Membership Inference Attacks against Vision Transformers: Mosaic MixUp Training to the Defense

Qiankun Zhang*
School of Cyber Science and Engineering
Huazhong University of Science nad Technology
Wuhan, Hubei, China
Key Laboratory of Cyberspace Security, Ministry of Education
Zhengzhou, Henan, China
qiankun@hust.edu.cn

Di Yuan
School of Cyber Science and Engineering
Huazhong University of Science nad Technology
Wuhan, Hubei, China
diyuan@hust.edu.cn

Boyu Zhang
School of Cyber Science and Engineering
Huazhong University of Science nad Technology
Wuhan, Hubei, China
boyu@hust.edu.cn

Bin Yuan
School of Cyber Science and Engineering
Huazhong University of Science nad Technology
Wuhan, Hubei, China
yuanbin@hust.edu.cn

Bingqian Du
School of Computer Science and Technology
Huazhong University of Science nad Technology
Wuhan, Hubei, China
bqdu@hust.edu.cn

Abstract

Vision transformers (ViTs) have demonstrated great success in various fundamental CV tasks, mainly benefiting from their self-attention-based transformer architectures, and the paradigm of pre-training followed by fine-tuning. However, such advantages may lead to significant data privacy risks, such as membership inference attacks (MIAs), which remain unclear. This paper presents the first comprehensive study on MIAs and corresponding defenses against ViTs. Our first contribution is a rollout-attention-based MIA method (RAMIA), based on an experimental observation that the attention, more precisely the rollout attention, behaves disproportionately for members and non-members. We evaluate RAMIA on the standard ViT architecture proposed by Google (ICLR 2021), achieving high accuracy, precision, and recall performance. Further, inspired by another experimental observation on a strong connection between positional embeddings (PEs) and attentions, we propose a novel framework for training ViTs, named Mosaic MixUp Training (MMUT), as a defense against RAMIA. Intuitively, MMUT mixes up private images and public ones at a patch level, and mosaics the corresponding PEs with a global learnable mosaic embedding. Our empirical results show MMUT achieves a much better accuracy-privacy trade-off than some common defense mechanisms. Extensive experiments are conducted to rigorously evaluate both RAMIA and MMUT.

CCS Concepts

• **Security and privacy**; • **Computing methodologies** → **Machine learning**;

Keywords

Membership inference attacks, Vision transformers

ACM Reference Format:
Qiankun Zhang, Di Yuan, Boyu Zhang, Bin Yuan, and Bingqian Du. 2024. Membership Inference Attacks against Vision Transformers: Mosaic MixUp Training to the Defense. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS '24), October 14–18, 2024, Salt Lake City, UT, USA*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3658644.3690268

1 Introduction

Large Vision Models are on their way!

Large Language Models (LLMs) such as GPT [4] and LLaMA [53] have emerged as a game-changing force in artificial intelligence. Large Vision Models (LVMs) revolution is later but arriving. For example, a very recent series of work [22, 55] since Bai et al. [3] makes breakthroughs in training LVMs without any linguistic data. Besides the power of data, the *transformer*-based architecture plays a vital role in their successes. In this paper, we study *vision transformers* [19] (ViTs), the skeletons of LVMs, from a security perspective.

*Qiankun Zhang is the corresponding author.

CCS '24, October 14–18, 2024, Salt Lake City, UT, USA

ACM ISBN 979-8-4007-0636-3/24/10
https://doi.org/10.1145/3658644.3690268

ViTs are gaining remarkable popularity as an alternative to convolutional neural networks (CNNs) across various computer vision tasks, such as image classification [19, 52], segmentation [5, 59], and object detection [6, 65]. The key ideas behind these breakthroughs can be characterized as two main aspects: (a) the transformer architectures based on the *self-attention* mechanism, which enable the model to learn global features of images; (b) initially *pre-training* on an upstream (un)labeled dataset through (self-)supervised learning, followed by *fine-tuning* on a local labeled dataset for a downstream task [19]. Most previous literature on ViTs targets on developing various pre-training objectives [9, 24, 60] or variants of self-attention mechanisms [20, 56, 63, 64]. However, such efforts in both aspects can lead to serious privacy risks of training data, which haven't been carefully studied yet. For example, ViTs may suffer *membership inference attacks* (MIAs) [47], which aim to infer whether an input image is in the *pre-training dataset*.

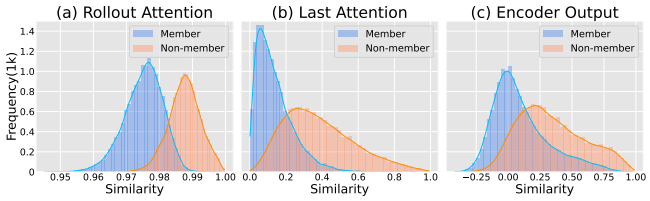


Figure 1: Histograms for the number of members vs. non-member images across different cosine similarity scores.

The principal component of a ViT is a *transformer encoder*, a multi-blocked neural network that processes input images to encode them as representative features containing global contextual information. The output of the ViT encoder can be used for various downstream tasks, such as image classification, where an extra classifier is added to the final block of the encoder as a decoder. To achieve high accuracy and wide scalability to various downstream tasks, the ViT encoder is typically pre-trained by organizations with substantial computational resources, such as Google and OpenAI. Subsequently, the entire ViT encoder (with or without a decoder) plus a model card, which describes its architecture, training dataset, training algorithm, etc., will be released in public on websites such as Hugging Face [1]. Such applications inspire our first research question: *given an incomplete ViT, say, a ViT encoder, can and how can an attacker proceed with MIAs?*

To answer the above, the only existing successful attempt is the EncoderMI proposed by Liu et al. [35]. EncoderMI attacks a ViT encoder, CLIP [42], based on the encoder output in a black-box manner, meaning that only the encoder output is known to an attacker. Their results are exemplary but yet preliminary in attacking ViTs: (a) in a restricted contrastive learning setting, which pre-trains an encoder using data augmentation in a self-supervised fashion on unlabeled datasets; and (b) under a strict black-box assumption, which may violate the aforementioned practical applications where downstream tasks can access full information on ViT encoders, in other words, in a *white-box* manner.

[1]A popular open-source hub for machine learning models, https://huggingface.co/models

In contrast, we study a more typical supervised pre-training for ViTs on labeled datasets, and focus on the core of a ViT encoder, the self-attention mechanism. In general, a ViT encoder starts by dividing an input image into fixed-size patches, and then linearly projects them into a sequence of vectors named patch embeddings. To understand the spatial arrangement of image patches, *positional embeddings* (PEs) are added to patch embeddings. The self-attention mechanism allows to integrate information across all patches, capturing global relationships in the image. That's why ViTs may surpass CNNs, which primarily focus on local features. However, we observe that the *attention*, an intermediate feature representation matrix to weigh the importance of different patches relative to each other, can lead to a membership leakage through an experiment.

We pre-train a standard ViT model provided by Google [19] from scratch on a *member dataset*. Then add an independent Gaussian noise to each training image and forward both images into the encoder. We compute a cosine similarity between their corresponding attention maps, as well as encoder outputs. Figure 1(b) and Figure 1(c) present histograms for the number of images across different similarity scores, respectively. We observe the distribution differences between members and non-members do exist, albeit not as pronounced. The coming challenge is: *Can and how can we enhance such differences and further apply them to MIAs?*

We adopt the *attention rollout* technique [2], a method to quantify a transformer encoder's attention flow. In a nutshell, attention rollout aggregates attention maps from the first to the final encoder blocks, producing a rollout attention (RA) map. The disparity in RA maps' similarity distribution is noticeably enhanced, as shown in Figure 1(a). Based on such observation, we propose a *rollout-attention-based MIA* (RAMIA) against ViTs. RAMIA follows the framework of *shadow training* [47], and uses a cosine similarity between RA maps for images before and after adding noises as a feature vector as a metric to distinguish members and non-members. We compare two variants of RAMIA depending on whether it trains a binary classifier [47] or determines a carefully selected threshold [43] for the inference. To evaluate RAMIA, we conduct experiments on CIFAR10, CIFAR100, ImageNet100, and ISIC2018 datasets [2], achieving higher accuracy, precision, and recall than baseline models, including EncoderMI. See Section 3 for details.

Subsequently, we question *how to defend RAMIA* in Section 4. We first observe a strong connection between PEs and RAs through an intuitive experiment. In particular, if we fix all PEs to identical learnable parameters, the difference in RA similarity before and after noise addition is significantly reduced compared to the standard ViT training method, where each PE is trained separately. Inspired by such findings, we design a novel defense method for training ViT, *Mosaic MixUp Training* (MMUT). MMUT mixes up *public dataset* and *private dataset* by replacing a certain percentage of patches for each training image. PEs corresponding to replaced patches will be replaced by a global learnable *mosaic embedding* when forwarding this image. By doing so, in the backward process, only the mosaic embedding, as well as those non-replaced PEs (except for replaced PEs), will be updated. We compare our MMUT to various typical defense mechanisms in literature, including label smoothing

[2]We consider four datasets in experiments for diversity. CIFAR10, CIFAR100, and ImageNet100 are standard image classification tasks in CV. ISIC2018 contains dermoscopy images used for disease classification.

[51], DP-SGD [1], RelaxLoss [10], and adversarial regularization [40]. Besides, experiments consider whether an attacker is *adaptive*, meaning that whether it knows the defense methods. Empirical results show the remarkable effectiveness of our MMUT. We highlight that our MMUT borrows the idea of mixing up images for data augmentation [48], however, a greater contributor to MMUT's success might be our special treatment on PEs, i.e., the mosaic embedding. To justify it, we compare MMUT with several standard image augmentation methods in literature, including merely mixing up images without mosaic embeddings. Empirical results show their less effectiveness in defending against RAMIAs. Nevertheless, coupling PE mosaics with image MixUp contributes to maintaining (or even surprisingly enhancing) the prediction accuracy of ViTs despite a high level of privacy. See Section 4 for details.

Practical Applications of Our Methods. Research in MIAs against ViTs is necessary and meaningful. One of the most significant applications of ViTs is medical image classification [45]. For instance, telemedicine platforms, such as Teladoc Health, use ViTs to identify patterns in chest X-rays that are indicative of conditions like pneumonia. In this scenario, a hospital might utilize MIAs to ensure that the patient's sensitive medical images used in pre-trained ViTs are handled in compliance with HIPAA and GDPR regulations, thereby maintaining patient confidentiality and data security. Another possible application is in content moderation on social media platforms [57]. For example, Facebook uses ViTs to moderate content by analyzing user-uploaded images. However, there's a risk that these models may inadvertently learn sensitive or private user data. Using MIAs, an independent watchdog organization could audit whether personal images uploaded by users have been used without consent to train these models. In addition, to ensure a more realistic fit, the ViT architecture studied in this paper is an official model [3] released by Google [19]. We pre-train it from scratch on CIFAR10, CIFAR100, ImageNet100, and ISIC2018 in experiments. Moreover, to see how our RAMIA performs in a ViT encoder directly downloaded from the website, we also attack against its original model with pre-trained parameters provided in Section 3.5.

Summary of Our Contributions:

- We provide the first comprehensive study on MIAs against ViTs.
- We propose and rigorously evaluate RAMIA, a white-box MIA method against ViT encoder, focusing on the self-attention mechanism.
- We propose MMUT, a training framework for ViTs. Extensive experiments show its effectiveness in accuracy-privacy trade-off preserving.

2 Preliminaries and Related Work

2.1 Vision Transformer

ViT [23, 30] is first proposed by Dosovitskiy et al. [19], whose core idea is to segment an image into a series of patches (typically 16x16 pixels) and then process these patches as a sequence of 'tokens' like words in NLP. The transformer architecture is trained to capture the long-range dependencies between these tokens. The self-attention mechanism helps ViTs effectively focus on the interactions between any parts of an image and achieve excellent performance in many CV tasks.

Figure 2 demonstrates a complete ViT model for a classification task. The most crucial part is the *transformer encoder*. The Transformer encoder initially divides the input image into multiple small patches and flattens each patch into a one-dimensional vector, named patch embedding, through a linear projection. Note that for a classification task, a special vector, known as a *class token* (CT), is added to the sequence of these patch embedding vectors. CT is typically used to capture the global information of an entire image, and after the final block of the ViT encoder, CT is the *only input* to the classifier because it has aggregated full information (both patch-wise and position-wise) from the entire image, though the series of transformer layers. Subsequently, a learnable *positional embedding* (PE) vector is then added to each patch embedding including CT. This combination, now imbued with positional information, is then trained through a *multi-head self-attention* (MSA) mechanism, and will be dynamically refined through learning and optimization with training data.

Multi-head Self-attention. MSA is the core component of the transformer encoder, containing *N transformer blocks*. In each block, MSA aggregates sequential tokens as:

$$t_j = \sum_i \mathbf{A}_i \mathbf{V}_{ij} = \sum_i \text{SoftMax}(\frac{\mathbf{QK}}{\sqrt{d}})_i \mathbf{V}_{ij} \,, \tag{1}$$

where $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are query, key and value matrices, respectively. s is the dimension of the query and key, and t_j is the j-th output token. We highlight that $\mathbf{A}$ is called the *attention map*, a matrix of scores indicating the relevance of each input embedding to others. Our attacks are based on such attention maps, more precisely, on the *rollout attention* (RA). Multiple attention heads separately run the attention mechanism in parallel, making the model focus on different parts of the input simultaneously and capture various aspects of the information. The outputs from each head are concatenated back together. The concatenated output is then passed through another linear transformation to produce the final output of the multi-head attention mechanism.

Classifier. In general, ViTs do not include a 'decoder' part. For a classification task, a classifier will be added at the end of the final encoder blocks. The classifier generally contains a multi-layer perceptron head (MLP), which usually consists of multiple layers of fully connected neural networks. The output is a *prediction vector* on every class label.

Rollout Attention. Attention rollout proposed by Abnar and Zuidema [2] is a methodology for quantifying the flow of attention of a transformer encoder, by tracing the progression of attention from initial to final block. Specifically, in each transformer block, an attention map $\mathbf{A}$ is computed. Each $\mathbf{A}_{ij}$ indicates how much attention flows from token j in the previous block to token i in the next block. An identity matrix $\mathbf{I}$ is then added to the layer attention $\mathbf{A} + \mathbf{I}$ to symbolize the unchanging attention mapping that results from the incorporation of residual connections between blocks. Consequently, the RA matrix, denoted as $\mathbf{RA}_\ell$ at block ℓ, can be computed recursively by matrices multiplication as:

$$\mathbf{RA}_\ell = (A_\ell + I)\mathbf{RA}_{\ell-1} \,. \tag{2}$$

[3] https://huggingface.co/google/vit-base-patch16-224-in21k

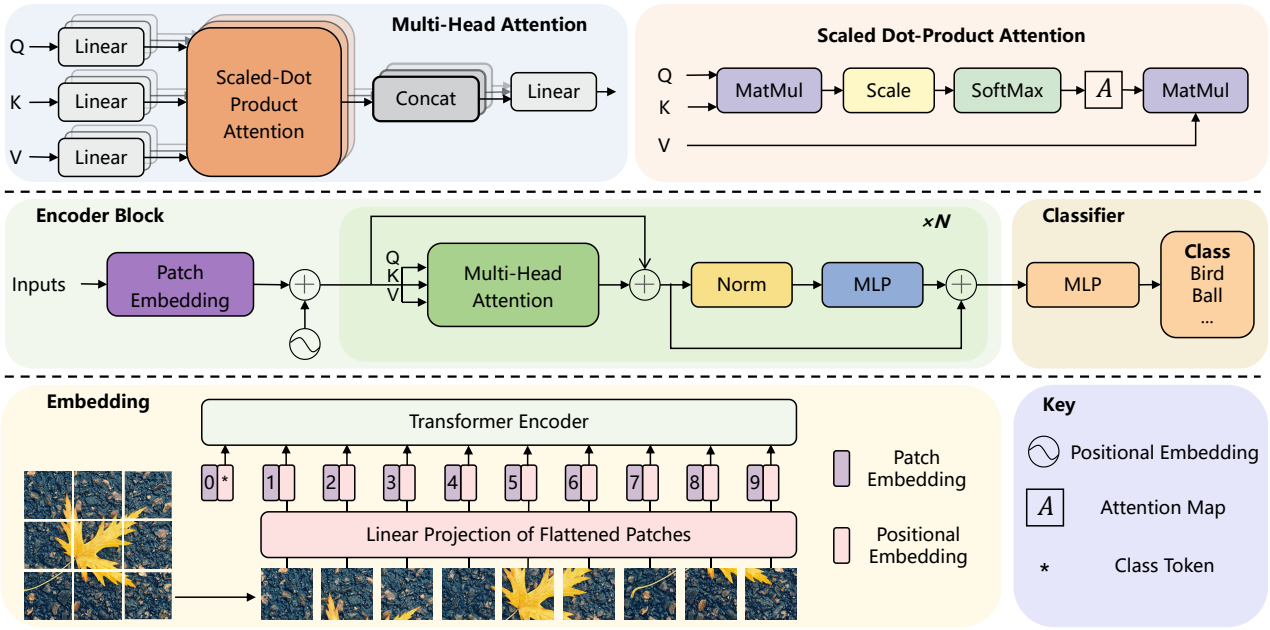


Figure 2: Overview of vision transformer achitecture.

Further, recall that transformer encoders often have multiple attention heads which separately compute self-attention in parallel. In this case, several different methods are utilized to compute the maximum (Max), minimum (Min), and average (Mean) of the heads. Abnar and Zuidema [2] suggest the best choice may vary among different tasks, and thus we shall evaluate all of them in Section 3.4.

Remarks. In our RAMIA, instead of using attention maps, we use the global rollout attention, meaning that all attention maps in every block are multiplied cumulatively according to Eqn. (2), to construct the feature vectors for training shadow encoders.

Most existing work on ViTs aims to enhance performance in various vision tasks, primarily focusing on three means: boosting the concept of locality within images [9, 24, 60], refining the self-attention mechanism [20, 56, 63, 64], and innovating in architectural design [14, 18, 21, 24, 28, 34, 37, 52]. Besides EncoderMI, the privacy risks in ViTs have been reported by Lu et al. [38], who point out a gradient leakage risk of self-attention-based mechanism. To the best of our knowledge, we provide the first comprehensive study on both membership inference attacks and defenses against ViTs.

2.2 Membership Inference Attack

MIAs on machine learning (ML) models aim to determine whether a specific sample is included in the training set of the target model. Depending on how the attack model is built, there are two main types of MIA strategies: those that train a binary classifier for members and non-members (*classifier-based*), and those that rely on a carefully-designed metric to separate members and non-members directly (*threshold-based*). For example, Shokri et al. [47] first propose a *shadow training* technique, which trains a shadow model to imitate the behavior of the target model using a shadow dataset. A classifier is then trained using the output of the shadow models as a feature. Such output-based classification often lies in *black-box* attacks. In contrast, *white-box* MIAs can get all the information, including the prediction vector, the intermediate computation (e.g., the feature map) at each hidden layer, the loss, and the gradient of the loss with respect to the parameters of each layer of the input image. An example is from Salem et al. [44], who use either the maximum prediction confidence or prediction loss of the target model as a metric. A threshold is then determined to classify members and non-members, named threshold-based MIAs. Recently, Pang et al. [41] employ model gradients in MIAs, and argue that they provide a more profound feature representation. Our RAMIA basically follows the idea of shadow training. We design both classifier-based and threshold-based variants of RAMIA, and compare them with the aforementioned attack methods. Numerous studies have applied MIAs to various models, including classification models [7, 36, 58], generative adversarial networks (GANs) [11, 25, 26], and diffusion models [8, 31, 41]. Readers may refer to [27] for a survey.

The existing work most relevant to ours is EncoderMI [35], a black-box MIA towards encoders in contrastive learning, which assumes the encoder is sufficiently overfitted. EncoderMI highly relies on such an assumption and observes a disparity in the outputs of the encoder for members and non-members. Although most of their experiments are conducted on CNNs, they also report results on CLIP with a ViT encoder. Compared to EncoderMI, our work differs in: (a) not relying on the contrastive learning assumption that may not be dominant in realistic scenarios where ViT encoders are pre-trained on labeled datasets; (b) assuming a more applicable white-box access. Full information on ViT encoders can be used for attacks; (c) a new defense method by a unified training framework.

2.3 Defenses against MIAs

Various defenses [13, 29, 33, 46, 47, 49, 62] against MIAs have been well studied. One of the most widely used privacy-preserving techniques is *differential privacy* (DP) [1], which provides a promising defense against MIAs by adding noise to the gradient (DP-SGD) or parameter during model training. DP faces a significant trade-off between accuracy and privacy: high degree of security may lead to poor performance in accuracy. Another defense more specific to MIAs is called *regularization* [25, 26, 33, 50], a more efficient way to ensure the privacy of overfitted models. It has been reported that most of the methods utilized to enhance the generalizability of an NN model can improve its privacy at the same time. For example, *label smoothing* [51] is proved to be effective in mitigating MIAs by minimizing the behavioral discrepancy between training and test data; *adversarial regularization* [40] incorporates membership inference gain into the target model's objective function, balancing classification loss with attack model accuracy; *RelaxLoss* [10] designs a more achievable learning objective, achieving easy implementation and negligible overhead. Details of these methods applying to ViTs will be discussed in Section 4.3. Our MMUT will view them as baselines for comparisons in Section 4.4.

3 Rollout Attention MIA

3.1 Attacker Assumptions

Recall that the objective of an attacker is to infer whether a given image x belongs to the pre-training dataset of a transformer encoder of a ViT (which we call the *target encoder* of a *target ViT*). in other words, to infer whether x is a member or non-member. In this paper, we assume that the attacker has *white-box* access to the encoder of the target ViT, with full information of (a) **the distribution of images for training**, meaning that the attacker can set up a *shadow dataset* with the same distribution as the *target dataset*; and (b) **the architecture and the learned parameters of the encoder**, meaning that the attacker can train a *shadow model* with the same architecture as the target encoder (without any information on remaining parts of the target ViT, say, the ViT classifier); (c) **how the target model is trained,** for example, using either a supervised or self-supervised learning on a labeled or unlabeled dataset, respectively. Note that such a threat model accurately captures the practical applications in Hugging Face, where information of a ViT encoder is recorded as a model card.

3.2 RAMIA Design

Overview. Figure 3 presents our RAMIA method, which follows the technique of *shadow training* proposed by Shokri et al. [47]. Recall that for a given image I, we forward it to the target encoder and tackle its rollout attention. Further recall that in Section 1, our experimental observation in Figure 1 indicates that the rollout attention maps are more susceptible to the influence of adding noises to a member of the target ViT encoder than a non-member. Therefore, inspired by such observation, our RAMIA classifies an input image I as a member when the rollout attention maps generated by the target encoder are significantly different for an image x and its noise-added counterpart $\tilde{x}$. Specifically, RAMIA follows the following three steps: (a) **shadow encoder pre-training.** We partition the attacker's shadow dataset, which has the same distribution as the target dataset, into two disjoint subsets, named *shadow members* and *shadow non-members*, respectively. The attacker then pre-trains a *shadow encoder* using shadow members; (b) **feature vectors construction.** We construct a feature vector for each image to be inferred using the similarity scores of rollout attention maps between the original input image and some of its neighbor images when independent random noises are added; (c) **membership inference.** We apply two common attack methods in literature to show the effectiveness of our RAMIA, that is, a *binary-classifier-based MIA* (named RAMIA-Classifier) and a *threshold-based* MIA (named RAMIA-Threshold). Specifically, RAMIA-Classifier trains a binary *inference classifier* to predict member/non-member for input based on the feature vectors obtained in (b), while RAMIA-Threshold directly determines a proper threshold θ. Only if the mean value of each component of the feature vector is smaller than θ, the input is inferred as a member, and vice versa. Algorithm 1 formally describes our RAMIA, which we will explain in detail in the rest of this subsection.

(a) Shadow Encoder Pre-training. Initially, the attacker splits a shadow dataset D^s, which is assumed to be independently and identically distributed to the target dataset D^t, into two disjoint subsets of shadow members D^s_{mem} and non-members $D^s_{\text{non-mem}}$. D^s_{mem} is then used to train a shadow ViT encoder E^s, which has the identical architecture to the target ViT encoder E^t.

(b) Feature Vectors Construction. For each input image x in the shadow dataset D^s, we compute its rollout attention, denoted as $\text{RA}(\text{E}^s, x)$, through the shadow encoder E^t. Subsequently, we construct n neighbor images of x, denoted as $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n$, by adding independent Gaussian noises as stated in Appendix A to x, and compute each $\text{RA}(\text{E}^s, \tilde{x}_i)$ for $\forall i \in [n]$. Next, a cosine similarity, denoted as $\text{sim}(x, \tilde{x}_i)$, between $\text{RA}(\text{E}^s, x)$ and $\text{RA}(\text{E}^s, \tilde{x}_i)$ is computed for each neighbor image $\tilde{x}_i$. Total n neighbor images produce a similarity feature vector $V^s(x)$ for input x, where $V^s(x) = [\text{sim}(x, \tilde{x}_1), \text{sim}(x, \tilde{x}_2), \ldots, \text{sim}(x, \tilde{x}_n)]$.

(c) Membership Inference. We adopt both binary-classifier-based and threshold-based techniques for attack.

- **RAMIA-Classifier:** we train the RAMIA-Classifier using the feature vectors V^s of the shadow dataset as inputs, and a binary label on whether they are shadow members or non-members. During the inference process, an inferred image is forwarded to the target encoder E^t, and produces a feature vector V^t. The inference classifier takes V^t as an input and predicts the membership of the inferred image.
- **RAMIA-Threshold:** the key insight is how to choose a proper threshold θ. We choose θ that maximizes the accuracy of predictions among all images in D^s. Subsequently, for an inferred image X, we forward it in the target encoder E^t and compute the corresponding feature vector $V^t(x)$. x is inferred as a member when the mean value of each component of $V^t(x)$ is smaller than θ, that is, when $\frac{||V^t(x)||_1}{|V^t(x)|} < \theta$, where $||\cdot||$ denotes ℓ_1-norm and $|\cdot|$ denotes the cardinality of a vector.

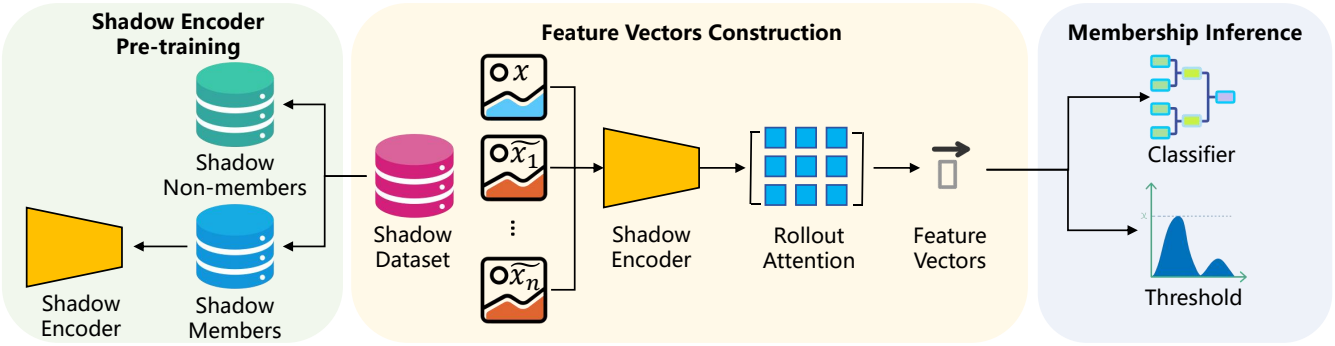


Figure 3: Overview of rollout-attention-based membership inference attack(RAMIA).

Algorithm 1 RAMIA

Require: Target Encoder E^t; Shadow Encoder E^s; Shadow Dataset D^s_{mem} and $D^s_{\text{non-mem}}$; Image X to be inferred.
Ensure: Indicator $\mathbb{I}$: True (member); False (non-member)

for each x in $D^s_{\text{mem}} \cup D^s_{\text{non-mem}}$ **do**
  $\mathrm{RA}(x) \leftarrow \mathrm{AttentionRollout}(x, \mathrm{E}^s)$
  for each i in $1, ..., n$ **do**
    $\tilde{x}_i \leftarrow \mathrm{AddNoise}(x)$
    $\mathrm{RA}(\tilde{x}_i) \leftarrow \mathrm{AttentionRollout}(\tilde{x}_i, \mathrm{E}^s)$
    $V^s(x)[i] \leftarrow \mathrm{Similarity}(\mathrm{RA}(x), \mathrm{RA}(\tilde{x}_i))$
  end for
end for
(a) RAMIA-Classifier
$\mathrm{Classifier} \leftarrow \mathrm{Train}(V^s)$
$\mathbb{I} \leftarrow \mathrm{Classifier}(V^t(X))$
(b) RAMIA-Threshold
$\theta = \mathrm{GetBestThreshold}(V^s)$
if $\frac{||V^t(x)||_1}{|V^t(x)|} < \theta$ **then**
  $\mathbb{I}$ = True
else
  $\mathbb{I}$ = False
end if
return $\mathbb{I}$

3.3 RAMIA Evaluation: Experiments Setup

Model. We evaluate the effectiveness of our RAMIA using a ViT encoder that comprises 12 transformer blocks, with 6 attention heads each. Unless stated otherwise, we default to using the average (Mean) of the heads when computing the rollout attention for multiple heads as we introduced in Section 2.1. But we shall compare it with the other two methods, including Max and Min, in Section 3.4.

Datasets. We consider four datasets in experiments for diversity as shown in Table 1. CIFAR10, CIFAR100, and ImageNet100 are standard image classification tasks in CV. ISIC2018 contains dermoscopy images used for disease classification.

Baselines. There are seven baseline attacks **(a-g)** in our experiments. First, to show the effectiveness of our use of rollout attention as a feature instead of a single attention map, we adapt our RAMIA to an attention-based MIA, which simply uses the attention map of the last transformer block for feature vector construction, and keeps other details the same as RAMIA. We call these baselines **(a) AMIA-Classifier** and **(b) AMIA-Threshold**, respectively.

Table 1: Dataset details including number of images, number of class labels, and size of images.

Dataset	Size	Categories	Image Size
CIFAR10 [32]	60000	10	$32 \times 32 \times 3$
CIFAR100 [32]	60000	100	$32 \times 32 \times 3$
ImageNet100 [17]	130000	100	$224 \times 224 \times 3$
ISIC2018 [15, 54]	12180	7	$224 \times 224 \times 3$

We next compare our RAMIA to **(c) EncoderMI** [35] directly, which also targets on attacking encoders. Note that EncoderMI uses the similarity of the encoder's output as a feature vector in CNN models (ResNet18). In the encoder of ViT, we compute the similarity of the class token, which serves as input for the decoder of ViT, as a feature vector for training the inference classifier. Besides, to evaluate the effectiveness of our idea in using the cosine similarity between the rollout attention of an input image and its neighbor images as a feature vector, we compare RAMIA to an encoder attack which uses the rollout attention as a feature vector for training the inference classifier directly. We call it **(d) Baseline-Classifier**. In particular, Baseline-Classifier is different from RAMIA in the feature vectors construction. Instead of using $V^s(x)$ to train a classifier, Baseline-Classifier trains it using the rollout attention maps $\mathrm{RA}(x)$ of each image x through the shadow encoder as input.

In addition, other MIA techniques in literature utilize prediction vectors of complete models for their attacks, so it is not straightforward to make direct comparisons. However, we aim to adapt or extend these prevailing MIA techniques to the encoder attacks. In particular, we complement the target encoder into a complete ViT, by adding a randomly initialized classifier and fine-tuning ViT on the shadow dataset. Note that the pre-trained target encoder is kept *frozen* through the fine-tuning process. We call such baselines *FullViT*. By doing so, we can get a prediction vector $P(y|x)$, the loss $\mathcal{L}(y, P(y|x))$, and the gradient of the loss concerning the parameters $\frac{\partial \mathcal{L}}{\partial \theta}$ for each input image x. FullViT trains a binary classifier using each of three metrics as a feature vector for member and

Table 2: Average accuracy, precision, and recall (%) of our methods for the target encoder pre-trained on four datasets, which are CIFAR10, CIFAR100, ImageNet100 and ISIC2018. C is for classifier, T is for threshold, P is for prediction, L is for LIRA, and G is for gradient. In this experiment, our multi-head fusion method is average (Mean).

Attack methods	CIFAR10			CIFAR100			ImageNet100			ISIC2018		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
RAMIA-T	**88.43**	**86.25**	**91.43**	**87.94**	**85.08**	**88.61**	**91.89**	**89.86**	**94.44**	**92.33**	**87.64**	**94.43**
RAMIA-C	**91.42**	**93.83**	**89.52**	**90.24**	**92.44**	**86.98**	**91.76**	**95.96**	**87.18**	**91.84**	**92.23**	**89.98**
AMIA-T	75.16	69.33	89.49	74.9	69.34	77.25	83.96	77.86	84.43	79.3	67.22	81.91
AMIA-C	75.77	62.97	82.35	80.91	88.54	78.9	84.36	87.34	80.73	80.94	77.19	84.5
EncoderMI	70.37	71.84	67.73	83.89	89.34	65.5	84.11	90.33	81.5	82.87	84.61	81.46
Baseline-C	63.86	68.05	62.79	52.63	31.51	54.55	55.22	50.4	55.78	52.31	50.81	53.7
FullViT-P	54.05	54.09	53.58	50.55	49.76	73.19	51.02	49.88	65.67	51.28	50.13	51.97
FullViT-L	55.29	52.21	58.54	53.06	50.06	55.92	51.78	50.64	53.65	53.36	54.47	51.61
FullViT-G	51.33	54.17	57.44	50.66	52.43	51.52	50.28	50.12	52.99	51.85	50.7	52.88

non-member, named **(e) FullViT-Prediction**, **(f) FullViT-LIRA** [7], which use a principled likelihood ratio test with Gaussian likelihood estimates and per-example difficulty scores to attack, and **(g) FullViT-Gradient**, respectively.

3.4 RAMIA Evaluation: Experimental Results

Effectiveness of RAMIA. Table 2 shows the attack performance of our RAMIAs and seven comparison methods from (a) to (g) on four different datasets: CIFAR10, CIFAR100, ImageNet100 and ISIC2018. The key observations are:

- The FullViT attack methods behave as random guessing because their accuracies are close to 50%. The results are not surprising because the classifier is trained on the shadow dataset, whose output features fail to capture whether the encoder is overfitted for images in the trarget dataset.
- EncoderMI, as well as the Baseline-Classifier, is also effective, but behaves worse than our RAMIA, which validates the effectiveness of using the rollout attention instead of the output of the encoder (class token). Readers may wonder if EncoderMI assumes black-box access to the encoder, which is more strict than RAMIA. Are the results comparable? Existing literature [39] reports and validates that a white-box attack may not be easier than a black-box attack as imagine. Besides, in practical scenarios, the encoders of ViTs are released on the website, such as Hugging Face. So, a white-box assumption is decent.
- Attention-based MIAs (AMIA-Threshold and AMIA-Classifier) behave worse than our RAMIAs, but yet beat EncoderMI in some cases. The reason is that compared to the attention in the last transformer block, rollout attention captures cumulative attention through a whole forwarding process, making it more sensitive to noises.

Besides, Figure 4 presents the ROC curves and AUC values of different attacks, indicating our RAMIA-Threshold outperforms other attacks significantly on all four datasets.

The impact of attention rollout methods. Recall that to handle multiple attention heads, the attention rollout technique aggregates the RA matrix of each head by computing the maximum (Max), minimum (Min), or average (Mean) of them. All three computations are entry-wise. Figure 5 shows their impacts on inference accuracy varying on four datasets. Among the three methods, the highest accuracy is achieved by Max on ImageNet100, because Max may enhance the disparity of feature maps for members and non-members. On the contrary, for a similar reason, Min behaves the worst because vanishing values in RA produce indistinguishable features. As a compromise, the Mean gives relatively stable accuracy among the four datasets.

The impact of model structures. Two important structural parameters of a ViT encoder are the number of encoder blocks and the number of attention heads in each block. We examine their impacts by fixing one and changing the other. Figures 6(a) and 6(b) show the impacts of encoder blocks and attention heads, respectively. Our main findings are:

- For different numbers of encoder blocks, accuracy does not seem to vary too much, meaning that more blocks do not weaken the performance of our RAMIA. That is the point why we use the RA maps, which aggregate all attention information in every block, for the feature construction.
- We notice a remarkable correlation between the number of attention heads and the inference accuracy of RAMIA. In particular, as the number of attention heads increases, accuracy rises first and falls later. More heads do provide much more information for constructing features. However, when there are too many, the aggregation policy may fail to capture the most salient features from RA maps. That is perhaps why the ViTs used in practice (with typically 6 heads) do not have too many attention heads in their encoders.

Other impacts. We also evaluate on other factors that may influence the performance of MMUT, including the number of neighbor images (denoted as n) with independent noises added and different choices of the metric when computing the similarity of RA maps, e.g., Pearson correlation coefficient (PCC). Details are reported in Appendix C. We highlight our main findings are: (1) in general, the larger n induces a higher attack accuracy; (2) both the cosine similarity and PCC are effective in RAMIAs.

3.5 Applying RAMIA to Real-world ViT Encoders

ViT encoders used in previous experiments mostly use an official model architecture released by Google. To further strengthen

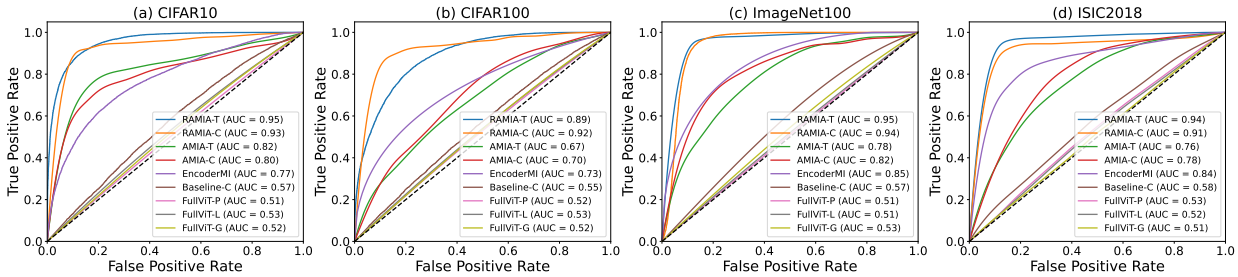


Figure 4: The ROC curves of nine attack methods on four datasets, CIFAR10, CIFAR100, ImageNet100 and ISIC2018.

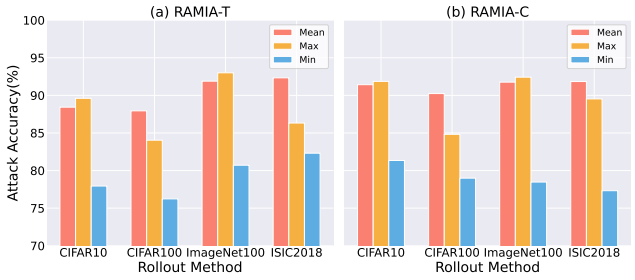


Figure 5: The impact of different attention rollout methods (Max, Min, and Mean) on the accuracy of our RAMIA on CIFAR10, CIFAR100, ImageNet100 and ISIC2018, either based on a threshold (T) or a classifier (C).

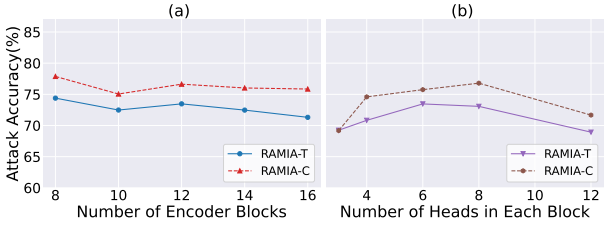


Figure 6: The impact of RAMIA on different structural models: (a) set the number of encoder blocks as 8, 10, 12, 14, 16, **fixing the number of heads as** 6**; (b) set the number of heads in each block as** 4, 6, 8, 12**, fixing the number of blocks as** 12**.**

the power of RAMIA, we conduct a RAMIA against the complete encoder with pre-trained parameters released by Google, named ViT-base. ViT-base is pre-trained on ImageNet-21k (14 million images, 21,843 classes) at resolution 224x224. We use ImageNet100 as the shadow dataset, which is i.i.d. to ImageNet21k. In addition, we also experiment on the non-i.i.d. shadow dataset to capture the scenarios where the target dataset is not released to the public. Table 3 shows the corresponding attack accuracy, precision, and recall. Results show the effectiveness of RAMIA, although accuracies are slightly lower than those presented in Table 2. This is because ImageNet100 only contains images of 100 classes, while ImageNet21k is considerably larger. We also point out that even if pre-trained on non-i.i.d. shadow datasets, our RAMIAs are still effective, with the lowest accuracy of 77.21% on ISIC2018.

Table 3: Attacking results (%) for Google's ViT-base.

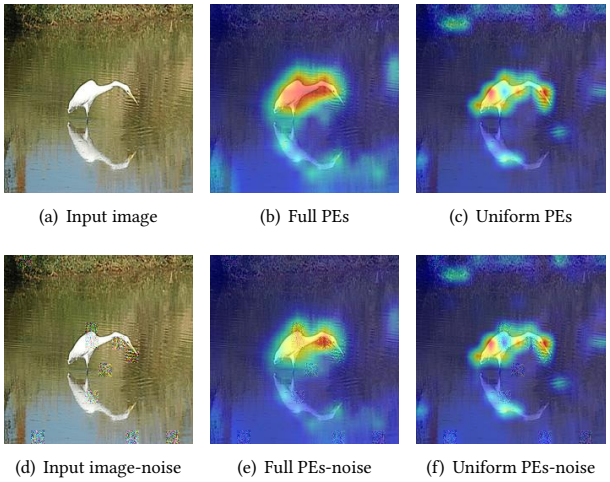
Method	Dataset	Accuracy	Precision	Recall
RAMIA-T	CIFAR10	78.56	62.21	90.86
	CIFAR100	80.87	66.04	87.73
	ImageNet100	81.35	75.91	84.36
	ISIC2018	77.21	77.91	78.36
RAMIA-C	CIFAR10	77.57	91.39	71.6
	CIFAR100	82.47	67.66	93.45
	ImageNet100	85.87	83.78	88.97
	ISIC2018	77.95	70.91	89.36

4 Mosaic MixUp Training against RAMIA

This section introduces our defense method against RAMIA in this section, which we call Mosaic MixUp Training (MMUT) for ViTs. MMUT is a novel unified framework for pre-training ViTs from scratch. Intuitively, MMUT enhances the robustness of the model to image noise by integrating private datasets with public datasets in a patch-level manner. Concurrently, the positional embeddings (PEs) of the corresponding replaced patches, through a shared training parameter scheme in forwarding this image, serve a dual purpose: it differentiates between private training data and integrated patches, and enhances the model's resilience against inference attacks. This approach not only fortifies the model against adversarial RAMIAs but also ensures that the prediction accuracy does not significantly diminish (or even increases). Before a thorough explanation of MMUT, we first introduce an intuitive experiment on the relationship between PEs and rollout attention (RA).

4.1 An Intuitive Experiment

To see how PEs affect RA maps for images in the training dataset (members), we compare two different ways to train a ViT from scratch using a member dataset: (a) updating parameters for each image patch separately as those standard training methods in ViT literature; (b) fixing the PE for all patches to identical learnable parameters. Compared to (a), (b) nullifies the spatial positional information typically conveyed by PEs, which is always viewed to be crucial in ViTs' success. We visualize images used for training and their respective RA maps in the form of heat maps. We refer to Appendix B for a detailed construction of heat maps. Figure 7(b) and 7(c) provide heat maps using the above two training methods,



(a) Input image (b) Full PEs (c) Uniform PEs

(d) Input image-noise (e) Full PEs-noise (f) Uniform PEs-noise

Figure 7: Rollout attention visualized as a heat map. An input image (a) is forwarded into a pre-trained ViT encoder and (b) is its corresponding RA heat map. Blue is for vanishing entries in the RA matrix, yellow for medium and red for large. (c) visualizes the RA generated by feeding (a) to a ViT encoder trained with identical PEs. (d) is from adding a random noise to (a). (e) and (f) are generated similarly to (b) and (c), respectively.

respectively. Besides, we add a small amount of noise to the input image such that the heat maps become Figure 7(e) and 7(f). The key observation is that heat maps from Figure 7(b) to 7(e) change much more dramatically than from Figure 7(c) to 7(f), meaning that the PE updating policy (b) produces more robust RA maps against noises adding to input images. Such a strong connection between PEs and RA inspires our MMUT.

4.2 MMUT Design

Overview. The design principle of our defense includes two aspects in both *privacy preservation* and *performance guarantee* for ViTs. On the one hand, the defense should significantly reduce the accuracy of MIAs, especially RAMIA proposed in Section 3; on the other hand, since the target ViT is pre-trained, whose encoder will be released online and be applied to downstream tasks, for example, a classification task, a high accuracy of the target ViT is also crucial. Our MMUT is based on the key idea of RAMIA that uses different behaviors of members and non-members in rollout attention before and after noise addition, respectively as a criterion for the attack, and the observation in Section 4.1 that indicates PEs may help weaken the sensitivity of rollout attention to noises. Figure 8 provides an overview. To illustrate our method, recall that ViTs divide an input image into a certain number of patches. Each patch is subsequently added to a PE. In general, MMUT first replaces α fractions of patches with patches of an image from a public image dataset, for example, ImageNet21k, for a given private image for training ViT. We call the replaced patches *mosaic patches*. In the forward step, we replace all PEs corresponding to mosaic patches with an extra *learnable mosaic embedding*. By doing so, in the backward step, the loss due to this image will only update those non-replaced PEs and the mosaic embedding (instead of replaced PEs).

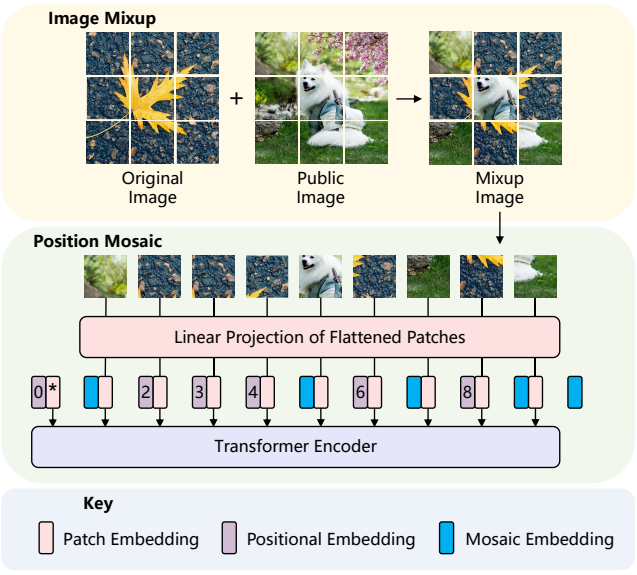


Figure 8: Overview of Mosaic MixUp Training (MMUT).

Algorithm 2 formally describes how MMUT works. In the beginning, MMUT prepares a training dataset, denoted as D, and a public dataset, denoted as D^{Pub}. When an image $x \in D$ is forwarded to ViT, MMUT mosaics x to image patches $\hat{x}^{\text{Pat}}$ as follows: (a) divide x into x^{Pat} with N image patches; (b) randomly sample an image $y \in D^{\text{Pub}}$, and divide y into y^{Pat} with N image patches[4]; (c) randomly pick α fractions in x^{Pat} and replace them by patches in y^{Pat} at the same position. In particular, MMUT defines an indicator vector $\mathbb{I} \in \{0,1\}^N$ and initializes it as a zero vector. α fractions of components in $\mathbb{I}$ will be randomly assigned as 1. MMUT then replaces all x_i^{Pat} by y_i^{Pat} for all $\mathbb{I}_i = 1$ to get $\hat{x}^{\text{Pat}}$; (d) when forwarding $\hat{x}^{\text{Pat}}$, replace the PEs corresponding to all mosaic patches (i.e., all PE_is such that $\mathbb{I}_i = 1$) with a learnable mosaic embedding ω; (e) In backward step, the loss due to $\hat{x}^{\text{Pat}}$ is computed to update those PE_is such that $\mathbb{I}_i = 0$, ω, and other learnable parameters. $\text{PE}_{i:\mathbb{I}_i=1}$s is not updated in this round. (f) train the ViT using $\hat{x}^{\text{Pat}}$s until a convergence. MMUT will be thoroughly evaluated in Section 4.4.

Remarks on MMUT. We highlight several details related to the above training process.

- A WarmUp before MMUT proceeds can be greatly helpful in speeding up the model convergence. WarmUp means training the ViT from scratch in a standard way on the private dataset for several epochs.
- The public dataset used in MMUT can be either i.i.d. or non-i.i.d. to the private dataset, even be random noises.
- Although MMUT borrows the idea of mixing up images for data augmentation, our novel mosaic embeddings are

[4]We may without loss of generalization assume x and y have the same size. Otherwise, we can resize all images in D^{Pub} at the beginning.

more crucial to MMUT's success than data augmentation techniques.

We shall set up experiments in Section 4.4 to justify the latter two arguments above.

Algorithm 2 MMUT

Require: Training Dataset D; Public Dataset D^{pub}; Mosaic Ratio: α; Learnable Positional Embedding: ω; Positional embedding: PE; Parameters other than PEs: θ
Ensure: Trained θ^* and PE*

```
for each x in D do
    𝕀 ← InitializeIndicator()
    𝕀̃ ← RandomFlip(𝕀, α)
    PE[𝕀] = ω, PE[𝕀̃] = PE[𝕀̃]
    y ← RandomSelect(D_pub)
    x[𝕀] = y[𝕀]
    θ*, PE*, ω ← GradientDescent(x, PE)
end for
return θ*, PE*
```

4.3 MMUT Evaluation: Experiments Setup

Models and Datasets. The same ViT structures and datasets (CIFAR10, CIFAR100, ImageNet100 and ISIC2018) as in Section 3.4 are used in experiments. The evaluations of defenses are against RAMIAs. For RAMIAs, we set the number of image neighbors to $n = 8$. Cosine similarity and Mean of RAs for multiple heads are used.

Baselines. We compare to four baseline defense methods (a-d) as mentioned in Section 2.3 in our experiments.

- **(a) Label Smoothing** [51]. The key idea is to soften the one-hot encoded target vector into a smooth one. In particular, we reduce the entry with respect to the class of label from 1 to $1 - \epsilon$, where ϵ is a smoothing hyper-parameter. Then, ϵ is equally distributed across all non-target classes. Label smoothing diminishes model overconfidence by preventing a 100% probability assignment to any single class, thereby promoting generalization and reducing overfitting to noisy or incorrect labels in the training data.
- **(b) Differentially Private Stochastic Gradient Descent (DP-SGD)** [1]. DP-SGD enforces privacy protections by altering the optimization routine. It involves two main actions: (i) clipping the gradients to ensure that their ℓ_2-norm is capped at a threshold value C during each training iteration; (ii) adding random noise to the gradients before the update step is applied. We adjust the noise scale β as a hyper-parameter while keeping the clipping threshold C fixed. Note that this is consistent with previous work [13]. We adopt small noise scale for maintaining target model's utility at a decent level, which leads to meaninglessly large ϵ values.
- **(c) RelaxLoss** [10]. RelaxLoss defends against MIAs by alternating gradient ascent and descent. It firstly adjusts the mean of the target loss, setting a target loss mean value γ that is more easily achievable by non-member data, thereby decreasing the distinguishability between member and non-member data and reducing the success rate of attacks. Secondly, to maintain the utility of the model, RelaxLoss does not maximize the predicted posterior score of the true class to 1. Instead, it flattens the posterior scores of the non-true classes, ensuring a significant margin between the true class score and others, thus preventing incorrect predictions, especially for challenging samples near decision.
- **(d) Adversarial Regularization (Adv-reg)** [40]. Adv-reg trains target models by blending traditional cross-entropy loss with adversarial loss. This method minimizes a composite loss, which is a weighted sum of the cross-entropy and adversarial losses. The weight of the adversarial loss δ is adjusted throughout the training to balance the contributions of both losses. The adversarial loss in Adv-Reg is generated by surrogate attack models, which are specifically trained on two types of data: the target model's training dataset and an additional, separate hold-out dataset. This training approach ensures these models are well-prepared to effectively challenge the target model, thus improving its robustness and overall performance.

Note that each of the aforementioned defense methods is governed by a hyper-parameter, including smoothing parameters ϵ in Label Smoothing, noise scale β in DP-SGD, target loss γ in RelaxLoss, and adversarial loss weight δ in Adv-Reg. Our MMUT is also governed by the mosaic ratio α, fractions of image patches that are replaced.

Attacker's information for defenses. We consider two typical assumptions on attacker's information for defense evaluation: *non-adaptive* and *adaptive*. For non-adaptive, the attacker lacks knowledge about the specific defense method employed by the model, which represents a more lenient form of defense. Conversely, under the adaptive assumption, the attacker possesses full awareness of both the defense method employed by the model and the specific parameters involved. In this case, defending against such attacks becomes even more challenging.

4.4 MMUT Evaluation: Experiments Result

Effectiveness of MMUT. We compare MMUT to the four mentioned baseline models, evaluating the trade-off between prediction accuracy and defense effectiveness. Figures 9 (against RAMIA-Threshold) and 10 (against RAMIA-Classifier) show such trade-offs using different defense methods. We consider both adaptive and non-adaptive attackers. For each defense method, we report the best four choices of its governed hyper-parameters. A lower attack accuracy with a higher prediction accuracy (closer to the bottom right) is better. Key observations are as follows:

- In both adaptive and non-adaptive settings against both RAMIAs, MMUT (blue square) points are roughly distributed closer to the bottom right, showing its effectiveness in preserving the trade-off.
- In the stricter adaptive setting, the four baselines show no significant effect. In the weaker non-adaptive setting, baselines other than DP-SGD have a minor effect in defending RAMIAs. The corresponding attack accuracies drop no greater than 10%. The only exception is DP-SGD, which behaves better

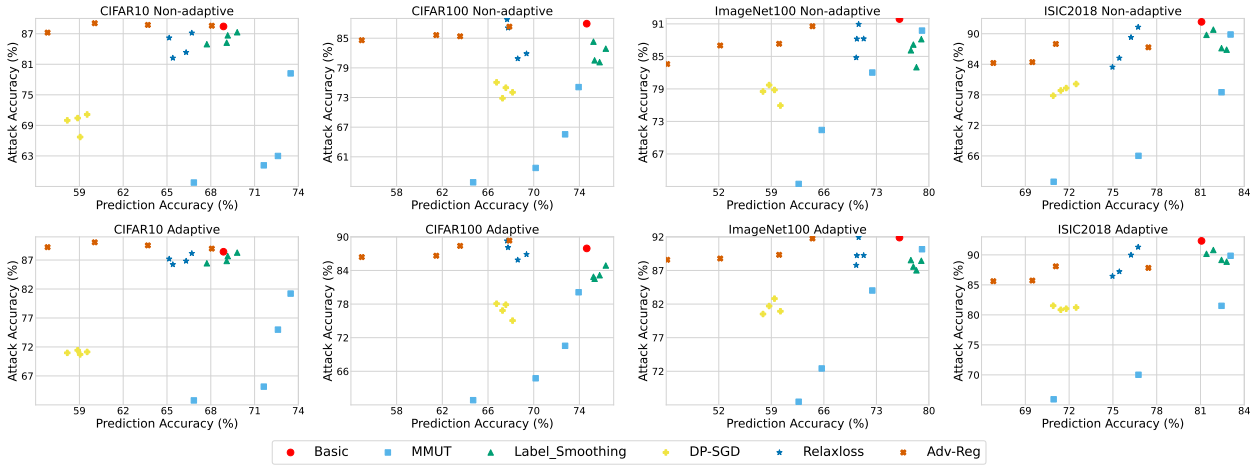


Figure 9: Comparison of the effectiveness of defenses with RAMIA-Threshold.

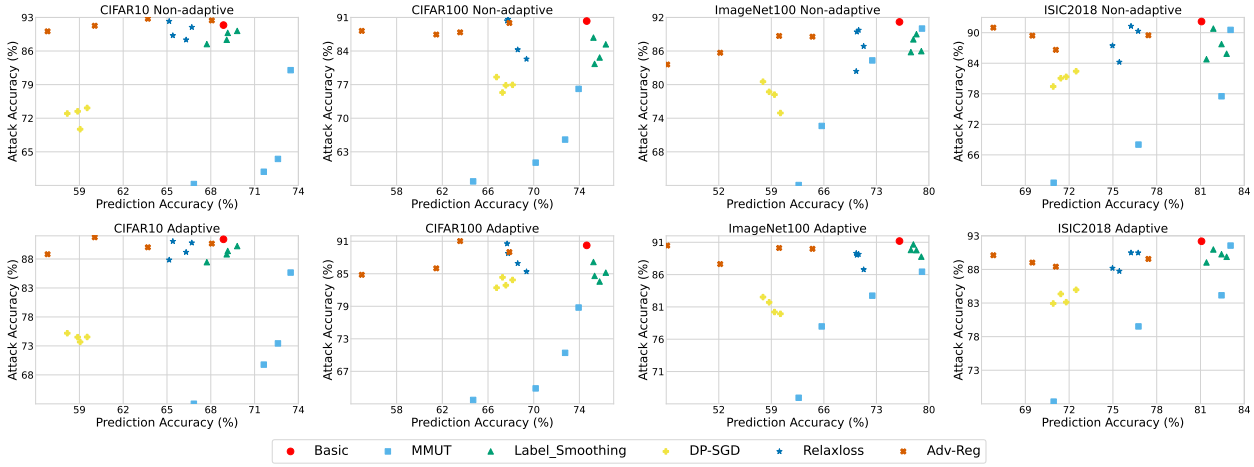


Figure 10: Comparison of the effectiveness of defenses with RAMIA-Classifier.

than other baselines in defense. However, it causes a dramatic drop in prediction accuracy.

- Remarkably, on CIFAR10 (the smallest dataset), our MMUT even improves the prediction accuracy while preserving great defense. The reason is the MixUp method can enhance the generalization ability of a ViT model.

The impact of mosaic ratio. The mosaic ratio α serves as a critical parameter that governs the privacy-performance trade-off. Larger α replaces more image patches from public data and achieves higher security while leading to worse performance in prediction, and vice versa. We evaluate MMUT on different values of α. Figure 11 shows the trends in attack accuracy when employing varying mosaic ratios in non-adaptive and adaptive scenarios, respectively. Figure 12 shows how the prediction accuracy changes with different mosaic ratios. Our defense method shows commendable performance across most attack scenarios. The key observations are:

- As mosaic proportion increases, the attack accuracy shows a downward trend in both settings. Such a trend becomes

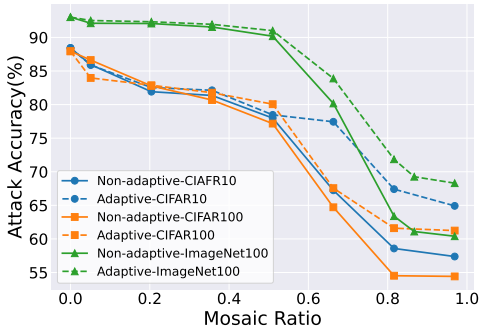


Figure 11: Performance of MMUT in defending against adaptive and non-adaptive RAMIA-Threshold on three datasets. A is for adaptive and N is for non-adaptive. Mosaic ratios α varies from 0 to 0.969.

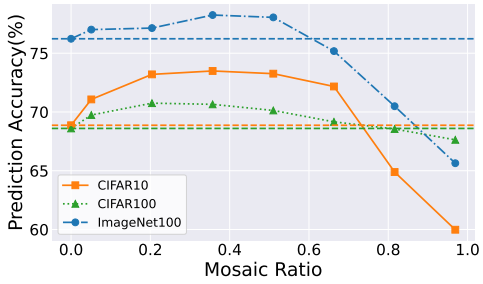


Figure 12: Performance of MMUT in prediction accuracy on three datasets. Mosaic ratios α varies from 0 to 0.969. The dotted lines show the prediction accuracy without any defense on three datasets.

more explicit when α exceeds roughly 0.5 for ImageNet100 and 0.4 for CIFAR10 and CIFAR100.

- For non-adaptive RAMIA, MMUT reduces the attack accuracy to a random guessing level, close to 50%. In an adaptive setting, even if the attacker knows the specific defense method and parameters, MMUT still reduces the attack accuracy remarkably.
- As α increases, the prediction accuracy first rises and drops later. More specifically, MMUT can improve the prediction accuracy of the model, with a maximum of 4.39% on CIFAR10, 2.15% on CIFAR100, and 2.02% on ImageNet100. Therefore, a good choice of α is crucial in designing MMUT.

The impact of public data distribution. We question on whether the public dataset used in MMUT is necessarily i.i.d. to private dataset. To verify its impact, we set the private dataset as CINIC10 [16], which has the same classification categories and the number of class labels as CIFAR10, meaning that their distributions are similar but not identical. We consider six different public datasets: CINIC10 (half for the target model and the rest for the shadow model), CIFAR10, ImageNet10 (a subset of ImageNet100 with 10 classes selected), ImageNet100 and random Gaussian noises. Figure 14 reports our experimental results. Our first finding is MMUT performs well even when replacing image patches with random noises. This is because the transformer-based model needs to learn robust attention allocations to distinguish the noise and original data patches, which is is exactly what we need to defend against RAMIA. Nevertheless, using other datasets for MixUp can be remarkably more effective. We observe all datasets can effectively defend against RAMIAs. However, much more interestingly, the best-performing public dataset is not CINIC10 itself, but CIFAR10, with a similar but not identical distribution to CINIC10. A possible reason is such a dataset can maximize the generalization ability of the model, while i.i.d. data provides a weaker contribution. On the other hand, ImageNet100 may not behave as well as expected, meaning that public data that differs too much does not help confuse the feature vectors constructed by RA maps.

MMUT vs. data augmentation (DA). We argue that simply adopting DA techniques without PE mosaics has much less effect in defending MIAs. We consider four well-known DA methods, including MixUp, PixelMix [61], GridMask [12], and Flipping. MixUp is the identical to MMUT, except for the use of mosaic position patches. PixelMix linearly combines multiple training images at pixel level by blending their features and labels. GridMask occludes image regions using a random grid-like mask. Table 4 shows our experimental results on the attack accuracies of RAMIA-Threshold. MixUp, PixelMix, and GridMask have extremely limited effects on defending agianst RAMIAs, while the simple geometric transformation of flipping images works slightly. Such findings in fact echo our insights on using mosaic position embbedings. Flipping messes up the patches of original image as well as the corresponding PEs, allowing the model to focus less on positional information and learn a similar dispersion of attentions as in Figure 7(c), leading to a more robust performance. Nevertheless, our MMUT shows much more effectiveness than merely using DA for defenses.

Table 4: Attack accuracy (%) of RAMIA-T against MMUT v.s. several DA techniques.

Dataset	CIFAR10	CIFAR100	ImageNet100	ISIC2018
MMUT	**55.63**	**61.23**	**63.99**	**62.73**
MixUp	80.69	81.64	83.62	83.29
PixelMix	88.38	87.45	89.64	87.21
GridMask	86.13	86.98	84.04	85.1
Flipping	76.27	77.57	78.28	75.05

Similarity of RA under MMUT. Finally, as a complement and another corroboration of our results, we explore how the RA maps change when noises are added to an input image on CIFAR10. Similar to Figure 1, Figure 13, which presents the difference in cosine similarities after MMUT is adopted, shows that MMUT is more effective than other methods in bridging the gap between members and non-members. Compared with the original model, MMUT significantly improves the RA's similarity level, whereas DP-SGD is exactly the opposite.

5 Conclusion

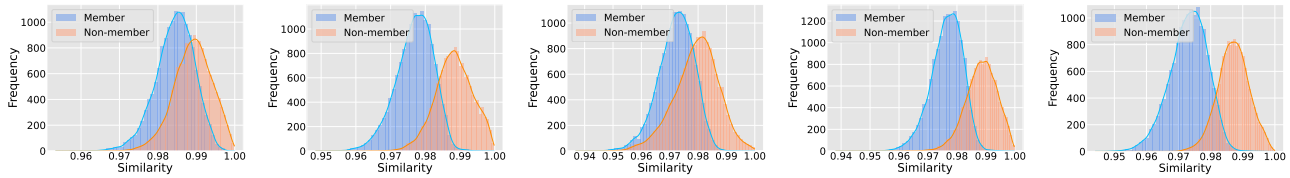
This work presents the first comprehensive study on the membership inference attacks and defenses against a powerful deep learning model, vision transformers. We use the information provided by the rollout attention maps and design a white-box attack against real-world ViT encoders. Based on an observation on the significance of positional embeddings, we design a unified framework of training ViTs as a defense method. Our defense can achieve a promising effectiveness in privacy-performance trade-off. Some possible future work includes: (a) extending our methods to language models; (b) attacking ViTs using another more expressive features; (c) designing a training-free defense method.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) (No. 62302183, No. 62372191, No. 62302187 and No. 62202197) and the Open Foundation of Key Laboratory of Cyberspace Security, Ministry of Education (No. KLCS20240401).

References

[1] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. B. McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. *Proceedings*



(a) MMUT (b) Label Smoothing (c) DP-SGD (d) RelaxLoss (e) Adv-Reg

Figure 13: Histograms for the number of members vs. non-member images in CIFAR10 across different cosine similarity scores under defenses on CIFAR10.

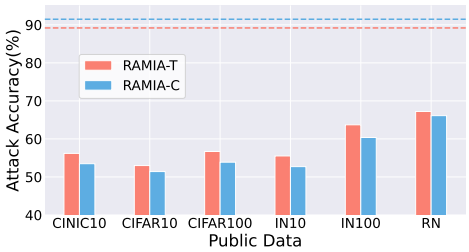


Figure 14: Attack accuracy (%) against MMUT on five different public datasets. IN10, IN100 and RN denote ImageNet10, ImageNet100 and random noise, respectively. Dotted lines show the attack accuracy without any defense.

of the 2016 ACM SIGSAC Conference on Computer and Communications Security (2016).

[2] Samira Abnar and Willem Zuidema. 2020. Quantifying Attention Flow in Transformers. In *Annual Meeting of the Association for Computational Linguistics*.

[3] Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan Yuille, Trevor Darrell, Jitendra Malik, and Alexei A. Efros. 2023. Sequential Modeling Enables Scalable Learning for Large Vision Models.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *ArXiv* abs/2005.14165 (2020).

[5] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. 2021. Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation. In *ECCV Workshops*.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*. Springer, 213–229.

[7] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, A. Terzis, and Florian Tramèr. 2022. Membership Inference Attacks From First Principles. *2022 IEEE Symposium on Security and Privacy (SP)* (2022), 1897–1914.

[8] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. 2023. Extracting Training Data from Diffusion Models. *ArXiv* abs/2301.13188 (2023).

[9] Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. 2021. RegionViT: Regional-to-Local Attention for Vision Transformers. *ArXiv* abs/2106.02689 (2021).

[10] Dingfan Chen, Ning Yu, and Mario Fritz. 2022. Relaxloss: Defending membership inference attacks without losing utility. *arXiv preprint arXiv:2207.05801* (2022).

[11] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. 2019. GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models. *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (2019).

[12] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. 2020. Gridmask data augmentation. *arXiv preprint arXiv:2001.04086* (2020).

[13] Christopher A. Choquette-Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. 2020. Label-Only Membership Inference Attacks. *ArXiv* abs/2007.14321 (2020).

[14] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. 2021. Twins: Revisiting the Design of Spatial Attention in Vision Transformers. In *Neural Information Processing Systems*.

[15] Noel C. F. Codella, Veronica M Rotemberg, Philipp Tschandl, M. E. Celebi, Stephen W. Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Armando Marchetti, Harald Kittler, and Allan C. Halpern. 2019. Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC). *ArXiv* abs/1902.03368 (2019). https://api.semanticscholar.org/CorpusID:60440592

[16] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. 2018. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505* (2018).

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009), 248–255.

[18] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. 2021. CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 12114–12124.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

[20] Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. 2021. XCiT: Cross-Covariance Image Transformers. In *Neural Information Processing Systems*.

[21] Jiemin Fang, Lingxi Xie, Xinggang Wang, Xiaopeng Zhang, Wenyu Liu, and Qi Tian. 2021. MSG-Transformer: Exchanging Local Spatial Information by Manipulating Messenger Tokens. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 12053–12062.

[22] Jianyuan Guo, Zhiwei Hao, Chengcheng Wang, Yehui Tang, Han Wu, Han Hu, Kai Han, and Chang Xu. 2024. Data-efficient Large Vision Models through Sequential Autoregression. *arXiv preprint arXiv:2402.04841* (2024).

[23] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. 2022. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence* 45, 1 (2022), 87–110.

[24] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. 2021. Transformer in Transformer. In *Neural Information Processing Systems*.

[25] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. 2017. LOGAN: Membership Inference Attacks Against Generative Models. *Proceedings on Privacy Enhancing Technologies* 2019 (2017), 133 – 152.

[26] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. 2019. Monte Carlo and Reconstruction Membership Inference Attacks against Generative Models. *Proceedings on Privacy Enhancing Technologies* 2019 (2019), 232 – 249.

[27] Hongsheng Hu, Zoran A. Salcic, Lichao Sun, Gillian Dobbie, P. Yu, and Xuyun Zhang. 2021. Membership Inference Attacks on Machine Learning: A Survey. *ACM Computing Surveys (CSUR)* 54 (2021), 1 – 37.

[28] Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. 2021. Shuffle Transformer: Rethinking Spatial Shuffle for Vision Transformer. *ArXiv* abs/2106.03650 (2021).

[29] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (2019).

[30] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2022. Transformers in vision: A survey. *ACM computing surveys (CSUR)* 54, 10s (2022), 1–41.

[31] Fei Kong, Jinhao Duan, Ruipeng Ma, Hengtao Shen, Xiao lan Zhu, Xiaoshuang Shi, and Kaidi Xu. 2023. An Efficient Membership Inference Attack for the Diffusion Model by Proximal Initialization. *ArXiv* abs/2305.18355 (2023).
[32] Alex Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images.
[33] Zheng Li and Yang Zhang. 2020. Membership Leakage in Label-Only Exposures. *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (2020).
[34] Hezheng Lin, Xingyi Cheng, Xiangyu Wu, Fan Yang, Dong Shen, Zhongyuan Wang, Qing Song, and Wei Yuan. 2021. CAT: Cross Attention in Vision Transformer. *2022 IEEE International Conference on Multimedia and Expo (ICME)* (2021), 1–6.
[35] Hongbin Liu, Jinyuan Jia, Wenjie Qu, and Neil Zhenqiang Gong. 2021. EncoderMI: Membership Inference against Pre-trained Encoders in Contrastive Learning. *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (2021).
[36] Yiyong Liu, Zhengyu Zhao, Michael Backes, and Yang Zhang. 2022. Membership Inference Attacks by Exploiting Loss Trajectory. *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (2022).
[37] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 9992–10002.
[38] Jiahao Lu, Xi Sheryl Zhang, Tianli Zhao, Xiangyu He, and Jian Cheng. 2021. APRIL: Finding the Achilles' Heel on Privacy for Vision Transformers. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 10041–10050.
[39] Milad Nasr, R. Shokri, and Amir Houmansadr. 2018. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. *2019 IEEE Symposium on Security and Privacy (SP)* (2018), 739–753.
[40] Milad Nasr, R. Shokri, and Amir Houmansadr. 2018. Machine Learning with Membership Privacy using Adversarial Regularization. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (2018).
[41] Yan Pang, Tianhao Wang, Xu Kang, Mengdi Huai, and Yang Zhang. 2023. White-box Membership Inference Attacks against Diffusion Models. *ArXiv* abs/2308.06405 (2023).
[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*.
[43] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2019. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*.
[44] A. Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. 2018. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. *ArXiv* abs/1806.01246 (2018).
[45] Fahad Shamshad, Salman Khan, Syed Waqas Zamir, Muhammad Haris Khan, Munawar Hayat, Fahad Shahbaz Khan, and Huazhu Fu. 2023. Transformers in medical imaging: A survey. *Medical Image Analysis* (2023).
[46] Virat Shejwalkar and Amir Houmansadr. 2021. Membership Privacy for Machine Learning Models Through Knowledge Transfer. In *AAAI Conference on Artificial Intelligence*.
[47] R. Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2016. Membership Inference Attacks Against Machine Learning Models. *2017 IEEE Symposium on Security and Privacy (SP)* (2016), 3–18.
[48] Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of big data* 6, 1 (2019), 1–48.
[49] Liwei Song and Prateek Mittal. 2020. Systematic Evaluation of Privacy Risks of Machine Learning Models. In *USENIX Security Symposium*.
[50] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15 (2014), 1929–1958.
[51] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the Inception Architecture for Computer Vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), 2818–2826.
[52] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herv'e J'egou. 2020. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*.
[53] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *ArXiv* abs/2302.13971 (2023).
[54] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. 2018. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data* 5 (2018). https://api.semanticscholar.org/CorpusID:263789934
[55] Jiahao Wang, Wenqi Shao, Mengzhao Chen, Chengyue Wu, Yong Liu, Kaipeng Zhang, Songyang Zhang, Kai Chen, and Ping Luo. 2024. Adapting LLaMA Decoder to Vision Transformer. *arXiv preprint arXiv:2404.06773* (2024).
[56] Pichao Wang, Xue Wang, F. Wang, Ming Lin, Shuning Chang, Wen Xie, Hao Li, and Rong Jin. 2021. KVT: k-NN Attention for Boosting Vision Transformers. *ArXiv* abs/2106.00515 (2021).
[57] Wenxuan Wang, Jingyuan Huang, Chang Chen, Jiazhen Gu, Jianping Zhang, Weibin Wu, Pinjia He, and Michael R. Lyu. 2023. Validating Multimedia Content Moderation Software via Semantic Fusion. *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis* (2023).
[58] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, and R. Shokri. 2021. Enhanced Membership Inference Attacks against Machine Learning Models. *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (2021).
[59] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. 2019. Cross-Modal Self-Attention Network for Referring Image Segmentation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 10494–10503.
[60] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis E. H. Tay, Jiashi Feng, and Shuicheng Yan. 2021. Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 538–547.
[61] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.
[62] Junxiang Zheng, Yongzhi Cao, and Hanpin Wang. 2021. Resisting membership inference attacks through knowledge distillation. *Neurocomputing* 452 (2021), 114–126.
[63] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Qibin Hou, and Jiashi Feng. 2021. DeepViT: Towards Deeper Vision Transformer. *ArXiv* abs/2103.11886 (2021).
[64] Daquan Zhou, Yujun Shi, Bingyi Kang, Weihao Yu, Zihang Jiang, Yuan Li, Xiaojie Jin, Qibin Hou, and Jiashi Feng. 2021. Refiner: Refining Self-attention for Vision Transformers. *ArXiv* abs/2106.03714 (2021).
[65] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2021. Deformable {DETR}: Deformable Transformers for End-to-End Object Detection. In *International Conference on Learning Representations*.

A Gaussian Noises

In this paper, noises are added to images as follows: for a given image, we linearly map $[0, 255] \mapsto [0, 1]$, and add a Gaussian with $\mu = 0$ and $\sigma = 0.2$ to each pixel value. For those smaller than 0 or larger than 1, recap them as 0 or 1, respectively. Finally remap $[0, 1] \mapsto [0, 255]$ with a nearest rounding.

B Rollout Attention Visualization

Heat map construction in Figure 7. The RA map obtained is a $(N + 1) \times (N + 1)$-sized matrix, which represents the attention allocation of each patch in a total of $N + 1$ patches (N patches plus one class token (CT)) to itself and other patches. We only use a one-dimensional vector with a size of $N + 1$ corresponding to CT, which records the importance of each image patch. Then only N components are kept, except one which represents the attention of CT on itself. Subsequently, we rearrange N attention values into a matrix of size $\sqrt{N} \times \sqrt{N}$, each value corresponds to a patch in the original image. And then adopt *interpolation* to restore the two-dimensional arranged attention to a matrix of size $(\sqrt{N} \times p) \times (\sqrt{N} \times p)$ (p is the size of each patch), such that it has the same height and width as the original image, to obtain a single channel heat map matrix h. We convert the single channel matrix into a color image using *JET color mapping* to obtain the final heat map H, a tensor of size $(\sqrt{N} \times p) \times (\sqrt{N} \times p) \times 3$. In JET, low values may be mapped to blue, intermediate values to green, and high values to red. Finally, we normalize the heat map and overlay it with the original image, and then restore it to the 0-255 area as displayed in Figure 7.

C Missing Empirical Results in Section 3

The impact of number of neighbor images. Recall that we use n neighbor images generated by adding independent noises to an original image from the dataset to construct the feature vector. Figure 15 shows the impact of n on the attack accuracy on CIFAR10, CIFAR100, and ImageNet100. Both RAMIA-Threshold and RAMIA-Classifier are evaluated. There are roughly consistent trends that the attack accuracies increase while n grows larger for the threshold-based model, which is not surprising because we compute an average value for every component in the feature vector, followed by determining a proper threshold. More neighbor images sharpen such disparity between members and non-members. However, for a classifier-based model, except for CIFAR10, the highest accuracy arises when $n = 8$. A possible reason is higher dimensional feature vectors make the inference classifier overfitting such that its generalization ability is reduced. Such observations inspire us that infinitely increasing the number of neighbor images is unreliable. A larger n may not help in accuracy but requires a longer computation time. Choosing a proper n is crucial.

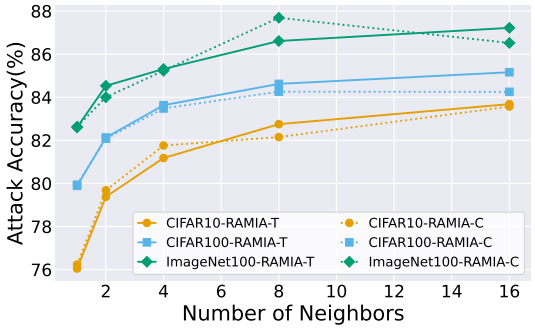


Figure 15: The impact of number of image neighbors n on attack accuracy. RAMIA-Threshold and RAMIA-Classifier are examined when n is set as 1, 2, 4, 8, 16 on CIFAR10, CIFAR100, and ImageNet100.

The impact of similarity metrics. We compute a cosine similarity of RA maps as a feature vector in our RAMIA. Besides, we consider another similarity metric named Pearson correlation coefficient (PCC) and evaluate its effect. Accuracy, precision, and recall of both RAMIA-Threshold and RAMIA-Classifier are presented in Table 5. Compared to the results in Table 2, there isn't a major difference in the performance of attack accuracies for the two metrics, providing another support to the effectiveness of our RAMIA.

D Missing Empirical Results in Section 4

The impact of mosaic embedding methods. In the process of position mosaic, in addition to using a global learnable mosaic embedding ω as presented in Algorithm 2, we propose two other alternative solutions: *(MMUT-Avg)* and *(MMUT-Zero)*. Neither of them learns a mosaic embedding during training. Instead, MMUT-Avg computes the average value of those non-mosaiced PEs as a mosaic, while MMUT-Zero mosaics PEs with a zero matrix. Table 6 presents their performances in both privacy preserving and prediction accuracy. Both MMUT-Avg and MMUT-Zero instead

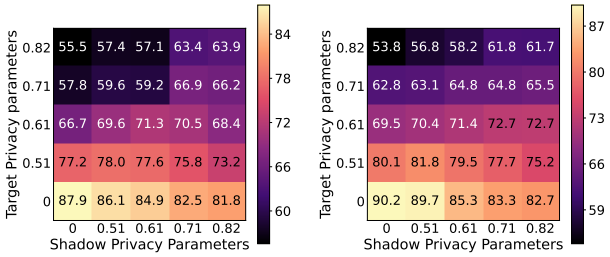
Table 5: Attack performance (%) using Pearson correlation coefficient for a similarity computation.

Method	Dataset	Accuracy	Precision	Recall
RAMIA-T	CIFAR10	87.66	83.99	90.43
	CIFAR100	88.33	80.49	93.45
	ImageNet100	90.25	85.66	95.73
RAMIA-C	CIFAR10	89.69	90.87	87.51
	CIFAR100	90.88	97.68	86.37
	ImageNet100	88.63	93.35	83.77

Table 6: Prediction and attack accuracies (%) using different mosaic methods. T for threshold and C for classifier.

Method	Dataset	MMUT	MMUT-Avg	MMUT-Zero
RAMIA-T	CIFAR10	60.63	67.19	61.79
	CIFAR100	55.23	79.4	56.51
	ImageNet100	71.99	89.25	89.05
RAMIA-C	CIFAR10	57.67	77.46	60.38
	CIFAR100	51.77	84.21	52.91
	ImageNet100	66.18	94.24	91.08

may also behave well on the smallest CIFAR10. Further, MMUT-Zero even achieves effectiveness similar to MMUT; however, on the largest ImageNet100, it demonstrates advantages over the other two methods.



(a) Defense against RAMIA-T (b) Defense against RAMIA-C

Figure 16: The attack accuracy (%) of semi-adaptive attackers attempting shadow models with different parameters on CIFAR100. T is for threshold and C is for classifier.

Defense against semi-adaptive RAMIA. Besides adaptive and non-adaptive assumptions on attacker's information, we examine defenses under a *semi-adaptive* assumption, where the attacker is aware of the defense method but lacks knowledge about the specific parameters associated with it. In this case, the attacker can only make a guess on such parameters. Figure 16 shows the attack accuracy of RAMIAs using different parameters in target models versus shadow models on CIFAR100. Note that blocks closer to the top left (closer to non-adaptive) are darker in color, with attack accuracies of approximate random guesses of 50%. Likewise, blocks closer to the diagonal (closer to adaptive) have lighter colors. Such assumptions are more strict for attackers, but MMUT can remarkably reduce the attack accuracy from 87.94% for RAMIA-Threshold and 90.24% for RAMIA-Classifier. In any case, the experimental results illustrate the effectiveness of our MMUT in defending RAMIAs.