This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3728414

Black-Box Membership Inference Attacks against Contrastive Learning via Aggressive Data Augmentations

Zixiong Wang, Lishuai Hou, Gaoyang Liu, *Member, IEEE*, Jianfeng Lu, *Member, IEEE*, Desheng Wang, *Member, IEEE*, Ahmed M. Abdelmoniem, *Senior Member, IEEE*, and Chen Wang, *Senior Member, IEEE*

***Abstract*—Contrastive learning (CL) usually leverages extensive quantities of unlabeled data to pre-train an encoder, but may confront significant data privacy vulnerabilities due to membership inference attacks (MIAs), where the adversary aims to infer whether a given sample was used to pre-train a target encoder. Existing MIAs against CL encoders often depend on prior knowledge of the target encoder's pre-training data or pre-training settings, which are typically kept confidential to protect the owner's interests. In this paper, we propose ADA-MIA, a novel black-box MIA against CL encoders, which requires only an auxiliary dataset comprising a mixture of certain training samples without the necessity of knowing their membership labels. We illustrate the inherent privacy risk of data augmentations in CL, and present evidence for the first time that aggressive data augmentations not utilized in the pre-training process can result in significant membership leakage in CL encoders. We further exploit the deliberately crafted aggressively augmented views of the target sample to construct the membership features, which can facilitate the execution of our inference in an unsupervised manner. Extensive experimental results on four datasets demonstrate that ADA-MIA can achieve inference performance comparable to that of seven existing MIAs, even with merely the black-box embedding access to the CL encoders.**

***Index Terms*—Membership inference attack, contrastive learning, aggressive data augmentation, black-box model.**

I. Introduction

CONTRASTIVE learning (CL) [1]–[3] is an emerging machine learning (ML) paradigm that can overcome the restrictions of labeled data. It usually makes use of a large amount of unlabeled data to pre-train an encoder, which maps different views of the same sample created through data augmentation to have close embeddings in the representation space, while ensuring that views from dissimilar data points are separated. The pre-trained encoder usually serves as a feature extractor for various downstream tasks, such as object detection and image classification, and many service providers have publicly released their pre-trained encoders for commercial benefits, e.g., SimCLR by Google [2] and MoCo by Meta [1].

In practice, the pre-trained CL encoder usually operates in an open environment accessible to anyone. This openness exposes the CL encoder and its training data to potential security risks. Recent works have demonstrated that CL encoders are vulnerable to different attacks, including adversarial attacks [4], [5], poisoning attacks [6], [7], and backdoor attacks [8], [9]. In this paper, we concentrate on the so-called membership inference attacks (MIAs) against the CL encoder, where the adversary aims to infer whether a given sample (i.e., the target sample) was used to pre-train a given encoder (i.e., the target encoder) or not. So far, only a few studies focus on MIAs against CL encoders [10]–[12]. To be specific, He et al. [10] propose to integrate a classification layer into CL encoders and treat these encoders as classification models to perform MIA. Liu et al. [11] and Zhu et al. [12] exploit the similarities among various standard augmented views or cropped patches of a target sample to determine its membership property. However, all these works rely on a shadow dataset which comes from the same distribution as the target models training dataset to train a shadow encoder or a subset of the actual training samples, to provide the necessary membership labels for constructing the inference attack model. In practice, the prior knowledge of the target encoder's pre-training data distribution or pre-training settings is typically kept confidential to protect the encoder's security. As a consequence, the applicability of existing MIAs against CL encoders is largely restricted and impractical.

In this paper, we take a further stride and propose ADA-MIA, which leverages the Aggressive Data Augmentation[1] to perform MIA against black-box pre-trained CL encoders. In contrast to existing MIAs which concentrate on the embedding layer or additional classification layer of CL encoders, we shift our attention to the data augmentation module. With ADA-MIA, we for the first time demonstrate that while the standard data augmentations tend to yield limited information

This work was supported in part by the National Natural Science Foundation of China under Grants 62506138, 62272183, 62071192 and 62372343; by the National Key R&D Program of China under Grant 2024YFE0103800; by the Key R&D Program of Hubei Province under Grants 2025EHA033, 2024BAA011, 2024BAB016 and 2024BAB031; and by the UKRI EPSRC Grant EP/X035085/1.

Z. Wang, L. Hou, G. Liu, D. Wang and C. Wang are with the Hubei Key Laboratory of Smart Internet Technology, School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China. Email: {zixwang, lishuaihou, liugaoyang, dswang, chenwang}@hust.edu.cn.

J. Lu is with School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China. Email: lujianfeng@wust.edu.cn.

A. M. Abdelmoniem is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, UK. Email: ahmed.sayed@qmul.ac.uk.

[1]We refer to data augmentations with greater strength than the standard augmentations in the pre-training process in CL as "aggressive data augmentations". The strength here refers to the intensity of transformation applied to the training data.

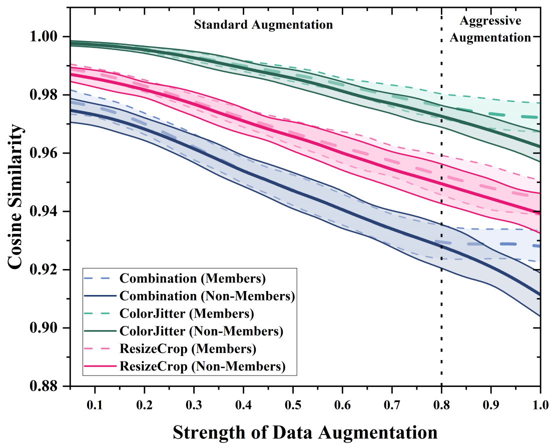

Fig. 1: Cosine similarities between member/non-member samples and their corresponding augmented views, using a CL encoder pre-trained with MoCo on CIFAR-10 dataset. It can be observed that the aggressively augmented views of the samples exhibit a more significant disparity in embedding similarities between members and non-members.

concerning membership, the aggressive data augmentations that have not been used in the pre-training process could lead to severe membership leakage in CL encoders.

Our key observation is that the aggressively augmented views of pre-training (*member*) and non-pre-training (*non-member*) samples exhibit notable differences in the embedding similarity to their respective original samples (c.f. Figure 1). Specifically, during the pre-training process, the CL encoder uses different data augmented views of the pre-training samples and tries to generate similar embeddings for these samples. In pursuit of this objective, it naturally tends to map the different detailed aspects of the original sample into a compact embedding space close to the original sample [13], [14]. Consequently, when presented with an aggressively augmented pre-training sample, the CL encoder prioritizes the detailed characteristics of the original sample, resulting in a heightened embedding similarity between this original sample and its aggressively augmented views. Conversely, this phenomenon is not exhibited in the case of non-training images. By generating aggressively augmented views of the target sample, we can compute the embedding similarities with the original target sample and employ these measurements as membership features to perform MIAs.

While the idea sounds simple, the realization of ADA-MIA confronts one major challenge: since we have no prior knowledge about the pre-training process of the target encoder, we cannot know the type and strength of the data augmentations used by the target encoder. As a consequence, it is difficult for us to directly select a data augmentation approach with a specified strength as an aggressive data augmentation, whose augmentation strength needs to be significantly stronger than that employed in the model pre-training process. To address this issue, we meticulously control the intensity of data augmentations from weak to strong to get a series of augmented samples for the target sample. Then by employing the original sample as a reference point, we can finally ensure that the resulting augmentation set includes transformations stronger than those used during pre-training of the target model. Then we successively calculate the embedding similarities between all augmented versions and the original sample, and use these similarities as the membership features to perform the MIA.

To summarize, we make the following contributions:

- We present ADA-MIA, a novel black-box MIA against CL encoders, which requires no prior knowledge about the target encoder or its pre-training data distribution, but only using the encoders embedding interface and an auxiliary dataset that mixes training and non-training samples, without the need for their ground truth membership labels.
- We observe the difference in the embedding similarities resulting from the aggressive data augmentations between the members and non-members of the target encoder's pre-training dataset, and show for the first time that the aggressive augmentation may leak the membership privacy and can be leveraged to facilitate MIA against CL encoders.
- We conduct extensive experiments on various datasets with different contrastive pre-training algorithms. The results provide compelling evidence that, with minimal prior knowledge of the CL encoders compared with existing methods, ADA-MIA can achieve an inference performance comparable to that of seven existing MIAs, while being minimally impacted by two existing defenses against MIAs. Our code is available for reproducibility[2].

II. Background

A. Preliminaries of Contrastive Learning

CL [1], [15], [16] represents an emerging deep learning paradigm designed to address the limitations of labeled data. CL utilizes various data augmentation techniques in its pre-training phase to create augmented views of input images through a series of random transformations. Two augmented views from the same (resp. different) image(s) are denoted as positive (resp. negative) pairs. The primary objective of CL is to train the base encoder with the goal of bringing the embeddings of positive pairs closer together while simultaneously pushing apart the features of negative pairs. To this end, the fundamental components of pre-training a CL encoder include multiple data augmentations, a base encoder, a projection head, and a contrastive loss. The conventional paradigm of CL is shown in Figure 2, and we will describe each component below.

Data Augmentation. In the pre-training process, the CL paradigm initially employs data augmentation methods to generate two augmented versions of a given image. It is noteworthy that there are several standard data augmentation methods [1], [2], including RandomResizedCrop, RandomHorizontalFlip, ColorJitter and RandomGrayscale[3]. Among them, ResizedCrop involves randomly selecting a patch from the image

[2]https://anonymous.4open.science/r/ADAMIA-64B2/

[3]In what follows, we will omit the prefix "Random" for short when referring to these data augmentation transformations.

TABLE I: A summary of priori knowledge for existing MIAs against CL encoders.

Attack Method	Prior Knowl. of Target Encoder		Prior Knowl. of Training Setting		Prior Knowl. of Pre-Training Data		
	Backbone Architecture	Projection Head	Training Algorithm	Data Augmentation	Data Distribution	Data Samples	Membership Labels
ContrastiveLeaks [10]	√	√	N/A	N/A	√	×	√
EncoderMI [11]	√	×	√	N/A	√	×	√
PartCrop [12]	×	×	×	×	×	√	√
ADA-MIA	×	×	×	×	×	√	×

Required Knowledge: √ Non-required Knowledge: × Not Available: N/A

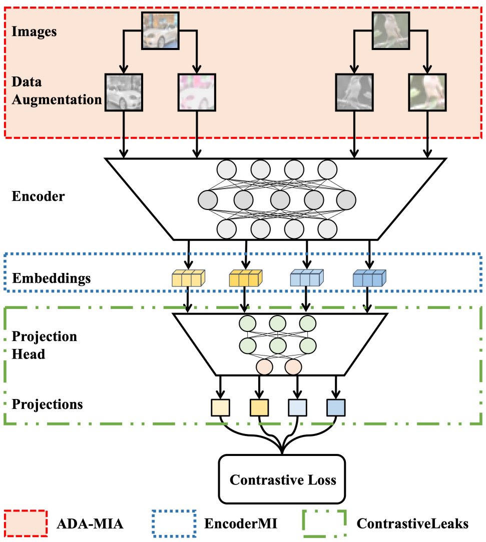

Fig. 2: CL pipeline with possible membership leakage. Previous MIAs utilize random augmentations and focus on an additional projection head (ContrastiveLeaks [10]) or the embedding part (EncoderMI [11]), while our ADA-MIA concentrates on the augmentation part, by manually presetting the intensity of augmentations, and exploring embedding similarities between original images and corresponding aggressively augmented views.

and resizing it back to the original size, with the cropping range controlled solely by a parameter named "scale". HorizontalFlip (resp. Grayscale) applies horizontal flipping (resp. grayscale) transformation to the images based on a probability parameter. ColorJitter randomly modifies the brightness, contrast, saturation, and hue of images, with each adjustment controlled by a specific parameter determining its strength.

Base Encoder. The base encoder is responsible for mapping the augmented samples into a representation space. Typically, it adopts a standard image model structure, such as ResNet [17] or Vision Transformer [18]. Following the pre-training process, the encoder is preserved and fine-tuned for use in downstream tasks.

Projection Head. The projection head is typically a simple neural network, comprising several fully connected layers. Its role is to transform the embeddings from the base encoder into a different latent space, aligning them with the contrastive loss to enhance the encoder's efficacy. After the pre-training stage of the CL encoder, it is typical to remove the projection head and keep only the encoder for further use.

Contrastive Loss. The objective of CL [19] is accomplished through the utilization of a contrastive loss function, which is derived from the Noise-Contrastive Estimation (NCE) loss [20], which can be formulated as:

$$\mathcal{L}_{NCE} = -\log\left(\frac{e^{S(f(\mathbf{x}),f(\mathbf{x}^+))}}{e^{S(f(\mathbf{x}),f(\mathbf{x}^+))} + e^{S(f(\mathbf{x}),f(\mathbf{x}^-))}}\right), \tag{1}$$

where f represents the CL base encoder, $\mathbf{x}^+$ is the augmented version of the given image $\mathbf{x}$ while $\mathbf{x}^-$ is the augmented version of other images. $S(\cdot)$ represents the embedding similarity measurement. Besides, various loss functions have been developed for different CL algorithms, such as NCE loss variants like infoNCE loss [21] and NT-Xent loss [22].

With these modules of CL, there are two widely used algorithms in CL pre-training process: one is MoCo [1] and the other is SimCLR [2]. MoCo employs a momentum encoder that updates slower than the base encoder using a momentum update strategy, and a memory bank that maintains a queue of embeddings from the momentum encoder for previous batches' augmented images in CL. This setup addresses embedding consistency issues and stabilizes the pre-training process. While SimCLR primarily studies the impact of various transformations on images and feature representation. It revealed that *ResizedCrop*, *HorizontalFlip*, *ColorJitter* and *Grayscale* yield the best results in CL.

B. Related Work on MIAs against CL Encoders

MIAs against ML models were pioneered by Shokri et al. [23], by establishing the novel shadow training paradigm. This line of research has been further exploited by many works [24]–[33]. For attack strategies devoid of shadow models, some studies leverages the probability threshold [34], predicted labels [35], loss threshold [35], [36], probability distribution [37], output confidence [38], adversarial robustness [39]–[41], and differential comparison [42] as membership features to perform MIAs.

Recent studies also expand the scope of MIAs to generative models, including generative adversarial networks (GANs) [43], [44], variational autoencoders (VAEs) [45], [46], diffusion models [47]–[52], large language models (LLMs) [53]–[56], and vision-language models [57], [58].

Differing from previous works, our focus lies on MIAs against CL encoders, an area that has been explored recently by only a few works [10]–[12]. Among them, ContrastiveLeaks [10] focuses on membership leakage in the projection layer (c.f. Figure 2). It appends an additional fully connected neural network to the target encoder and treats this encoder as a standard classification model to perform MIA.

EncoderMI [11] investigates membership information on the outputs of CL embedding layer (c.f. Figure 2). This work posits that the embedding similarities between augmented versions of training data is significantly higher than those of testing samples. Recently, PartCrop [12] finds that CL encoders show a stronger part-aware capability on the training data compared with testing data. This method requires a part of pre-training samples of the target encoders along with the ground truth membership labels.

Summary. Compared to existing MIAs targeting CL encoders, ADA-MIA differs from these methods in two key aspects. Firstly, ADA-MIA makes novel observations and explorations regarding membership leakage of data augmentation within CL frameworks. Specifically, while existing researches [10], [11] predominantly focus on exploiting classification head or output embedding correlations, ADA-MIA reveals that data augmentations with high intensity largely contributes to distinguishing between members and non-members (c.f. Figure 1). Moreover, ADA-MIA explores which augmentation methods lead to more leakage (Section V-E1) and whether membership information will be exposed when adversaries obtain no priori knowledge about pre-training augmentations (Section V-E2). Secondly, at the threat model level, ADA-MIA significantly reduces unrealistic prior knowledge dependencies and eliminates the need for shadow model training. Existing MIAs [10], [11] unpractically rely on data similar to target encoders' pre-training distribution while also necessitate exact membership labels, otherwise their effectiveness will exhibit significant reduction. However, ADA-MIA circumvents these limitations with information derived from aggressively augmented views, developing a more practical attack and exploring the potential to bypass conventional shadow training paradigm. This paves the way for developing more practical and efficient MIAs in future works.

Our ADA-MIA introduces a novel black-box MIA method against CL encoders. In contrast to existing methods, it operates solely with the black-box embedding access and an auxiliary dataset comprising pre-training samples without exact membership labels (c.f. Table I). Our attack employs the aggressive augmentations to extract membership features, which does not require identical data augmentations as those used in the pre-training process or necessitate information about the model and the training data to generate the augmented samples. Furthermore, we focus on the membership leakage risk stemming from the data augmentation layer of CL. We discover, for the first time, that the aggressive augmentation has the potential to magnify the membership features, which is not observed in previous works using only standard augmentations.

III. Threat Model

This paper considers an adversary whose objective is to determine whether a data sample was included in the pre-training dataset of a target CL encoder with merely the black-box access to the target encoder's embedding interface.

Adversary's Knowledge. ADA-MIA focuses on attacking against CL encoders, regardless of what training algorithm and model structure the encoder uses. Therefore, it assumes the adversary to have the following knowledge:

- *Black-box access to target encoder*: i.e., the access to the embedding interface of the target encoder. When querying the target encoder f with a data sample $\mathbf{x}$, it will output the corresponding embedding: $\mathbf{z} = f(\mathbf{x})$.
- *An auxiliary dataset*: i.e., the access to an auxiliary dataset D_a composed of a mixture of both pre-training samples and non-training samples, without any knowledge of their true membership labels. Note that the composition of this auxiliary dataset does not require a balanced ratio of members to non-members at 1:1, which is usually used in current MIAs (c.f. Section V-D).

On the contrary, ADA-MIA assumes the adversary cannot obtain any prior knowledge of the following aspects, which is necessary for existing MIAs against CL encoders [10], [11], [59]:

- *Target encoder's structure:* including the architecture of backbone and projection head.
- *Target encoder's training setting:* including the CL training algorithms, the information of data augmentation, as well as the parameter settings.
- *Target encoder's pre-training dataset:* including any dataset sharing the same distribution as the pre-training dataset used to create f, the distribution of the training data, as well as any information about the membership property of each sample.

The comparison of adversary's knowledge between our ADA-MIA and existing works is presented in Table I.

Adversary's Capacity. The adversary equipped with ADA-MIA can employ different data augmentation methods to generate augmented versions of a target sample, and obtain the embeddings of a target encoder for the target sample and its augmented versions as:

$$\mathbf{z}_t = f(t(\mathbf{x})). \tag{2}$$

Here, t represents the data augmentation. With the access to the target encoder, the adversary can obtain the embeddings corresponding to the augmented versions of the target sample.

Adversary's Goal. Given a target encoder f, an auxiliary dataset D_a, and the target sample $\mathbf{x}_t$, the adversary's objective is to determine whether $\mathbf{x}_t$ was used for pre-training f:

$$\mathcal{A}(\mathbf{x}_t; D_a, f) \longrightarrow \textit{Member / Non-Member}, \tag{3}$$

where $\mathcal{A}$ represents the attacking functionality of ADA-MIA. The label *Member* (resp. *Non-Member*) means our inference that $\mathbf{x}_t$ was (resp. not) used in the pre-training process of f.

IV. Design of ADA-MIA

The inference process of ADA-MIA can be formulated as follows (c.f. Figure 3 and Algorithm 1): given the embedding access to the target encoder f, a target image $\mathbf{x}_t$, and an auxiliary dataset D_a which contains a part of member images of f but without exact ground truth membership labels, ADA-MIA first applies multiple data augmentations $\mathbf{T}$ on each image $\mathbf{x}_a$ from D_a, and obtain a series of augmented images X_a. For each augmentation t from $\mathbf{T}$, ADA-MIA then queries f with

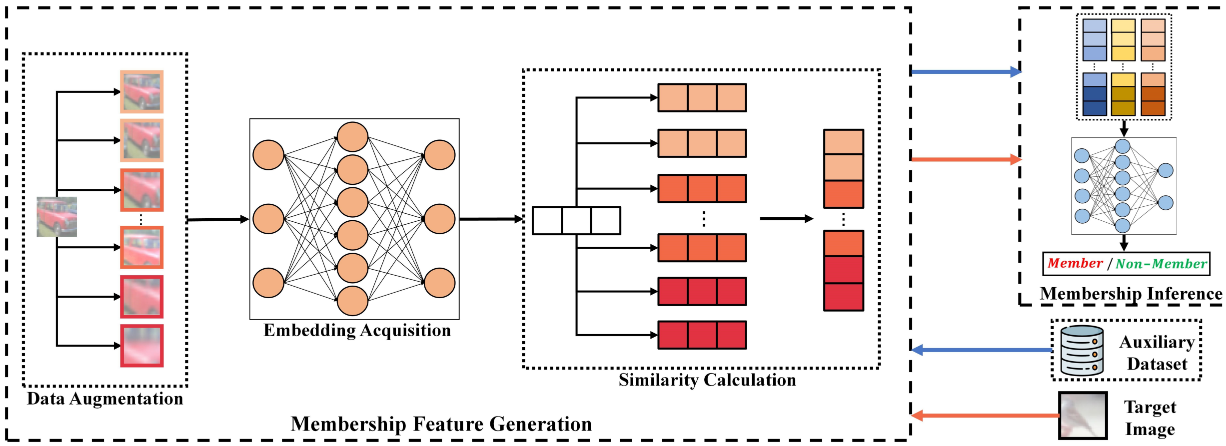

Fig. 3: The framework of ADA-MIA. (a) Obtaining a series of augmented images of the target image from weak to strong. (b) Querying the target black-box encoder with the augmented images and the original target image to obtain their respective embeddings. (c) Calculating similarities between embeddings of the target image and each augmented version, and then concatenating these similarities into a membership feature vector. (d) Leveraging an unsupervised clustering algorithm to train the attack model and conduct membership property inference.

$\mathbf{x}_a$ and corresponding augmented version $t(\mathbf{x}_a)$, obtaining the corresponding embeddings $f(\mathbf{x}_a)$ and $f(t(\mathbf{x}_a))$. Subsequently, ADA-MIA generates a membership feature vector $\mathbf{v}_a$ for $\mathbf{x}_a$ by calculating the similarities between original embedding and all augmented embeddings. Finally, with the set of feature vectors V_a for all the images in D_a, an unsupervised clustering algorithm $\mathcal{C}$ can be adopted to train the attack model $\mathcal{M}$ and subsequently infer the membership property of $\mathbf{x}_t$. In general, ADA-MIA includes two major modules: membership feature generation and unsupervised membership inference.

A. Membership Feature Generation

CL encoders tend to capture intricate details of pre-training samples and align different augmented versions of an image closely in the representation space during the pre-training. Therefore, the key idea of the membership feature generation is to design aggressive augmentations to prompt CL encoders prioritizing the detailed characteristics or sample-specific regions (e.g., a bird and its beak). Thus, only pre-training images can maintain feature alignment with their extreme distorted views [60]. To obtain these aggressively augmented samples, we employ several data augmentation techniques with extreme high intensity [61], [62]. Subsequently, we derive the augmented versions using our precisely controlled augmentation methods, and then obtain the similarity of embeddings between an image and its aggressive augmented versions, which can serve as a vital signal to the membership features for this image.

Although the idea is simple, different augmentations would unveil different aspects of one image, rendering it challenging to precisely construct the membership features. The choice of data augmentation method and the sequence of augmentation strength orchestration thus significantly impact the performance of ADA-MIA. Furthermore, since the adversary lacks prior knowledge about the pre-training configuration of the target encoder, we consider utilizing the standard data augmentations and then incorporate a parameter λ to regulate the

Algorithm 1 ADA-MIA

Input: The target encoder f, a target image $\mathbf{x}_t$, an auxiliary dataset D_a, the number of types of augmentations N, the interval of augmentation λ_{int}, the set of data augmentations $\mathbf{T}$, the similarity metric $S(\cdot)$, and the clustering algorithm $\mathcal{C}$.
Output: Membership status of $\mathbf{x}_t$.

$M = 1/\lambda_{int}$
$\mathbf{T} = \{t_1^1, t_1^2, \cdots, t_N^M\}$
▷ Augmentation intensity presetting with Eq. (4)
$V_a = \emptyset$
for each sample $\mathbf{x}_a \in D_a$ **do**
  $\mathbf{v}_a = \emptyset$
  $X_a = \{t_1^1(\mathbf{x}_a), t_1^2(\mathbf{x}_a), \cdots, t_N^M(\mathbf{x}_a)\}$
  for $n \in \{1, \cdots, N\}$ and $m \in \{1, \cdots, M\}$ **do**
    $\mathbf{v}_a = \mathbf{v}_a \cup S(f(\mathbf{x}_a), f(t_n^m(\mathbf{x}_a)))$
    ▷ Membership feature generation with Eq. (5)
  end for
  $V_a = V_a \cup \mathbf{v}_a$
end for
Attack Model $\mathcal{M} \leftarrow \mathcal{C}(V_a)$
▷ Unsupervised attack model training with Eq. (7)
$\mathbf{v}_t = \emptyset$
$X_t = \{t_1^1(\mathbf{x}_t), t_1^2(\mathbf{x}_t), \cdots, t_N^M(\mathbf{x}_t)\}$
for $n \in \{1, \cdots, N\}$ and $m \in \{1, \cdots, M\}$ **do**
  $\mathbf{v}_t = \mathbf{v}_t \cup S(f(\mathbf{x}_t), f(t_n^m(\mathbf{x}_t)))$
end for
Member / *Non-Member* $\leftarrow \mathcal{M}(\mathbf{v}_t)$
▷ Membership inference
return *Member* / *Non-Member*

strength of these data augmentations. Specifically, we select N augmentation methods $[t_1, t_2, \cdots, t_N]$ (e.g., *ResizedCrop* and *ColorJitter* when $N = 2$) from the standard data augmentations commonly employed in the research field of computer vision. Then we precisely control the augmentation strength by configuring the augmentation magnitude to λ. We suppose λ to range from 0 to 1 with an interval λ_{int}. With our collection of augmentations $\mathbf{T} \in \mathbb{R}^{M \times N}$, we can obtain the augmentation methods as:

$$\mathbf{T} = \begin{pmatrix} t_1^1 & t_1^2 & t_1^3 & \cdots & t_1^M \\ t_2^1 & t_2^2 & t_2^3 & \cdots & t_2^M \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_N^1 & t_N^2 & t_N^3 & \cdots & t_N^M \end{pmatrix}, \tag{4}$$

where $M = 1/\lambda_{int}$ represents the number of occurring times for each augmentation method. t_n^m is the element in the n-th row and m-th column of $\mathbf{T}$, which represents the n-th augmentation method with the augmentation intensity of $\lambda = m\lambda_{int}$. For example, when we set $N = 1$ (e.g., we select *ColorJitter* as the augmentation method), $\lambda_{int} = 0.25$, $\mathbf{T}$ can be denoted as $\{t_1^1, t_1^2, t_1^3, t_1^4\}$, where t_1^3 denotes the transformation of *ColorJitter* with an intensity of 3×0.25. Then $\mathbf{T}$ will be utilized to derive augmented views for $\mathbf{x}_a$.

For an image $\mathbf{x}_a$ from D_a, we then obtain the augmented version $t_n^m(\mathbf{x}_a)$ using the augmented method t_n^m from $\mathbf{T}$. Afterward, ADA-MIA generates the membership features for $\mathbf{x}_a$. In particular, for one certain sample $\mathbf{x}_a$ and all its augmented samples $X_a = \mathbf{T}(\mathbf{x}_a)$, we query the target encoder f and collect the corresponding embeddings $f(\mathbf{x}_a)$ and $f(t_n^m(\mathbf{x}_a))$. Then, for each augmented sample $t_n^m(\mathbf{x}_a)$ in X_a, we calculate the similarity between $f(t_n^m(\mathbf{x}_a))$ and $f(\mathbf{x}_a)$ as:

$$v_n^m = S(f(\mathbf{x}_a), f(t_n^m(\mathbf{x}_a))). \tag{5}$$

Finally, all these similarities are concatenated together and flattened into a vector, which is denoted as the membership feature vector $\mathbf{v}_a$ of $\mathbf{x}_a$:

$$\mathbf{v}_a = [v_1^1, v_1^2, \cdots, v_n^m, \cdots, v_N^M]. \tag{6}$$

In this way, we can obtain the membership feature vector $\mathbf{v}_a$ for a certain sample $\mathbf{x}_a$ and obtain a feature bank V_a, which would serve as a solid basis for the subsequent membership inference process.

B. Unsupervised Membership Inference

Providing the membership feature generation method for one sample, ADA-MIA proceeds to obtain membership feature vectors for all samples in the auxiliary dataset. Subsequently, it can leverage these vectors to train the attack model and conduct membership inference.

According to the considered threat model, the pre-training data distribution of CL encoders is typically kept confidential to protect the owners benefits. Consequently, the adversary equipped with ADA-MIA can only access a relatively small auxiliary dataset that contains a mixture of member and non-member samples but does not share the same distribution as the target encoders pre-training data. Note that the adversary is assumed not to know the exact membership labels of samples in this auxiliary set (see Table I). Due to both the distribution mismatch and the insufficient data scale, the adversary cannot rely on traditional shadow model training pipelines or supervised attack model construction[4] [10], [11].

[4]We further verify this claim empirically in Section V-B3.

Considering that the auxiliary dataset contains member samples without their membership labels, and that samples with similar embedding compactness to their augmented versions are more likely to share the same membership property, ADA-MIA utilizes an unsupervised clustering technique $\mathcal{C}$ to train the attack model, based on the membership feature vectors V_a of the samples in the auxiliary dataset D_a:

$$\mathcal{M} \leftarrow \mathcal{C}(V_a). \tag{7}$$

After obtaining the attack model employing an unsupervised manner, it can leverage the attack model $\mathcal{M}$ to conduct targeted membership inference. Given a target sample $\mathbf{x}_t$, ADA-MIA first obtains the membership feature $\mathbf{v}_t$ from the similarity of embeddings between the target sample and its aggressive augmented version, and then leverages $\mathcal{M}$ to infer $\mathbf{x}_t$'s membership property.

Intuitively, any unsupervised clustering algorithms (e.g., K-Means [63] or DBSCAN [64]) can be used in the inference step, and we choose to utilize DeepCluster [65], mainly renowned for its adaptable performance in unsupervised clustering tasks. Specifically, DeepCluster adopts an alternating iterative mechanism: it first performs clustering by using the unsupervised clustering algorithm on the high-dimensional features extracted by the feature extraction model, and then uses the resulting cluster assignments as pseudo-labels to update the model parameters, thereby enabling iterative optimization of unsupervised feature representations. This iterative process progressively refines the feature space, ultimately yielding more accurate clustering results than methods that rely on fixed, predefined features.

Empirical analysis reveals the membership feature distribution overlap between member and non-member images, with multiple images clustered in ambiguous regions between the member and non-member centroids. This may be due to information loss during the membership feature extraction. Consequently, clustering effectiveness degrades when the cluster number is directly set to only two. To address this, we introduce a multi-level clustering paradigm to DeepCluster. Specifically, the clustering algorithm of DeepCluster first decomposes the membership features V_a into K clusters $\{c_1, \cdots, c_K\}$, each cluster corresponds to a centroid $\{\mu_1, \cdots, \mu_K\}$. It subsequently groups these features into two final clusters. For the membership feature $\mathbf{v}_a$ of $\mathbf{x}_a$, the distance $\mathbf{d}_a$ is calculated by:

$$\mathbf{d}_a = \{\mathrm{dist}(\mathbf{v}_a, \mu_1), \cdots, \mathrm{dist}(\mathbf{v}_a, \mu_K)\} \in \mathbb{R}^K \tag{8}$$

where $\mathrm{dist}(\cdot)$ denotes the distance measurement method. Finally, the cluster exhibiting less mean latent distance between original samples and their aggressively augmented views is labeled as *Member*, and the other as *Non-Member*. Relevant implementations are detailed in Section V-A5.

C. Supervised ADA-MIA

Under real-world attack scenarios, it occasionally remains possible for adversaries to obtain access to a part of the training data of target encoder with exact membership labels, as exemplified in the settings of existing works [12], [25], [66], [67]. In this case, ADA-MIA can be simply extended to a supervised version ADA-MIA$_{\text{sup}}$, whose threat model is defined as (c.f. Table I and Section III):

Adversary's Knowledge. ADA-MIA$_{\text{sup}}$ assumes the adversary to obtain the similar priori knowledge as ADA-MIA, only acquiring the exact membership labels for samples in D_a.

Adversary's Capacity. Same as ADA-MIA, the adversary can query f with $\mathbf{x}_a$ and its augmented view $t(\mathbf{x}_a)$, obtaining corresponding embedding $f(\mathbf{x}_a)$ and $f(t(\mathbf{x}_a))$.

Adversary's Goal. The adversary maintains the same goal as ADA-MIA, determining whether a given sample $\mathbf{x}_t$ belongs to the pre-training data of f with black-box query access.

ADA-MIA$_{\text{sup}}$ retains the membership feature generation pipeline of ADA-MIA, only allows the adversary to train the attack model $\mathcal{M}$ with membership features and exact labels of samples from D_a using a supervised manner:

$$\mathcal{M} \leftarrow \mathcal{T}(V_a|\{0,1\}). \tag{9}$$

Here $\mathcal{T}$ denotes the supervised training strategy (e.g., via XGBoost, SVM, or Random Forest) utilized by the adversary. After training, ADA-MIA$_{\text{sup}}$ leverages $\mathcal{M}$ for inferring membership status of any target input $\mathbf{x}_t$.

V. Evaluation

A. Experiment Setting

1) Datasets: We evaluate the performance of ADA-MIA on different datasets, which remains consistent with EncoderMI [11], PartCrop [12] and various existing MIA studies on image classification [30]–[32], [41].

CIFAR-10 [68] consists of $60,000$ images from 10 object classes. It comprises $50,000$ training images and $10,000$ testing images, with each image having a size of 32×32 pixels.

CIFAR-100 [68] encompasses $60,000$ images spanning 100 object categories, and is subdivided into $50,000$ training images and $10,000$ testing images. Each image in CIFAR-100 is rendered in a 32×32 resolution.

STL-10 [69] includes a total of $13,000$ labeled images categorized into 10 distinct classes, alongside an additional $100,000$ unlabeled images. The labeled subset is specifically partitioned into $5,000$ training images and $8,000$ testing images, with each image having a size of 96×96 pixels.

Tiny-ImageNet-200 [70] consists of $100,000$ training images and $10,000$ testing images from 200 classes. Each class has 500 training images and 50 testing images. Each image has a size of 64×64 pixels.

2) Target Encoders: We evaluate ADA-MIA on ten different target encoders: eight of which are locally implemented, and two are obtained from open-source encoder providers. In our experiments, we treat all the target encoders as black-boxes.

Local target encoders. We conduct local pre-training for eight contrastive encoders using the four datasets, employing the MoCo [1] and SimCLR [2] algorithms, respectively. Our pre-training process utilizes publicly available implementations of MoCo v3[5] and SimCLR[6] with default parameter settings. Unless otherwise specified, each target encoder utilizes ResNet-18 [17] as the backbone architecture and undergoes training for a duration of $1,000$ epochs. After finishing the pre-training phase, we remove the projection head and retain the feature extraction component of each CL encoder. This retained component enables us to extract embeddings from the input images.

Public target encoders. We also assess the performance of ADA-MIA using two publicly available encoders to gauge its practical utility. We utilize the well-known library for self-supervised methods in unsupervised representation learning, Solo-Learn[7], which provides the weights and checkpoints for released pre-trained CL encoders. Specifically, we select two representative encoders from Solo-Learn. One of them is trained with MoCo v3 on the CIFAR-10 dataset[8], and the other is trained with SimCLR on the CIFAR-100 dataset[9]. We download their pre-trained checkpoints and perform MIAs against them, treating them as black-box models.

3) Evaluation Metrics: We utilize the standard metrics in ML field, which include *accuracy*, *precision*, and *recall*, to assess the performance of ADA-MIA, as in previous MIA works [10], [11]. Specifically, accuracy represents the proportion of target samples whose membership are correctly inferred. Precision is the ratio of predicted members that are indeed members. Recall presents the fraction of training records which are correctly inferred as members. To provide a more intuitive and precise evaluation, we also report their *F1-scores*.

4) Comparison Methods: We consider the MIAs for CL in [10]–[12] for comparisons. We also expand four MIAs originally designed for Vision Transformers [59], face recognition [11], person identification [71], and text embedding models [72] to the CL framework as comparison methods.

PartCrop[10] [12] calculates the embedding similarities between the target image and its multiple cropped patches, and then uses these similarities to infer whether this target image is in the target model's training dataset.

RAMIA [59] targets MIAs against Vision Transformers, using the shadow training paradigm to leverage internal attention activations along with the prediction output embedding of a target image to infer its membership status.

EncoderMI [11] applies multiple random augmentations to each target image and utilizes similarities between these embeddings to infer the membership property of this image.

ContrastiveLeaks[11] [10] adds a classification layer to the target encoder, and then treats the encoder as a standard classification model.

EmbasCon [11] is proposed as a baseline in [11], which directly treats the embeddings of the target image generated by the CL encoder as the prediction probabilities of standard classification models in order to perform MIA.

[5]https://github.com/facebookresearch/moco-v3

[6] https://github.com/google-research/simclr

[7]https://github.com/vturrisi/solo-learn

[8]https://drive.google.com/drive/folders/1KwZTshNEpmqnYJcmyYPvfIJ_DNwqtAVj?usp=sharing

[9]https://drive.google.com/drive/folders/13pGPcOO9Y3rBoeRVWARgbM_FEp8OXxZa0?usp=sharing

[10]https://github.com/JiePKU/PartCrop

[11]https://github.com/xinleihe/ContrastiveLeaks

ReID-MIA [71] targets the inference of whether any image from the target user was used to train the ReID model. It evaluates the similarities between the embeddings of the target user's images and its corresponding embedding centroids to conduct the user-level MIA.

PatasWord [72] focuses on the text domain and seeks to infer the membership property of individual sentences. It uses the similarity between embeddings of the central word and other words within a sentence to perform MIAs.

Notably, PartCrop, EncoderMI, ContrastiveLeaks, and EmbasCon all require the shadow data for training shadow models to construct their supervised attack models. To ensure a fair comparison of our membership features derived from aggressive data augmentation with those of other methods, we treat the target model itself as the shadow model in our experiments, eliminating the potential interference of shadow models on the results. As for RAMIA, we combine the activations of the final layer and the first to the fourth layers together as membership features to perform MIAs. Note that this is consistent with the baseline-C proposed in [59]. For ReID-MIA, we treat the target image and all its augmented versions as belonging to the same user, enabling us to derive inference results using its user-level attack. For Patasword, we first divide one image into multiple patches and view the center patch as the center "word", and then compute its cosine similarity with remaining patches to perform MIAs.

5) Implementation Details: We specify the following implementation details of experimental setup for reproducibility:

Data Augmentation. To obtain augmented images, we select four types of commonly employed data augmentations by default, namely *ResizedCrop*, *HorizontalFlip*, *ColorJitter*, and *Grayscale*. For *ResizedCrop* and *ColorJitter*, the intensity parameter λ varies from 0.1 to 1, with a default interval of λ_{int} set to 0.05. For *HorizontalFlip* and *Grayscale*, we always apply them to the target images. All augmentations are implemented using the Torchvision[12] library provided by PyTorch.

Similarity Calculation. We leverage cosine similarity for calculating $S(\cdot,\cdot)$, and Euclidean distance in the embedding space for measuring $\text{dist}(\cdot,\cdot)$ during cluster semantic assignment.

Pre-training Dataset Composition. Following the implementation in [10], [11], we first randomly sample four equal parts from each dataset, i.e., $D^{\text{train}}_{\text{target}}$, $D^{\text{test}}_{\text{target}}$, $D^{\text{train}}_{\text{shadow}}$, and $D^{\text{test}}_{\text{shadow}}$. Each part consists of $10,000$ images. $D^{\text{train}}_{\text{target}}$ is utilized to pre-train the target encoder. $D^{\text{train}}_{\text{target}}$ and $D^{\text{test}}_{\text{target}}$ are treated as ground-truth members and non-members. $D^{\text{train}}_{\text{shadow}}$ is utilized to train the shadow encoder, while $D^{\text{train}}_{\text{shadow}}$ and $D^{\text{test}}_{\text{shadow}}$ are employed to train attack model in a supervised manner. For CIFAR-10, CIFAR-100 and Tiny-ImageNet datasets, we randomly sample $D^{\text{train}}_{\text{target}}$ and $D^{\text{train}}_{\text{shadow}}$ from their respective training datasets, while $D^{\text{test}}_{\text{target}}$ and $D^{\text{test}}_{\text{shadow}}$ are selected from testing datasets. For STL-10 dataset, we randomly sample all four sets from its unlabeled dataset.

Auxiliary Dataset Composition. The auxiliary dataset consists of $1,000$ images, with 500 images randomly selected from $D^{\text{train}}_{\text{target}}$ and $D^{\text{test}}_{\text{target}}$, respectively. During evaluation, we dedicate 800 samples for training the inference attack model, and 200 images for evaluating MIA effectiveness. The ratio of members to non-members is maintained at 1:1 by default (further discussed in Section V-D). It is worth noting that ADA-MIA operates without access to the ground-truth membership status of all images from the auxiliary dataset.

Unsupervised Attack Model Training. ADA-MIA utilizes multi-level DeepCluster[13] to train the unsupervised attack model. It initially decomposes all membership features into 10 distinct clusters, and subsequently group them into 2 final clusters. Within DeepCluster, we employ a fully connected neural network with two hidden layers, each comprising 256 neurons and *ReLU* activation, as the feature extraction models. We employ K-Means as the default algorithm for both levels of clustering. The second-level K-Means is applied to distance vectors to obtain two final clusters. During clustering assignment, the cluster exhibiting a more compact embedding distribution is labeled as members. The number of maximum iterations is set to 800. The tolerance is set to 1×10^{-2}. For training, we utilize cross-entropy as the loss function and employ the Adam optimizer with a learning rate of 1.5×10^{-4} and the weight decay of 1×10^{-4} for 100 training epochs. The default batch size is set to 64. For ADA-MIA, the membership labels are only used for evaluation, not for attack model training.

Supervised Attack Model Training. ADA-MIA$_{\text{sup}}$ retains the similar settings as ADA-MIA in membership feature generation and auxiliary dataset. We utilize the XGBoost[14] [73] algorithm to construct the inference attack model, with a learning rate of 3.5×10^{-3} and the number of estimators as 100. The maximum depth is set to 4, and γ is set to 0.025. Other configurations are kept as default.

B. Performance of ADA-MIA

1) Performance on Local Target Encoders: For local target encoders, Table II shows that ADA-MIA$_{\text{sup}}$ outperforms all comparison method, while ADA-MIA with minimal prior knowledge performs slightly worse than EncoderMI and PartCrop but better than the remaining methods.

For MoCo encoders, ADA-MIA$_{\text{sup}}$ improves the attack F1-score over EncoderMI by 4.98% without requiring detailed pre-training settings. On CIFAR-10, it achieves a mean recall of 89.81%, outperforming all comparison methods by 25%, and reaches a mean accuracy of 83.98% across all MoCo encoders. However, ADA-MIA$_{\text{sup}}$ is not consistently the best: on MoCo/CIFAR-100 setting, EncoderMI achieves 89.20% recall, exceeding ADA-MIA$_{\text{sup}}$ by 7.49%. ADA-MIA ranks slightly below ADA-MIA$_{\text{sup}}$, EncoderMI, and PartCrop. Nevertheless, ADA-MIA ows the smallest standard deviations among all methods, nearly below 4%. Without additional shadow data, it still outperforms various supervised attack methods, achieving a mean accuracy of 72.30%, nearly 20% higher than ContrastiveLeaks. Its recall is slightly lower than ReID-MIA and PartCrop on CIFAR-100 and STL-10.

[12]https://github.com/pytorch/vision/blob/main/torchvision/transforms/transforms.py

[13]https://github.com/facebookresearch/deepcluster

[14]https://github.com/dmlc/xgboost

TABLE II: The performances (%) of our ADA-MIA and all comparison methods against local target encoders.

Method	Metric	MoCo CIFAR-10	MoCo CIFAR-100	MoCo STL-10	MoCo ImageNet	SimCLR CIFAR-10	SimCLR CIFAR-100	SimCLR STL-10	SimCLR ImageNet
ADA-MIA	Accuracy	73.60±1.96	72.01±1.79	72.40±1.50	71.20±2.04	71.60±1.96	71.60±2.33	72.01±1.79	72.80±2.04
	Precision	71.65±2.39	70.99±1.75	71.22±1.01	70.59±2.02	68.15±2.70	72.30±3.16	72.48±2.58	71.74±1.29
	Recall	78.40±4.08	74.40±1.96	75.20±3.92	72.80±4.66	80.80±1.60	70.40±4.08	71.20±3.92	75.20±4.66
	F1-Score	74.78±1.99	72.65±1.77	73.10±2.07	71.59±2.52	74.21±1.09	71.23±2.44	71.74±2.04	73.37±2.64
ADA-MIA$_{\text{sup}}$	Accuracy	**85.56**±1.87	**85.36**±2.83	**82.58**±1.79	**82.42**±0.74	**87.28**±4.14	**83.69**±3.38	**86.67**±4.09	**80.67**±1.33
	Precision	79.76±11.90	**91.43**±11.43	**88.81**±9.81	81.99±1.63	**88.61**±9.95	**92.67**±9.04	74.22±6.22	**84.29**±13.93
	Recall	**89.81**±8.52	81.71±10.63	72.81±11.05	**81.33**±1.63	**89.17**±9.72	75.22±8.09	**86.67**±16.33	**82.00**±16.55
	F1-Score	**83.45**±4.34	**84.92**±2.19	**78.80**±4.72	**82.66**±1.33	**87.96**±3.86	**82.31**±4.43	79.72±10.67	**80.33**±3.06
PartCrop [12]	Accuracy	81.07±4.32	79.07±3.67	78.07±4.14	74.07±4.49	81.42±1.39	80.73±1.52	80.07±5.96	77.40±3.28
	Precision	81.60±3.13	79.48±5.04	78.65±4.16	74.29±4.56	80.22±1.67	79.60±1.84	79.97±5.01	76.58±4.27
	Recall	80.40±3.28	80.73±3.82	**78.07**±4.03	72.07±4.99	88.73±2.16	**85.40**±1.56	80.07±8.98	79.40±2.83
	F1-Score	80.98±3.16	80.07±3.33	77.85±4.23	73.04±4.79	84.24±1.31	81.38±1.38	**79.88**±6.64	77.88±2.79
EncoderMI [11]	Accuracy	82.19±4.57	82.35±4.49	80.35±4.43	78.80±4.41	81.06±5.57	79.80±5.04	81.50±2.98	75.95±2.01
	Precision	**82.80**±9.87	79.87±8.25	86.74±8.76	**82.59**±9.59	87.89±4.52	78.41±6.02	**85.54**±8.31	75.18±8.18
	Recall	84.39±9.94	**89.20**±8.41	73.69±10.02	76.90±15.80	80.00±9.62	82.80±4.68	75.90±9.54	81.89±14.62
	F1-Score	82.59±3.93	83.56±3.47	78.77±5.37	77.68±7.07	83.11±4.29	80.45±4.61	79.68±3.71	76.78±4.28
Contrastive -Leaks [10]	Accuracy	69.57±1.25	63.53±7.19	63.20±7.68	61.87±6.16	68.71±2.28	63.87±5.66	61.20±5.49	60.53±4.91
	Precision	69.73±7.42	63.42±7.39	64.75±8.67	63.93±7.68	68.82±4.72	62.24±6.16	60.07±4.89	59.64±4.56
	Recall	69.87±6.39	65.67±4.19	57.67±8.35	55.07±6.12	69.33±6.97	73.00±6.71	67.00±5.82	65.67±4.19
	F1-Score	59.28±1.39	56.26±3.68	58.16±0.97	52.09±3.26	52.71±0.88	59.32±0.84	58.31±0.70	56.84±2.15
RAMIA [59]	Accuracy	66.57±2.04	67.24±2.99	64.29±0.98	66.76±2.53	66.36±17.02	64.82±11.63	66.48±0.98	66.41±1.96
	Precision	71.29±10.83	66.85±7.75	61.33±1.12	70.73±4.54	64.98±17.83	64.69±13.66	64.96±0.63	63.54±0.17
	Recall	64.80±19.82	75.20±16.47	76.40±23.75	55.20±1.60	80.80±10.55	75.20±16.67	72.85±2.99	51.20±5.88
	F1-Score	69.54±1.92	64.47±5.87	60.99±8.47	66.07±6.49	68.76±2.32	66.92±4.57	63.32±5.14	62.49±4.32
ReID-MIA [71]	Accuracy	56.70±0.48	61.14±2.10	57.80±0.40	57.50±1.94	56.20±1.47	59.29±1.59	58.93±1.86	55.69±2.42
	Precision	58.07±0.76	56.04±2.21	58.45±0.93	52.93±2.24	55.46±1.18	63.43±0.53	62.85±1.07	59.35±1.86
	Recall	60.40±1.96	56.77±6.32	54.00±2.19	51.67±5.98	50.80±0.98	54.03±0.67	52.42±1.39	53.85±3.63
	F1-Score	64.91±6.23	68.97±4.68	63.31±6.46	61.92±1.89	71.02±3.57	67.99±10.99	63.49±1.36	60.19±4.23
PatasWord [72]	Accuracy	51.00±0.81	50.46±0.29	50.56±0.67	50.86±0.71	50.86±0.50	51.02±0.63	50.38±0.37	51.16±0.63
	Precision	50.78±6.94	54.01±2.78	51.33±1.32	55.77±5.72	50.80±0.92	51.71±1.99	50.39±0.37	51.60±1.16
	Recall	30.40±4.69	6.44±1.55	11.28±7.19	16.80±13.29	14.40±11.91	20.96±12.64	15.88±6.10	47.08±17.47
	F1-Score	56.47±10.86	11.44±2.49	17.26±12.36	27.24±22.15	60.37±8.89	46.31±13.76	34.68±5.19	47.27±9.93
EmbasCon [11]	Accuracy	49.90±2.14	51.20±2.02	49.95±1.53	48.89±1.02	49.59±1.35	50.00±0.72	51.50±1.52	50.15±1.02
	Precision	49.73±1.86	51.09±1.67	50.36±1.78	50.99±3.68	48.56±3.55	50.03±0.59	51.16±1.86	50.18±1.12
	Recall	64.39±12.42	64.10±19.56	42.40±9.79	60.80±42.75	56.80±30.00	47.69±19.19	53.00±12.89	49.60±20.37
	F1-Score	55.70±5.86	55.36±8.57	45.23±5.41	45.16±25.16	48.34±17.51	46.95±9.65	51.25±6.27	47.99±9.42

The **bold** numbers represent the best attack effectiveness, while the underlined numbers indicate the second-best. The numbers following "plus-minus sign" are standard deviations in five trials.

For SimCLR encoders, ADA-MIA performs similarly to that on MoCo encoders, achieving an F1-score above 71%. It attains an attack accuracy of 72.01% and a precision of 72.48%, outperforming the RAMIA by at least 5%, although its recall is slightly lower than ReID-MIA. ADA-MIA$_{\text{sup}}$ outperforms EncoderMI and PartCrop by 5%, achieving accuracy of 87.28% and 86.67% on CIFAR-10 and STL-10, respectively. On CIFAR-100 and Tiny-ImageNet, it yields similar performances with EncoderMI and PartCrop, while still outperforming the other method.

2) Performance on Public Target Encoders: For public target encoders, our method achieves the best attack performance, as shown in Table III ADA-MIA$_{\text{sup}}$ achieves F1-scores above 80% on both encoders, surpassing PartCrop and EncoderMI by approximately 5% and 17%, respectively. On SimCLR/CIFAR-100, its recall reaches 93.10%, though its precision remains at least 10% lower than EncoderMI. ADA-MIA maintains nearly 70% across four metrics. Compared with local encoders, its performance gap with EncoderMI & PartCrop further narrows.

The advantage of ADA-MIA may stem from its reduced reliance on training details. Its diverse data augmentations provide more opportunities to capture membership information, regardless of the augmentations used during pre-training. Moreover, exploiting similarity between differently augmented views appears more effective than the similarity between different images and corresponding embedding centroids.

TABLE III: The effectiveness (%) of our method and all comparisons against public target encoders.

(a) Performance on MoCo encoders pre-trained on CIFAR-10.

Method	Accuracy	Precision	Recall	F1-Score
ADA-MIA	70.40	70.38	71.20	70.58
ADA-MIA$_{\text{sup}}$	**82.98**	84.67	**81.95**	**81.47**
PartCrop	76.87	77.47	76.33	76.59
EncoderMI	74.20	**94.88**	52.40	64.21
ContrastiveLeaks	52.88	53.23	47.64	50.26
RAMIA	61.20	59.75	64.63	62.01
ReID-MIA	66.00	72.43	60.80	61.25
PatasWord	51.00	50.94	51.02	51.16
EmbasCon	50.88	51.18	51.00	51.08

The encoder's test accuracy is 92.94%.

(b) Performance on SimCLR encoders pre-trained on CIFAR-100.

Method	Accuracy	Precision	Recall	F1-Score
ADA-MIA	69.60	69.22	72.80	70.19
ADA-MIA$_{\text{sup}}$	**82.60**	71.80	**93.10**	**80.10**
PartCrop	75.20	72.68	82.33	76.32
EncoderMI	76.35	**88.34**	61.80	71.50
ContrastiveLeaks	52.64	52.45	56.60	54.44
RAMIA	62.80	62.89	62.40	62.64
ReID-MIA	65.20	71.94	64.80	62.24
PatasWord	51.08	51.11	51.10	51.08
EmbasCon	51.60	51.60	51.70	51.70

The encoder's test accuracy is 65.78%.

3) Impact of Training Data Amount: To validate the data efficiency of ADA-MIA$_{\text{sup}}$, we evaluate supervised MIAs on the public MoCo encoder pretrained on CIFAR-10. We vary the auxiliary set size of ADA-MIA$_{\text{sup}}$ from 500 to 800 samples, and evaluate EncoderMI and PartCrop using both their original data budgets (10K and 25K samples, respectively) and the

TABLE IV: Performance comparison (%) on MoCo encoder trained on CIFAR-10 with varying additional data amount.

Method	ADA-MIA$_{sup}$				EncoderMI			PartCrop	
Amount	500	600	700	800	10K	10K*	800	25K	800
Accuracy	63.20	71.30	75.80	82.98	74.20	60.60	51.10	76.87	61.65
Precision	62.31	71.52	75.49	84.67	94.88	61.73	51.09	77.47	61.95
Recall	66.80	70.80	76.40	81.95	52.40	55.80	51.40	76.63	60.40
F1-score	64.48	71.16	75.94	81.47	64.21	58.61	51.25	76.59	61.16

10K*: It utilizes SimCLR/CIFAR-100 shadow encoder.

TABLE V: The attack F1-score (%) of ADA-MIA and ADA-MIA$_{sup}$ with different attack model training algorithms. *w.* & *w/o.* denote with and without multi-level clustering, respectively.

Method	MoCo / CIFAR-10		SimCLR / CIFAR-100	
	w.	*w/o.*	*w.*	*w/o.*
Clustering Methods for ADA-MIA				
K-Means	67.69	64.86	67.21	63.69
DBSCAN	67.68	63.53	67.25	62.96
Spectral Clustering	66.54	64.06	66.08	61.44
Agglomerative Clustering	67.84	65.43	66.97	64.95
DeepCluster	**70.58**	**67.33**	**70.19**	**66.79**
Supervised Methods for ADA-MIA$_{sup}$				
Neural Network	81.16		**80.23**	
Random Forest	80.37		79.46	
SVM	80.34		79.89	
Logistic Regression	80.05		79.77	
KNN	80.13		79.78	
XGBoost	**81.47**		80.10	

same 800-sample budget. For EncoderMI, the target encoder itself serves as the shadow model, while 10K* instead uses a public SimCLR encoder pretrained on CIFAR-100. Table IV show that, with only 800 auxiliary samples, ADA-MIA$_{sup}$ matches or outperforms baselines trained with substantially larger datasets, whereas reducing to 800 samples degrades EncoderMI's performance toward random guessing. These results also support our threat model where the auxiliary set cannot be directly used for shadow training.

C. Impact of Attack Model Training Algorithm

To examine how the selection of attack model training algorithm affects MIA performance, we evaluate ADA-MIA and ADA-MIA$_{sup}$ using different clustering and supervised methods. Table V suggests that DeepCluster and XGBoost exhibit the best overall performance among various algorithms. Furthermore, it validates that our proposed multi-level clustering introduces effectiveness improvement for ADA-MIA.

D. Impact of Member-to-Non-Member Ratio

Conventional MIAs [11], [23] commonly train the attack model on balanced shadow datasets with a 1:1 member-to-non-member ratio. In contrast, ADA-MIA utilizes an unlabeled auxiliary dataset containing some member samples, which in practice is often skewed toward non-members [29]. To evaluate ADA-MIA under such imbalanced conditions, we use an auxiliary set of $1,000$ samples and and vary the member-to-non-member ratio from 0.1 to 0.9 with an interval of 0.2.

Following LiRA [29], we evaluate the attack using ROC curves and AUC, where an ideal MIA has an AUC of 1 and random guessing yields 0.5. These metrics offer an intuitive and comprehensive assessment of the attack's performance.

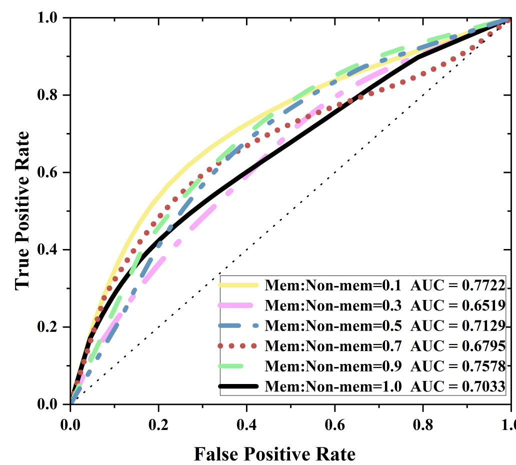

Fig. 4: The ROC curves evaluating ADA-MIA's performance under varying member-to-non-member ratios in the auxiliary dataset during unsupervised membership inference (Section IV-B). The black dotted line denotes random guessing.

Figure 4 demonstrates that ADA-MIA mains robust under highly imbalanced ratios, achieving an AUC of 0.772 at a ratio of 0.1. Its robustness can be attributed to DeepClusters effectiveness in handling outliers [74], which helps ADA-MIA handle clusters containing few member samples.

E. Impact of Data Augmentations

1) Intersection of Data Augmentations: To evaluate the impact of discrepancies between data augmentations used by the attacker and those used by target encoders on attack performance, we retain the four augmentations for generating membership features while excluding one augmentation during the local encoder's pre-training phase. These experiments are conducted on encoders pre-trained on CIFAR-10 using MoCo and SimCLR algorithms. This setup is designed to mimic real-world scenarios in which potential attackers utilize established augmentation combinations to target encoders that have been pre-trained with undisclosed augmentation techniques. Furthermore, the results of these experiments enable us to identify which data augmentation plays the most significant role in leaking the membership information.

From the results in Table VI we can see that even when the data augmentations employed by the attacker do not align with the pre-training augmentations, ADA-MIA can achieve the robust attack performance. Specifically, when omitting ResizedCrop during the pre-training process of target encoders, the F1-score of ADA-MIA exhibits a decrease of nearly 5%. In contrast, omitting HorizontalFlip has the minimal impact on the inference performance, resulting in a mere 0.65% decrease in the F1-score for the SimCLR encoder, compared to other augmentation omissions. These experimental results indicate that the utilization of ResizedCrop during the pre-training process poses a greater risk of membership leakage, whereas HorizontalFlip presents the lowest risk for membership privacy breaches.

One possible explanation lies in the extremely distinctive views generated by ResizedCrop in comparison to the original

TABLE VI: The attack performance (%) of ADA-MIA. We retain the four augmentations when generating membership features while excluding one augmentation during pre-training of the target encoder.

(a) Performance on MoCo encoders pre-trained on CIFAR-10.

Augmentation	Test Acc.	Acc.	Pre.	Rec.	F1
w/o ——	76.12	72.83	73.10	72.33	72.69
w/o ResizedCrop	71.98	65.50	65.25	66.33	65.78
w/o HorizontalFlip	74.46	70.50	70.44	70.67	70.53
w/o ColorJitter	72.62	68.67	67.78	71.33	69.48
w/o Grayscale	73.58	70.50	68.86	75.00	71.76

(b) Performance on SimCLR encoders pre-trained on CIFAR-100.

Augmentation	Test Acc.	Acc.	Pre.	Rec.	F1
w/o ——	46.62	71.60	68.15	80.80	71.23
w/o ResizedCrop	40.78	64.40	64.79	66.40	64.82
w/o HorizontalFlip	45.91	70.00	70.21	72.80	70.58
w/o ColorJitter	43.58	66.80	67.62	64.80	66.01
w/o Grayscale	44.34	69.60	71.24	67.20	68.71

image. Consequently, excluding this transformation from the pre-training stage renders target encoder less adept at handling strongly augmented images. Therefore, querying the target encoder with such augmented samples leads to the embeddings of both augmented members and non-members diverging greatly from the original images, yielding the decreased performance of ADA-MIA.

2) Non-Overlap of Data Augmentations: Further, to rigorously evaluate the influence of a priori knowledge concerning data augmentation types on ADA-MIA, we categorized the four standard data augmentations into two groups: one is employed for pre-training the target encoders, while the other is utilized for generating membership features. In this section, we perform experiments using the MoCo and SimCLR algorithms on the CIFAR-10 dataset.

From the results in Table VIII, it becomes evident that ADA-MIA maintains a mean attack accuracy of approximately 68.53%, even when employing different data augmentations compared to those used by the target encoder. Furthermore, ADA-MIA consistently exhibits a F1-score of nearly 70% across all data augmentation configurations. Specifically, when we employ HorizontalFlip and Grayscale for encoder pre-training and utilize ResizedCrop and ColorJitter for generating membership features, ADA-MIA appears to exhibit slightly inferior performance compared to other settings. This can be attributed to the fact that, in comparison to HorizontalFlip and Grayscale, the use of ResizedCrop and ColorJitter in the pre-training process introduces a higher risk of membership privacy leakage, as shown by the results in Section V-E1. These experiments demonstrate that our ADA-MIA does not require pre-knowledge of the data augmentation settings of the target models, and further highlight the potential risks associated with membership information leakage within data augmentation in CL.

F. Impact of Data Augmentation Parameters

In ADA-MIA, the intervals and the range of the parameters of data augmentation determine how many augmented images of the target samples will be generated. So next, we evaluate the impact of the data augmentation parameters in ADA-MIA.

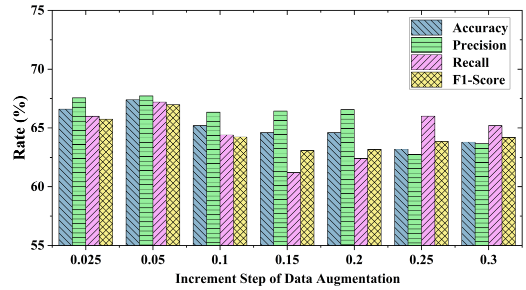

Fig. 5: Performance of ADA-MIA under different interval settings during membership feature generation (Section IV-A). The interval is controlled by λ_{int}. The target encoder is pretrained on CIFAR-10 with MoCo.

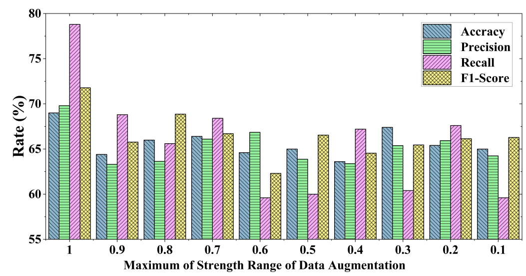

Fig. 6: Performances of ADA-MIA under different maximum augmentation intensity settings during membership feature generation (Section IV-A). The maximum intensity is controlled by the maximum value of λ. The target encoder is pretrained on CIFAR-10 with MoCo.

1) Impact of the Strength Interval of Augmentation: To assess the impact of the data augmentation strength interval used by ADA-MIA, we vary the interval parameter λ_{int} from 0.025 to 0.5 and examine its impact on our MIA's inference performance. The experimental results in Figure 5 indicate that as the data augmentation interval increases, the MIA's inference performance gradually decreases. In particular, when λ_{int} is set at 0.025, we achieve an attack accuracy of 66.53% and an F1-score of 65.84%. However, as we increase λ_{int} beyond 0.05, ADA-MIA's performance starts to decline. When the interval ranges from 0.025 to 0.3, all performance metrics drop by approximately 4%. This decline in performance is mainly due to the larger intervals, which yield fewer augmented images and less finely focus on the detailed aspects of the target image. Consequently, this approach leads to less refined membership features, thereby reducing the performance of our MIA.

2) Impact of Maximum Strength of Augmentation: To evaluate the impact of this strength parameter on our MIA, we maintain the default interval of 0.05 while modifying only the maximum value of λ, which is set to range from 0.1 to 1. From the results in Figure 6, it is evident that when the maximum value of λ is below 0.9, the performance of our MIA experiences a dramatic decline, particularly in terms of the recall metric. However, within the range of 0.1 to 0.9 for the maximum value of λ, the accuracy and precision of ADA-MIA remain relatively stable, which are all around 65%.

TABLE VII: Performance (%) of ADA-MIA against target encoders pre-trained with early-stopping.

Methods	Metric	MoCo				SimCLR			
		CIFAR-10	CIFAR-100	STL-10	ImageNet	CIFAR-10	CIFAR-100	STL-10	ImageNet
ADA-MIA	Accuracy	**68.80**(4.80)	67.20(4.81)	**68.40**(4.00)	**68.80**(2.40)	**68.80**(2.80)	**69.20**(2.40)	**67.20**(4.81)	66.40(6.40)
	Precision	**71.78**(0.13)	69.26(1.73)	67.59(3.63)	**68.85**(1.74)	**68.46**(0.31)	**72.94**(0.64)	**70.46**(2.02)	64.39(7.35)
	Recall	62.40(16.00)	62.40(12.00)	**71.20**(4.00)	70.40(2.40)	68.81(11.99)	**61.60**(8.80)	**60.80**(10.40)	73.60(1.60)
	F1-Score	66.62(8.16)	65.44(7.21)	**69.18**(3.92)	69.21(2.38)	68.39(5.82)	**66.45**(4.78)	**64.62**(7.12)	**68.62**(4.75)
ADA-MIA$_{sup}$	Accuracy	68.57(16.99)	**67.94**(17.42)	66.84(15.74)	67.79(14.63)	66.48(20.80)	66.48(17.21)	60.20(26.47)	**67.41**(13.26)
	Precision	68.03(11.73)	**70.61**(20.82)	**69.27**(19.54)	67.61(14.38)	67.66(20.95)	70.06(22.61)	61.08(13.14)	**73.94**(10.35)
	Recall	74.79(15.02)	**71.47**(10.24)	66.19(6.62)	**73.27**(8.06)	**75.04**(14.13)	60.84(14.38)	58.80(27.87)	65.20(16.80)
	F1-Score	**71.07**(12.38)	**70.32**(14.60)	67.09(11.71)	**69.46**(13.20)	**70.51**(17.45)	65.04(17.27)	58.92(20.80)	68.49(11.84)
EncoderMI	Accuracy	59.05(23.14)	57.65(24.70)	54.25(26.10)	54.50(24.30)	60.00(21.06)	58.75(21.05)	54.40(27.10)	52.40(23.55)
	Precision	60.52(22.28)	64.26(15.61)	67.85(18.89)	61.63(20.96)	65.43(22.46)	62.99(15.42)	61.59(23.95)	52.36(22.82)
	Recall	57.44(26.95)	48.90(40.30)	57.01(16.68)	66.19(10.71)	54.70(25.30)	55.40(27.40)	52.79(23.11)	**73.69**(8.20)
	F1-Score	55.89(26.70)	50.37(33.19)	47.04(31.73)	54.03(23.65)	55.22(27.89)	52.31(28.14)	49.11(30.57)	58.73(18.05)
Contrastive -Leaks	Accuracy	53.70(15.87)	52.50(11.03)	52.70(10.50)	51.10(10.77)	51.90(16.81)	52.80(11.07)	52.40(8.80)	50.80(9.73)
	Precision	53.81(15.92)	52.58(10.84)	52.57(12.18)	51.13(12.80)	52.16(16.66)	52.48(9.76)	52.38(7.69)	50.46(9.18)
	Recall	52.20(17.67)	51.00(14.67)	55.20(2.47)	49.80(5.27)	45.80(23.53)	51.21(21.79)	44.35(22.65)	43.95(21.72)
	F1-Score	52.99(6.29)	51.78(4.48)	53.85(4.31)	50.46(1.63)	48.78(3.93)	51.84(7.48)	48.03(10.28)	46.98(9.86)
ReID-MIA	Accuracy	54.80(1.90)	54.00(7.14)	47.20(10.60)	57.20(0.30)	54.80(1.40)	58.80(0.49)	52.80(6.13)	56.00(0.31)
	Precision	53.22(4.85)	63.16(7.12)	48.27(10.18)	59.88(6.95)	58.28(2.82)	62.32(1.11)	66.66(3.81)	63.59(4.24)
	Recall	**81.60**(21.20)	41.60(15.17)	68.01(14.01)	54.40(2.73)	54.40(3.60)	52.00(2.03)	53.60(1.18)	42.40(11.45)
	F1-Score	64.38(0.53)	40.12(28.85)	60.60(2.71)	50.18(11.74)	49.93(21.09)	52.79(15.20)	52.00(11.49)	43.16(17.03)

The numbers in parentheses means the percentage by which the efficacy of each MIA has altered compared with no defense. Red number represents a reduction in effectiveness, while green number signifies an increase.

TABLE VIII: Performance (%) of ADA-MIA under different exclusions of augmentations (***R***, ***H***, ***C***, and ***G*** respectively stand for *ResizedCrop*, *HorizontalFlip*, *ColorJitter*, and *Grayscale*).

(a) Performance on MoCo encoders pre-trained on CIFAR-10.

Pre-training	Feature Generation	Test Acc.	Acc.	Pre.	Rec.	F1
R, ***H***, ***C***, ***G***	***R***, ***H***, ***C***, ***G***	76.12	72.83	73.10	72.33	72.69
R, ***H***	***C***, ***G***	72.98	68.50	67.75	71.00	69.27
R, ***C***	***H***, ***G***	72.42	70.67	69.35	74.00	71.57
R, ***G***	***H***, ***C***	71.85	69.17	68.69	70.67	69.63
H, ***C***	***R***, ***G***	71.26	71.17	70.18	73.67	71.84
H, ***G***	***R***, ***C***	71.18	65.67	65.03	68.67	66.74
C, ***G***	***H***, ***R***	72.67	70.67	69.64	73.33	71.43

(b) Performance on SimCLR encoders pre-trained on CIFAR-100.

Pre-training	Feature Generation	Test Acc.	Acc.	Pre.	Rec.	F1
R, ***H***, ***C***, ***G***	***R***, ***H***, ***C***, ***G***	46.62	71.60	68.15	80.80	71.23
R, ***H***	***C***, ***G***	42.68	69.60	71.31	61.60	66.46
R, ***C***	***H***, ***G***	43.16	68.00	64.03	82.40	72.04
R, ***G***	***H***, ***C***	41.87	70.00	70.66	71.20	70.21
H, ***C***	***R***, ***G***	40.12	67.20	67.91	67.20	66.99
H, ***G***	***R***, ***C***	39.69	65.20	64.27	72.00	67.01
C, ***G***	***H***, ***R***	41.66	67.60	68.35	67.20	67.25

These findings suggest that the aggressive data augmentation applied to the target sample indeed provides useful membership information. By combining the aggressive augmentation with standard augmentation techniques, we can enhance the performance of our MIA against CL encoders.

VI. Performance Against Defenses

A. Reducing Overfitting via Early Stopping

Liu et al. [11] illustrate that a crucial factor contributing to membership information leakage is the overfitting of the target encoder to its training data, and they demonstrate that the use of early stopping [75] can effectively mitigate MIAs. Specifically, early stopping is a classic method for reducing overfitting on training samples, only requiring the training of target encoders for fewer epochs.

To evaluate the defense effect of early stopping, we conduct an evaluation of ADA-MIA's attack performance on MoCo and SimCLR with three MIA comparisons. Specifically, we train eight local target encoders from scratch on four datasets using MoCo and SimCLR, incorporating early stopping during the training process. The experiment results are presented in Table VII. Notably, the effectiveness of EncoderMI diminishes rapidly when applied to less overfitted encoders. In contrast, ADA-MIA and ADA-MIA$_{sup}$ manage to maintain respectable precision and accuracy. This underscores that EncoderMI heavily relies on overfitting, while the basis for inference in ADA-MIA is not closely tied to overfitting.

From our perspective, this phenomenon can be attributed to the extensive use of data augmentation by CL encoders during training. As a result, the model inevitably "memorizes" a part of details of the training samples. Even with the early stopping technique, the employed aggressive data augmentation enables our method to effectively extract the memorized details from the training data, thus enhancing the performance of our method to conduct the attack.

B. Pre-training with Differential Privacy

Differential privacy [76], [77] has been established as a framework offering formal data privacy guarantees for ML models through probabilistic privacy mechanisms. In order to evaluate the defense capabilities of differential privacy against MIAs, we apply DP-SGD [78] to all model parameters and pre-train the local target encoders. Specifically, we utilize the open-source PyTorch library Opacus[15] [79] with default parameters $(\epsilon, \delta) = (50.0, 10^{-5})$ within our local pre-training pipeline, which is consistent with the implementations in PartCrop [12].

The results regarding DP-SGD's defense performance are outlined in Table IX. From the results we could see that DP-SGD can effectively defend against EncoderMI and ReID-MIA. However, ADA-MIA and ADA-MIA$_{sup}$ achieve an attack accuracy of approximately 70% on CIFAR-10

[15] https://github.com/pytorch/opacus/blob/main/tutorials/building_image_classifier.ipynb

TABLE IX: Performance (%) of ADA-MIA against target encoders pre-trained with DP-SGD.

Methods	Metric	MoCo				SimCLR			
		CIFAR-10	CIFAR-100	STL-10	ImageNet	CIFAR-10	CIFAR-100	STL-10	ImageNet
ADA-MIA	Accuracy	68.00(5.60)	68.80(3.21)	65.01(7.39)	66.19(5.01)	**68.39**(3.21)	68.80(2.80)	**63.59**(8.42)	64.20(8.60)
	Precision	70.88(0.77)	67.88(3.11)	66.55(4.67)	**68.87**(1.72)	**70.75**(2.60)	69.06(3.24)	62.93(9.55)	**67.19**(4.55)
	Recall	62.40(16.00)	**74.40**(0.00)	61.19(14.01)	60.40(12.40)	66.40(14.40)	72.00(1.60)	**68.80**(2.40)	56.80(18.40)
	F1-Score	65.87(8.91)	70.04(2.61)	63.48(9.62)	64.03(7.56)	**67.47**(6.74)	68.89(2.34)	64.95(6.79)	61.29(12.08)
ADA-MIA$_{sup}$	Accuracy	**71.32**(14.24)	**71.39**(13.97)	**69.24**(13.34)	**70.41**(12.01)	67.74(19.54)	**70.61**(13.08)	62.98(23.69)	**70.12**(10.55)
	Precision	**71.03**(8.73)	**73.18**(18.25)	**69.04**(19.77)	67.40(14.59)	64.92(23.69)	64.43(28.24)	**76.79**(2.57)	63.36(20.93)
	Recall	71.67(18.14)	71.82(9.89)	**73.19**(0.38)	**78.61**(2.72)	**69.16**(20.01)	**77.14**(1.92)	64.75(21.92)	**78.61**(3.39)
	F1-Score	**70.47**(12.98)	**71.82**(13.1)	**70.19**(8.61)	**72.28**(10.38)	66.61(21.35)	**70.05**(12.26)	**69.87**(9.85)	**68.99**(11.34)
EncoderMI	Accuracy	57.10(25.09)	54.25(28.10)	53.75(26.60)	53.25(25.55)	59.45(21.61)	57.55(22.25)	54.20(27.30)	57.00(18.95)
	Precision	66.23(16.57)	54.12(25.75)	57.89(28.85)	62.79(19.80)	66.23(21.66)	**71.76**(6.65)	62.51(23.03)	62.67(12.51)
	Recall	**76.89**(7.50)	64.89(24.31)	63.80(9.89)	41.00(35.90)	65.15(14.85)	51.20(31.60)	57.09(18.81)	56.20(25.69)
	F1-Score	63.98(18.61)	57.61(25.95)	52.72(26.05)	39.11(38.57)	59.85(23.26)	49.41(31.04)	48.90(30.78)	51.79(24.99)
Contrastive -Leaks	Accuracy	52.94(16.63)	51.76(11.77)	49.41(13.79)	50.59(11.28)	51.76(16.95)	52.94(10.93)	49.41(11.79)	47.06(13.47)
	Precision	48.78(20.95)	47.73(15.69)	45.00(19.75)	45.45(18.48)	47.62(21.20)	48.94(13.30)	43.33(16.74)	42.86(16.78)
	Recall	51.28(18.59)	53.85(11.82)	46.15(11.52)	38.46(16.61)	51.28(18.05)	58.97(14.03)	33.33(33.67)	46.15(19.52)
	F1-Score	50.00(9.28)	50.60(5.66)	45.57(12.59)	41.67(10.42)	49.38(3.33)	53.49(5.83)	37.68(20.63)	44.46(12.38)
ReID-MIA	Accuracy	53.60(3.10)	54.00(7.14)	54.80(3.00)	52.00(5.50)	54.00(2.20)	57.20(2.09)	54.80(4.13)	53.60(2.09)
	Precision	54.73(3.34)	57.46(1.42)	58.64(0.19)	63.74(10.81)	53.89(1.57)	60.02(3.41)	55.20(7.65)	54.76(4.59)
	Recall	65.60(5.20)	47.20(9.57)	39.20(14.80)	61.60(9.93)	65.60(14.80)	55.19(1.16)	67.19(14.77)	48.80(5.05)
	F1-Score	56.00(8.91)	46.28(22.69)	45.59(17.72)	44.57(17.35)	56.99(14.03)	53.65(14.34)	58.61(4.88)	48.19(12.00)

TABLE X: MIA F1-Score (%) on MoCo encoder under varying privacy budget ϵ. δ is set to 10^{-5}.

Methods	Privacy Budget ϵ					
	5.0	10.0	20.0	30.0	40.0	50.0
ADA-MIA	50.20	50.20	51.30	53.04	60.16	65.87
ADA-MIA$_{sup}$	51.29	50.79	50.79	54.76	62.77	70.47
EncoderMI	51.19	50.74	51.25	52.28	56.33	63.98

TABLE XI: Performance (%) of ADA-MIA against target encoders using contrastive adversarial training.

Methods	Metric	MoCo CIFAR-10	SimCLR CIFAR-100
ADA-MIA	Accuracy	64.20(9.40)	64.60(7.00)
	Precision	64.37(7.28)	65.47(6.73)
	Recall	63.60(14.80)	61.80(8.60)
	F1-Score	63.98(10.80)	63.58(7.65)
ADA-MIA$_{sup}$	Accuracy	**67.90**(17.66)	**67.70**(15.99)
	Precision	**68.08**(11.68)	**68.48**(24.19)
	Recall	67.40(22.41)	**65.60**(9.62)
	F1-Score	**67.74**(15.71)	**67.01**(15.30)
EncoderMI	Accuracy	61.30(20.89)	60.50(19.30)
	Precision	62.26(20.54)	59.89(18.52)
	Recall	57.40(26.99)	63.60(19.20)
	F1-Score	59.73(22.86)	61.69(18.76)
PartCrop	Accuracy	61.70(19.37)	60.80(19.93)
	Precision	59.45(22.15)	60.27(19.33)
	Recall	**73.60**(6.80)	63.40(22.00)
	F1-Score	65.77(15.21)	61.79(19.59)

and CIFAR-100, representing only a marginal 3% decrease compared to the performance of the undefended encoders. Nonetheless, our methods still experience an accuracy degradation of 5% when attacking against the target encoders trained on STL-10 and Tiny-ImageNet datasets. This outcome can be primarily attributed to the presence of the perturbations introduced by DP-SGD during optimization, making it challenging for the encoders to learn the patterns hidden behind the training data. Moreover, we also conduct evaluation on MoCo encoder trained on CIFAR-10 with varying privacy budgets ϵ. Table X demonstrate that all MIAs fail to determine membership status precisely as privacy budgets get stricter (i.e., $\epsilon < 30.0$).

C. *Pre-training Using Adversarial Training*

Adversarial training (AT) techniques have been increasingly adopted to enhance the privacy robustness of CL encoders [60], [80], [81]. To further evaluate the resilience of ADA-MIA against adversarially trained CL encoders, we select two open-source encoders from the Model Zoo in AutoLoRa [82]. One of them is pretrained on CIFAR-10 using MoCo with Adversarial Contrastive Learning (ACL) [80], and the other trained on CIFAR-100 using SimCLR with DYNACL [60]. We treat them as publicly released black-box encoders, and we can only obtain the features corresponding to input images. Empirical results are outlined in Table XI. The results demonstrate that compared to EncoderMI, ADA-MIA and ADA-MIA$_{sup}$ preserve attack performance better. Specially, the average F1-score of ADA-MIA only decreases by nearly 10%, denoting the effectiveness of aggressive data augmentation under extreme scenarios.

VII. Conclusion

In this paper, we have presented ADA-MIA, a novel black-box MIA for CL pre-trained encoders. We demonstrate that the aggressive data augmentations that have not been used in the pre-training process could lead to severe membership leakage in CL encoders. ADA-MIA extracts embedding similarities between the target image and its aggressively augmented versions as membership features, leveraging only the embedding interface without requiring prior knowledge of the pre-training settings or the pre-training data distribution of the target encoders. Experiments on different datasets with real-world CL encoders demonstrate that ADA-MIA achieves comparable performance, even with less prior knowledge, compared to existing MIAs, while showing minimal degradation in inference performance under two classic MIA defenses. We envision our work as a solid step towards revealing the privacy leakage risks of the pre-training data in CL and shed light on designing more effective defenses against MIAs in CL.

References

[1] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of IEEE/CVF CVPR*, 2019, pp. 9726–9735.

[2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of ICML*, 2020, pp. 1597–1607.

[3] M. C. Schiappa, Y. S. Rawat, and M. Shah, "Self-supervised learning for videos: A survey," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–37, 2023.

[4] Z. Zhou, S. Hu, R. Zhao, Q. Wang, L. Y. Zhang, J. Hou, and H. Jin, "Downstream-agnostic adversarial examples," in *Proceedings of the IEEE/CVF ICCV*, 2023, pp. 4345–4355.

[5] Y. Wang, Y. Zhu, and X.-S. Gao, "Efficient availability attacks against supervised and contrastive learning simultaneously," in *Proceedings of NeurIPS*, 2024.

[6] Z. Wang, J. Yu, M. Gao, H. Yin, B. Cui, and S. Sadiq, "Unveiling vulnerabilities of contrastive recommender systems to poisoning attacks," in *Proceedings of ACM SIGKDD*, 2024, pp. 3311–3322.

[7] J. Zhang, H. Liu, J. Jia, and N. Z. Gong, "Data poisoning based backdoor attacks to contrastive learning," in *Proceedings of IEEE/CVF CVPR*, 2024, pp. 24 357–24 366.

[8] S. Liang, M. Zhu, A. Liu, B. Wu, X. Cao, and E.-C. Chang, "Badclip: Dual-embedding guided backdoor attack on multimodal contrastive learning," in *Proceedings of IEEE/CVF CVPR*, 2024, pp. 24 645–24 654.

[9] G. Tao, Z. Wang, S. Feng, G. Shen, S. Ma, and X. Zhang, "Distribution preserving backdoor attack in self-supervised learning," in *Proceedings of IEEE S&P*, 2024, pp. 2029–2047.

[10] X. He and Y. Zhang, "Quantifying and mitigating privacy risks of contrastive learning," in *Proceedings of ACM CCS*, 2021, pp. 845–863.

[11] H. Liu, J. Jia, W. Qu, and N. Z. Gong, "EncoderMI: Membership inference against pre-trained encoders in contrastive learning," in *Proceedings of ACM CCS*, 2021, pp. 2081–2095.

[12] J. Zhu, J. Zha, D. Li, and L. Wang, "A unified membership inference method for visual self-supervised encoder via part-aware capability," in *Proceedings of ACM CCS*, 2024.

[13] X. Wang and G.-J. Qi, "Contrastive learning with stronger augmentations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5549–5560, 2022.

[14] J. Wu, J. Chen, J. Wu, W. Shi, X. Wang, and X. He, "Understanding contrastive learning via distributionally robust optimization," in *Proceedings of NeurIPS*, 2024.

[15] W. Huang, A. Han, Y. Chen, Y. Cao, Z. Xu, and T. Suzuki, "On the comparison between multi-modal and single-modal contrastive learning," in *Proceedings of NeurIPS*, 2024, pp. 81 549–81 605.

[16] L. Wu, J. Zhuang, and H. Chen, "Voco: A simple-yet-effective volume contrastive learning framework for 3d medical image analysis," in *Proceedings of IEEE/CVF CVPR*, 2024, pp. 22 873–22 882.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE/CVF CVPR*, 2015, pp. 770–778.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of ICLR*, 2020.

[19] J. Kossen, M. Collier, B. Mustafa, X. Wang, X. Zhai, L. Beyer, A. Steiner, J. Berent, R. Jenatton, and E. Kokiopoulou, "Three towers: Flexible contrastive learning with pretrained image models," in *Proceedings of NeuIPS*, 2024.

[20] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of AISTATS*, 2010, pp. 297–304.

[21] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR, arXiv:1087.03748*, 2018.

[22] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Proceedings of NeurIPS*, vol. 29, 2016.

[23] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proceedings of IEEE S&P*, 2017, pp. 3–18.

[24] 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*. ACM, 2019, p. 241257.

[25] L. Song and P. Mittal, "Systematic evaluation of privacy risks of machine learning models," in *Proceedings of USENIX Security*, 2021.

[26] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," in *Proceedings of ICML*, 2021, pp. 1964–1974.

[27] H. Yan, S. Li, Y. Wang, Y. Zhang, K. Sharif, H. Hu, and Y. Li, "Membership inference attacks against deep learning models via logits distribution," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[28] X. Yuan and L. Zhang, "Membership inference attacks and defenses in neural network pruning," in *Proceedings of USENIX Security*, 2022.

[29] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," in *Proceedings of IEEE S&P*, 2022, pp. 1897–1914.

[30] Y. Liu, Z. Zhao, M. Backes, and Y. Zhang, "Membership inference attacks by exploiting loss trajectory," in *Proceedings of ACM CCS*, 2022, pp. 2085–2098.

[31] H. Liu, Y. Wu, Z. Yu, and N. Zhang, "Please tell me more: Privacy impact of explainability through the lens of membership inference attack," in *Proceedings of IEEE S&P*, 2024, pp. 4791–4809.

[32] H. Li, Z. Li, S. Wu, Y. Ye, M. Zhang, D. Feng, and Y. Zhang, "Enhanced label-only membership inference attacks with fewer queries," in *Proceedings of USENIX Security*, 2025, pp. 5465–5483.

[33] M. Lassila, J. Östman, K.-H. Ngo, and A. Graell i Amat, "Practical bayes-optimal membership inference attacks," in *Proceedings of NeurIPS*, vol. 38, 2026, pp. 34 278–34 312.

[34] 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," in *Proceedings of NDSS*, 2019.

[35] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *Proceedings of IEEE CSF*, 2018, pp. 268–282.

[36] B. Wu, C. Chen, S. Zhao, C. Chen, Y. Yao, G. Sun, L. Wang, X. Zhang, and J. Zhou, "Characterizing membership privacy in stochastic gradient langevin dynamics," in *Proceedings of AAAI*, 2020, pp. 6372–6379.

[37] S. Zarifzadeh, P. Liu, and R. Shokri, "Low-cost high-power membership inference attacks," in *Proceedings of ICML*, 2024.

[38] M. Bertran, S. Tang, A. Roth, M. Kearns, J. H. Morgenstern, and S. Z. Wu, "Scalable membership inference attacks via quantile regression," in *Proceedings of NeurIPS*, vol. 36, 2023, pp. 314–330.

[39] Z. Li and Y. Zhang, "Membership leakage in label-only exposures," in *Proceedings of ACM CCS*, 2021, pp. 880–895.

[40] G. Liu, Z. Tian, J. Chen, C. Wang, and J. Liu, "Tear: Exploring temporal evolution of adversarial robustness for membership inference attacks against federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 4996–5010, 2023.

[41] Y. Peng, J. Roh, S. Maji, and A. Houmansadr, "Oslo: One-shot label-only membership inference attacks," in *Proceedings of NeurIPS*, 2024.

[42] B. Hui, Y. Yang, H. Yuan, P. Burlina, N. Z. Gong, and Y. Cao, "Practical blind membership inference attack via differential comparisons," in *Proceedings of NDSS*, 2021.

[43] M. Zhang, N. Yu, R. Wen, M. Backes, and Y. Zhang, "Generated distributions are all you need for membership inference attacks against generative models," in *Proceedings of IEEE/CVF WACV*, 2024, pp. 4839–4849.

[44] Y. Chen, S. Pang, and R. Zhang, "Anti-asyndgan: Black-box membership inference attacks against medical distributed generation models," in *Proceedings of IEEE IJCNN*, 2024, pp. 1–10.

[45] L. Wang, J. Wang, J. Wan, L. Long, Z. Yang, and Z. Qin, "Property existence inference against generative models," in *Proceedings of USENIX Security*, 2024, pp. 2423–2440.

[46] L. Hu, H. Yan, Y. Peng, H. Hu, S. Wang, and J. Li, "Vae-based membership cleanser against membership inference attacks," *IEEE Transactions on Dependable and Secure Computing*, 2024.

[47] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramer, B. Balle, D. Ippolito, and E. Wallace, "Extracting training data from diffusion models," in *Proceedings of USENIX Security*, 2023, pp. 5253–5270.

[48] J. Duan, F. Kong, S. Wang, X. Shi, and K. Xu, "Are diffusion models vulnerable to membership inference attacks?" in *Proceedings of ICML*, vol. 202, 2023, pp. 8717–8730.

[49] S. Zhai, H. Chen, Y. Dong, J. Li, Q. Shen, Y. Gao, H. Su, and Y. Liu, "Membership inference on text-to-image diffusion models via conditional likelihood discrepancy," in *Proceedings of NeurIPS*, vol. 37, 2024, pp. 74 122–74 146.

[50] X. Fu, X. Wang, Q. Li, J. Liu, J. Dai, J. Han, and X. Gao, "Unlocking generative priors: A new membership inference framework for diffusion

models," *IEEE Transactions on Information Forensics and Security*, 2025.

[51] Y. Peng, A. Naseh, and A. Houmansadr, "Diffence: Fencing membership privacy with diffusion models," in *Proceedings of NDSS*, 2025.

[52] Y. Chen, K. Zhang, Y. Du, E. Stoppa, C. Fleming, A. Kundu, B. Ribeiro, and N. Li, "Membership inference attacks against fine-tuned diffusion language models," in *Proceedings of ICLR*, 2026.

[53] Y. He, B. Li, Y. Wang, M. Yang, J. Wang, H. Hu, and X. Zhao, "Is difficulty calibration all we need? towards more practical membership inference attacks," in *Proceedings of the ACM CCS*, 2024, pp. 1226–1240.

[54] W. Fu, H. Wang, C. Gao, G. Liu, Y. Li, and T. Jiang, "Membership inference attacks against fine-tuned large language models via self-prompt calibration," in *Proceedings of NeurIPS*, 2024.

[55] J. Zhang, J. Sun, E. Yeats, Y. Ouyang, M. Kuo, J. Zhang, H. F. Yang, and H. Li, "Min-k%++: Improved baseline for pre-training data detection from large language models," in *Proceedings of ICLR*, 2025.

[56] R. Kuditipudi, J. Huang, S. Zhu, D. Yang, C. Potts, and P. Liang, "Blackbox model provenance via palimpsestic membership inference," in *Proceedings of NeurIPS*, vol. 38, 2026, pp. 162 337–162 367.

[57] Y. Hu, Z. Li, Z. Liu, Y. Zhang, Z. Qin, K. Ren, and C. Chen, "Membership inference attacks against vision-language models," in *Proceedings of USENIX Security*, 2025.

[58] Y. Liu, X. Lyu, D. Wang, Y. Li, and B. Xiao, "Lomia: Label-only membership inference attacks against pre-trained large vision-language models," in *Proceedings of NeurIPS*, vol. 38, 2026, pp. 153 714–153 737.

[59] Q. Zhang, D. Yuan, B. Zhang, B. Yuan, and B. Du, "Membership inference attacks against vision transformers: Mosaic mixup training to the defense," in *Proceedings of ACM CCS*, 2024, pp. 1256–1270.

[60] R. Luo, Y. Wang, and Y. Wang, "Rethinking the effect of data augmentation in adversarial contrastive learning," in *Proceedings of ICLR*, 2023.

[61] J. Zhang and K. Ma, "Rethinking the augmentation module in contrastive learning: Learning hierarchical augmentation invariance with expanded views," in *Proceedings of IEEE/CVF CVPR*, 2022, pp. 16 650–16 659.

[62] W. Huang, M. Yi, X. Zhao, and Z. Jiang, "Towards the generalization of contrastive self-supervised learning," in *Proceedings of ICLR*, 2023.

[63] K. Krishna and M. N. Murty, "Genetic k-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 29, no. 3, pp. 433–439, 1999.

[64] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Proceedings of ACM KDD*, no. 34, 1996, pp. 226–231.

[65] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of IEEE/CVF ECCV*, 2018, pp. 132–149.

[66] 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 *Proceedings of IEEE S&P*, 2019.

[67] X. Tang, S. Mahloujifar, L. Song, V. Shejwalkar, M. Nasr, A. Houmansadr, and P. Mittal, "Mitigating membership inference attacks by self-distillation through a novel ensemble architecture," in *Proceedings of USENIX Security*, 2021.

[68] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[69] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of AISTATS*, 2011, pp. 215–223.

[70] M. A. mnmoustafa, "Tiny imagenet," 2017. [Online]. Available: https://kaggle.com/competitions/tiny-imagenet

[71] G. Li, S. Rezaei, and X. Liu, "User-level membership inference attack against metric embedding learning," in *Proceedings of ICLR Workshop on PAIR2̂Struct*, 2022.

[72] C. Song and A. Raghunathan, "Information leakage in embedding models," in *Proceedings of ACM CCS*, 2020, pp. 377–390.

[73] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of ACM KDD*, 2016, p. 785794.

[74] Q. Qian, "Stable cluster discrimination for deep clustering," in *Proceedings of IEEE/CVF ICCV*, 2023, pp. 16 645–16 654.

[75] S. Yuan, L. Feng, and T. Liu, "Early stopping against label noise without validation data," in *Proceedings of ICLR*, 2024.

[76] Z. Kazan and J. Reiter, "Prior-itizing privacy: A bayesian approach to setting the privacy budget in differential privacy," in *Proceedings of NeurIPS*, 2024, pp. 90 384–90 430.

[77] L. Demelius, R. Kern, and A. Trügler, "Recent advances of differential privacy in centralized deep learning: A systematic survey," *ACM Computing Surveys*, vol. 57, no. 6, pp. 1–28, 2025.

[78] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of ACM CCS*, 2016, pp. 308–318.

[79] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao *et al.*, "Opacus: User-friendly differential privacy library in pytorch," in *NeurIPS 2021 Workshop Privacy in Machine Learning*.

[80] M. Kim, J. Tack, and S. J. Hwang, "Adversarial self-supervised contrastive learning," in *Proceedings of NeurIPS*, 2020, pp. 2983–2994.

[81] R. Zhang, S. Tang, and J. Cao, "Self-supervised adversarial training via diverse augmented queries and self-supervised double perturbation," in *Proceedings of NeurIPS*, 2024, pp. 43 788–43 808.

[82] X. Xu, J. Zhang, and M. Kankanhalli, "Autolora: An automated robust fine-tuning framework," in *Proceedings of ICLR*, 2024.

Zixiong Wang received the B.E. degree from Central China Normal University, China, in 2022. He is currently pursuing the Ph.D. degree in Electronics and Information Engineering at Huazhong University of Science and Technology, China. His research interests focus on data privacy and security issues of Large Language Models.

Lishuai Hou received the B.E. degree from Dalian University of Technology, China, in 2022. He is currently pursuing the Ph.D. degree in Information and Communication Engineering at Huazhong University of Science and Technology, China. His research interests focus on machine unlearning and AI security.

Gaoyang Liu (S'20-M'21) received the B.S. and Ph.D. degrees from Huazhong University of Science and Technology, China, in 2015 and 2021, respectively. From 2021 to 2024, he was a post-doctoral researcher at the School of Computing Science, Simon Fraser University, British Columbia, Canada. Thereafter, he joined the faculty of Huazhong University of Science and Technology where he is currently a lecturer. His research interests include machine learning, mobile sensing and data privacy protection. He is a member of IEEE.

Jianfeng Lu (M'10) received the Ph.D. degree in computer application technology from the Huazhong University of Science and Technology, in 2010. He was with Zhejiang Normal University, Jinhua, China, from 2010 to 2021. He was a Visiting Researcher with the University of Pittsburgh, Pittsburgh, USA, in 2013. He is currently a Professor with the School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China. His research interests include federated learning, crowd computing and game theory.

Desheng Wang (M'12) received the Ph.D. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, in 2004. He is currently a full professor with the Huazhong University of Science and Technology. His current research interests lie in the field of cooperative communication, radio resource allocation, and wireless radio link protocol design. He is a member of the China Broadband Wireless IP Standard Group and FuTURE, and a member of IEEE.

Ahmed M. Abdelmoniem (M'17-SM'25) received his Ph.D. in Computer Science and Engineering from Hong Kong University of Science and Technology, Hong Kong in 2017. He is an Associate Professor at the Queen Mary University of London, UK and leads the Scalable Adaptive Yet Efficient Distributed (SAYED) Systems Research Group. Formerly, he was a Research Scientist at KAUST, Saudi Arabia, and a Senior Researcher with Huawei's Future Networks Lab in Hong Kong. He is an investigator on several UK and international grants totaling nearly USD 1.5mil in funding. His research interests lie in the intersection of distributed systems, machine learning, and computer networks. He is a member of ACM, IEEE and USENIX.

Chen Wang (S'10-M'13-SM'19) received the B.S. and Ph.D. degrees from the Department of Automation, Wuhan University, China, in 2008 and 2013, respectively. From 2013 to 2017, he was a postdoctoral research fellow in the Networked and Communication Systems Research Lab, Huazhong University of Science and Technology, China. Thereafter, he joined the faculty of Huazhong University of Science and Technology where he is currently an associate professor. His research interests are in the broad areas of wireless networking, Internet of Things, and mobile computing, with a recent focus on privacy issues in intelligent systems. He is a senior member of IEEE and ACM.