# Analyzing and Defending against Membership Inference Attacks in Natural Language Processing Classification

Yijue Wang*§, Nuo Xu†§, Shaoyi Huang*, Kaleel Mahmood*, Dan Guo‡, Caiwen Ding*,
Wujie Wen†, Sanguthevar Rajasekaran*
University of Connecticut*,
Lehigh University†,
Northeastern University‡,
Email: {yijue.wang, shaoyi.huang, kaleel.mahmood, caiwen.ding, sanguthevar.rajasekaran}@uconn.edu§
{nux219,wuw219}@lehigh.edu†,
guo.dan@northeastern.edu‡

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020711

***Abstract*—The risk posed by Membership Inferen (MIA) to deep learning models for Computer Vision (C well known, but MIA has not been addressed or explor the Natural Language Processing (NLP) domain. In thi analyze the security risk posed by MIA to NLP models that NLP models are at great risk to MIA, in some more so than models trained on Computer Vision (CV This includes an 8.04% increase in attack success rate for NLP models (as compared to CV models and dat determine that there are some unique issues in NLP cla tasks in terms of model overfitting, model complexity diversity that make the privacy leakage severe and ver from CV classification tasks. Based on these findings, v a novel defense algorithm - Gap score Regularization Pruning (GRIP), which can protect NLP models agains achieve competitive testing accuracy. Our experimen show that GRIP can decrease the MIA success rate b as 31.25% when compared to the undefended model. In addition, when compared to differential privacy, GRIP offers 7.81% more robustness to MIA and 13.24% higher testing accuracy. Overall our experimental results span four NLP and two CV datasets, and are tested with a total of five different model architectures.**

## I. Introduction

As the global machine learning market grows, Machine Learning as a Service (MLaaS) [1] is gaining increasing popularity from cloud computing providers such as Amazon [2], Microsoft [3], and Google [4]. Using black-box interfaces, MLaaS allows users to upload data easily, leverage powerful large-scale DNNs, and deploy analytic services [5]. Examples of MLaaS in NLP include companies (as well as individuals) putting their data in deep learning models for speech recognition, word sense disambiguation, sentiment analysis, and other tasks. In parallel to the deep learning developments in NLP, deep learning has also been applied to achieve state-of-the-art results on Computer Vision (CV) tasks [6]–[8]. CV models have been shown to suffer from a privacy leakage attack (see

§: Equal Contribution

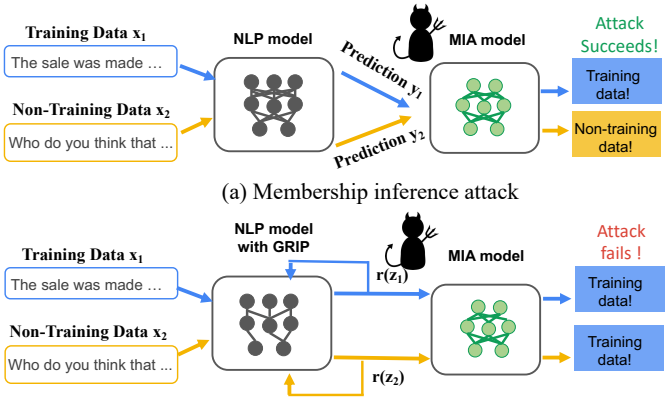


Fig. 1. (a) MIA in NLP. (b) Our proposed method against MIA: Gap score Regularization Integrated Pruning (GRIP).

Figure 1) known as Membership Inference Attack (MIA). CV models are vulnerable to black-box MIAs due to multiple reasons, such as overfitting and large model complexity [9]–[13]. However, to the best of our knowledge, the vulnerability of NLP models to MIA has not been thoroughly studied. From these observations, several important questions arise.

1) *Are NLP models vulnerable to MIA attacks like CV models?*
2) *What makes NLP models vulnerable to MIA?*
3) *What can be done to defend against MIA in the NLP domain?*

We have carried out a thorough literature search and found the aforementioned issues lack an in-depth investigation. These are pertinent questions to the future security of deep learning for NLP and are precisely the questions we seek to answer.

To answer the first question, we experiment with the text classification tasks in NLP domain and image classification tasks in CV domain. The text classification tasks have a smaller number of classes than the image classification tasks. Thus, the outputs of the NLP models for text classification

contain less information. Despite this fact, the results on various NLP datasets suggest that the privacy risk of membership inference is severe for NLP models. As shown in Table I, similar to general CV models, NLP models are vulnerable to two types of MIA, neural network (NN) MIAs and metric-based MIAs. However, differences arise in MIA between the CV and NLP domains due to a variety of issues such as overfitting, model complexity, and data diversity, which we analyze and discuss in depth later in the paper.

Due to the severity of MIA in NLP, the next natural question in our investigation is how to defend against this threat. We propose a novel defense algorithm, Gap score Regularization Integrated Pruning (GRIP), that is optimized by finding a sub-network from the original over-parameterized NLP model (see Figure 1). GRIP can prevent privacy leakage from MIA and achieve similar accuracy to the original NLP model. As an additional side benefit, GRIP can also reduce the model storage and the computation overhead. In summary, we make the following contributions.

1) **Comprehensive MIA Analysis in the NLP Domain:** We illustrate the classification tasks to compare MIAs in the CV and NLP domains and find that NLP models are also vulnerable to MIA attacks. We then analyze the causes of MIAs from three perspectives: overfitting, model complexity, and data diversity.
2) **Novel MIA Defense for NLP Models:** We develop a new MIA defense that works across all NLP datasets we studied in this paper. Our proposed defense algorithm GRIP reduces the attack success rate of MIA by as much as $31.25\%$ compared to undefended models and models with differential privacy.

Having listed our majority contributions, we outline the structure for the rest of the paper. In Section 2, we discuss relevant background information and related literature. In Section 3, we compare MIAs on classification tasks in the CV and NLP domain and analyze the causes of MIAs in NLP. We propose a novel defense strategy to MIAs in Section 4 and evaluate defense on various datasets and models in Section 5.

## II. Related Work

### A. Membership Inference Attack (MIA)

The MIA attempts to determine whether a given data is from the training dataset or not for a target model [10], [14]–[17]. This attack can lead to serious privacy problems that leak the individual's private information like the health data, financial state, etc., in different scenarios [18]. There are two basic types of adversarial attacks for MIA, i.e., the white-box and the black-box MIA. In this paper, we consider the black-box attack that the attacker assumes can only access the model outputs. Recent studies have shown that multiple realistic machine learning classifiers are vulnerable to such black-box MIAs [17], [18]. There are two types of black-box MIAs, i.e., Neural Network (NN) MIA and metric-based MIA.

**NN MIAs.** Multiple current MIA algorithms work by training a machine learning MIA model that leverages the statistical differences between members of the training set and non-training set to distinguish between the two [9], [10]. In this paper, we present a general machine learning MIA model for NLP classification models and formulate the optimization problem to defend against an adversary in this setting.

TABLE I
Membership inference attack accuracy for different models on some representative datasets for classification tasks in the NLP and CV domain.

| NLP | | | CV | | |
|---|---|---|---|---|---|
| **Model Dataset** | **NN MIA** | **Metric MIA** | **Model Dataset** | **NN MIA** | **Metric MIA** |
| **BERT RTE** | 84.37% | 69.00% | **Alexnet CIFAR10** | 71.70% | 66.80% |
| **BERT MRPC** | 71.88% | 59.10% | **MobilenetV2 CIFAR100** | 62.75% | 55.01% |
| **BERT CoLA** | 68.75% | 63.70% | **Resnet18 CIFAR100** | 69.85% | 73.02% |
| **BERT SST2** | 73.44% | 58.50% | **Vgg16 CIFAR100** | 61.99% | 68.24% |

**Metric MIAs.** Unlike NN attacks, metric-based attacks directly use the prediction vectors to compute customized metrics as a way to infer membership or non-membership in comparison with preset thresholds. We follow the state-of-the-art works [14], [19], [20] and experiment with four metric MIAs based on *correctness*, *confidence*, *entropy* and *modified entropy*. The detailed explanations of these four metric MIAs can be found in Appendix A.

### B. Current Defense Mechanism

There are several mechanisms that have been developed to address MIA in general classification tasks. Differential privacy (DP) [21], [22] is a major privacy-preserving mechanism against general inference attack. It is based on adding noises into gradients or objective functions when training the model and has been applied in different machine learning models [23]–[25]. Another mechanism to address MIA is adding regularization during the model training. Existing regularization methods are mainly proposed to reduce the overfitting problem, which is one of the main causes of MIAs [10], [26]. However, it is common to load large pre-trained NLP models with private training data and then fine-tune the models on a smaller task-specific dataset. Due to this training regime, it is necessary to reevaluate how severe the overfitting problem is in the NLP classification domain. As a result, these regularization methods are difficult to incorporate into NLP models to create a feasible defense against MIA. We use DP training to compare the effectiveness of defense against MIA in NLP classification tasks as it is a general adversarial defense mechanism in transfer learning with provable privacy guarantees [13], [23].

### C. Weight Pruning

Weight pruning techniques have traditionally been used to increase model performance (i.e., speed up inference time) and reduce the model size (save space) while still maintaining

high fidelity (high prediction accuracy) [27]–[30]. State-of-the-art DNNs contain multiple cascaded layers and millions of parameters (i.e., weights) for the entire model [31], [32].

In natural language processing, irregular magnitude weight pruning (IMWP) has been evaluated on BERT, where $30\% - 40\%$ weights with a magnitude close to zero are set to be zero [33], [34]. Irregular reweighted proximal pruning (IRPP) [35] adopts iteratively reweighted $l_1$ minimization with the proximal algorithm and achieves $59.3\%$ more overall pruning ratio than irregular magnitude weight pruning without accuracy loss. [36] investigates the model general redundancy and task-specific redundancy on BERT and XLNet [37].

## III. Membership Inference Attack in the NLP Domain

Even though MIA has been comprehensively studied in computer vision, the same cannot be said of NLP. This raises a critical question, *how vulnerable are NLP models to Membership Inference Attacks?*

We consider the MIA problems in the context of a black-box adversary. We assume that the adversary has access to part of the data records from the training and testing set and the predictions from the black-box DNN target model.

### A. MIAs in NLP vs. MIAs in CV

We summarize the best attack accuracy of NN MIAs and metric MIAs for different classification tasks in NLP and CV domains in Table I. The NLP models and all MIA experiments are conducted according to the settings in Section V-A, and the CV models are trained based on settings in [31], [38]–[40]. Our first set of results shows a unique difference between models trained on CV tasks and models trained on NLP tasks. Specifically, in Table I, we show that privacy leakage in the NLP classification tasks is significant. For example, the BERT-RTE task has an $84.37\%$ NN attack success rate.

Besides, we can observe that, NN MIAs could be different from CV domains MIA. NN MIAs consistently outperform metric MIAs in NLP models. Even when the overfitting is not severe and the metric MIAs are weak, they still show superior attack ability with potential privacy leakage risk.

### B. Causes of MIAs in the NLP

In the following, we discuss the causes of MIAs in NLP from three perspectives: overfitting, model complexity, and data diversity.

**(1) Overfitting.** Overfitted models perform much better on training data than on non-training data (i.e., validation or test data) and it is one of the main factors that cause privacy leakage. We find that despite the fact that NLP models are pre-trained, overfitting can also occur. Evidence of this claim can be seen in Figure 2, where we show the accuracy gap between training and testing data for a BERT model trained on different NLP datasets. In Figure 2, we can see that the NN MIA is stronger than the metric MIA for all datasets. For example, on the RTE datasets, the accuracy gap is 25.73%,

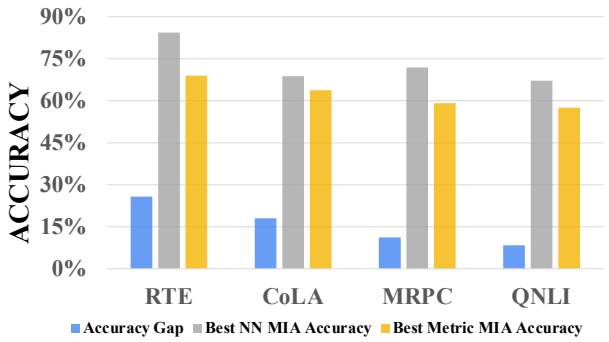


Fig. 2. The membership inference attack accuracy as well as the accuracy gap between training and testing set on different datasets.

and the NN MIA accuracy is almost 85%. This performance is consistent with previous studies in the CV field. Moreover, NN MIAs show more robustness on the MRPC and SST-2 datasets when the overfitting is not significant. Unlike metric MIAs that decrease when the accuracy gap is small, the NN attack remains strong. This suggests more causes for privacy breaches in the NLP models.

**(2) Model Complexity.** NLP classification models are often over-parameterized with high complexity. For example, the BERT model contains 12 encoder blocks and 110 million parameters in total. This on the one hand gives them the ability to learn efficiently from hard NLP tasks, but on the other hand also leads to the possibility that they may have a high parameters redundancy to remember noise or details of the training dataset. On the other hand, for CV classification models, VGG16 has 16 layers, 13 million parameters, and ResNet-18 has 18 layers, 11 million parameters. The NLP classification model structures could be very different from the CV classification model, and their parameter sizes could be much larger.

**(3) Data Diversity.** There are many dataset properties that may boost the performance of MIA. First, the number of classes in NLP classification tasks is limited, e.g., most of the GLUE datasets are binary or ternary classification tasks, while there are 10 to 1000 classification tasks in the CV domain. Second, the size of both training and non-training data in NLP tasks can be limited. For example, RTE has only 2490 training samples, which is 20 times less than MNIST. Due to the limited amount of training data and categories, the learned distribution of the dataset may be less representative and induced. Therefore, MIAs can achieve high accuracy even if the model is not overfitted.

## IV. How to Prevent MIA in NLP?

### A. Defense Problem Formulation

The first goal of designing an MIA defense is finding a target model $g$ to minimize the privacy leakage and the second

goal is to ensure that the target model $g$'s prediction accuracy remains high. Mathematically, the objective is:

$$\min_g \; G_g(f_A) + \mathcal{L}(g) \tag{1}$$

Where $f$ is the classification model, $f_A$ is the attack model, $\mathcal{L}(g)$ is the classification loss of model $g$, and $G_g(f_A)$ is the adversary's gain function that quantitatively present how much privacy leakage information the adversary can obtain. According to [9], [41], $G_g(f_A)$ can be written as:

$$\begin{aligned} G_g(f_A) = \int_{x,y} & [P_D(x,y)p_g(f(x))\log(f_A(x,y,g(x)))+ \\ & P_{D'}(x,y)p'_g(g(x))\log(1-f_A(x,y,g(x))]dxdy \\ & = -\log(4) + 2\cdot JS(p_g(g(x))||p'_g(g(x))) \end{aligned} \tag{2}$$

Where $D$ is the training set and $D'$ is the non-training set. $p_g$ and $p'_g$ are the probability distribution of the classification model $g$'s output for training data and non-training data. $JS(p_g(g(x))||p'_g(g(x)))$ is the Jensen–Shannon divergence between the two distributions and it is always non-negative. The global minimum value that $G_g(f_A)$ can possibly have is -log(4) if and only if:

$$p_g(g(x)) = p'_g(g(x')) \tag{3}$$

This means that the prediction of classification model $g$ has the same probability distribution for both the training set and non-training set. In this case, the attack fails in the sense the attacker can do no better than a random guess.

### B. Proposed Defense Strategy

Since overfitting and model complexity are the two main reasons for MIA, we design our defense strategy to reduce the overfitting and the model complexity and while maintaining competitive accuracies of the classification model $g$. In terms of reducing the model complexity, the main issue that arises is the question of finding a sub-network. Specifically, *can we find a sub-network from the original over-parameterized NLP model that can prevent privacy leakage from MIA while maintaining accuracy similar to that of the original NLP model?* Next, we will introduce the original network and analysis the strategy to find such sub-network.

We define the original NLP network $g^*(x)$:

$$g^*(x) = \mathbf{E}_n^g \circ \mathbf{E}_{n-1}^g \circ ... \circ \mathbf{E}_1^g(M(x))) \tag{4}$$

where $\mathbf{E}_j^g$ is the $j$th block in model $g^*$ For example, in the BERT model, there are twelve building blocks, each building block contains a self-attention layer and a fully connected feed-forward network. Symbol $\circ$ stands for the connection between neighboring blocks. $M$ is the embedding block connected with the data input and the first block. In defense design, we want to find a sub-network $\hat{g^*}(x)$ that has competitive prediction accuracy similar to the target network $g(x)$.

We propose a weight pruning method to find the sub-network $\hat{g^*}(x)$. Moreover, to reduce the overfitting, we use a gap score-based regularization to minimize the prediction gap between training and non-training data. In total, our defense strategy contains two components: 1) weight pruning to reduce model complexity and overfitting, and 2) gap score regularization to reduce overfitting. Next, we present a theoretical analysis of the existence of a sub-network $\hat{g^*}(x)$ on the regularization term.

### C. Accuracy analysis of weight pruned sub-network

We first analyze and ensure the pruned model can still maintain the classification accuracy. A pruned network $\hat{g}(x)$ can be presented as $\hat{g^*}(x)$:

$$\hat{g^*}(x) = \hat{\mathbf{E}}_n^g \circ \hat{\mathbf{E}}_{n-1}^g \circ ... \circ \hat{\mathbf{E}}_1^g(\mathbf{E}(x))) \tag{5}$$

where $\mathbf{P}_i$ is the pruning matrix in $i$-th layer.

**Theorem 1.** *For every network g defined in Eq. 4 with depth l and $\forall i \in \{1,2,\ldots,n\}$. Consider $g^*$ as a randomly initialized neural network, and width $poly(d,n,m,1/\epsilon,log1/\delta)$, where d is input size, n is number of layers in $g^*$, m is the maximum number of neurons in a layer. For the weights in $\mathbf{E}_i^g$, the weight initialization distribution belongs to uniform distribution in range [-1,1]. Then with probability at least $1-\delta$ there is a weight-pruned sub-network $\hat{g^*}$ of g such that:*

$$\sup_{x\in\chi, \|W\|\leq 1} \left\| g(x) - \hat{g^*}(x) \right\| \leq \epsilon \tag{6}$$

Based on Theorem 1, we know that for every bounded distribution and every target network with bounded weights, there is a sub-network with an accuracy that is close to the original over-parameterized neural networks. Next, we analyze two different types of modules in transformer, i.e., the feed-forward linear layer and the self-attention layer.

*1) Feed-forward Linear Network:* In this case, $g(x) = \mathbf{W}\cdot x$ , and $g^*(x) = \left(\sum_{i=1}^d \mathbf{W}_i\right) x$.

**Theorem 2.** *Let $\mathbf{W}_1^*, ..., \mathbf{W}_n^*$ belongs to i.i.d. Uniform distribution over [-1,1], where $n \geq C\cdot log\frac{2}{\delta}$, where $\delta \leq min\{1,\epsilon\}$. Then, with probability at least 1-δ, we have*

$$\begin{aligned} & \exists S \subset \{1,2,...,n\}, \forall W \in [-0.5, 0.5], \\ & s.t \left| \mathbf{W} - \sum_{i\in S} \mathbf{W}_i^* \right| \leq \epsilon \end{aligned} \tag{7}$$

Lueker et al. [42] proposed this theorem and had given a proof.

*2) Self-attention Layer: General case:* Consider a model $g(x)$ with only one self-attention layer, when the token size is n, $\mathbf{x} = (x_1, x_2, ..., x_n)$. let $(h_{..})_{n\times n} = \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{(d_k)}}$, then

$$\begin{aligned} g(x_i) &= softmax((h_{i.})_{1\times n})\mathbf{V}_i \\ &= (\frac{\sum_j e^{h_{ij}}}{\sum_i \sum_j (e^{h_{ij}})})\mathbf{V}_i \\ &= (\frac{\sum_j e^{h_{ij}}}{\sum_i \sum_j (e^{h_{ij}})})\mathbf{W}^{\mathbf{V}_i} x_i \\ &= \mathbf{W}^{h_{i\cdot}} x_i \end{aligned} \tag{8}$$

**Corollary 1** *Let* $\mathbf{W}_1^{g^*}, ..., \mathbf{W}_d^{g^*}$ *belongs to i.i.d. uniform distribution over [-1,1], where* $d \geq Clog\frac{2}{\delta}$*, where* $\delta \leq min\{1, \epsilon\}$*. Then, with probability at least 1-δ, we have*

$$\forall\, i \in \{1, 2, ..., n\}, \mathbf{W}_l^{g^*} \in [-1, 1], \exists\, p_l \in \{0, 1\},$$
$$s.t. \left| \mathbf{W}^{h_i} - (\sum_{l=1}^{d} p_l \mathbf{W}_l^{g^*}) \right| < \epsilon \tag{9}$$

### D. Analysis of Gap Score Regularization

To prevent privacy leakage, our goal is to find the target model $g$ that minimizes the adversary's gain by adding a regularization term into the loss function, we consider this problem as:

$$\min\ \mathcal{L}(g) + \alpha \cdot r(\mathbf{z}_{max} - \mathbf{z}_{min}) \tag{10}$$

where $\mathcal{L}(g)$ is the classification loss of $g$. $r$ represents the regularization objective function and $\alpha$ is the coefficient to tune the impact between the training objective and privacy objective. Let $\mathbf{z}$ be the one-hot encoding prediction of the model, $\mathbf{z}_{max}$ is the highest probability value from all individuals in $\mathbf{z}$ and $\mathbf{z}_{min}$ is the lowest probability value from all individuals in $\mathbf{z}$. To represent the gap score in the multi-class classification case, we show:

$$\begin{aligned} &r(\mathbf{z}_{max} - \mathbf{z}_{min}) = \mathbf{z}_{max} - \mathbf{z}_{min} \\ &s.t.\ \mathbf{z}_{max} - \mathbf{z}_{min} \in [0, 1] \end{aligned} \tag{11}$$

so we have

$$\alpha \cdot r(\mathbf{z}_{max} - \mathbf{z}_{min}) \in [0, \alpha] \tag{12}$$

the update gradient can be calculated as:

$$\begin{aligned} \nabla \mathbf{W} &= \frac{\partial L(\mathbf{W})}{\partial \mathbf{W}} + \alpha \cdot \frac{\partial r(\mathbf{z})}{\partial \mathbf{W}} \\ &= \frac{\partial L(\mathbf{W})}{\partial \mathbf{W}} + \alpha \cdot \frac{\partial (\mathbf{z}_{max} - \mathbf{z}_{min})}{\partial \mathbf{W}} \\ &= \frac{\partial L(\mathbf{W})}{\partial \mathbf{W}} + \alpha \cdot (\frac{\partial \mathbf{z}_{max}}{\partial \mathbf{W}} - \frac{\partial \mathbf{z}_{min}}{\partial \mathbf{W}}) \end{aligned} \tag{13}$$

In this case, when we update the model by minimizing the loss function, the gap score is also minimized. So the distribution of $p_f(f(x))$ and $p'_f(f(x'))$ are more similar than each other, i.e., $JS(p_f(f(x))||p'_f(f(x)))$ decreases and is closer to 0. Thus, the adversary has minimum gain for the trained model and privacy leakage is prevented.

### E. Proposed Method: GRIP

We show our proposed method Gap score Regularization Integrated Pruning (GRIP) in Algorithm 1. For a fixed NLP classification model $g$, we set target sparsity for different layers. Let $P_k^s$ be the target sparsity for self-attention layer and $P_k^{fc}$ for feed-forward network. In $i$th iteration and $k$th block, we set the sparsity $P_{ik}^s$ for self-attention layer and $P_{ik}^{fc}$ for feed-forward network. Then, inspired by [27], [43] and instead of pruning the weights directly to the target sparsity, we systematically prune the weights of each block in multiple iterations gradually by satisfying the following Equations to minimize the utility loss from weight pruning.

**Algorithm 1** The Process of GRIP

**for** *epoch* in *Epochs* **do**
  Get a random mini-batch $S$.
  **for** i in Iterations: **do**
    **for** Encoder $k$ : **do**
      **for** self-attention layer: **do**
        Prune $\{\mathbf{W}^Q\}$ to $\{P_{ik}^s \odot \mathbf{W}^Q\}$ by Eq.14
        Prune $\{\mathbf{W}^K\}$ to $\{P_{ik}^s \odot \mathbf{W}^K\}$ by Eq.15
      **end for**
      **for** feed-forward network: **do**
        Prune $\{\mathbf{W}\}$ to $\{P_{ik}^{fc} \odot \mathbf{W}\}$
      **end for**
    **end for**
  **end for**
  Get $\{\mathbf{z}_{max}\}$ and $\{\mathbf{z}_{min}\}$
  Calculate $r(\mathbf{z}_{max}, \mathbf{z}_{min})$
  Update $\{\mathbf{W}\}$, $\{\mathbf{W}^Q\}$,$\{\mathbf{W}^K\}$,$\{\mathbf{W}^V\}$
  by minimizing $\mathcal{L}(f) + \alpha \cdot r(\mathbf{z}_{max} - \mathbf{z}_{min})$
**end for**
OUTPUT $\{\mathbf{W}\}$, $\{\mathbf{W}^Q\}$,$\{\mathbf{W}^K\}$,$\{\mathbf{W}^V\}$

$$P_{ik}^s = P_k^s + (1 - P_k^s) * (1 - \frac{i}{n})^3 \tag{14}$$

$$P_{ik}^{fc} = P_k^{fc} + (1 - P_k^{fc}) * (1 - \frac{i}{n})^3 \tag{15}$$

When updating these weights, we minimize the loss function in Eq. 10 with gap score regularization.

## V. Evaluation

### A. Experimental Setup

**Datasets.** For the proposed sparse progressive distillation, we conduct experiments on General Language Understanding Evaluation (GLUE) benchmarks [44] including RTE, CoLA, MRPC and SST-2, which are grouped into three categories of natural language understanding tasks (single-sentence tasks, similarity matching tasks, and natural language inference tasks) according to the purpose of tasks and difficulty level of datasets.

**Models.** We use the fine-tuned $\text{BERT}_{\text{BASE}}$ as a teacher and also initialize the student with the fine-tuned $\text{BERT}_{\text{BASE}}$. Specifically, we fine-tune the pre-train $\text{BERT}_{\text{BASE}}$ on four GLUE tasks for 4 epochs, including SST-2, CoLA, MRPC, and RTE. We select the learning rate with best performance from $\{2e^{-}5, 3e^{-}5, 4e^{-}5, 5e^{-}5\}$. Batch size and maximum sequence length are set as 32 and 128, respectively.

**Membership Inference Attacks Setup.** To evaluate the neural network (NN) MIAs, we follow the model structure and setup in [9] to construct and train the attack classifier. The attack classifier takes two pieces of information as input. One is the unsorted confidence score vector, and the other one is the label of the input data that is one hot encoded (all elements

TABLE II
COMPARISON OF CLASSIFICATION ACCURACY AND MEMBERSHIP ATTACK ACCURACY BETWEEN REGULAR TRAINING, DIFFERENTIAL PRIVATE TRAINING AND GRIP TRAINING ON BERT MODEL.

| | RTE | | | MRPC | | | CoLA | | | SST-2 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Defense | None | DP | **GRIP** | None | DP | **GRIP** | None | DP | **GRIP** | None | DP | **GRIP** |
| Testing Accuracy | 70.28% | 53.79% | 61.01% | 84.39% | 68.38% | 81.62% | 81.09% | 71.80% | 81.20% | 92.89% | 81.77% | 91.17% |
| Accuracy Gap | 28.11% | 2.75% | 12.28% | 13.62% | 0.93% | 5.27% | 15.53% | 1.00% | 9.00% | 6.48% | 1.31% | 2.83% |
| NN MIA | 84.38% | 59.38% | 53.13% | 71.88% | 53.13% | 53.13% | 60.94% | 57.81% | 50.00% | 73.44% | 60.94% | 57.81% |
| Metric MIA | 69.00% | 54.20% | 57.80% | 59.10% | 52.00% | 53.70% | 63.70% | 51.50% | 56.90% | 58.50% | 55.30% | 52.50% |

except the one that corresponds to the label index are 0). The classifier consists of three fully connected sub-networks. The one operates on the confidence score vectors has three layers with size 1024,512 and 64. One network with two layers with 512 and 64 neurons works on the label. The third network is the combined network that takes the outputs of the two networks as a concatenate input and has five layers with sizes 512, 256, 128, 64, and 1. The final output will predict whether the input belongs to the trainset or not with a probability (larger than 0.5 will count as a member). We use the ReLu activation function for the network except for the final output layer with the sigmoid activation function. We train the attack classifier with Adam optimizer and mean squared error (MSE) criterion for a total of 300 epochs. To better generate the model, we set the initial learning rate to 0.001 and decays by 0.1 in the 30th epoch. For the metric MIAs evaluation, we adopt four metric attacks following the [14] and show the best attack accuracy in the tables.

**Defense Training Setup.** In our evaluation, we conduct the canonical implementation of training a model with differential privacy (DP) [23] and the associated analysis in Pytorch implementation from Opacus [45] library. We adopt the DP training into the original fine-tuning process and set the clipping bound to be 1.0 based on standard practices and report the best testing accuracy results in Table II.

In our GRIP defense, we give different sparsity in different iterations and different blocks. We gradually prune weight for both self-attention layers and feed-forward networks by Eq. 14 and 15, then we will reach the sparsity after all iterations. In detail, we use sparsity $40\%$ for CoLA and sparsity $60\%$ pruning rate for the other datasets on the last 6 encoders and $\alpha = 1$ for all datasets on the pre-trained BERT model with 4 to 12 fine-tuning epochs and record the best accuracy results.

### *B. Results and Analysis*

**GRIP can significantly reduce the membership inference attack success rate.** As shown in Table II, our defense leads to a significant reduction in privacy risks in both NN and metric MIAs. For all evaluated datasets, we can reduce the MIA accuracy with neural network to $\sim 50\%$, which is close to a random guess and performs much better compared to the high attack accuracy of the undefended model, from $60.94\%$ (CoLA) to $84.38\%$ (RTE). Our defense can also outperform the DP training on the NN MIAs. For metric MIAs, although the attack accuracy with GRIP is not always close to random guesses, we can still observe a $5 \sim 10\%$ decrease in attack accuracy even when the original MIA risk is not high.

**GRIP achieves privacy protection with a small utility cost.** With all the benefits of the privacy defense from our proposed methods, the utility loss is limited in a small range at most times. Our GRIP training maintains the classification accuracy at the same level on CoLA and SST-2 dataset and causes $2.77\%$ accuracy decrease on MRPC. Defense on the RTE dataset leads to $10\%$ utility loss, but it is a very small dataset with limited training and testing data. The model is unstable with random separation on the training and testing data in each time of training and attack. Even in the worst cases, our approach can still largely outperform DP training as it leads to $10 \sim 20\%$ utility loss on all the datasets with very limited privacy protection on the NN MIAs. This is a case where the privacy budget is large and the model utility will be further reduced when the theoretical guarantees of DP training are obtained.

**GRIP has significantly reduced model complexity.** Tabel III summaries the weights reduction ratio of GRIP fine-tuned model on different datasets. Except for the benefit of privacy defense, our GRIP has an additional advantage on model storage and computations. Table III show that our GRIP has over $1.18\times$ ratio over different datasets.

In summary, we have the following analysis:

1. Reducing the overfitting of the NLP classification problem does not completely eliminate the membership privacy risk, which is consistent with the observation in Section III-A. Taking the DP-trained model as an example, it successfully reduces overfitting as the accuracy gap is only $0.93 \sim 2.75\%$ on all datasets, which helps the models limit the metric MIAs to $55\%$. However, the NN MIAs remain at $60\%$, indicating that there is still privacy leakage on the poor utility models.

2. Our GRIP works during training for both constraint of output prediction and reduction of model complexity of intermediate structures. As a result, we not only reduce model overfitting but also yield similar performance in terms of confidence and robustness for both training and test samples. For 'free lunch', we also reduce the model storage and the computations. Thus, our defenses can effectively resist MIAs and maintain good model utility.

TABLE III
MODEL COMPLEXITY REDUCTION BY GRIP FOR DIFFERENT TASKS.

| Data | Model | Weights (#) | Weights after prunning (#) | Weights reduction ratio |
|---|---|---|---|---|
| RTE | BERT | 110 M | 77 M | 1.30 × |
| MRPC | BERT | 110 M | 77 M | 1.30 × |
| CoLA | BERT | 110 M | 88 M | 1.18 × |
| SST-2 | BRET | 110 M | 77 M | 1.30 × |

TABLE IV
BEST CLASSIFICATION ACCURACY AND NN MIA ACCURACY ON BERT MODELS FINE-TUNING WITH MIA-PRUNING OR GAP SCORE REGULARIZATION.

| Defense | Proposed Pruning | | Gap Score Regularization | |
|---|---|---|---|---|
| Accuracy | Testing Accuracy | NN MIA | Testing Accuracy | NN MIA |
| RTE | 63.05% | 62.50% | 58.12% | 59.37% |
| MRPC | 81.86% | 65.63% | 77.21% | 57.81% |
| CoLA | 80.50% | 59.37% | 80.70% | 51.56% |
| SST-2 | 92.66% | 67.18% | 93.46% | 57.81% |

### C. Hyperparameter Analysis

In this subsection, we investigate the contribution of the proposed pruning and the proposed gap score regularization, respectively.

We first show the classification accuracy and NN MIA results on the four datasets using proposed pruning and proposed gap score regularization in Table IV. Compared to the baseline model results in Table II, we can observe that each component of the proposed method can help reduce the attack accuracy with some utility loss. The proposed pruning methods achieve at most $31.25\%$ (RTE) and on average $19.14\%$ attack accuracy decrease for NN MIA with $0.23 \sim 7.23\%$ utility loss. The gap score regularization achieves better defense against MIAs ($16.02\%$ decrease on average) while leading to a little bit more classification accuracy loss ($0 \sim 12.16\%$). In following subsections, we will demonstrate the effects of the individual proposed methods with more detailed ablation studies.

*1) Proposed Pruning Algorithm:* We investigate how our proposed pruning affects defense performance by pruning ratios. As shown in Figure 3, the attack accuracy of metric MIA decreases along with the higher pruning ratio when the pruning ratio is over 70%. However, the attack accuracy of NN MIA presents a fluctuation pattern when varying the pruning ratio. It reaches the minimum value when the pruning ratio is 70%.

*2) Gap Score Regularization:* In order to show the effects of the gap score regularization on the classification accuracy and MIAs defense, we tune the hyperparameter $\alpha$ that controls the impact of the regularization in training on RTE dataset as shown in Figure 4. $\alpha$ trades off the utility and privacy. With the increase of $\alpha$, the constraint on the gap score becomes tighter and the gap score of the final result becomes smaller. Hence, the accuracy gap and classification accuracy decrease

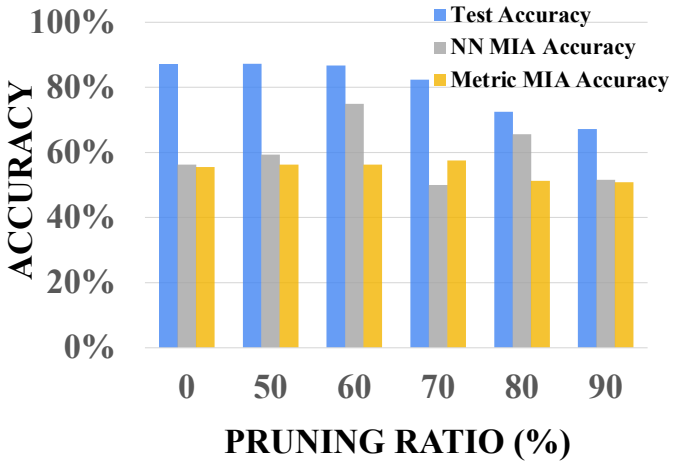


Fig. 3. The effects of different pruning ratio on BERT for MRPC task.

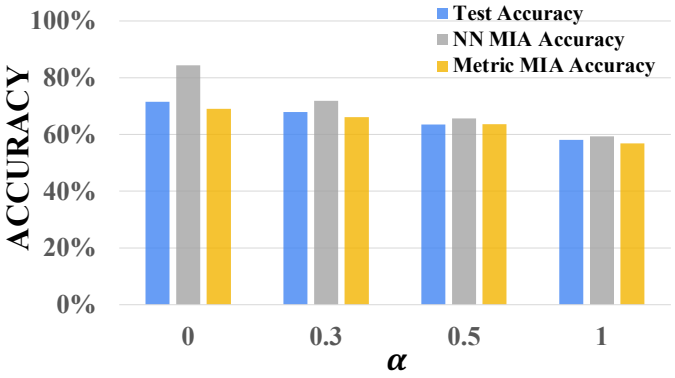


Fig. 4. Different $\alpha$ for gap score regularization on BERT model for RTE task

while the model can better defend against NN and metric MIA. Specifically, $\alpha = 0.3$ in Figure 4 shows the case when the constraint is not large enough. The regularization starts to control the output and shows defensiveness, and this effect is first shown in a decrease in test accuracy, while the training data accuracy remains close to $100\%$ and consequently the accuracy gap might increase.

**Key takeaways:** Our GRIP defense achieves a much better privacy-utility trade-off than using the proposed pruning or gap score regularization alone. This is because GRIP is a combinatorial approach that benefits from pruning to derive a finer and sparser model structure. And GRIP can better learn the proposed regularization and loss minimization during the fine-tuning process to control the final prediction distributions.

## VI. CONCLUSION

In this work, we explore NN MIAs and metric MIAs on NLP models. Our experiments show that MIA represents a significant threat to NLP models and in some cases this vulnerability is even greater than that of CV models and datasets. To better understand this issue we further analyzed the MIA

in NLP models in terms of overfitting, model complexity and data diversity. We then developed a defense method GRIP, specifically for NLP that is based on weight pruning and gap score regularization. Our evaluations of the BERT model on RTE, MRPC, CoLA, SST-2 datasets show that GRIP achieves privacy protection against MIAs with a substantially smaller cost on the utility loss compared with DP. Specifically, GRIP can reduce the MIA success rate by $31.25\%$ as compared to the undefended model. When compared to DP, GRIP offers $7.81\%$ more robustness to MIA and $13.24\%$ higher testing accuracy.

In addition, GRIP significantly reduces the model storage and computation cost, *e.g.*, it has approximately $1.30\times$ weight reduction ratio on RTE, MRPC, and SST-2 datasets. Overall, our MIA analyses and proposed MIA NLP defense serve as important steps toward developing efficient and privacy-preserving deep learning models in NLP.

## VII. Acknowledgement

This research has been supported in part by the National Science Foundation (NSF) Grants 1743418 and 1843025.

## References

[1] M. Ribeiro, K. Grolinger, and M. A. Capretz, "Mlaas: Machine learning as a service," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2015, pp. 896–902.

[2] A. Kurniawan, *Learning AWS IoT: Effectively manage connected devices on the AWS cloud using services such as AWS Greengrass, AWS button, predictive analytics and machine learning*. Packt Publishing Ltd, 2018.

[3] D. Gollob, *Microsoft Azure-Planning, Deploying, and Managing Your Data Center in the*. Springer-verlag Berlin And Hei, 2015.

[4] A. Ravulavaru, *Google Cloud AI Services Quick Start Guide: Build Intelligent Applications with Google Cloud AI Services*. Packt Publishing Ltd, 2018.

[5] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, "Demystifying membership inference attacks in machine learning as a service," *IEEE Transactions on Services Computing*, 2019.

[6] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "Coatnet: Marrying convolution and attention for all data sizes," *arXiv preprint arXiv:2106.04803*, 2021.

[7] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. V. Le, "Rethinking pre-training and self-training," *arXiv preprint arXiv:2006.06882*, 2020.

[8] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2918–2928.

[9] M. Nasr, R. Shokri, and A. Houmansadr, "Machine learning with membership privacy using adversarial regularization," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 634–646.

[10] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[11] Y. Wang, C. Wang, Z. Wang, S. Zhou, H. Liu, J. Bi, C. Ding, and S. Rajasekaran, "Against membership inference attack: Pruning is all you need," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conferences on Artificial Intelligence Organization, 2021.

[12] Y. Wang, J. Deng, D. Guo, C. Wang, X. Meng, H. Liu, C. Shang, B. Wang, Q. Cao, C. Ding, and S. Rajasekaran, "Variance of the gradient also matters: Privacy leakage from gradients," in *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022.

[13] R. Shokri, "Privacy games: Optimal user-centric data obfuscation," *Proceedings on Privacy Enhancing Technologies*, 2015.

[14] L. Song and P. Mittal, "Systematic evaluation of privacy risks of machine learning models," in *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[15] L. Song, R. Shokri, and P. Mittal, "Privacy risks of securing machine learning models against adversarial examples," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 241–257.

[16] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. IEEE, 2018, pp. 268–282.

[17] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," *arXiv preprint arXiv:1806.01246*, 2018.

[18] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[19] V. Shejwalkar, H. A. Inan, A. Houmansadr, and R. Sim, "Membership inference attacks against nlp classification models," in *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021.

[20] N. Xu, B. Wang, R. Ran, W. Wen, and P. Venkitasubramaniam, "Neuguard: Lightweight neuron-guided defense against membership inference attacks," in *Annual Computer Security Applications Conference*, 2022.

[21] C. Dwork, "Differential privacy," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2006, pp. 1–12.

[22] ——, "Differential privacy: A survey of results," in *International conference on theory and applications of models of computation*. Springer, 2008, pp. 1–19.

[23] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016.

[24] X. Zhang, C. Huang, M. Liu, A. Stefanopoulou, and T. Ersal, "Predictive cruise control with private vehicle-to-vehicle communication for improving fuel consumption and emissions," *IEEE Communications Magazine*, 2019.

[25] M. A. Rahman, T. Rahman, R. Laganière, N. Mohammed, and Y. Wang, "Membership inference attack against differentially private deep learning model." *Transactions on Data Privacy*, vol. 11, no. 1, pp. 61–79, 2018.

[26] K. Leino and M. Fredrikson, "Stolen memories: Leveraging model memorization for calibrated white-box membership inference," in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020, pp. 1605–1622.

[27] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," *arXiv preprint arXiv:1506.02626*, 2015.

[28] M. Augasta and T. Kathirvalavakumar, "Pruning algorithms of neural networks—a comparative study," *Open Computer Science*, vol. 3, no. 3, pp. 105–115, 2013.

[29] S. Huang, D. Xu, I. Yen, Y. Wang, S.-E. Chang, B. Li, S. Chen, M. Xie, S. Rajasekaran, H. Liu *et al.*, "Sparse progressive distillation: Resolving overfitting under pretrain-and-finetune paradigm," in *ACL*, 2022.

[30] H. Peng, S. Huang, S. Chen, B. Li, T. Geng, A. Li, W. Jiang, W. Wen, J. Bi, H. Liu *et al.*, "A length adaptive algorithm-hardware co-design of transformer on fpga through sparse attention and dynamic pipelining," in *DAC*, 2022.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017.

[33] M. Gordon, K. Duh, and N. Andrews, "Compressing BERT: Studying the Effects of Weight Pruning on Transfer Learning," in *Proceedings of the 5th Workshop on Representation Learning for NLP*, 2020, pp. 143–155.

[34] S. Chen, S. Huang, S. Pandey, B. Li, G. R. Gao, L. Zheng, C. Ding, and H. Liu, "Et: re-thinking self-attention for transformer models on gpus," in *SC*, 2021.

[35] F.-M. Guo, S. Liu, F. S. Mungall, X. Lin, and Y. Wang, "Reweighted proximal pruning for large-scale language representation," *arXiv preprint arXiv:1909.12486*, 2019.

[36] F. Dalvi, H. Sajjad, N. Durrani, and Y. Belinkov, "Analyzing redundancy in pretrained transformer models," in *Proceedings of the 2020 Confer-*

*ence on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 4908–4926.
[37] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc.
[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012.
[41] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
[42] G. S. Lueker, "Exponentially small bounds on the expected optimum of the partition and subset sum problems," *Random Structures & Algorithms*, vol. 12, no. 1, pp. 51–62, 1998.
[43] M. Zhu and S. Gupta, "To prune, or not to prune: exploring the efficacy of pruning for model compression," *arXiv preprint arXiv:1710.01878*, 2017.
[44] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A Multi-task Benchmark and Analysis Platform for Natural Language Understanding," in *7th International Conference on Learning Representations, ICLR 2019*, 2019.
[45] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao, G. Cormode, and I. Mironov, "Opacus: User-friendly differential privacy library in PyTorch," *arXiv preprint arXiv:2109.12298*, 2021.

## APPENDIX A
## METRIC MIAS

**Correctness based MIA.** This attack infers the membership according to whether a given input data $x$ is classified correctly by the target model $g$ [16]. The intuition is that training data are more likely to be correctly classified than test data. The attack $\mathcal{M}_{\text{corr}}$ is defined as follows, where $I(\cdot)$ indicates the indicator function.

$$\mathcal{M}_{\text{corr}}\,(g; x, y) = I(\operatorname{argmax} g(x) = y) \tag{16}$$

**Confidence based MIA.** This attack determines the membership of the input $x$ by comparing the most significant confidence score with the preset threshold. It is intuitive that the prediction confidence score $g(x)$ for the training data should be close to 1, while the prediction confidence for the test data is usually lower. The attack is first designed by [17] with a single threshold for all classes. [14] further improves it by applying class-wise thresholds to minimize the effect of inter-class confidence differences.The attack $\mathcal{M}_{\text{conf}}$ is defined as follows, where $\tau_y$ represents the threshold for the class $y$.

$$\mathcal{M}_{\text{conf}}(g; x, y) = I(\max g(x)_y \geq \tau_y) \tag{17}$$

**Entropy based MIA.** The entropy based MIA attack is first presented by [17], then followed by an enhanced version that uses the class-wise threshold $\tau_y$ [14]. It is based on the fact that the prediction entropy of the test set should be much larger than that of the training set. It identifies the input $x$ as a member if the prediction entropy is lower than the preset threshold. The attack $\mathcal{M}_{\text{entr}}(f; x, y)$ can be expressed as:

$$\mathcal{M}_{\text{entr}}(g; x, y) = I(-\sum_{i=0}^{k} g(x)_i \log\left(g(x)_i\right) \leq \hat{\tau}_y) \tag{18}$$

Here $\hat{\tau}_y$ denotes the threshold for class $y$, and $k$ is the number of output classes.

**Modified prediction entropy based MIA.** [14] mentioned that prediction entropy attack has a major limitation that it does not contain any labeling information. As a result, only the confidence score is important in the calculation of the prediction entropy attack, without considering the correctness of the prediction. Both a highly correct label with a score close to 1 and a totally wrong predict with an incorrect label score close to 1 can lead to zero prediction entropy values. Modified prediction entropy [14] fixes this issue by: 1) only correct predictions with high probability 1 can be calculated to 0, and 2) incorrect predictions with high confidence scores are calculated to infinity. [14]. Then such modified entropy $\text{ME}(f(x), y)$ is presented as:

$$\begin{aligned}\text{ME}(g(x), y) = &-(1 - g(x)_y)\log\left(g(x)_y\right) \\ &- \sum_{i \neq y} g(x)_i \log\left(1 - g(x)_i\right)\end{aligned} \tag{19}$$

The adversary determines an input data as a member if Eqn. is smaller than the preset class-related threshold–$\check{\tau}_y$ for class $y$. The attack $\mathcal{M}_{\text{Mentr}}(f; x, y)$ is defined as:

$$\mathcal{M}_{\text{Mentr}}\,(g; x, y) = I(\text{ME}(g(x), y) \leq \check{\tau}_y) \tag{20}$$

## APPENDIX B
## ANALYSIS ON FEED-FORWARD NETWORKS

### *A. Analysis on Feed-Forward Networks: A simple layer with activation*

In this case, $g(x) = w \cdot x$, $g^*(x) = \mathbf{u}\sigma(\mathbf{w}^g x)$ . Base on [32], we consider $\sigma$ as ReLU activation function, we have $w = \sigma(w) - \sigma(-w)$. So that the a single ReLU neuron can be written as:

$$x^* \mapsto \sigma\left(wx\right) = \sigma\left(\sigma(wx) - \sigma(-wx)\right) \tag{21}$$

On the other hand, this neuron can be present by a width m two layer network with a pruning matrix $p^*$ for the first layer as:

$$x^* \mapsto \mathbf{u}\sigma\left(\mathbf{p} \odot \mathbf{w}^g x\right) \tag{22}$$

we define $\mathbf{w}^+ = max\{\mathbf{0}, \mathbf{w}\}$, $\mathbf{w}^- = min\{\mathbf{0}, \mathbf{w}\}$, $\mathbf{w}^+ + \mathbf{w}^- = \mathbf{w}^g$. Combine Eq. 21 and 22 we have:

$$x^* \mapsto \mathbf{u}\sigma\left(\sigma\left(\mathbf{p} \odot \mathbf{w}^+ x\right) - \sigma\left(\mathbf{p} \odot -\mathbf{w}^- x\right)\right) \tag{23}$$

Base on Theorem 2, when $n \geq Clog\frac{4}{\epsilon}$, there exist a pattern of $\mathbf{w}$, such that, with probability $1 - \epsilon/2$,

$$\begin{aligned}&\forall\, w^g \in [0, 1], \exists\, p \in 0, 1^n, \\ &s.t.\ \left|w^g - \mathbf{u}\sigma(\mathbf{p} \odot \mathbf{w}^+)\right| < \epsilon/2\end{aligned} \tag{24}$$

Similarly, we have $\mathbf{w}$, such that, with probability $1 - \epsilon/2$,

$$\begin{aligned}&\forall\, w^g \in [0, 1], \exists\, p \in 0, 1^n, \\ &s.t.\ \left|w^g - \mathbf{u}\sigma(\mathbf{p} \odot \mathbf{w}^-)\right| < \epsilon/2\end{aligned} \tag{25}$$

so combine Eq.30 and 25, we have:

$$\begin{aligned}
&\sup|w^g x - \mathbf{u}\sigma(\mathbf{p}\odot\mathbf{w}x)| \\
&\le \left|\sigma(w^g)x - \sigma(-w^g)x - \mathbf{u}\sigma(\mathbf{p}\odot\mathbf{w}^+x) - \mathbf{u}\sigma(\mathbf{p}\odot\mathbf{w}^-x)\right| \\
&\le \sup\left|\sigma(w^g)x - \mathbf{u}\sigma(\mathbf{p}\odot\mathbf{w}^+x)\right| + \\
&\quad \sup\left|\sigma(w^g)x - \mathbf{u}\sigma(\mathbf{p}\odot\mathbf{w}^-x)\right| \\
&\le \epsilon/2 + \epsilon/2 \\
&\le \epsilon
\end{aligned} \tag{26}$$

### *B. The analysis in Entire Feed-Forward Networks*

For general case , $g(x)$ is target model, $g^*(x)$ is defined as Eq.4. so with the probability over $1-\epsilon$, we have:

$$\begin{aligned}
&\sup\|g(x) - \hat{g}^*(x)\| \\
&= \left\|\mathbf{W}_n\mathbf{x}_n - \mathbf{P}_{2n}\odot\mathbf{W}^g_{2n}\mathbf{x}^g_n\sigma(\mathbf{P}_{2n-1}\odot\mathbf{x}^g_{2n-1})\right\| \\
&\le \|\mathbf{W}_n\mathbf{x}_n - \mathbf{W}_n\mathbf{x}^g_n\| + \\
&\quad \left\|\mathbf{W}_n\mathbf{x}^g_n - \mathbf{P}_{2n}\odot\mathbf{W}^g_{2n}\mathbf{x}^g_n\sigma(\mathbf{P}_{2n-1}\odot\mathbf{x}^g_{2n-1})\right\| \\
&\le \|\mathbf{x}_n - \mathbf{x}^g_n\| + \\
&\quad \left\|\mathbf{W}_n\mathbf{x}^g_n - \mathbf{P}_{2n}\odot\mathbf{W}^g_{2n}\mathbf{x}^g_n\sigma(\mathbf{P}_{2n-1}\odot\mathbf{x}^g_{2n-1})\right\| \\
&\le \epsilon/2 + \epsilon/2 \\
&\le \epsilon
\end{aligned} \tag{27}$$

## APPENDIX C
## THE ANALYSIS IN SELF-ATTENTION LAYER: A SIMPLE CASE

the self-attention layer can be present as:

$$\mathbf{Z} = softmax(\frac{QK^T}{\sqrt{(d_k)}})V \tag{28}$$

Where $Q = W^Q x$ , $K = W^K x$ , $V = W^V x$ Here, we start from a simple example. Consider a model $g(x)$ with only one self-attention layer, when the token size of input x is 1, $softmax(\frac{QK^T}{\sqrt{(d_k)}}) = 1$, we have

$$g(x) = W^V x \tag{29}$$

consider $g(x) = \left(\sum_{i=1}^{d} w_i^g\right)x$. and a pruning vector $\mathbf{p} = (p_1, p_2, ..., p_d)$. Base on Theorem 2, when $d \ge Clog4/\epsilon$, there exist a pattern of $p_i w_i^g$, such that, with probability $1-\epsilon$,

$$\begin{aligned}
&\forall\, w_i^g \in [-1,1], \exists\, p_i \in \{0,1\}, \\
&s.t. \left|W^V - (\sum_{i=1}^{d} p_i w_i^g)\right| < \epsilon
\end{aligned} \tag{30}$$

## APPENDIX D
## MIA FORMULATION

For the target machine learning model, we consider the classification model in this work. Let $f$ denotes the target classification model, $x$ denotes a data point, and $g(x)$ denotes the output of $g$ on data $x$. $g(x)$ is a one-hot vector of probabilities of $x$ belonging to $k$ classes. We consider the MIA problems in a black-box condition, which means the adversary can not access the classification model's parameters but can only observe the input and output of the classification model. We assume that the adversary has access to some data records from the training set and the predictions from the black-box DNN target model. Based on the difference between the model's prediction on the training dataset and the non-training dataset, the adversary can determine whether a data record belongs to the model's training dataset or not. We use $f_A$ to denote the adversarial inference model $f_A : x \times y \times g(x) \longrightarrow [0,1]$. $f_A$ takes the feature of the data $x$, the label of the data $y$, and the prediction of the classification model $g(x)$ as inputs. $f_A$ outputs the probability of data $(x,y)$ belonging to the training set $D$ or the non-training set $D'$. The probability distributions of samples in $D$ and $D'$ are $P_D$ and $P_{D'}$, respectively. The gain function of the inference model $f_A$ given the classification model $g$ can be written as:

$$\begin{aligned}
G_g(f_A) = &\mathop{\mathbb{E}}_{(x,y)\sim P_D}\left[\log(f_A(x,y,g(x)))\right] \\
&+ \mathop{\mathbb{E}}_{(x,y)\sim p_{D'}}\left[\log(1 - f_A(x,y,g(x)))\right]
\end{aligned} \tag{31}$$

According to [9], we rewrite the gain function of the inference model in the form of probability distribution:

$$\begin{aligned}
&G_g(f_A) = \\
&\int_{x,y}[P_D(x,y)p_g(g(x))\log(f_A(x,y,g(x))) + \\
&P_{D'}(x,y)p'_g(g(x))\log(1 - f_A(x,y,g(x))]dxdy
\end{aligned} \tag{32}$$

where $D$ is the training set and $D'$ is the non-training set. $p_g$ and $p'_g$ are the probability distribution of the classification model $g$'s output for training data and non-training data.

For a given classification model $g$ and data sampled from a known probability distribution, the optimal determination solution for the inference model $f_A$ is [9], [41]:

$$f_A^*(x,y,g(x)) = \frac{p_g(g(x))}{p_g(g(x)) + p'_g(g(x'))} \tag{33}$$

Therefore, by substituting $f_A^*$ in the Equation 31, the gain function of $f_A^*$ can be written as:

$$\begin{aligned}
&G_g(f_A^*) \\
&= \mathop{\mathbb{E}}_{(x,y)\sim P_D}\left[\log(\frac{p_g(g(x))}{p_g(g(x)) + p'_g(g(x))})\right] + \\
&\quad \mathop{\mathbb{E}}_{(x,y)\sim p_{D'}}\left[\log(1 - \frac{p_g(g(x))}{p_g(g(x)) + p'_g(g(x))})\right] \\
&= -\log(4) + 2\cdot JS(p_f(g(x))||p'_f(g(x)))
\end{aligned} \tag{34}$$

Where $JS(p_g(g(x))||p'_g(g(x)))$ is the Jensen–Shannon divergence between the two distributions. Since $JS(p_g(g(x))||p'_g(g(x)))$ is always non-negative and equals 0 if and only if $p_g(g(x)) = p'_g(g(x'))$, the global minimum value that $G_g(f_A^*)$ can possibly have is -log(4) if and only if $p_g(g(x)) = p'_g(g(x'))$ [41].