeak <sub>SR</sub> 2 (LR)	-	56.8	56.6	-	56.8	55.5
CompLeak <sub>SR</sub> 1 (RF)	-	54.9	54.2	-	57.2	56.5
CompLeak <sub>SR</sub> 2 (RF)	-	<b>57.0</b>	<b>56.8</b>	-	<b>59.5</b>	<b>59.6</b>

Table 15: Attack performance of different attacks against MobileNetV2 trained and clustered at various levels on Tiny-ImageNet.

Attack Method	TPR @ 0.1% FPR (%)			Balanced Accuracy (%)			AUC (%)					
	original	32	16	8	original	32	16	8	original	32	16	8
CompLeak <sub>NR</sub> [60] (LR)	0.2	0.1	0.3	0.2	51.3	50.9	51.9	51.7	51.3	52.0	52.3	52.3
CompLeak <sub>NR</sub> [51] (LR)	0.0	0.0	0.0	0.1	51.5	51.5	51.5	51.9	51.6	52.0	52.1	52.2
CompLeak <sub>NR</sub> [60] (RF)	<b>5.1</b>	5.3	6.3	5.8	62.5	62.2	62.3	61.3	67.8	67.3	67.7	66.2
CompLeak <sub>NR</sub> [51] (RF)	4.8	5.2	6.3	5.5	62.2	61.9	62.1	61.1	67.7	67.0	67.5	66.0
CompLeak <sub>NR</sub> [73]	0.1	0.1	0.1	0.1	55.3	55.7	53.6	49.4	53.9	54.1	51.8	45.6
CompLeak <sub>NR</sub> [63]	0.1	0.1	0.1	0.1	61.5	61.7	59.6	57.1	63.4	63.9	62.2	57.8
CompLeak <sub>NR</sub> [74]	4.9	4.2	4.9	5.2	<b>67.2</b>	67.0	66.0	61.0	<b>72.6</b>	73.0	72.5	67.1
CompLeak <sub>SR</sub> 1 (LR)	-	27.7	22.4	17.0	-	69.0	69.2	68.2	-	77.3	77.4	76.5
CompLeak <sub>SR</sub> 2 (LR)	-	14.9	11.8	12.0	-	68.0	69.2	69.6	-	80.4	80.9	80.5
CompLeak <sub>SR</sub> 1 (RF)	-	87.2	55.5	28.7	-	<b>96.9</b>	<b>93.0</b>	85.9	-	99.6	98.2	93.7
CompLeak <sub>SR</sub> 2 (RF)	-	<b>88.2</b>	<b>59.3</b>	<b>30.6</b>	-	96.7	<b>93.0</b>	<b>86.3</b>	-	<b>99.6</b>	<b>98.3</b>	<b>94.3</b>

## D Attack Against DP-SGD

Table 16: Attack performance against 70%-pruned ResNet18 trained on Mini-ImageNet with DP-SGD ( $\sigma = 0.2$  and  $\sigma = 0.5$ ). The values in parentheses represent the attack performance on the original ResNet18.

Attack Method	TPR @ 0.1% FPR (%)			Balanced Accuracy (%)			AUC (%)		
	no defense	$\sigma = 0.2$	$\sigma = 0.5$	no defense	$\sigma = 0.2$	$\sigma = 0.5$	no defense	$\sigma = 0.2$	$\sigma = 0.5$
CompLeak <sub>NR</sub> [60]	1.1 (1.2)	0.7 (0.7)	0.4 (0.6)	57.1 (57.8)	55.7 (55.8)	55.2 (55.4)	60.6 (60.8)	58.2 (58.4)	57.6 (58.1)
CompLeak <sub>NR</sub> [51]	1.0 (0.9)	0.7 (0.8)	0.5 (0.7)	57.6 (57.7)	56.0 (56.0)	54.5 (55.0)	61.0 (60.9)	58.5 (58.6)	56.7 (57.1)
CompLeak <sub>SR</sub> 1	40.0	13.8	9.4	86.0	73.4	71.5	93.4	81.1	79.1
CompLeak <sub>SR</sub> 2	40.4	13.2	10.2	86.5	73.4	71.6	93.9	81.2	79.2

The structure of the attack meta-classifier is all based on the RF.

Corresponding privacy budget ( $\epsilon$ ) for given noise multiplier ( $\sigma$ ): ( $\sigma = 0.2, \epsilon = 707.15$ ) and ( $\sigma = 0.5, \epsilon = 32.92$ ).

## E Ablation Study

**Overfitting Level of the Victim Model.** It is widely recognized that MIA is closely related to the overfitting level of the victim model. Following [40], we quantify the overfitting level by measuring the gap between training and testing accuracy and regulate it by varying the size of the training dataset. Specifically, we consider two distinct levels  $L_o \in \{29\%, 10\%\}$  to explore their influence on MIA performance.

As described in Table 18, all MIAs demonstrate weaker performance against models with low overfitting, which conclusion aligns with [40]. In addition, we observe that, under low-overfitting conditions, the 80%-pruned model is less susceptible to CompLeak<sub>NR</sub> than the original model, which aligns with our earlier findings in Section 4.3. However, as overfitting increases, this trend reverses. Notably, CompLeak<sub>SR</sub> is always

Table 17: Attack performance on Compression vs Duplication (VGG16+Mini-ImageNet).

Attack Method	TPR @ 0.1% FPR (%)						Balanced Accuracy (%)					AUC (%)																			
	original	R1	R2	R3	R4	prune	original	R1	R2	R3	R4	prune	original	R1	R2	R3	R4	prune													
CompLeak <sub>NR</sub> [60] (RF)	1.6	1.0	1.1	1.2	1.1	1.5	59.3	58.4	59.7	59.1	59.6	59.1	63.2	62.5	63.7	63.0	63.8	62.7													
CompLeak <sub>NR</sub> [51] (RF)	1.5	1.1	1.0	1.0	1.0	1.4	59.2	58.2	59.4	58.7	59.3	59.2	63.3	62.3	63.3	62.7	63.4	62.8													
CompLeak <sub>SR</sub> 1 (RF)	-	12.4	15.0	15.8	13.3	42.1	-	72.2	71.9	72.0	71.8	84.2	-	81.0	80.9	81.0	80.9	92.8													
CompLeak <sub>SR</sub> 2 (RF)	-	12.6	16.3	16.6	13.6	41.7	-	72.7	72.6	72.7	72.3	83.9	-	81.8	81.7	81.9	81.7	93.0													
CompLeak <sub>MR</sub>	20.6						61.3					73.1					94.8					83.0					98.9				

The training and test accuracies of the four reference models are as follows: R1 (90.58%, 75.13%), R2 (92.31%, 75.39%), R3 (90.90%, 75.30%), and R4 (92.85%, 75.78%). As a comparison, **prune** report results under 70% pruning for CompLeak<sub>NR</sub> and CompLeak<sub>SR</sub>, and aggregate multiple pruning rates (60%, 70%, 80%, and 90%) for CompLeak<sub>MR</sub>.

superior over CompLeak<sub>NR</sub> on the original/compressed model, with the performance gap becoming particularly pronounced in high-overfitting scenarios.

Table 18: Attack performance against 80%-pruned ResNet-18 trained on CIFAR-10 with different overfitting levels (values in parentheses represent the performance on the original ResNet18).

Attack Method	Balanced Accuracy (%)		AUC (%)	
	$L_o = 29\%$	$L_o = 10\%$	$L_o = 29\%$	$L_o = 10\%$
CompLeak <sub>NR</sub> [60]	68.5 (65.6)	57.8 (60.3)	74.5 (71.4)	60.2 (64.2)
CompLeak <sub>NR</sub> [73]	68.8 (65.5)	60.5 (61.0)	71.9 (69.1)	61.6 (62.2)
CompLeak <sub>SR</sub> 2	<b>72.2</b>	<b>61.7</b>	<b>77.2</b>	<b>65.4</b>

When  $L_o = 29\%$ , the training accuracy is 99%, the test accuracy is 70%, and the training dataset contains 13500 samples, following [74]. When  $L_o = 10\%$ , the training accuracy is 99%, the test accuracy is 89%, and the training dataset contains 20000 samples. Attack meta-classifier adopts the RF architecture.

**Dataset of the Victim Model.** Here, we focus on examining the influence of datasets on attacks by using the same model architecture with different datasets. Specifically, we trained the VGG16 model on Mini-ImageNet and CIFAR-10. As shown in Table 19, the attack performance of the CompLeak<sub>NR</sub> on both the original model and the pruned model is consistently lower for Mini-ImageNet compared to CIFAR-10, regardless of the attack meta-model used. However, with our CompLeak<sub>SR</sub>, which leverages the variation introduced by the compression operation, the attack performance on Mini-ImageNet is significantly higher than on CIFAR-10. We hypothesize that this is because Mini-ImageNet is a more complex dataset than CIFAR-10, causing larger compression-induced variation differences between members and non-members. Therefore, under the same model architecture, compressing more complex datasets can lead to more severe privacy leakage.

Table 19: The attack performance of CIFAR-10/Mini-ImageNet on VGG16.

Attack Method	Balanced Accuracy (%)			AUC (%)		
	original	pruned (80%)	quantized	original	pruned (80%)	quantized
CompLeak <sub>NR</sub> [60] (LR)	50.3/48.3	50.5/48.5	50.8/48.3	50.1/47.3	50.3/48.0	50.3/47.4
CompLeak <sub>NR</sub> [51] (LR)	49.8/48.3	50.1/50.0	50.0/51.1	49.6/50.1	50.2/51.0	50.0/50.5
CompLeak <sub>NR</sub> [60] (RF)	61.7/59.3	59.7/58.6	61.6/59.4	65.9/63.2	63.0/62.1	65.9/63.4
CompLeak <sub>NR</sub> [51] (RF)	61.8/59.2	60.5/58.7	61.9/59.3	66.4/63.3	63.9/62.3	66.2/63.5
CompLeak <sub>SR</sub> 2 (LR)	-	62.1/59.9	61.8/60.8	-	61.2/68.8	61.1/70.8
CompLeak <sub>SR</sub> 2 (RF)	-	<b>62.4/83.4</b>	<b>62.1/90.3</b>	-	<b>66.6/92.1</b>	<b>66.4/98.3</b>

## F Performance of Individual Component

As CompLeak<sub>MR</sub> utilizes the loss concatenation and posterior concatenation to form meta-data. To validate the contribution of each meta-data component, we evaluate attack performance using each component individually. As shown in Table 20, the contribution of posterior concatenation is the most pronounced, and the combination of both components surpasses the performance of any individual component. This underscores the essential contribution of each component to the overall attack performance.

Table 20: Attack performance of CompLeak<sub>MR</sub> using the individual meta-data component under Adversary 1.

Component	TPR @ 0.1% FPR (%)	Balanced Accuracy (%)	AUC (%)
Loss	1.0	72.0	73.7
Posterior	55.6	93.6	98.3
Loss & Posterior	61.3	94.8	98.9

## G Other Results

The results for Compression vs Duplication are shown in Table 17, while the prediction accuracy of the original and compressed BERT/GPT-2 models is presented in Figure 9.

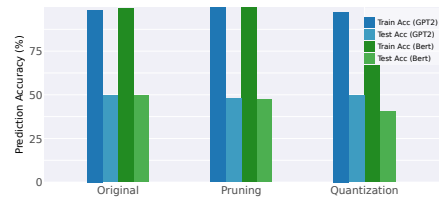


Figure 9: The prediction accuracy of the original and compressed BERT/GPT-2 fine-tuned on SST-5.



Figure 10: The same weight matrix subjected to pruning, quantization and weight clustering separately.