Saarland University
Department of Computer Science

On the Privacy Risks of Machine Learning Models

Dissertation
zur Erlangung des Grades
des Doktors der Ingenieurwissenschaften
der Fakultät für Mathematik und Informatik
der Universität des Saarlandes

von
Zheng Li

Saarbrücken, 2023

Tag des Kolloquiums:	05 Oktober 2023
Dekan:	Prof. Dr. Jürgen Steimle

Prüfungsausschuss:

Vorsitzender:	Pro. Dr. Ingmar Weber
Berichterstattende:	Dr. Yang Zhang
	Prof. Dr. Mario Fritz
	Prof. Dr. Konrad Rieck
Akademischer Mitarbeiter:	Dr. Zhiqiu Jiang

Zusammenfassung

Das maschinelle Lernen (ML) hat in den letzten zehn Jahren enorme Fortschritte gemacht und wurde für eine breite Palette wichtiger Anwendungen eingesetzt. Durch den zunehmenden Einsatz von Modellen des maschinellen Lernens ist die Bedeutung von Datenschutzrisiken jedoch wichtiger denn je geworden. Diese Risiken können je nach der Rolle, die ML-Modelle spielen, in zwei Kategorien eingeteilt werden: in eine, in der die Modelle selbst anfällig für das Durchsickern sensibler Informationen sind, und in die andere, in der die Modelle zur Verletzung der Privatsphäre missbraucht werden.

In dieser Dissertation untersuchen wir die Datenschutzrisiken von Modellen des maschinellen Lernens aus zwei Blickwinkeln, nämlich der Anfälligkeit von ML-Modellen und dem Missbrauch von ML-Modellen. Um die Anfälligkeit von ML-Modellen für Datenschutzrisiken zu untersuchen, führen wir zwei Studien zu einem der schwerwiegendsten Angriffe auf den Datenschutz von ML-Modellen durch, nämlich dem Angriff auf die Mitgliedschaft (membership inference attack, MIA). Erstens erforschen wir das Durchsickern von Mitgliedschaften in ML-Modellen, die sich nur auf Labels beziehen. Wir präsentieren den ersten "label-only membership inference"-Angriff und stellen fest, dass das "membership leakage" schwerwiegender ist als bisher gezeigt. Zweitens führen wir die erste Analyse der Privatsphäre von Netzwerken mit mehreren Ausgängen durch die Linse des Mitgliedschaftsverlustes durch. Wir nutzen bestehende Angriffsmethoden, um die Anfälligkeit von Multi-Exit-Netzwerken für Membership-Inference-Angriffe zu quantifizieren und schlagen einen hybriden Angriff vor, der die Exit-Informationen ausnutzt, um die Angriffsleistung zu verbessern. Unter dem Gesichtspunkt des Missbrauchs von ML-Modellen zur Verletzung der Privatsphäre konzentrieren wir uns auf die Manipulation von Gesichtern, die visuelle Fehlinformationen erzeugen können. Wir schlagen das erste Abwehrsystem `UnGANable` gegen GAN-basierte Gesichtsmanipulationen vor, indem wir den Prozess der GAN-Inversion gefährden, der ein wesentlicher Schritt für die anschließende Gesichtsmanipulation ist.

Alle Ergebnisse tragen dazu bei, dass die Community einen Einblick in die Datenschutzrisiken von maschinellen Lernmodellen erhält. Wir appellieren an die Gemeinschaft, eine eingehende Untersuchung der Risiken für die Privatsphäre, wie die unsere, im Hinblick auf die sich schnell entwickelnden Techniken des maschinellen Lernens in Betracht zu ziehen.

Abstract

Machine learning (ML) has made huge progress in the last decade and has been applied to a wide range of critical applications. However, driven by the increasing adoption of machine learning models, the significance of privacy risks has become more crucial than ever. These risks can be classified into two categories depending on the role played by ML models: one in which the models themselves are vulnerable to leaking sensitive information, and the other in which the models are abused to violate privacy.

In this dissertation, we investigate the privacy risks of machine learning models from two perspectives, i.e., the vulnerability of ML models and the abuse of ML models. To study the vulnerability of ML models to privacy risks, we conduct two studies on one of the most severe privacy attacks against ML models, namely the membership inference attack (MIA). Firstly, we explore membership leakage in label-only exposure of ML models. We present the first label-only membership inference attack and reveal that membership leakage is more severe than previously shown. Secondly, we perform the first privacy analysis of multi-exit networks through the lens of membership leakage. We leverage existing attack methodologies to quantify the vulnerability of multi-exit networks to membership inference attacks and propose a hybrid attack that exploits the exit information to improve the attack performance. From the perspective of abusing ML models to violate privacy, we focus on deepfake face manipulation that can create visual misinformation. We propose the first defense system `UnGANable` against GAN-based face manipulation by jeopardizing the process of GAN inversion, which is an essential step for subsequent face manipulation.

All findings contribute to the community's insight into the privacy risks of machine learning models. We appeal to the community's consideration of the in-depth investigation of privacy risks, like ours, against the rapidly-evolving machine learning techniques.

Background of this Dissertation

This dissertation is based on the following papers. I am the primary contributor to each paper.

The proposal to explore membership leakage in the label-only scenario and the idea of label-only membership inference attacks [P1] were contributions of Zheng Li. The design, implementation, and evaluation were carried out by Zheng Li. Zheng Li and Yang Zhang participated in writing and reviewing the paper.

The initial idea of auditing membership leakage of multi-exit networks [P2] originated from Yang Zhang. Zheng Li and Yang Zhang then jointly discussed and determined the three research perspectives presented in the paper, i.e., measurement, attack, and defense. For the three research perspectives, Zheng Li designed them and was responsible for their implementation and evaluation. Yiyong Liu, Xinlei He, and Ning Yu provided valuable feedback on the three research perspectives. All authors participated in the writing and reviewing of the paper.

The general idea of defending against face manipulation [P3] was proposed by Yang Zhang and later refined to defending GAN inversion, an essential step for face manipulation, by Zheng Li, Ning Yu, and Ahmed Salem. Zheng Li designed defense methodologies with the support of Ning Yu, Ahmed Salem, Michael Backes and Mario Fritz. The implementation and evaluation were done by Zheng Li. All authors participated in the writing and reviewing of the paper.

[P1] Li, Z. and Zhang, Y. Membership Leakage in Label-Only Exposures. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2021, 880–895.

[P2] Li, Z., Liu, Y., He, X., Yu, N., Backes, M., and Zhang, Y. Auditing Membership Leakages of Multi-Exit Networks. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2022, 1917–1931.

[P3] Li, Z., Yu, N., Salem, A., Backes, M., Fritz, M., and Zhang, Y. UnGANable: Defending Against GAN-based Face Manipulation. In: *USENIX Security Symposium (USENIX Security)*. USENIX, 2023.

Further Contributions of the Author

The author was also able to contribute to the following: [S1, T1, T2, T3, T4, T5]

Published Papers:

[S1] Liu, Y., Li, Z., Backes, M., Shen, Y., and Zhang, Y. Backdoor Attacks Against Dataset Distillation. In: *Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2023.

Technical Reports:

[T1] He, X., Li, Z., Xu, W., Cornelius, C., and Zhang, Y. Membership-Doctor: Comprehensive Assessment of Membership Inference Against Machine Learning Models. *CoRR abs/2208.10445* (2022).

[T2] Sha, Z., Li, Z., Yu, N., and Zhang, Y. DE-FAKE: Detection and Attribution of Fake Images Generated by Text-to-Image Generation Models. *CoRR abs/2210.06998* (2022).

[T3] Shen, X., He, X., Li, Z., Shen, Y., Backes, M., and Zhang, Y. Backdoor Attacks in the Supply Chain of Masked Image Modeling. *CoRR abs/2210.01632* (2022).

[T4] Wu, Y., Yu, N., Li, Z., Backes, M., and Zhang, Y. Membership Inference Attacks Against Text-to-image Generation Models. *CoRR abs/2210.00968* (2022).

[T5] Yang, Z., He, X., Li, Z., Backes, M., Humbert, M., Berrang, P., and Zhang, Y. Data Poisoning Attacks Against Multimodal Encoders. *CoRR abs/2209.15266* (2022).

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Yang Zhang, for his unwavering support and guidance throughout my doctoral studies. His encouragement, expertise, and commitment to excellence have been instrumental in helping me achieve this milestone.

I would also like to thank the members of my dissertation committee, Dr. Yang Zhang, Prof. Dr. Mario Fritz, and Prof. Dr. Konrad Rieck, for their invaluable feedback and insights. Their expertise and dedication to their respective fields have greatly enriched my work.

I extend my heartfelt gratitude to my colleagues and friends who have been a constant source of support, motivation, and inspiration throughout my doctoral journey. Their encouragement and positive energy have been instrumental in helping me overcome the challenges that I faced. In particular, I am grateful for the stimulating discussions, ideas, and feedback that we shared together. Their knowledge and expertise have enriched my work and broadened my perspectives.

Finally, I would like to acknowledge my family for their unconditional love, support, and patience throughout this journey. Their sacrifices and encouragement have been the driving force behind my success. I dedicate this thesis to them.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

Machine learning (ML) has made tremendous progress in the past decade, leading to significant advancements in many real-world applications, such as medical image analysis [74, 131, 25], automatic driving [150, 22], and artificial intelligence generated content (AIGC) [45, 71, 103, 105, 109]. In addition, AI-based systems are also ubiquitous in smart devices used by individuals in daily life, e.g., smartphones [87] and IoT devices [90], which rely on machine learning techniques to offer personalized recommendations and enhance user experiences.

Accordingly, the rapid development and widespread adoption of machine learning models have led to two significant trends: the demand for large-scale data and higher quality of generated multimedia content. The demand for large-scale data in machine learning cannot be overstated, as it allows machine learning to learn patterns and make predictions with greater accuracy and reliability. In addition, with the ability to generate higher-quality multimedia content, AIGC opens the door to a range of exciting applications in areas as diverse as creative arts, advertising, filmmaking, and video games.

However, as these trends continue to grow, the significance of privacy risks has become more crucial than ever. Data privacy refers to personal and confidential information, such as medical records, personal identities, facial attributes, and other personally identifiable information. Recent research has shown that privacy in machine learning can be violated through various attacks, which can compromise sensitive information and undermine the trustworthiness of machine learning systems. Concretely, the demand for large-scale data has led to large amounts of sensitive data (e.g., medical records and biometric data) being collected and used to train ML models. ML models trained on these sensitive data have proven to be vulnerable to leaking sensitive information about the data. Some examples of such privacy risks are membership inference attacks [116, 124, 128, P1, 34], model inversion [42, 148], and training data reconstruction [115]. In addition, AIGC, which enables the creation of higher-quality multimedia content, is vulnerable to being used as a tool for privacy violations, i.e., the adversary abuses ML models for malicious purposes. The most representative privacy risk is visual misinformation through deepfake technology based on machine learning and especially, generative models such as Generative Adversarial Networks (GANs). For instance, malicious editing of face images based on GAN-based face manipulation [141, 160, 122, 63, 48] can create false impressions, deceive people, or even trick biometric systems.

1.1 Our Contributions

In this dissertation, we evaluate the privacy risks of ML models. Abstractly, the privacy risks of ML models can be divided into two categories depending on the role played by ML models: one in which the models themselves are vulnerable to leaking sensitive information and the other in which the models are abused to violate privacy.

More concretely, in this dissertation, we comprehensively investigate the privacy risks of the two perspectives, i.e., the vulnerability of ML models and the abuse of ML models. For the vulnerability of ML models, we explore one of the most severe attacks against ML models, namely the membership inference attack (MIA). Specially, we conduct two works on membership inference attacks. For the abuse of ML models,

we focus on deepfake face manipulation that abuses ML models. We explore how to protect facial images from malicious face manipulation. All the above work stretches across the following peer-reviewed publications [P1, P2, P3].

Label-Only Membership Inference Attacks. In our first work [P1], we focus on the membership inference attack, one of the most representative privacy attacks against ML models to infer sensitive data information. Existing membership inference attacks [124, 85, 116, 146, 129, 60, 78] rely on the confidence scores (e.g., class probabilities or logits) returned by a target ML model as their inputs, i.e., an ML model is more confident facing a data sample it was trained on, and this confidence is reflected in the model's output scores. However, these attacks can be easily mitigated if the model only exposes the predicted label. In this work, we propose label-only membership inference attacks and demonstrate that label-only exposures are also vulnerable to membership leakage. In particular, we develop two types of label-only attacks, namely transfer-based attack and boundary-based attack. Empirical evaluation shows that our label-only membership inference attacks can achieve remarkable performance, and even outperform the previous score-based attacks in some cases. We further present new insights on the success of membership inference based on quantitative and qualitative analysis, i.e., member samples of a model are more distant to the model's decision boundary than non-member samples. Finally, we evaluate multiple defense mechanisms against our decision-based attacks and show that our two types of attacks can bypass most of these defenses.

Auditing Membership Leakage of Multi-Exit Networks. In our second work [P2], we explore the vulnerability of multi-exit networks, which endow a backbone model with early exits, allowing to obtain predictions at intermediate layers of the model and thus save computation time and/or energy. However, current various designs of multi-exit networks are only considered to achieve the best trade-off between resource usage efficiency and prediction accuracy, the privacy risks stemming from them have never been explored. In this work, we perform the first privacy analysis of multi-exit networks through the lens of membership leakages. In particular, we first leverage the existing attack methodologies to quantify the multi-exit networks' vulnerability to membership leakages. Our experimental results show that multi-exit networks are less vulnerable to membership leakages and the exit (number and depth) attached to the backbone model highly correlates with the attack performance. Furthermore, we propose a hybrid attack that exploits the exit information to improve the performance of existing attacks. We evaluate membership leakage threat caused by our hybrid attack under three different adversarial setups, ultimately arriving at a model-free and data-free adversary. These results clearly demonstrate that our hybrid attack is very broadly applicable, thereby the corresponding risks are much more severe than shown by existing membership inference attacks. We further present a defense mechanism called *TimeGuard* specifically for multi-exit networks and show that *TimeGuard* mitigates the newly proposed attacks perfectly.

Defending Against GAN-inversion-based Face Manipulation. In this work [P3], we focus on one of the most representative privacy risks caused by the abuse of ML models, namely deepfake, which poses severe threats of visual misinformation to our society. One popular deepfake application is face manipulation which modifies a victim's

facial attributes in an image (e.g., changing her age or hair color), and the state-of-the-art face manipulation techniques rely on Generative Adversarial Networks (GANs). In this paper, we propose the first defense system, namely `UnGANable`, against GAN-based face manipulation. In specific, `UnGANable` focuses on defending GAN inversion, an essential step for face manipulation. Its core technique is to search for alternative images (called cloaked images) around the original images (called target images) in image space. When posted online, these cloaked images can jeopardize the GAN inversion process. We consider two state-of-the-art inversion techniques including optimization-based inversion and hybrid inversion, and design five different defenses under five scenarios depending on the defender's background knowledge. Extensive experiments on four popular GAN models trained on two benchmark face datasets show that `UnGANable` achieves remarkable effectiveness and utility performance, and outperforms multiple baseline methods. We further investigate four adaptive adversaries to bypass `UnGANable` and show that some of them are slightly effective.

1.2 Organization

The rest of this dissertation is organized as the following. We first present the preliminaries and background in chapter 2. chapter 3 presents our label-only membership inference attacks. Then, we explore the membership leakage risk of multi-exit networks in chapter 4. We next investigate deepfake face manipulation and propose our defenses against face manipulation in chapter 5. Finally, we presents related works in chapter 6, and chapter 7 concludes the dissertation.

2
Preliminaries and Background

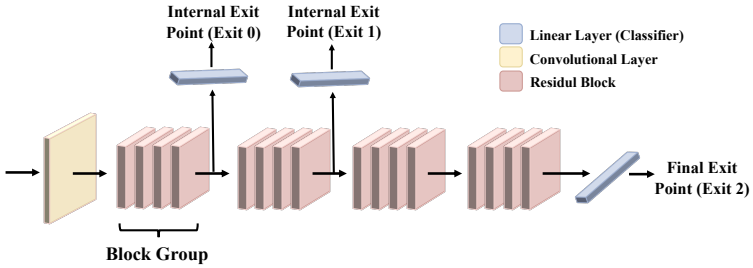


Figure 2.1: An illustration of the multi-exit network with three exits inserted, including two internal and one final exit point.

In this chapter, we present the preliminaries and background related to this dissertation. We start by introducing the machine learning classification task and multi-exit networks. We then introduce the most representative privacy attack against machine learning models. namely membership inference attack. Lastly, we introduce the deepfake based on GAN-base face manipulation

2.1 Machine Learning Preliminaries

2.1.1 Machine Learning Classifiers

The classification task is a fundamental aspect of machine learning that is applied in various domains, such as face recognition [73, 154], medical image analysis [74, 131, 25], and spam filtering [30]. The goal of the classification task is to train a machine learning classifier that can identify the decision boundary between classes and predict the category or class for new data points based on a set of features or attributes. This is typically achieved through supervised learning, where the machine learning classifier is trained on a labeled dataset that associates each data point with a specific class or category label.

2.1.2 Multi-Exit Networks

Relying on the fact that not all inputs require the same amount of computation to yield a confident prediction, the multi-exit network [132, 58, 72] is gaining attention as a prominent approach for pushing the limits of efficient deployment. Multi-exit networks save computation by making input-specific decisions about bypassing the remaining layers once the model becomes confident. More concretely, a multi-exit network applies multiple lightweight classifiers on a vanilla ML model to allow the inference to preemptively finish at one of the exit points when the network is sufficiently confident with a predefined stopping criterion. See Figure 2.1 for an illustration of the design of multi-exit networks.

Backbone Initialisation. As aforementioned, multi-exit networks modify the vanilla ML model by adding multiple lightweight classifiers at certain placements throughout the network. Here, vanilla ML models are also referred to as backbone models. A backbone

model can be any regular machine learning model architecture, such as VGG [125], ResNet [50], and MobileNet [117].

Exit Placement. For simplicity, exit placements are restricted to be at the output of individual network blocks, following an approximately equidistant workload distribution.

Multi-Exit Network Training. Given a training dataset, a multi-exit model is optimized by minimizing the loss function of all training samples and exit points. The training process consists of two steps: the feedforward pass and the backward pass. In the former, a data sample is passed through the model, including both the final exit point and internal exit points, the output from the network at all exit points is recorded, and the loss of the network is then calculated. In backward propagation, the loss is passed back through the network and the model's weights are updated using gradient descent.

Early-Exit Criteria. Given a data sample, it will leave at one of the exit points when the network is sufficiently confident with a predefined stopping criterion. To quantify confidence, we use the estimated probability of the sample belonging to the predicted class. We deem a prediction confident if this probability exceeds the threshold τ. The threshold facilitates on-the-fly adjustment of the early exits based on resource availability and performance requirements. Following most previous works [132, 58, 72, 101, 55, 110], the principle of threshold selection is to guarantee the same or similar classification performance as vanilla models while gaining a lower computational cost.

2.1.3 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) [45] is a type of deep learning algorithm that has attracted tremendous attention recently. GANs consist of a generator and a discriminator, and these two neural networks are trained in tandem to produce real outputs that closely resemble the distribution of the input data. Besides, GANs are often used to model data points, estimate probabilities, and use these probabilities to distinguish between categories. The probability distributions learned by the GAN from the dataset can be used as a guide for creating new data points. Specially, GANs have been successfully applied in various fields, including image and video synthesis [45, 69], natural language processing [147], and image manipulation [141, 37, 160, 122]. Currently, researchers are actively working to improve GANs and further design various promising GANs, such as DCGAN [104], WGAN [46], StyleGANv1 [70], and StyleGANv2 [71]. These GAN models are built with different architectures, losses, and training schemes.

2.2 Membership Leakage of Machine Learning Models

In this section, we introduce one of the most severe privacy risks of ML models, namely membership leakage. The corresponding attack is called membership inference attack (MIA).

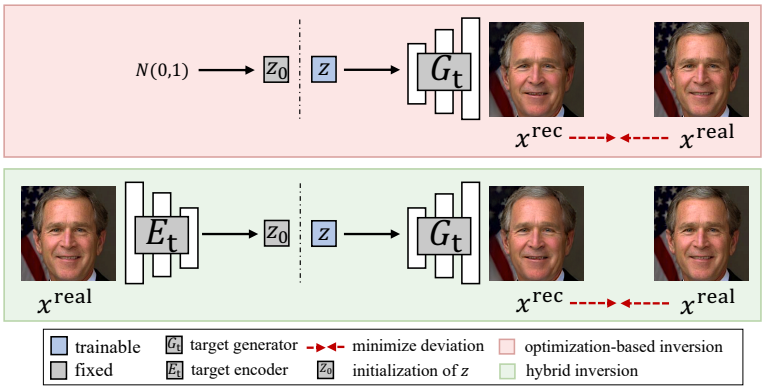


Figure 2.2: Illustration of GAN inversion methods. The upper is the optimization-based inversion. The bottom is the hybrid inversion.

2.2.1 Membership Inference Attack

Membership inference attack occurs when an adversary tries to determine if a particular data sample was used to train a given model. More formally, given a candidate data sample x, a well-trained ML model $\mathcal{M}$, and external knowledge of an adversary, denoted by Ω, the membership inference attack $\mathcal{A}$ can be defined in the following form.

$$\mathcal{A} : x, \mathcal{M}, \Omega \rightarrow \{0, 1\}.$$

Here, 1 represents x as a member of $\mathcal{M}$'s training set, and 0 represents x as not. Besides, the attack model $\mathcal{A}$ is actually a binary classifier. This type of privacy attack is called membership inference attack [124].

Successful membership inference attacks can cause severe privacy consequences, as they may reveal sensitive information such as human identifications. For instance, if the model is trained on sensitive data (e.g., diseases), identifying the person in the training dataset directly reveals this individual's health status.

2.3 GAN-based Face Manipulation

We here present one of the most representative privacy risks caused by abusing ML models, called GAN-based face manipulation. In particular, GAN-based face manipulation consists of two steps, namely GAN inversion and latent code manipulation. In the threat model we considered in chapter 5, we refer to the malicious face manipulator as the adversary, the adversary-controlled generator as the target generator G_{t}, and the adversary-controlled encoder as the target encoder E_{t}.

2.3.1 GAN Inversion

In chapter 5, we consider two representative and most widely-used techniques of GAN inversion, i.e., optimization and hybrid formulations, as shown in Figure 2.2.

Optimization-based Inversion. Existing optimization-based inversions [14, 15] typically reconstruct a target image by optimizing the latent vector

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \mathcal{L}_{\text{rec}}(\mathbf{x}, G_{\text{t}}(\mathbf{z})) \tag{2.1}$$

where $\mathbf{x}$ is the target image and G_{t} is the target generator. Starting from a Gaussian initialization $\mathbf{z}$, we search for an optimized vector $\mathbf{z}^*$ to minimize the reconstruction loss $\mathcal{L}_{\text{rec}}$ which measures the similarity between the given image $\mathbf{x}$ and the image generated from $\mathbf{z}^*$. $\mathcal{L}_{\text{rec}}$ is a weighted combination of the perceptual loss [68] and MSE loss:

$$\mathcal{L}_{\text{rec}} = \mathcal{L}_{\text{percept}}(G_{\text{t}}(\mathbf{z}), \mathbf{x}) + \mathcal{L}_{\text{mse}}(G_{\text{t}}(\mathbf{z}), \mathbf{x})$$

where $\mathcal{L}_{\text{percept}}$ measures the similarity of features extracted from a pre-trained neural network, such as VGG-16 [125], and $\mathcal{L}_{\text{mse}}$ measures the pixel-wise similarity.

Hybrid Inversion. An important issue for optimization-based inversion is initialization. Since Equation 2.1 is highly non-convex, the reconstruction quality strongly relies on a good initialization of $\mathbf{z}$. Consequently, researchers [159, 159, 139, 133] propose to use an encoder to provide better initialization $\mathbf{z}$ for optimization, namely hybrid inversion.

Hybrid inversion first predicts $\mathbf{z}$ of a given image $\mathbf{x}$ by training a separate encoder, then uses the obtained $\mathbf{z}$ as the initialization for optimization. The learned predictive encoder serves as a fast bottom-up initialization for the non-convex optimization problem Equation 2.1.

2.3.2 Latent Code Manipulation

Considering that a given image has been successfully inverted into the latent space, the editing of the image can be easily executed. There are multiple methods [141, 160, 122, 63, 48, 123, 149, 99, 31, 44] to manipulate the latent code, most of them are based on algebraic operations on the latent code. For instance, in InterFaceGAN [122], the authors move the latent code $\mathbf{z}$ along a certain semantic direction n to edit the corresponding attribute of the image $(\mathbf{z} + n)$. As the adversary has full control over the manipulation step, it is extremely difficult to defend this step. Therefore, we only focus on defending against the GAN inversion step - the adversary can only obtain a misleading latent code that is already far from its exact one. In this way, the latent code manipulation step will not achieve its ideal result.

2.4 Datasets Description

We now present the datasets used for the evaluations in this dissertation.

CIFAR-10/CIFAR-100. CIFAR-10 [1] and CIFAR-100 [1] are benchmark datasets used to evaluate image recognition algorithms. CIFAR-10 is composed of 32×32 color images in 10 classes, with 6000 images per class. In total, there are 50000 training images and 10000 test images. CIFAR-100 has the same format as CIFAR-10, but it has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class.

GTSRB. The GTSRB [2] dataset comprises 43 traffic signs in RGB-encoded color images of varying sizes, ranging from 15×15 to 250×250 pixels (not necessarily square). It contains 51,839 images, with 39,209 for training and 12,630 for testing. To ensure consistency, all images are resized to 64×64 before classification.

Face. The Face [3] dataset includes more than 13,000 face images obtained from the Internet and contributed by 1,680 individuals with at least two images each. In our evaluation, we focus on 19 classes containing individuals with more than 40 images. This dataset is particularly difficult for the facial recognition task because the images were not captured in a controlled laboratory environment. In addition, the distribution of data between classes is unbalanced.

TinyImageNet. TinyImageNet [4] is a benchmark dataset utilized for assessing image recognition algorithms, comprising of 100,000 64x64 colored images categorized into 200 classes with 500 images per class. The dataset also includes 500 training, 50 validation, and 50 test images for each class.

Purchases. The dataset for the "acquire valued shoppe" challenge on Kaggle contains 197,000 customer records with 600 binary features indicating their purchase history. The records are categorized into 100 clusters, each representing a distinct purchasing style, and the goal is to predict which purchase style a customer belongs to. This dataset is also widely used to evaluate membership inference attacks in [124, 60, 64, 78, 93, 106, 34].

Locations. The dataset used in this study is a pre-processed version of the Foursquare dataset[1], containing 5,010 data samples with 446 binary features. Each feature represents whether a user visited a particular region or location type. The objective is to classify the users into one of 30 geosocial types based on their record. This dataset is used to evaluate membership inference attacks in [66, 124, 34].

Texas. The dataset comprises 67,330 instances with 6,170 binary features derived from the Discharge Data public use files of the Texas Department of State Health Services.[2] The features related to external causes of injury (e.g., drug misuse, suicide), diagnosis, procedures performed on the patient, and generic information such as age, gender, and race. In line with [58], our study focuses on the top 100 procedures (i.e., 100 classes) by frequency, and the task is to predict a patient's procedure using their data. This dataset is used to evaluate membership inference attacks in [60, 78, 66, 34, 124, 121].

CelebA. CelebA [83] is a collection of 200,000 images of celebrities' faces, with each image having 40 attributes annotated.

FFHQ. The Flickr-Faces-HQ (FFHQ) dataset [70, 71], sourced from Flickr, comprises 70,000 high-resolution human face images with a resolution of 1024 × 1024 pixels. The dataset includes significant variations in age, ethnicity, and image background quality. It is considered a high-quality image dataset for human faces.

[1] https://sites.google.com/site/yangdingqi/home/foursquare-dataset
[2] https://www.dshs.texas.gov/THCIC/Hospitals/Download.shtm

3

Membership Leakage in Label-Only Exposure

3.1 Introduction

Machine learning (ML) has witnessed tremendous progress over the past decade and such developments rely on not only novel training algorithms and architectures, but also access to large-scale data, which typically consists of sensitive and private data, such as health data. Various recent research [114, 116, 124, 146, 135, 54, 85, 129, 60, 86, 78, 153] has shown that ML models are vulnerable to leaking sensitive information about the data. One major privacy risk of ML models is membership inference: An adversary aims to determine whether or not a data sample is used to train a target ML model. Membership inference attacks can reveal sensitive information about an individual. For example, if an ML model is trained on data collected from individuals with a specific disease, an adversary who knows that a victim's data was part of the training data can infer the victim's health status. Membership inference attacks have been demonstrated in various domains, including biomedical data [19] and mobility data [102].

Existing membership inference attacks [124, 85, 116, 146, 129, 60, 78] rely on the confidence scores (e.g. class probabilities or logits) returned by a target ML model as their inputs. The success of membership inference is due to the inherent overfitting property of ML models, i.e., an ML model is more confident facing a data sample it was trained on, and this confidence is reflected in the model's output scores. See Figure 3.1 for an illustration of accessible components of an ML model for such score-based threat model. A major drawback for these score-based attacks is that they can be trivially mitigated if the model only exposes the predicted label, i.e., the final model decision, instead of confidence scores. The fact that score-based attacks can be easily averted makes it more difficult to evaluate whether a model is truly vulnerable to membership inference or not, which may lead to premature claims of privacy for ML models.

3.1.1 Contributions

This motivates us to focus on a new category of membership inference attacks that have so far received fairly little attention, namely *label-only attacks*. Here, the adversary solely relies on the final prediction of the target model, i.e., the top-1 predicted label, as their attack model's input. It is more realistic to evaluate the vulnerability of a machine learning system under label-only attacks with sole access to the model's final prediction. First, compared to score-based attacks, label-only attacks are much more relevant in real-world applications where confidence scores are rarely accessible. Furthermore, label-only attacks have the potential to be much more robust to state-of-the-art defenses, such as confidence score perturbation [66, 143, 93]. In label-only exposure, a naive *baseline attack* [146] infers that a candidate sample is a member of a target model if it is predicted correctly by the model. However, this baseline attack cannot distinguish between members and non-members that are both correctly classified.

In this work, we propose two types of label-only attacks under different scenarios, namely *transfer-based label-only attack* and *boundary-based label-only attack*. We outline the threat models considered in this work in Table 3.1. In the following, we abstractly introduce our proposed two label-only membership inference attacks.

Transfer-Based Attack. We assume the adversary has an auxiliary dataset (namely

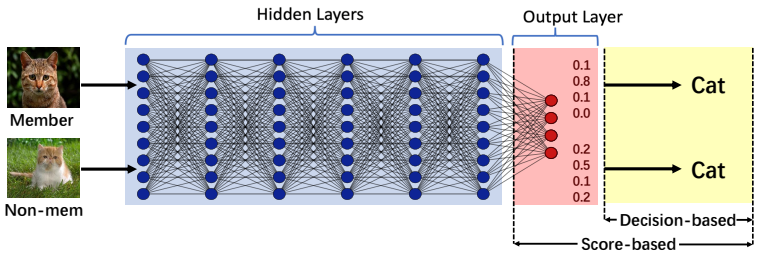


Figure 3.1: An illustration of accessible components of the target model for each of the two threat models. A score-based threat model assumes access to the output layer; a label-only threat model assumes access to the predicted label alone.

shadow dataset) that comes from the same distribution as the target model's training set. The assumption also holds for previous score-based attacks [124, 85, 116, 129]. The adversary first queries the target model in a manner analog to cryptographic oracle, thereby relabeling the shadow dataset by the target model's predicted labels. Then, the adversary can use the relabeled shadow dataset to construct a local shadow model to mimic the behavior of the target model. This way, the relabeled shadow dataset contains sufficient information from the target model, and membership information can also be transferred to the shadow model. Finally, the adversary can leverage the shadow model to locally launch a score-based membership inference attack.

Boundary-Based Attack. Collecting data, especially sensitive and private data, is a non-trivial task. Thus, we consider a more difficult and realistic scenario in which no shadow dataset and model are available. To compensate for the lack of information in this scenario, we shift the focus from the target model's output to the input. Here, our key intuition is that it is harder to perturb member data samples to different classes than non-member data samples. The adversary queries the target model on candidate data samples and perturbs them to change the model's predicted labels. Then the adversary can exploit the magnitude of the perturbation to differentiate member and non-member data samples.

Extensive experimental evaluation shows that both of our attacks achieve strong performance. In particular, in some cases, our boundary-based attack even outperforms the previous score-based attacks. This demonstrates the severe membership risks stemming from ML models. In addition, we present a new perspective on the success of current membership inference and show that the distance between a sample and an ML model's decision boundary is strongly correlated with the sample's membership status.

Finally, we evaluate our attacks on multiple defense mechanisms: generalization enhancement [130, 135, 116], privacy enhancement [13] and confidence score perturbation [93, 66, 143]. The results show that our attacks can bypass most of the defenses unless heavy regularization is applied. However, heavy regularization can significantly affect the model's accuracy.

In general, our contributions can be summarized as the following:

- We systematically investigate membership leakage in label-only exposures of ML models and introduce label-only membership inference attacks, which are highly relevant for real-world applications and important to gauge model privacy.

Table 3.1: An overview of membership inference threat models. "✓" means the adversary needs the knowledge and "-" indicates the knowledge is not necessary.

Category	Attacks	Data Distribution	Shadow Model	Detailed Prediction (e.g. probabilities)	Final Prediction (e.g. class label)
Score-Based	[124, 85, 116, 146, 129, 60, 78]	✓or -	✓or -	✓	✓
Label-Only	Baseline [146]	✓	-	-	✓
	Transfer-Based	✓	✓	-	✓
	Boundary-Based	-	-	-	✓

- We propose two types of label-only attacks under different scenarios, including transfer-based attack and boundary-based attack. Extensive experiments demonstrate that our two types of attacks perform better than the baseline attack and even outperform the previous score-based attacks in some cases.
- We propose a new perspective on the reasons for the success of membership inference and perform a quantitative and qualitative analysis to demonstrate that members of an ML model are more distant from the model's decision boundary than non-members.
- We evaluate multiple defenses against our label-only attacks and show that our novel attacks can still achieve reasonable performance unless heavy regularization is applied.

3.1.2 Organization

The rest of this work is organized as follows. Section 3.2 presents the threat models, key intuition, attack methodology, and evaluation of the transfer-based attack. Section 3.2 presents the treat models, key intuition, attack methodology, and evaluation of the label-only attack. In Section 3.4, we provide an in-depth analysis of the success of membership inference. Section 3.5 provides multiple defenses against label-only attacks.

3.2 Transfer-Based Label-Only Membership Inference Attack

In this section, we present the first type of label-only membership inference attacks, i.e., transfer-based attack. We start by introducing our key intuition. Then, we describe the attack methodology. Finally, we present the evaluation results.

3.2.1 Threat Model

In the transfer-based label-only membership inference attack, we define that the adversary only has black-box access to the target model. Concretely, the adversary cannot access the target model's confidence scores but relies on the final predictions, i.e., the predicted label, to launch the membership inference attack.

Based on our key intuition (Section 3.2.2), we assume that the adversary trains a local model (called shadow model) to mimic the behavior of the target model and relies on the shadow model to infer membership information. We further assume that the shadow model has the same architectures as the target model. Note that we show this assumption can be relaxed in Section 3.2.

To train the shadow model, we make another assumption that the adversary has an auxiliary dataset (namely, shadow dataset) that comes from the same distribution as the target model's training set. Note that both the shadow model and shadow dataset assumptions hold for previous score-based attacks [124, 85, 116, 129].

3.2.2 Key Intuition

The intuition of this attack is that the transferability property holds between the shadow model and the target model. Almost all related works [96, 36, 82, 92] focus on the transferability of adversarial examples, i.e., adversarial examples can transfer between models trained for the same task. Unlike these works, we focus on the transferability of membership information for benign data samples, i.e., the member and non-member data samples behaving differently in the target model will also behave differently in the shadow model. Then we can leverage the shadow model to launch a score-based membership inference attack.

Algorithm 1: Transfer-based label-only attack algorithm.

Input: shadow dataset $\mathcal{D}_{shadow}$, shadow model $\mathcal{S}$, target model $\mathcal{M}$, a candidate sample (x, y), threshold τ, minibatch m, membership indicator T;

Output: Trained shadow model $\mathcal{S}$, x is member or not;

Initialize the parameters of *shadow*;
Relabel $\mathcal{D}_{shadow}$ by querying to $\mathcal{M}$;
for *number of training epochs* **do**
  for $i = 1; i \leq \frac{|\mathcal{D}_{shadow}|}{m}; i++$ **do**
    sample minibatch of m samples from $\mathcal{D}_{shadow}$;
    update $\mathcal{S}$ by descending its Adam gradient
  end
end
Feed x into $\mathcal{S}$ to obtain p_i;
calculate loss: $l = -\sum_{i=0}^{K} \mathbf{1}_y \log(p_i)$;
if $l \leq \tau$ **then**
  $T = 1$; ; /* x is a member */
else
  $T = 0$; ; /* x is a non-member */
end
return $\mathcal{S}$, T;

3.2.3 Attack Methodology

The transfer-based attack methodology can be divided into four stages: shadow dataset relabeling, shadow model architecture selection, shadow model training, and membership inference. The algorithm can be found in Algorithm 1.

Shadow Dataset Relabeling. As aforementioned, the adversary has a shadow dataset $\mathcal{D}_{shadow}$ drawn from the same distribution as the target model $\mathcal{M}$'s dataset $\mathcal{D}_{target}$. To train a shadow model, the first step is to relabel these data samples using the target model $\mathcal{M}$ as an oracle. In this way, the adversary can establish a connection between the shadow dataset and the target model, which facilitates the shadow model to be more similar to the target model in the next step.

Shadow Model Architecture Selection. As the adversary knows the main task of the target model, it can build the shadow model using high-level knowledge of the classification task (e.g., convolutional networks are appropriate for vision). As in prior score-based attacks, we also use the same architecture of target models to build our shadow models. Note that we emphasize that the adversary does not have the knowledge of the concrete architecture of the target model, and in Section 3.2.5, we also show that a wide range of architecture choices yield similar attack performance.

Shadow Model Training. The adversary trains the shadow model $\mathcal{S}$ with the relabeled shadow dataset $\mathcal{D}_{shadow}$ in conjunction with classical training techniques.

Membership Inference. Finally, the adversary feeds a candidate data sample into the shadow model $\mathcal{S}$ to calculate its cross-entropy loss with the ground truth label.

$$\text{CELoss} = -\sum_{i=0}^{K} \mathbf{1}_y \log(p_i), \tag{3.1}$$

where $\mathbf{1}_y$ is the one-hot encoding of the ground truth label y, p_i is the probability that the candidate sample belongs to class i, and K is the number of classes. If the loss value is smaller than a threshold, the adversary then determines the sample being a member and vice versa. The adversary can pick a suitable threshold depending on their requirements, as in many machine learning applications [20, 102, 47, 43, 116, 66]. In our evaluation, we mainly use the area under the ROC curve (AUC), which is threshold independent as our evaluation metric.

3.2.4 Experimental Setup

Following the attack strategy, we split each dataset into $\mathcal{D}_{target}$ and $\mathcal{D}_{shadow}$: One is used to train and test the target model, and the other is used to train the shadow model $\mathcal{S}$ after relabeled by the target model. For evaluation, $\mathcal{D}_{target}$ is also split into two: One is used to train the target model $\mathcal{M}$, i.e., $\mathcal{D}_{train}$, and serves as the member samples of the target model, while the other $\mathcal{D}_{test}$ serves as the non-member samples.

It is well known that the inherent overfitting drives ML models to be vulnerable to membership leakage [124, 116]. To show the variation of the attack performance on each dataset, we train 6 target models $\mathcal{M}$-0, $\mathcal{M}$-1, ..., $\mathcal{M}$-5 using different sizes of the training set $\mathcal{D}_{train}$, exactly as performed in the prior work by Shokri et al. [124] and

many subsequent works [135, 116, 129, 85]. The sizes of $\mathcal{D}_{train}$, $\mathcal{D}_{test}$, and $\mathcal{D}_{shadow}$ are summarized in Appendix Table A.1.

We execute the evaluation on randomly reshuffled data samples from $\mathcal{D}_{target}$, and select sets of the same size (i.e, equal number of members and non-members) to maximize the uncertainty of inference. Thus the baseline performance is equivalent to random guessing. We adopt AUC as our evaluation metric, which is threshold independent. In addition, we further discuss methods to pick the threshold for our attack later in this section.

3.2.5 Evaluation

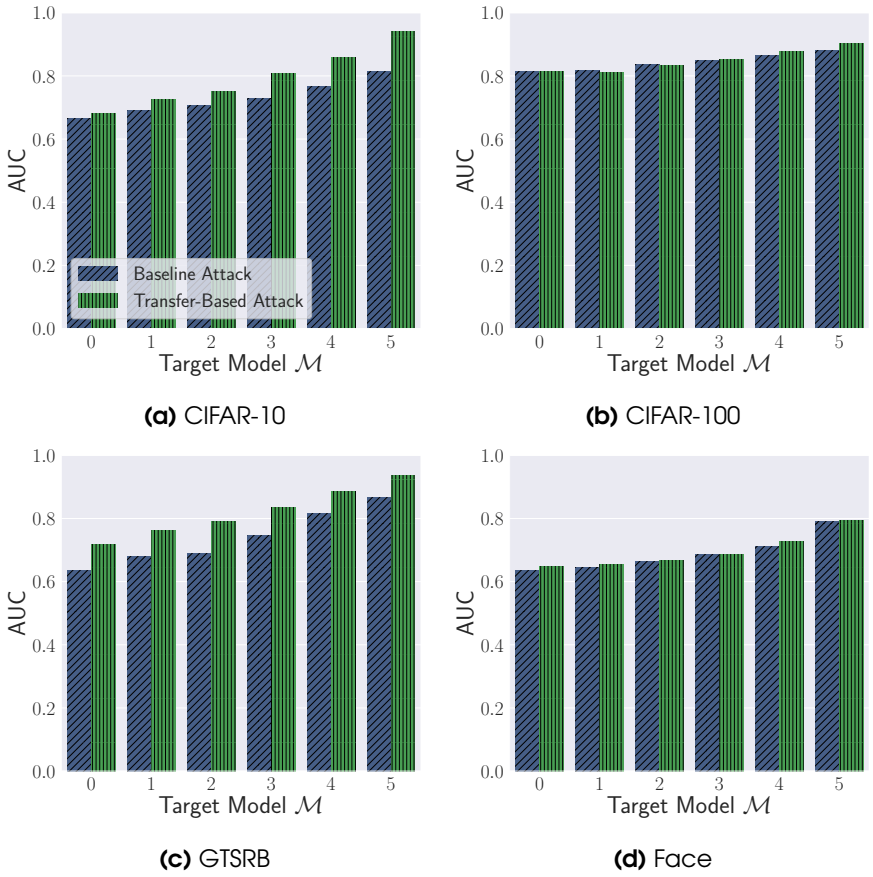
Experimental Setup. Following the attack strategy, we split each dataset into $\mathcal{D}_{target}$ and $\mathcal{D}_{shadow}$: One is used to train and test the target model, and the other is used to train the shadow model $\mathcal{S}$ after relabeled by the target model. For evaluation, $\mathcal{D}_{target}$ is also split into two: One is used to train the target model $\mathcal{M}$, i.e., $\mathcal{D}_{train}$, and serves as the member samples of the target model, while the other $\mathcal{D}_{test}$ serves as the non-member samples.

It is well known that the inherent overfitting drives ML models to be vulnerable to membership leakage [124, 116]. To show the variation of the attack performance on each dataset, we train 6 target models $\mathcal{M}$-0, $\mathcal{M}$-1, ..., $\mathcal{M}$-5 using different sizes of the training set $\mathcal{D}_{train}$, exactly as performed in the prior work by Shokri et al. [124] and many subsequent works [135, 116, 129, 85]. The sizes of $\mathcal{D}_{train}$, $\mathcal{D}_{test}$, and $\mathcal{D}_{shadow}$ are summarized in Appendix Table A.1.

We execute the evaluation on randomly reshuffled data samples from $\mathcal{D}_{target}$, and select sets of the same size (i.e, equal number of members and non-members) to maximize the uncertainty of inference. Thus the baseline performance is equivalent to random guessing. We adopt AUC as our evaluation metric which is threshold independent. In addition, we further discuss methods to pick the threshold for our attack later in this section.

Attack AUC Performance. Figure 3.2 depicts the performance of our transfer-based attack and baseline attack. First, we can observe that our transfer-based attack performs at least on-par with the baseline attack. More encouragingly, on the CIFAR-10 and GTSRB datasets, our transfer-based attack achieves better performance than the baseline attack. For example, in Figure 3.2 ($\mathcal{M}$-5, CIFAR-10), the AUC score of the transfer-based attack is 0.94, while that of the baseline attack is 0.815. The reason why our transfer-based attack outperforms the baseline attack on CIFAR-10 and GTSRB rather than on CIFAR-100 and Face, is that the size of the shadow dataset for the first two datasets is relatively larger than that of the latter two, compared to the size of each dataset (see Appendix Table A.1). In the next experiments, we make the same observation that a larger shadow dataset implies better attack performance.

Effects of the Shadow Dataset and Model. We further investigate the effects of shadow dataset size and shadow model complexity (structure and hyper-parameter) on the attack performance. More concretely, for the target model ($\mathcal{M}$-0, CIFAR-100), we vary the size of the shadow dataset $\mathcal{D}_{shadow}$ from 5,000 to 42,000, where the target



(a) CIFAR-10 **(b)** CIFAR-100

(c) GTSRB **(d)** Face

Figure 3.2: Comparison of our transfer-based attack performance with the baseline attack by Yeom et al. (146). The x-axis represents the target model being attacked and the y-axis represents the AUC score.

training set $\mathcal{D}_{train}$ is 7,000. We also vary the complexity of the shadow model from 0.86M (number of parameters) and 26.01M (FLOPs,[1] computational complexity) to 4.86M and 418.88M, where the complexity of the target model is 3.84M and 153.78M, respectively. We conduct extensive experiments to simultaneously tune these two hyper-parameters and report the results in Figure 3.3. Through investigation, we make the following observations.

- Larger shadow dataset implies more queries to the target model which leads to better attack performance.
- Even simpler shadow models and fewer shadow datasets (bottom left part) can achieve strong attack performance.
- In general, the transfer-based attack is robust even if the shadow model is much different from the target model.

[1]FLOPs represent the theoretical amount of floating-point arithmetic needed when feeding a sample into the model.

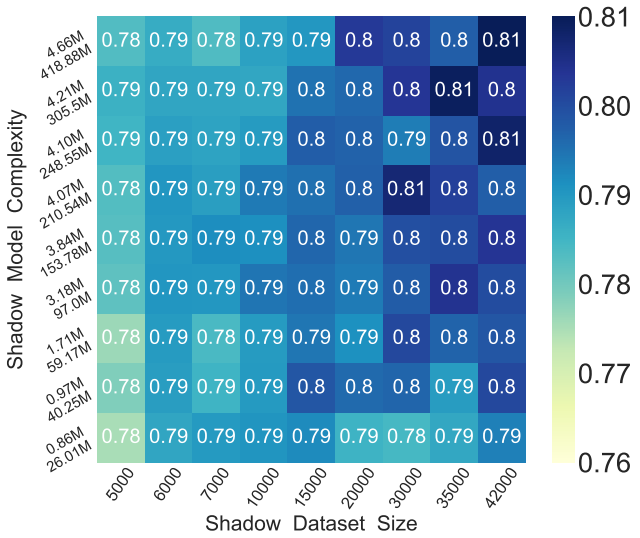
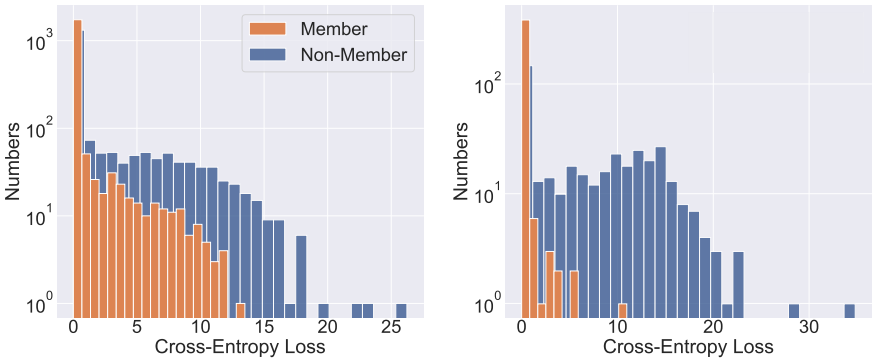


Figure 3.3: Attack AUC under the effect of changing the dataset size and shadow model complexity (upper is the number of parameters, lower is the computational complexity FLOPs). The target model ($\mathcal{M}$-0, CIFAR-100)'s training set size is 7,000, and the complexity is 3.84M parameters and 153.78M FLOPs.

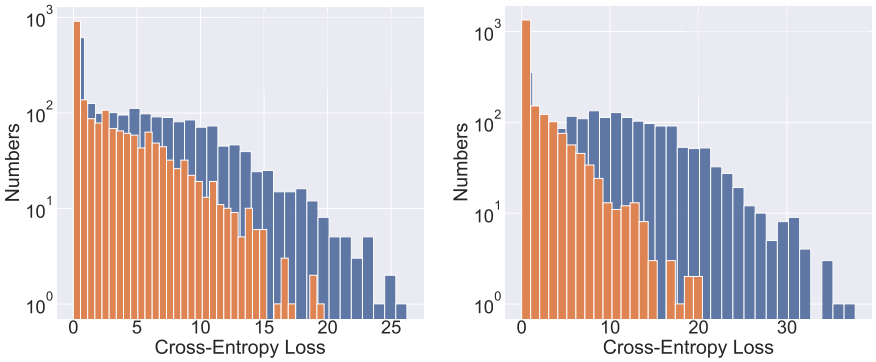
Effects of Statistical Metrics. As prior works [124, 116] also use other statistical metrics, i.e., maximum confidence scores $Max(p_i)$ and normalized entropy $\frac{-1}{\log(K)} \sum_i p_i \log(p_i)$. Here, we also conduct experiments with these statistical metrics. Figure 3.5 reports the AUC on the CIFAR-10 and CIFAR-100 datasets. We can observe that the loss metric achieves the highest performance with respect to the different target models. Meanwhile, the AUC score is very close between the maximum confidence score and entropy. This indicates that the loss metric contains the strongest signal on differentiating member and non-member samples. We will give an in-depth discussion on this in Section 3.4.2.

Loss Distribution of Membership. To explain why our transfer-based attack works, Figure 3.4 further shows the loss distribution of member and non-member samples from the target model calculated on the shadow model ($\mathcal{M}$-0 and $\mathcal{M}$-5 on CIFAR-10 and CIFAR-100). Though both member and non-member samples are never used to train the shadow model, we still observe a clear difference between their loss distribution. This verifies our key intuition aforementioned: The transferability of membership information holds between shadow model $\mathcal{S}$ and target model $\mathcal{M}$, i.e., the member and non-member samples behaving differently in $\mathcal{M}$ will also behave differently in $\mathcal{S}$.

Threshold Choosing. As mentioned before, in the membership inference stage, the adversary needs to make a manual decision on which threshold to use. For the transfer-based attack, since we assume that the adversary has a dataset that comes from the same distribution as the target model's dataset, it can rely on the shadow dataset to estimate a threshold by sampling a certain part of that dataset as non-member samples.

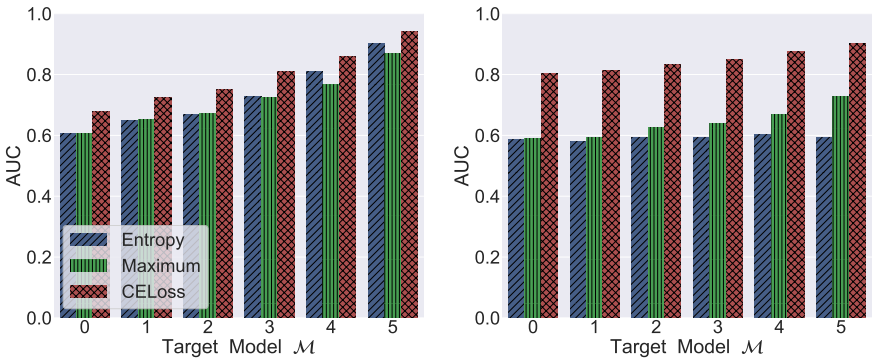


(a) CIFAR-10, $\mathcal{M}$-0 **(b)** CIFAR-10, $\mathcal{M}$-5



(c) CIFAR-100, $\mathcal{M}$-0 **(d)** CIFAR-100, $\mathcal{M}$-5

Figure 3.4: The cross-entropy loss distribution obtained from the shadow model. The x-axis represents the loss value, and the y-axis represents the loss number.



(a) CIFAR-10 **(b)** CIFAR-100

Figure 3.5: Attack AUC for three different statistical measures. The x-axis represents the target model being attacked, and the y-axis represents the AUC score.

3.3 Boundary-Based Label-Only Membership Inference Attack

This section presents our second type of label-only membership inference attack, i.e., boundary-based attack. We start with the threat model description. Then, we introduce

Table 3.2: The cross-entropy between the confidence scores and other labels except for the predicted label. ACE represents the Average Cross Entropy.

Status	Truth Label	Predicted Label	Cross Entropy 0	1	2	···	7	8	9	ACE
(a) Member	6	6	7.8156	8.3803	4.1979	···	7.6328	1.5522	1.2923	**4.4946**
(b) Non-member	8	8	2.3274	0.8761	0.8239	···	1.1152	-	5.0451	1.2218
(c) Member	3	3	1.2995	5.2842	5.4212	···	7.1547	3.2411	4.7910	**4.2334**
(d) Non-member	7	9	2.8686	1.8325	3.6480	···	0.6866	3.1071	-	2.1766

the key intuition and attack methodology. In the end, we present the evaluation results.

3.3.1 Threat Model

Since curating auxiliary data requires significant time and monetary investment. Thus, we relax the assumptions of both the shadow dataset and shadow model in this attack. The adversary does not have a shadow dataset to train a shadow model. All the adversary could rely on is the predicted label from the target model. To the best of our knowledge, this is by far the most strict setting for membership inference against ML models.

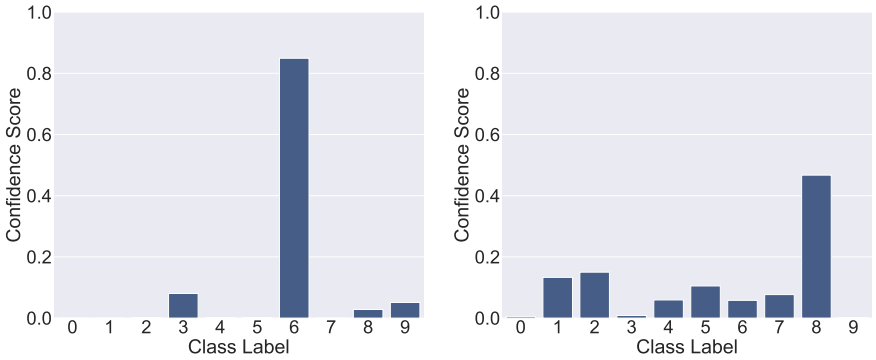
3.3.2 Key Intuition

Our intuition behind this attack follows a general observation of the overfitting nature of ML models. Concretely, an ML model is more confident in predicting data samples that it is trained on. In contrast to the prior score-based attacks[124, 85, 116, 146, 129, 60, 78] that directly exploit confidence scores as analysis objects, we place our focus on the antithesis of this observation, i.e., since the ML model is more confident on member data samples, it should be much harder to change its mind.

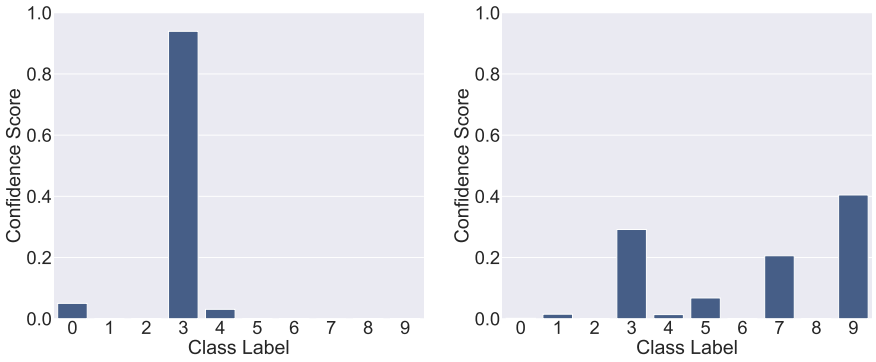
Intuitively, Figure 3.6 depicts the confidence scores for two randomly selected member data samples (Figure 3.6a, Figure 3.6c) and non-member data samples (Figure 3.6b, Figure 3.6d) with respect to $\mathcal{M}$-0 trained on CIFAR-10. We can observe that the maximal score for member samples is indeed much higher than the one for non-member samples. We further use cross entropy (Equation 3.1) to quantify the difficulty for an ML model to change its predicted label for a data sample to other labels.

Table 3.2 shows the cross entropy between the confidence scores and other labels for these samples. We can see that the member samples' cross-entropy is significantly larger than non-member samples. This leads to the following observation on membership information.

Observation. Given an ML model and a set of data samples, the cost of changing the target model's predicted labels for member samples is larger than the cost for non-member samples. Furthermore, consider the label-only exposures in a black-box ML model, which means the adversary can only perturb the data samples to change the target model's predicted labels, thus the perturbation needed to change a member sample's predicted label is larger than non-members. Then, the adversary can exploit



(a) Member data sample **(b)** Non-member data sample



(c) Member data sample **(d)** Non-member data sample

Figure 3.6: The probability distribution of the target model ($\mathcal{M}$-0, CIFAR-10) on member samples and non-member samples.

the magnitude of the perturbation to determine whether the sample is a member or not.

3.3.3 Attack Methodology

Our attack methodology consists of the following three stages, i.e., decision change, perturbation measurement, and membership inference. The algorithm can be found in Algorithm 2.

Decision Change. The goal of changing the final model decision, i.e., predicted label, is similar to that of adversarial attack [23, 98, 97, 27, 134, 119], For simplicity, we utilize adversarial example techniques to perturb the input to mislead the target model. Specifically, we utilize two state-of-the-art black-box adversarial attacks, namely HopSkipJump [29] and QEBA [77], which only require access to the model's predicted labels.

Perturbation Measurement. Once the final model decision has changed, we measure the magnitude of the perturbations added to the candidate input samples. In general, adversarial attack techniques typically use L_p distance (or Minkowski Distance), e.g.,

Algorithm 2: Boundary-based label-only attack algorithm.

Input: adversarial attack technique *HopSkipJump*, target model $\mathcal{M}$, a candidate sample (x, y), threshold τ, membership indicator T;
Output: x is member or not;

for *number of query* **do**
  Feed x into $\mathcal{M}$ to obtain predicted label y';
  if $y' \neq y$ **then**
    $x' = \mathrm{x}$; ; /* perturbed sample x' */
  else
    Apply *HopSkipJump* to perturb x ;
  end
end
calculate perturbation $P = |x - x'|_2$;
if $P \leq \tau$ **then**
  $T = 0$; ; /* x is a non-member */
else
  $T = 1$; ; /* x is a member */
end
return T;

L_0, L_1, L_2, and L_∞, to measure the perceptual similarity between a perturbed sample and its original one. Thus, we use L_p distance to measure the perturbation.

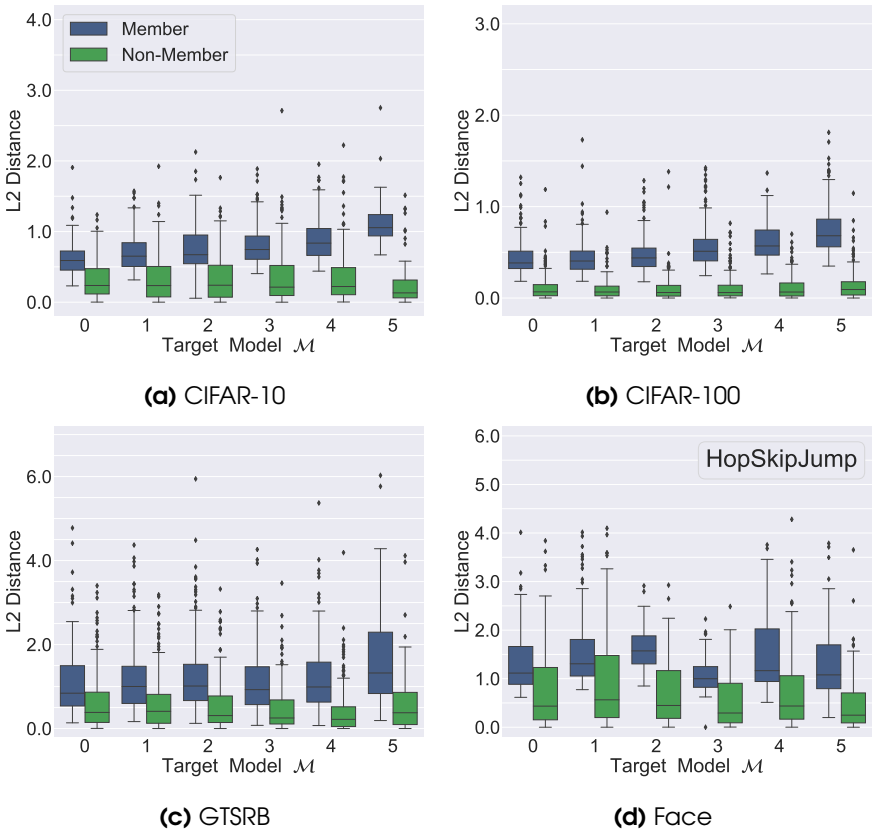
Membership Inference. After obtaining the magnitude of the perturbations, the adversary simply considers a candidate sample with perturbations larger than a threshold as a member sample and vice versa. Similar to the transfer-based attack, we mainly use AUC as our evaluation metric. We also provide a general and simple method for choosing a threshold in Section 3.3.4.

3.3.4 Evaluation

Experiment Setup. We use the same experimental setup as presented in Section 3.2.5, such as the dataset splitting strategy and 6 target models trained on different sizes of the training set $\mathcal{D}_{train}$. In the decision change stage, we use the implementation of a popular python library (ART[2]) for HopSkipJump. Note that we only apply untargeted decision change, i.e., changing the initial decision of the target model to any other random decision. Besides, as HopSkipJump requires multiple queries to perturb data samples to change their predicted labels, we set 15,000 as the default. We further study the influence of the number of queries on the attack performance.

Distribution of Perturbation. First, we show the distribution of perturbation between a perturbed sample and its original one for member and non-member samples in Figure 3.7. Due to the decision change scheme, i.e., HopSkipJump, applies L_2 distance to limit the magnitude of perturbation, thus we report results of L_2 distance as well.

[2] https://github.com/Trusted-AI/adversarial-robustness-toolbox



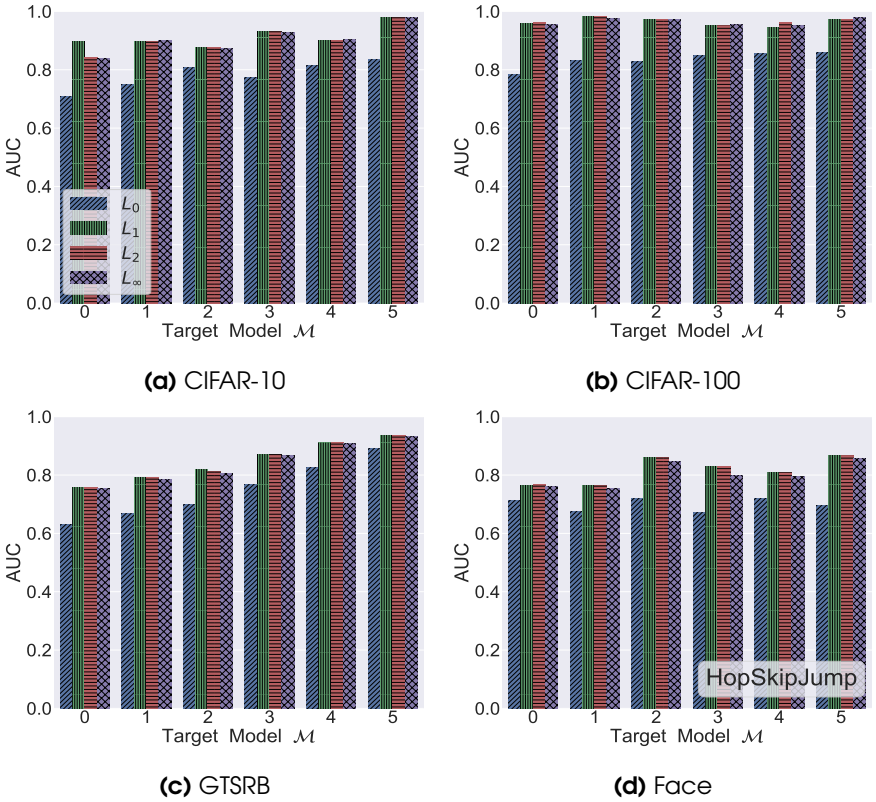
(a) CIFAR-10 **(b)** CIFAR-100

(c) GTSRB **(d)** Face

Figure 3.7: L_2 distance between the original sample and its perturbed samples generated by the HopSkipJump attack. The x-axis represents the target model being attacked, and the y-axis represents the L_2 distance.

As expected, the magnitude of the perturbation on member samples is indeed larger than that on non-member samples. For instance, in Figure 3.7 ($\mathcal{M}$-5, CIFAR-10), the average L_2 distance of the perturbation for member samples is 1.0755, while that for non-member samples is 0.1102. In addition, models with a larger training set, i.e., lower overfitting level, require less perturbation to change the final prediction. As the overfitting level increases, the adversary needs to modify more on the member sample. The reason is that an ML model with a higher overfitting level has remembered its training samples to a larger extent, thus it is much harder to change their predicted labels, i.e., larger perturbation is required.

Attack AUC Performance. We report the AUC scores over all datasets in Figure 3.8. In particular, we compare 4 different distance metrics, i.e., L_0, L_1, L_2, and L_∞, for each decision change scheme. From Figure 3.8, we can observe that L_1, L_2, and L_∞ metrics achieve the best performance across all datasets. For instance in Figure 3.8 ($\mathcal{M}$-1, CIFAR-10), the AUC scores for L_1, L_2, and L_∞ metrics are 0.8969, 0.8963, and 0.9033, respectively, while the AUC score for L_0 metric is 0.7405. Therefore, an adversary can simply choose the same distance metric adopted by adversarial attacks to measure the magnitude of the perturbation.



(a) CIFAR-10 **(b)** CIFAR-100

(c) GTSRB **(d)** Face

Figure 3.8: Attack AUC for four different L_p distances between the original samples and its perturbed samples generated by the HopSkipJump attack. The x-axis represents the target model being attacked, and the y-axis represents the AUC score.

Effects of Number of Queries. To mount the boundary-based attack in real-world ML applications such as Machine Learning as a Service (MLaaS), the adversary cannot issue as many queries as they want to the target model, since a large number of queries increase the cost of the attack and may raise the suspicion of the model provider. Now, we evaluate the attack performance with the different number of queries. Here, we show the results of the HopSkipJump scheme for $\mathcal{M}$-5 over all datasets. We vary the number of queries from 0 to 15,000 and evaluate the attack performance based on the L_2 metric. As we can see in Figure 3.9, the AUC increases sharply as the number of queries increases in the beginning. After 2,500 queries, the attack performance becomes stable. From the results, we argue that query limiting would likely not be a suitable defense. For instance, when querying 131 times, the AUC for CIFAR-10 is 0.8228 and CIFAR-100 is 0.9266. At this time, though the perturbed sample is far away from its origin's decision boundary, the magnitude of perturbation for member samples is still relatively larger than that for non-member samples. Thus, the adversary can still differentiate between member and non-member samples.

Threshold Choosing. Here, we focus on the threshold choosing for our boundary-based attack where the adversary is not equipped with a shadow dataset. We provide a simple and general method for choosing a threshold. Concretely, we generate a set

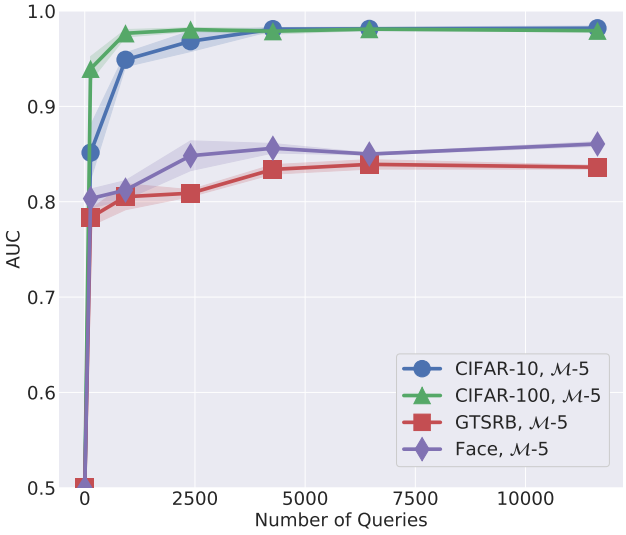


Figure 3.9: Attack AUC under the effect of the number of queries. The x-axis represents the number of queries, and the y-axis represents the AUC score for our boundary-based attack.

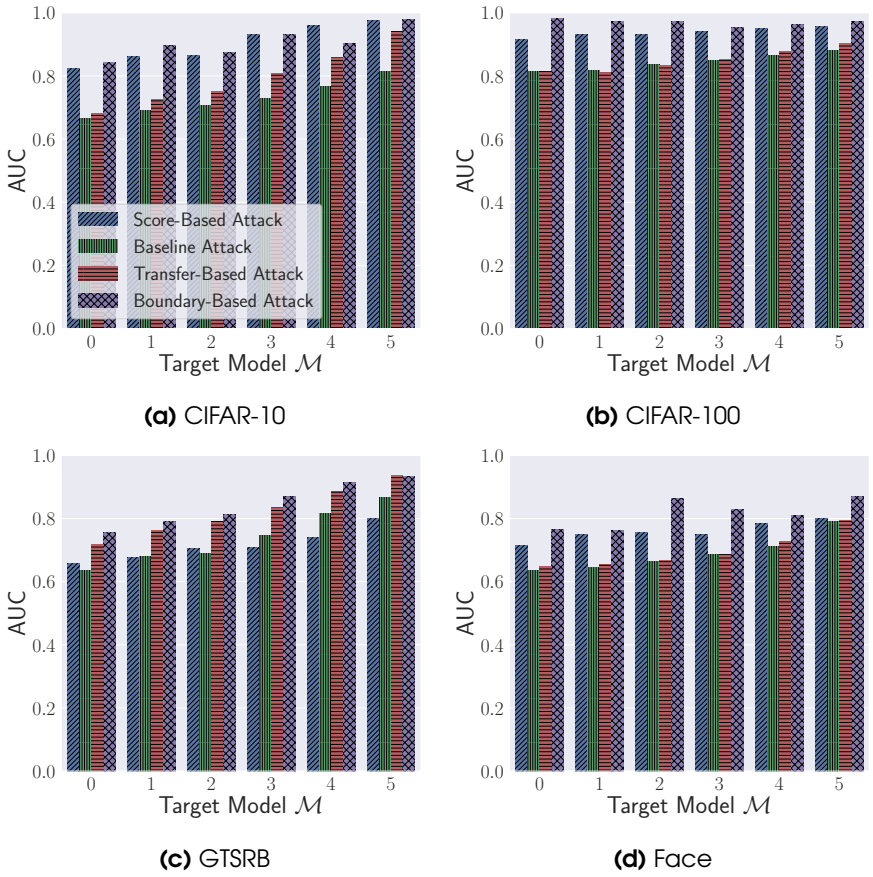
of random samples in the feature space as the target model's training set. In the case of image classification, we sample each pixel for an image from a uniform distribution. Next, we treat these randomly generated samples as non-members and query them to the target model. Then, we apply adversarial attack techniques on these random samples to change their initial predicted labels by the target model. Finally, we use these samples' perturbation to estimate a threshold, i.e., finding a suitable top t percentile over these perturbations.

We experimentally generate 100 random samples for $\mathcal{M}$-5 trained across all datasets, and adopt HopSkipJump in the decision change stage. We again use the L_2 distance to measure the magnitude of perturbation and F1 score as our evaluation metric. From Figure 3.11, we make the following observations:

- The peak attack performance is bounded between $t = 0\%$ and $t = 100\%$, which means the best threshold can definitely be selected from these random samples' perturbation.
- The powerful and similar attack performance ranges from $t = 30\%$ to $t = 80\%$, reaching half of the total percentile, meaning a suitable threshold can be easily selected.

Therefore, we conclude that our threshold-choosing method is effective and can achieve excellent performance.

Comparison of Different Attacks. Now we compare the performance of our two attacks and previous existing attacks. In particular, we also compare our attacks against prior score-based attacks. Following the score-based attack proposed by Salem



(a) CIFAR-10 **(b)** CIFAR-100

(c) GTSRB **(d)** Face

Figure 3.10: Comparison of our two types of attacks with the baseline attack and score-based attack. The x-axis represents the target model being attacked, and the y-axis represents the AUC score.

et al. [116], we train one shadow model using half of $\mathcal{D}_{shadow}$ with its ground truth labels, and one attack model in a supervised manner based on the shadow model's output scores. Here, we do not assume that the attacker knows the exact training set size of the target model, which is actually a strong assumption. Note that this is not a fair comparison, as our label-only attacks only access to the final model's prediction, rather than the confidence scores.

We report attack performance for our boundary-based attack using the L_2 metric in the HopSkipJump scheme. From Figure 3.10, we can find that our boundary-based attack achieves similar or even better performance than the score-based attack in some cases. This demonstrates the efficacy of our proposed label-only attack, thereby the corresponding membership leakage risks stemming from ML models are much more severe than previously shown.

As for cost analysis, the attack logic is different for each method, so it is difficult to evaluate the cost with standard metrics. Besides the adversarial knowledge acquired for each attack, we mainly report training and query costs in Table 3.3. We can find the baseline attack only queries once for a candidate sample. However, in our transfer-based

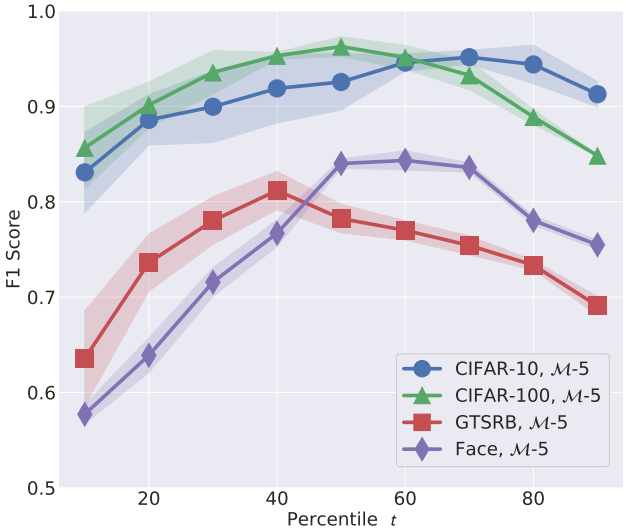


Figure 3.11: The relation between the top t percentile of the L_2 distance, i.e., threshold, and the attack performance. The x-axis represents the top t percentile and the y-axis represents the F1 score.

attack, once a shadow model is built, the adversary will only query the shadow model for candidate samples without making any other queries to the target model. Therefore, we cannot prematurely claim that the baseline attack has the lowest cost, but should consider the actual situation.

Table 3.3: The cost of each attack. Query cost is the number of queries to the target model.

Attack Type	Shadow Model Training Epochs	Query for $\mathcal{D}_{shadow}$	Query for a candidate sample
Score-Based Attack	200	-	1
Baseline Attack	-	-	1
Transfer-Based Attack	200	$\lvert\mathcal{D}_{shadow}\rvert$	-
Boundary-Based Attack	-	-	Multiple

3.4 Membership Leakage Analysis

The above results fully demonstrate the effectiveness of our label-only attacks. Here, we delve more deeply into the reasons for the success of membership inference. Our boundary-based attack utilizes the magnitude of the perturbation to determine whether the sample is a member or not, and the key to stopping searching perturbations is the final decision change of the model. Here, the status of decision change actually contains information about the decision boundary, i.e., the perturbed sample crosses the decision boundary. This suggests a new perspective on the relationship between member samples and non-member samples, and we intend to analyze membership leakage from

this perspective. Since previous experiments have verified our key intuition that the perturbation required to change the predicted label of a member sample is larger than that of a non-member, we argue that the distance between the member sample and its decision boundary is typically larger than that of the non-member sample. Next, we will verify it both quantitatively and qualitatively.

3.4.1 Quantitative Analysis

We introduce the neighboring L_p-radius ball to investigate the membership leakage of ML models. This neighboring L_p-radius ball, also known as *Robustness Radius*, is defined as the L_p robustness of the target model at a data sample, which represents the radius of the largest L_p ball centered at the data sample in which the target model does not change its prediction, as shown in Figure 3.12d. Concretely, we investigate the L_2 robustness radius of the target model $\mathcal{M}$ at a data sample x. Unfortunately, computing the robustness radius of a ML model is a hard problem. Researchers have proposed many certification methods to derive a tight lower bound of robustness radius $R(\mathcal{M}; x, y)$ for ML models. Here, we also derive a tight lower bound of robustness radius, namely *Certified Radius* [152], which satisfies $0 \leq CR(\mathcal{M}; x, y) \leq R(\mathcal{M}; x, y)$ for any $\mathcal{M}, x$ and its ground truth label $y \in \mathcal{Y} = \{1, 2, \cdots, K\}$.

ACR of Members and Non-members. In particular, the value of the certified radius can be estimated by repeatedly sampling Gaussian noises [152]. Thus, for the target model $\mathcal{M}$ and a data sample (x, y), we can also estimate the certified radius $CR(\mathcal{M}; x, y)$. Here, we use the *average certified radius* (ACR) as a metric to estimate the average certified radius for members and non-members separately, i.e.,

$$ACR_{member} = \frac{1}{|\mathcal{D}_{train}|} \sum_{(x,y)\in\mathcal{D}_{train}} CR(\mathcal{M}; x, y), \tag{3.2}$$

$$ACR_{non-member} = \frac{1}{|\mathcal{D}_{test}|} \sum_{(x,y)\in\mathcal{D}_{test}} CR(\mathcal{M}; x, y). \tag{3.3}$$

We randomly select an equal number of members and non-members for target models and report the results in Table 3.4. Note that the certified radius is actually an estimated value representing the lower bound of the robustness radius, not the exact radius. Therefore, we analyze the results from a macroscopic perspective and can draw the following observations.

- The ACR of member samples is generally larger than the ACR of non-member samples, which means that in the output space, the ML model maps member samples further away from its decision boundary than non-member samples.
- As the level of overfitting increases, the macroscopic trend of the gap between the ACR of members and non-members is also larger, which exactly reflects the increasing attack performance in the aforementioned AUC results.

Furthermore, we also feed the equal member and non-member samples into each corresponding shadow model and obtain the ACR. Note that member and non-member

Table 3.4: Average Certified Radius (ACR) of members and non-members for target models.

Target	CIFAR-10		CIFAR-100		GTSRB		Face	
Model	Member	Non-mem	Member	Non-mem	Member	Non-mem	Member	Non-mem
$\mathcal{M}$-0	0.1392	0.1201	0.0068	0.0033	0.0300	0.0210	0.0571	0.0607
$\mathcal{M}$-1	0.1866	0.1447	0.0133	0.0079	0.0358	0.0215	0.0290	0.0190
$\mathcal{M}$-2	0.1398	0.1170	0.0155	0.0079	0.0692	0.0463	0.0408	0.0313
$\mathcal{M}$-3	0.1808	0.1190	0.0079	0.0074	0.0430	0.0348	0.1334	0.1143
$\mathcal{M}$-4	0.1036	0.1032	0.0141	0.0116	0.0212	0.0176	0.0392	0.0292
$\mathcal{M}$-5	0.1814	0.0909	0.0157	0.0080	0.0464	0.0385	0.1242	0.1110

Table 3.5: Average Certified Radius (ACR) of members and non-members for shadow models.

Shadow	CIFAR-10		CIFAR-100	
Model	Member	Non-mem	Member	Non-mem
$\mathcal{M}$-0	0.1392	0.1301	0.0091	0.0039
$\mathcal{M}$-1	0.1873	0.1516	0.0150	0.0071
$\mathcal{M}$-2	0.1416	0.1463	0.0177	0.0068
$\mathcal{M}$-3	0.1962	0.1452	0.0121	0.0047
$\mathcal{M}$-4	0.1152	0.1046	0.0099	0.0092
$\mathcal{M}$-5	0.1819	0.0846	0.0176	0.0087

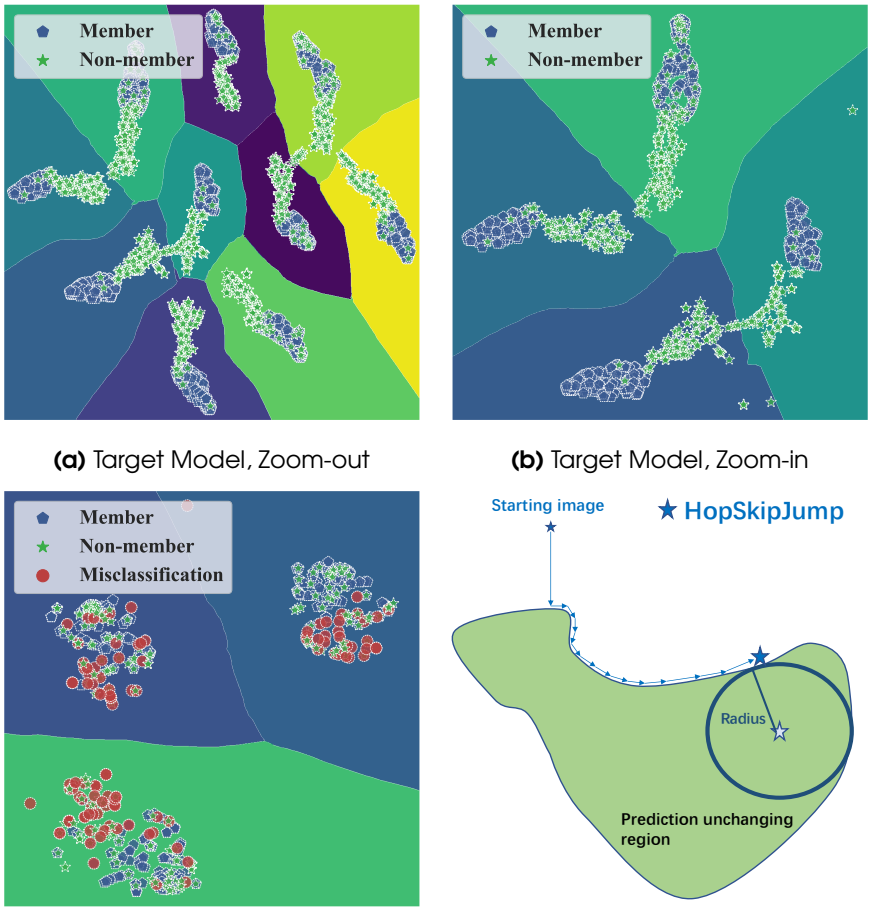
samples are never used to train the shadow model. We report the results in Table 3.5, and we can draw the same observations as for the target model. In other words, this again verifies our key intuition for the transfer-based attack: The transferability of membership information holds between shadow model $\mathcal{S}$ and target model $\mathcal{M}$, i.e., the member and non-member samples behaving differently in $\mathcal{M}$ will also behave differently with high probability in $\mathcal{S}$.

3.4.2 Qualitative Analysis

Next, we investigate the membership leakage of ML models from a visualization approach. We study the decision boundary of the target model (CIFAR-10, $\mathcal{M}$-3) with a given set of data samples, including 1,000 member samples and 1,000 non-member samples. To better visualize the decision boundary, there are two points to note:

- Both member and non-member samples are mapped from the input space to the output space, presenting the membership signal. Thus, we visualize the decision boundary in the output space, i.e., the transformed space of the last hidden layer fully connected with the final model decision.
- Due to the limitation of the target dataset size, we further sample a large number of random data points in the output space and label them with different colors according to their corresponding classes. This can visualize the decision boundary that distinguishes between different class regions.

To this end, we map the given data samples into the transformed space and embed the output logits or scores into a 2D space using t-Distributed Stochastic Neighbor



(a) Target Model, Zoom-out

(b) Target Model, Zoom-in

(c) Shadow Model, Zoom-in

(d) Adversarial attack process

Figure 3.12: The visualization of decision boundary for target model (a, b) and shadow model (c), and the search process of the perturbed sample by HopSkipJump (d).

Embedding (t-SNE) [5]. Figure 3.12a shows the results for 10 classes of CIFAR-10. We can see that the given data samples have been clearly classified into ten classes and mapped to 10 different regions. For the sake of analysis, we purposely zoom in four different regions in the left of the whole space. From Figure 3.12b, we can make the following observations:

- The member and non-member samples belonging to the same class are tightly divided into 2 clusters, explaining why the previous score-based attacks can achieve effective performance.
- More interestingly, we can see that the member samples are further away from the decision boundary than the non-member samples. That is, the distance between the members and the decision boundary is larger than that of the non-members. Again, this validates our key intuition.

Recall that in the decision change stage of boundary-based attack, we apply black-

box adversarial attack techniques to change the final model decision. Here, we give an intuitive overview of how HopSkipJump and QEBA schemes work in Figure 3.12d. As we can see, though these two schemes adopt different strategies to find the perturbed sample, there is one thing in common: The search ends at the tangent samples between the neighboring L_p-radius ball of the original sample and its decision boundary. Only in this way, can they mislead the target model and also generate a small perturbation. Combined with Figure 3.12b, we can find that the magnitude of perturbation is essentially a reflection of the distance from the original sample to its decision boundary.

We again feed the 1,000 member samples and 1,000 non-member samples to the shadow model (CIFAR-10, $\mathcal{M}$-3) and visualize its decision boundary in Figure 3.12c. In particular, we mark in red the misclassified samples from non-members. First, looking at the correctly classified samples, we can also find that the member samples are relatively far from the decision boundary, i.e., the loss is relatively lower than that of non-member samples. As for the misclassified samples, it is easy to see that their loss is much larger than any other samples. Therefore, we can leverage the loss as a metric to differentiate members and non-members. However, we should also note that compared to Figure 3.12b, the difference between members and non-members towards the decision boundary is much smaller. Thus, if we do not adopt the loss metric, which considers the ground truth label, then the maximum confidence scores $Max(p_i)$ and normalized entropy $\frac{-1}{\log(K)} \sum_i p_i \log(p_i)$ which are just based on self-information will lead to a much lower difference between members and non-members. This is the reason why the loss metric achieves the highest performance.

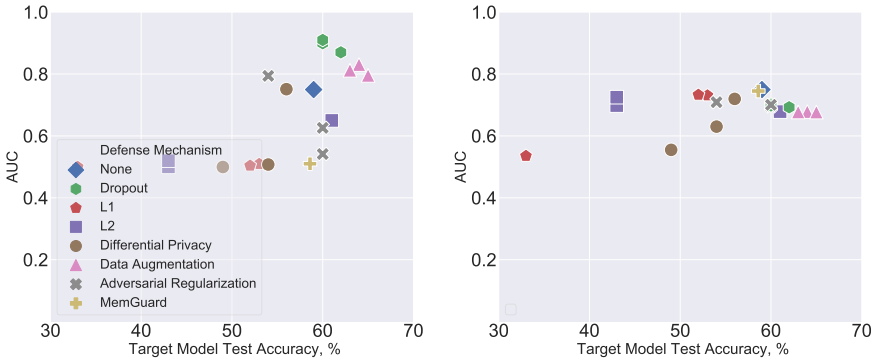
Summarizing the above quantitative and qualitative analysis, we verify our argument that the distance between the member sample and its decision boundary is larger than that of the non-member sample, thus revealing the reasons for the success of the membership inference, including score-based and label-only attacks. In addition, we verify that membership information remains transferable between the target and shadow models. Last but not least, we also show the reason why the loss metric of the transfer-based attack achieves the best performance.

3.5 Defense Evaluation

To mitigate the threat of membership leakage, a large body of defense mechanisms has been proposed in the literature. In this section, we evaluate the performance of current membership inference attacks against state-of-the-art defenses. We summarize existing defenses in the following three broad categories.

Generalization Enhancement. As overfitting is the major reason for membership inference to be successful, multiple approaches have been proposed to reduce overfitting, which were first introduced by the machine learning community to encourage generalization. The standard generalization enhancement techniques, such as weight decay (L1/L2 regularization) [135, 116], dropout [130], and data augmentation, have been shown to limit overfitting effectively, but may lead to a significant decrease in model accuracy.

Privacy Enhancement. Differential privacy [28, 40, 62] is widely adopted for



(a) Score-based attack **(b)** Baseline attack

(c) Transfer-based attack **(d)** Boundary-based attack

Figure 3.13: Attack AUC of transfer-based and boundary-based attacks against multiple defense mechanisms.

mitigating membership privacy. Many differential privacy-based defense techniques add noise to the gradient to ensure privacy in the training process of the ML model. A representative approach in this category is DP-Adam [13], and we adopt an open-source version of its implementation in our experiments.[3]

Confidence Score Perturbation. Previous score-based attacks have demonstrated that the confidence score predicted by the target model clearly presents a membership signal. Therefore, researchers have proposed several approaches to alter the confidence score. We focus on two representative approaches in this category: MemGuard [66] and adversarial regularization [93], which changes the output probability distribution so that both members and non-members look like similar examples to the inference model built by the adversary. We adopt the original implementation of MemGuard,[4] and an open-source version of the adversarial regularization.[5]

For each mechanism, we train 3 target models (CIFAR-10, $\mathcal{M} - 2$) using different hyper-parameters. For example, in L2 regularization, the λ used to constrain the regularization loss is set to 0.01, 0.05, and 0.1, and the λ in L1 regularization is set

[3] https://github.com/ebagdasa/pytorch-privacy
[4] https://github.com/jjy1994/MemGuard
[5] https://github.com/SPIN-UMass/ML-Privacy-Regulization

Table 3.6: Attack AUC performance under the defense of MemGuard.

Attack	CIFAR-10, $\mathcal{M}$-2		Face, $\mathcal{M}$-2	
	None	MemGuard	None	MemGuard
score-based	0.8655	0.5151	0.755	0.513
baseline attack	0.705	0.705	0.665	0.665
transfer-based attack	0.7497	0.7497	0.6664	0.6664
boundary-based attack	0.8747	0.8747	0.8617	0.8617

to 0.0001, 0.001, and 0.005, respectively. In differential privacy, the noise is randomly sampled from a Gaussian distribution $\mathcal{N}(\epsilon, \beta)$, wherein ϵ is fixed to 0 and β is set to 0.1, 0.5, and 1.1, respectively.

We report the attack performance against models trained with a wide variety of different defensive mechanisms in Figure 3.13, and we make the following observations.

- Our label-only attacks. i.e., both transfer-based attack and boundary-based attack, can bypass most types of defense mechanisms.
- Strong differential privacy (β=1.1), L1 regularization ($\lambda = 0.005$) and L2 regularization ($\lambda = 0.1$) can reduce membership leakage but, as expected, lead to a significant degradation in the model's accuracy. The reason is that the decision boundary between members and non-members is heavily blurred.
- Data augmentation can definitely reduce overfitting, but it still does not reduce membership leakage. This is because data augmentation drives the model to strongly remember both the original samples and their augmentations.

In Table 3.6, we further compare the performance of all attacks against MemGuard [66], which is the latest powerful defense technique and can be easily deployed. We can find that MemGuard cannot defend against label-only attacks at all, but is very effective against previous score-based attacks.

3.6 Conclusion

In this chapter, we perform a systematic investigation on membership leakage in label-only exposures of ML models and propose two novel label-only membership inference attacks, including transfer-based attack and boundary-based attack. Extensive experiments demonstrate that our two attacks achieve better performances than the baseline attack, and even outperform prior score-based attacks in some cases. Furthermore, we propose a new perspective on the reasons for the success of membership inference and show that members are further away from the decision boundary than non-members. Finally, we evaluate multiple defense mechanisms against our label-only attacks and show that our novel attacks can still achieve reasonable performance unless heavy regularization has been applied. In particular, our evaluation demonstrates that confidence score perturbation is an infeasible defense mechanism in label-only exposures.

4

Auditing Membership Leakage of Multi-Exit Network

4.1 Introduction

To achieve better performance, large ML models with increasing complexity are proposed. The improvement in performance stems from the fact that the deeper ML model fixes the errors of the shallower one. However, such progression to deeper ML models has dramatically increased the latency and energy required for feedforward inference, as some samples that are already correctly classified or recognized by the shallow ML model do not require additional complexity. This reality has motivated research on input-adaptive mechanisms, i.e., multi-exit network [132], which is an emerging direction for fast inference and energy-efficient computing.

The multi-exit model consists of a backbone model (i.e., a large vanilla model) and multiple exits (i.e., lightweight classifiers) attached to the backbone model at different depths. The backbone model is used for feature extraction and the lightweight classifiers allow data samples to be predicted and to exit at an early layer of the model based on tunable early-exit criteria. Multi-exit architecture can be applied to many critical applications as it can effectively reduce computational costs. For example, exiting early means low latency, which is crucial for operating under real-time constraints in robotics applications, such as self-driving cars. Furthermore, exiting early can improve energy efficiency, which directly influences battery life and heat release, especially on mobile devices.

4.1.1 Contributions

Multi-exit networks, despite their low latency and high energy efficiency, also rely on large-scale data to train themselves, as the way vanilla ML models are trained. As described in chapter 3, these data typically contain sensitive and private information of individuals. Various studies have already shown that vanilla ML models, represented by image classifiers, are vulnerable to leaking sensitive information about the data [94, 76, 124, 116, 78, 128, P1].

However, current various designs of multi-exit networks are only considered to achieve the best trade-off between resource usage efficiency and prediction accuracy, the privacy risks stemming from them have never been explored. This prompts the need for a comprehensive investigation of privacy risks in multi-exit networks, such as the vulnerability of multi-export networks to data privacy attacks, the reasons inherent in this vulnerability, the factors that affect attack performance, and whether or how these factors can be exploited to improve or reduce attack performance.

In this work, we take the first step to audit the privacy risks of multi-exit networks through the lens of membership inference. More specifically, we focus on machine learning classification, which is the most common machine learning task, and conduct experiments with 3 types of membership inference attacks, 6 benchmark datasets, and 8 model architectures.

Main Findings. We first leverage the existing attack methodologies (gradient-based, score-based, and label-only) to audit the multi-exit networks' vulnerability through membership inference attacks. We conduct extensive experiments and the empirical results demonstrate that multi-exit models are less vulnerable to membership inference

attacks than vanilla ML models. For instance, considering the score-based attacks, we achieve an attack success rate of 0.5413 on the multi-exit model trained on CIFAR-10 with the backbone model being ResNet-56, while the result is 0.7122 on the corresponding vanilla ResNet-56. Furthermore, we delve more deeply into the reasons for the lower vulnerability and reveal that the reason behind this is that the multi-exit models are less likely to be overfitted.

We also find that the number of exits is negatively correlated with the attack performance, i.e., multi-exit models with more exits are less vulnerable to membership inference. Besides, a more interesting observation is that considering a certain multi-exit model, exit depth is positively correlated with attack performance, i.e., exits attached to the backbone model at deeper locations are more vulnerable to membership inference. These observations are due to the fact that different depths of exits in the backbone model actually imply different capacity models, and that deeper exits imply higher capacity models, which are more likely to be overfitted by memorizing properties of the training set.

Hybrid Attack. The above findings render us a new factor to improve the attack performance. More concretely, we propose a novel hybrid attack against multi-exit networks that exploit the exit information as new knowledge of the adversary. The hybrid attack's methodology can be divided into two stages:

- Hyperparameter stealing: the adversary's goal is to steal the hyperparameters, i.e., the number of exits and the exit depth of a given multi-exit network designed by the model owner.
- Enhanced membership inference: the adversary then exploits the stolen exit information as new knowledge to launch more powerful member inference attacks.

In particular, we study three different adversaries for obtaining exit information by starting with some strong assumptions, and gradually relaxing these assumptions in order to show that far more broadly applicable attack scenarios are possible. Our investigation shows that indeed, our proposed hybrid attack can achieve better attack performance by exploiting extra exit information, compared to original membership inference attacks.

Adversary 1. For the first adversary, we assume they have direct access to the exit information, i.e., exit depth, as well as train a shadow model of the same architecture (especially the exit placements) as the target model. Further, the adversary trains the shadow models on a shadow dataset that comes from the same distribution as the target dataset. The assumption of the same architecture and same distribution also holds for almost all existing membership inference attacks [94, 76, 124, 116, 78, 128, P1].

We start by querying the target model using a large number of data samples to determine the number of exits attached to the model. Then we propose different methods (e.g., one-hot encoding) based on the attack models adopted by existing attacks to exploit this exit information. Extensive experimental evaluation shows that extra exit information indeed leaks more membership information about training data. For example, our hybrid attack achieves an attack success rate of 0.7681 on a multi-exit WideResNet-32 trained on CIFAR-100, while the result of the original attack is 0.6799.

Adversary 2. For this adversary, we relax the assumption that they have direct access to exit information and keep the assumption of the same architecture and same distribution unchanged. This is a more challenging scenario compared to the previous one.

In this scenario, we propose *time-based hyperparameter stealing* to obtain the exit information. Concretely, we feed a set of samples to the target multi-exit model and record the inference time of these samples. We then propose a simple but effective unsupervised method to cluster the samples based on different inference times. Thus, the number of clusters implies the number of exits, and the index of the cluster implies the exit depth.

The intuition is that the goal of multi-exit models is to reduce computational costs by allowing data samples to be predicted and to exit at an early point. Therefore, the inference time for data samples inevitably varies with the depth of the exit, i.e., data samples leaving deeper exit points imply longer inference times. Thus, we can determine the exit depths by observing the magnitude of inference time. Experimental results show that our hybrid attack achieves a strong performance as our *time-based hyperparameter stealing* can achieve almost 100% prediction accuracy of exit depths.

Adversary 3. This adversary works without any knowledge about the target models and target datasets, that is, the adversary can only construct a shadow model that is different from the target model or a dataset from a different distribution from the target dataset. Meanwhile, the different architectures between the shadow model and the target model will inevitably lead to different exit placements between them. Encouragingly, our hybrid attack still has better attack performance than the original attacks, suggesting that the extra exit information has a broader range of applicable attack scenarios.

Finally, we propose a simple but effective defense mechanism called *TimeGuard*, which postpones giving the prediction, rather than giving them immediately. Our in-depth analysis shows that *TimeGuard* can reduce attack performance to a lower bound and maintain high efficiency, i.e., achieve the best trade-off between privacy and efficiency.

Abstractly, our contributions can be summarized as:

- We take the first step to audit the privacy risks of multi-exit networks through the lens of membership inference attacks.
- Our empirical evaluation shows that the multi-exit networks are less vulnerable to member inference, and the exit information is highly correlated with the attack performance.
- We propose a hybrid attack that exploits the exit information to improve the attack performance of membership inference.
- We evaluate the membership leakage threat caused by the proposed hybrid attack under three different adversarial setups, ultimately arriving at a model-free and data-free adversary, which further enlarges the scope of the hybrid attack.
- We propose *TimeGuard* to mitigate privacy risks stemming from our attack and empirically evaluate its effectiveness.

4.1.2 Organization

The rest of the chapter is organized as follows. In Section 4.2, we conduct a comprehensive measurement of the vulnerability of multi-exit networks to membership inference. Section 4.3, Section 3.2.5, and Section 3.3.4 present the threat models, attack methodologies, and evaluations of our proposed hybrid attack under different types of adversaries, respectively. In Section 4.6, we introduce the defense mechanism. Finally, Section 4.7 concludes the chapter.

4.2 Quantifying Membership Leakage Risks

In this section, we quantify the privacy risks of multi-exit networks through the lens of membership inference attacks. We start by defining the threat model. Then, we describe the attack methodology. Finally, we present the evaluation results. Note that our goal here is not to propose a novel membership inference attack. Instead, we aim to quantify the membership leakages of multi-exit networks. Therefore, we follow the existing attacks and their threat models.

4.2.1 Threat Model

Here, we outline the threat models considered in this work. There are three existing categories of scenarios, i.e., white-box scenario, black-box scenario, and label-only scenario.

Given a target model, we assume the adversary has an auxiliary dataset (namely shadow dataset) that comes from the same distribution as the target model's training set. The shadow dataset is used to train a shadow model, the goal of which is to mimic the behavior of the target model to perform the attack. Furthermore, we assume the shadow model has the same architecture as the target model following previous works [94, 76, 124, 116, 78, 128, P1]. In particular, the exit placements of the shadow multi-exit model are also the same as that of the target multi-exit model.

4.2.2 Attack Methodologies

We leverage existing membership inference attacks, which are designed for vanilla ML models, to multi-exit models. More specifically, for three different scenarios, we consider three representative attacks, namely gradient-based attacks [94, 76] in the white-box scenario, score-based attacks [124, 116, 78, 128] in the black-box scenario, and label-only attacks [P1, 34] in the label-only scenario.

Gradient-based Attacks. In gradient-based attacks [94, 76], the adversary obtains all adversarial knowledge and has full access to the target model. This means for any data sample x, the adversary not only obtains the prediction (score and label) but also knows the intermediate computations (features and gradients) of x on the target model. Given a shadow dataset $\mathcal{D}_{shadow}$, the adversary first splits it into two disjoint sets, i.e., shadow training set $\mathcal{D}^{train}_{shadow}$ and shadow testing set $\mathcal{D}^{test}_{shadow}$. Then the adversary queries the shadow model $\mathcal{S}$ on each data sample x from $\mathcal{D}^{train}_{shadow}$, and computes the prediction score, the feature of the second to the last layer, the loss in a forward pass,

and the gradient of the loss with respect to the last layer's parameters in the backward pass. These computations, in addition to the one-hot encoding of the true label, are concatenated into a flat vector and labeled as a member if x is in the shadow training set $\mathcal{D}_{shadow}^{train}$, otherwise labeled as a non-member. In this way, the adversary can derive all data samples of $\mathcal{D}_{shadow}$ as an attack training data set. With the attack training dataset, the adversary then trains the attack model, which is a binary classifier. Once the attack model is trained, the adversary can perform the attack to query the target model $\mathcal{T}$ to differentiate members and nonmembers of the target dataset $\mathcal{D}_{target}$.

Score-based Attacks. Score-based attacks [124, 116, 78, 128] need to train the shadow model as well. Unlike gradient-based attacks, score-based attacks do not require intermediate features or gradients of the target model, but only access to the output scores of the model. The adversary also derives the attack training dataset by querying the shadow model using the shadow training dataset (labeled as members) and the shadow test dataset (labeled as non-members). The adversary can then use the attack training set to construct an attack model.

Label-only Attacks.. Label-only attacks [P1, 34] consider a more restricted scenario where the target model only exposes the predicted label instead of intermediate features or gradients, or even output scores. Thus, label-only attacks solely rely on the target model's predicated label as their attack model's input. Similar to previous attacks, this attack requires the adversary to train a shadow model. The adversary queries the target model on a data sample and perturbs it to change the model's predicted labels. Then, the adversary measures the magnitude of the perturbation and considers the data samples as members if their magnitude is greater than a predefined threshold, which can be derived by perturbing the shadow dataset on the shadow model.

4.2.3 Experimental Settings

Datasets. We consider six benchmark datasets of different tasks, sizes and complexity to conduct our experiments. Concretely, we adopt three computer vision tasks, namely CIFAR-10 [1], CIFAR-100 [1], TinyImageNet [4], and three non-computer vision tasks, namely Purchases [6], Locations [7] and Texas [8]. In particular, the latter three datasets are privacy-sensitive: Purchases relate to shopping preferences, Locations relate to social connections, and Texas relate to health status. Details of all six datasets can be found in Section 2.4.

Datasets Configuration. For a given dataset $\mathcal{D}$, we randomly split it into four disjoint equal parts: $\mathcal{D}_{target}^{train}$, $\mathcal{D}_{target}^{test}$, $\mathcal{D}_{shadow}^{train}$, and $\mathcal{D}_{shadow}^{test}$. We use $\mathcal{D}_{target}^{train}$ to train the target model $\mathcal{T}$ and treat it as the members of the target model. We treat $\mathcal{D}_{target}^{test}$ as the non-members of the target model. Similarly, we use $\mathcal{D}_{shadow}^{train}$ to train the shadow model $\mathcal{S}$ and treat it as the members of the shadow model. We again treat $\mathcal{D}_{shadow}^{test}$ as the non-members of the shadow model. We feed all $\mathcal{D}_{shadow}^{train}$ and $\mathcal{D}_{shadow}^{test}$ to the shadow model to create an attack training dataset to train the attack models.

Attack Model. Here we establish three types of attack models and each type for one attack.

- **Gradient-based.** This attack has five inputs for the attack model, like the one used by Nasr et al. [94], including the target sample's prediction score, the feature of the second to last layer, classification loss, gradients of the last layer's parameters, and one-hot encoding of its true label. Each input is fed into a different MLP (2 or 3 layers), and the resulting embeddings are concatenated together as one vector to a 4-layer MLP.

- **Score-based.** The score-based attack utilizes the predicted score as input to the attack model, which is constructed as a 4-layer MLP with one input component.

- **Label-only..** Here, the attack model is not a specific MLP but a decision function that measures the magnitude of the perturbation and considers data samples as members if their magnitude is larger than a predefined threshold, which can be derived by perturbing the shadow dataset on the shadow model.

Target Model (Multi-Exit Model). For computer vision tasks, we adopt four popular architectures as the backbone to construct multi-exit models, including VGG-16 [125], ResNet-56 [50], MobileNetV2 [117], and WideResNet-32 [151]. For non-computer vision tasks, we designed four 18-layer fully connected networks (FCN-18) with different numbers of hidden neural units (1024, 2048, 3072, 4096), named FCN-18-1/2/3/4 throughout the work. For the exit placement, we follow the principle of Kaya et al. [72] by attaching an additional lightweight classifier (2- or 3-layer MLP) as an exit, i.e., exit placements are restricted to be at the output of individual network blocks, following an approximately equidistant workload distribution. In particular, for each backbone model, we construct 5 different target models with the number of exits varying from 2 to 6. Note that here we consider the backbone model's own classifier as the final exit point and count it in the total number of exits. For early-exit threshold τ ($0 \leq \tau \leq 1$), we manually search for suitable τ value (among 0 to 1 in 0.05 steps) that achieve the same or similar classification performance as vanilla (backbone) models while gaining lower computational cost. To evaluate computational cost, we calculate the number of mathematical operations (denoted as *ops*) in the feedforward pass process by averaging over 10,000 images. The early-exit threshold we set for multi-exit models can be found in Appendix Table A.2 and Table A.3.

Baseline (Vanilla Model). To fully understand the membership leakages of multi-exit models, we further use the vanilla model as the baseline model. We train eight models from scratch for all datasets, including both computer vision and non-computer vision models. In all cases, including vanilla and multi-exit models, we adopt cross-entropy as the loss function and Adam as the optimizer, and train them for 100 epochs. Our code is implemented in Python 3.8 and PyTorch 1.8.1 and runs on an NVIDIA HGX-A100 server with Ubuntu 18.04.

Metric. Following previous work, we adopt the accuracy, i.e., attack success rate (denoted as `ASR`) through the work, as the attack model's training and testing datasets are both balanced with respect to membership distribution. Note that we average the performance of different multi-exit models with the number of exits varying from 2 to 6

(a) Classification Accuracy

(b) Computational Cost

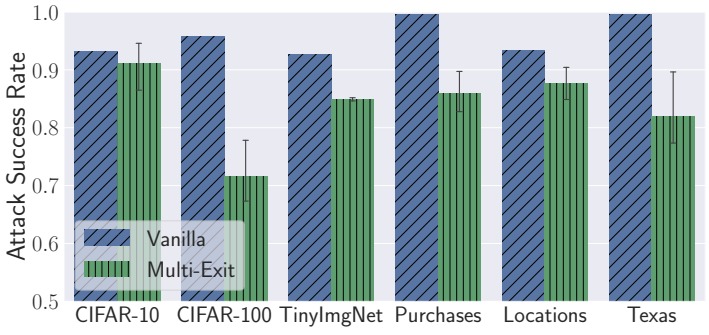
Figure 4.1: The performance of original classification tasks and computational costs for both vanilla and multi-exit models. Computer vision tasks are on VGG-16, and non-computer vision tasks are on FCN-18-1.

and report the mean and standard deviation. Besides, our evaluation adopts different datasets, architectures, and attack methods, which inevitably lead to a wide variety of results.

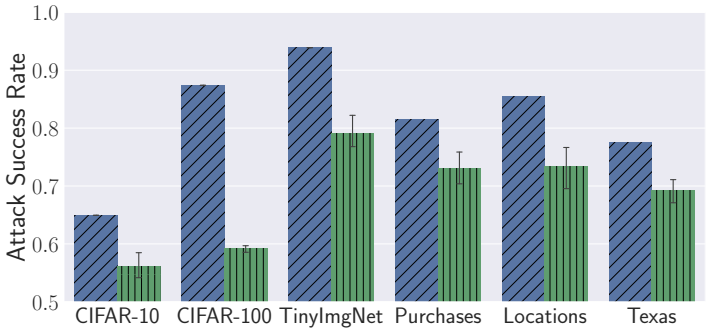
4.2.4 Evaluation

Classification Accuracy and Computational Cost. We first show the performance of vanilla and multi-exit models on their original classification tasks and computational costs in Figure 4.1. See more results of other morels in our peer-review publication [P2]. We observe that the multi-exit model performs at least on par with the vanilla model on the classification task, but is much better in terms of computational cost. For instance, the multi-exit VGG-16 trained on CIFAR-10 achieves 80.558% accuracy, which is better than the 80.04% accuracy of vanilla VGG-16. As for the computational cost, the multi-exit VGG-16 achieves 0.3125 `Gops` while the vanilla model achieves 0.6283 `Gops`.

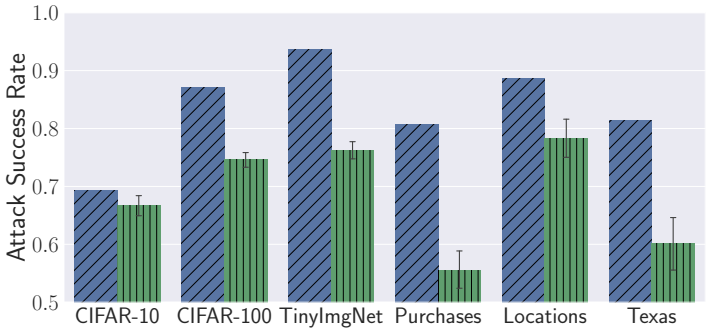
Attack `ASR` Score. Regarding membership inference against vanilla and multi-exit models, we report `ASR` score on all datasets and model architectures in Figure 4.2. We can observe that all the multi-exit models have lower `ASR` than the vanilla models. For example, score-based `ASR` on vanilla VGG-16 trained on CIFAR-100 is 0.8738, while the



(a) Gradient-based



(b) Score-based

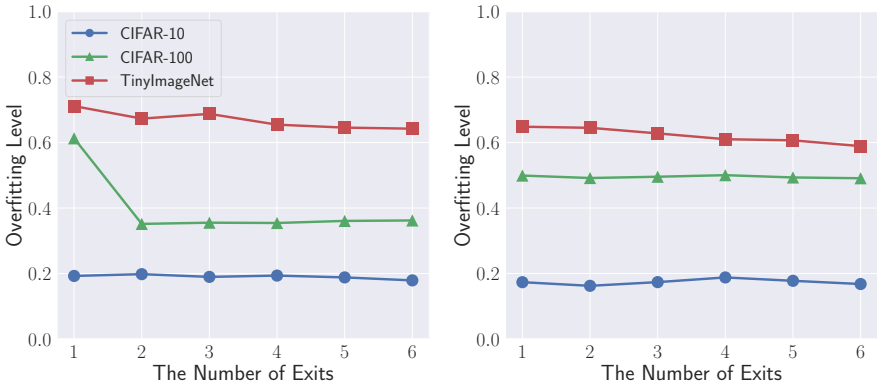


(c) Label-only

Figure 4.2: The attack performance of original membership inference attacks against vanilla and multi-exit models. Computer vision tasks are on VGG-16, and non-computer vision tasks are on FCN-18-1.

mean ASR on multi-exit VGG-16 is only 0.5914. Label-only ASR on vanilla FCN-18-1 trained on Locations is 0.8866, while the mean ASR on multi-exit VGG-16 is 0.7831. However, these results may lead to premature claims of privacy. Section 4.3 presents that the membership leakage risks stemming from our hybrid attack are much more severe than shown by existing attacks.

Overfitting Level. Here, we delve more deeply into the reasons for the less vulnerability of multi-exit models. As almost all previous works [116, 124, P1, 128, 78, 52, 146, 93] claim that the overfitting level is the main factor contributing to the vulnerability of the



(a) VGG-16 **(b)** WideResNet-32

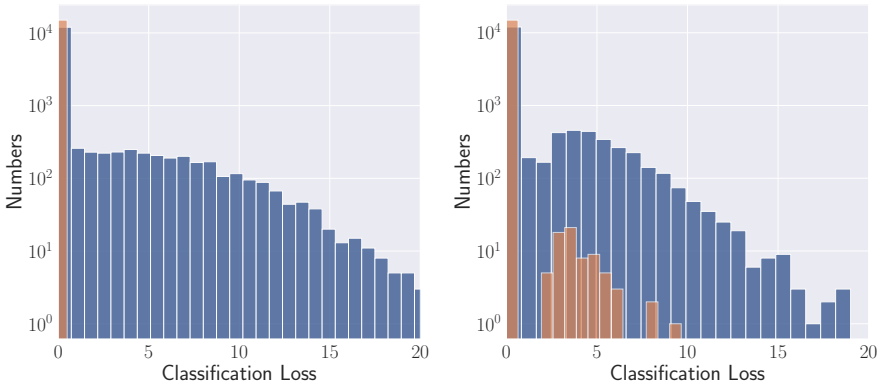
Figure 4.3: Comparison of overfitting levels between vanilla and multi-exit model. Note that 1 exit represents the vanilla model, and 2-6 exits represent different multi-exit models.

model to membership inference, i.e., a lower overfitting level leads to less vulnerability to membership inference. Here, we also relate this to the different overfitting levels of ML models. The overfitting level of a given model is measured by calculating the difference between its training accuracy and testing accuracy, i.e., subtracting testing accuracy from training accuracy, which is adopted by previous works. In Figure 4.3, however, we see that the overfitting level of multi-exit models remains almost the same compared to the vanilla model, especially in VGG-16 and WideResNet-32 trained CIFAR-10 dataset.

This observation which is contradictory to the previous conclusion inspires us to rethink the relationship between overfitting levels and vulnerability to membership inference. More precisely, we argue that the current calculation, i.e., subtracting test accuracy from training accuracy, is not the best way to characterize overfitting level, which leads to no strong correlation between overfitting level and vulnerability to membership inference, at least for multi-exit models.

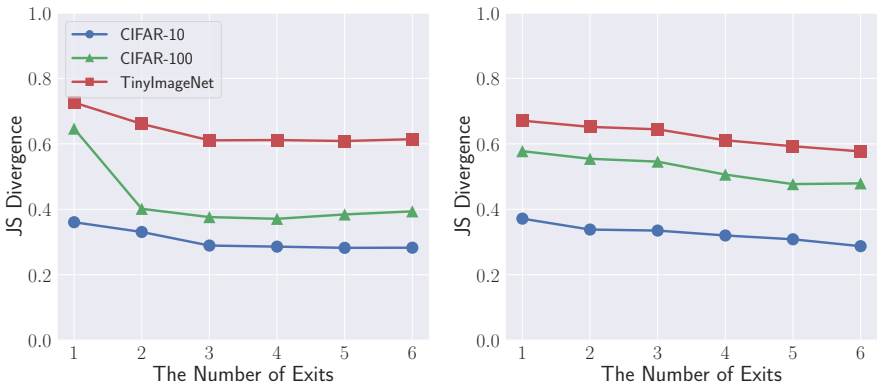
Loss Distribution. To find a more appropriate way to characterize the overfitting level and also to further investigate why the multi-exit model is less vulnerable to membership inference, we analyze the loss distribution between members and non-members in both vanilla and multi-exit models. Due to space limitations, we only show the results of VGG-16 trained on the CIFAR-10 dataset in Figure 4.4. A clear trend is that compared to the vanilla VGG-16, the multi-exit VGG-16 has a much lower divergence between the classification loss (cross-entropy) for members and non-members, especially the classification loss of members becomes larger. Note that in Figure 4.3a, the overfitting level calculated by subtracting test accuracy from training accuracy is almost the same between vanilla and multi-exit VGG-16 trained on CIFAR-10.

Based on the above observation, we believe that calculating the divergence between the loss distribution of members and non-members can better characterize the overfitting level. More concretely, we leverage Jensen-Shannon (denoted as JS) divergence, a widely used metric, to measure the distance of two probability distributions [45]. In Figure 4.5, we display JS divergence between the classification loss of members and non-members



(a) Vanilla **(b)** Multi-Exit

Figure 4.4: The loss distribution of original classification tasks for member and non-member samples between the vanilla VGG-16 and the 4-Exit VGG-16 on CIFAR-10.



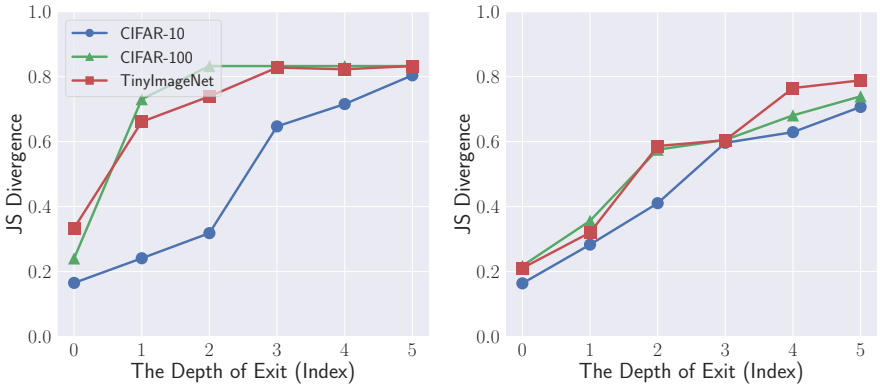
(a) VGG-16 **(b)** WideResNet-32

Figure 4.5: Comparison of JS divergence between vanilla and multi-exit models. Note that 1 exit represents the vanilla model and 2-6 exits represent different multi-exit models.

with respect to the number of exits. We can see that the JS divergence of multi-exit models is clearly lower than that of vanilla models. These results show that JS divergence is indeed a better way to characterize the overfitting level, compared to subtracting test accuracy from training accuracy.

Effects of the Number of Exits. We further investigate the effects of the number of exits attached to the backbone models. More interestingly, in Figure 4.5, we can also find the model with more number of exits leads to lower divergence. This indicates that the number of exits is negatively correlated to the vulnerability to membership leakages. The reason is that more exits attached to the backbone model mean that more data samples leave the earlier exit points than the final exit points, which makes the model less likely to be overfitted.

Effects of the Depth of Exits. Here we investigate the effects of the depth of exits attached to the backbone models. Given a backbone model with 6 exits, we use the exit



(a) VGG-16 (6 Exits) **(b)** WideResNet-32 (6 Exits)

Figure 4.6: The JS divergence of classification loss with respect to the depth of exits. The x-axis represents the depth of exit. The y-axis represents the JS divergence.

index (from 0 to 5) to represent the depth of exits. We calculate the JS divergence for members and non-members of each exit point separately. As shown in Figure 4.6, we can see that the JS divergence increases with the depth of exit. These results indicate that the depth of exits is positively correlated to the vulnerability to membership leakages, i.e., the samples leaving from deeper exit points were easier to distinguish between members and non-members. The reason for this observation is that deeper exit points imply higher capacity models, which are more likely to be overfitted to the training set.

4.3 Hybrid Membership Inference Attack with Exit Information (Adversary 1)

After quantifying membership leakages of multi-exit models, we conclude that the multi-exit models are less vulnerable to membership leakages, and, more interestingly, we find that exit information is highly correlated with attack performance. The latter motivates us to present a new research question: *Can extra exit information (number and depth) of the multi-exit model leak more membership information about the training set?*. Before answering the above research question, more importantly, we need to answer these two-step questions first:

- how to obtain the exit information of target multi-exit models, especially in black-box and label-only scenarios.
- how to leverage the exit information to improve existing membership inference attacks.

In the next section, we propose a novel hybrid attack that first steals exit information and then exploits the exit information as new knowledge for the adversary. In particular, we study three different scenarios for hybrid attacks by starting with some strong assumptions and gradually relaxing them to show that far more broadly applicable

attack scenarios are possible. Next, we describe our first adversary considered for leveraging exit information to mount membership inference attacks.

4.3.1 Threat Model

For this adversary, we mainly make a strong assumption about the adversary's knowledge. We assume that the adversary has direct access to exit information, i.e., exit depth. More concretely, given a data sample and a 6-exit model, the model outputs not only predictions (score or label) but also exit information, e.g., predictions from the first exit point (*exit 0*) or the sixth exit point (*exit 5*). Note that, here we directly consider the exit index as the exit depth. *exit 0* means the shortest path from the entry point to the first exit, while *exit 5* means the longest path from the entry point to the final exit.

In addition, we make the same assumptions for other settings, such as data knowledge, training knowledge, model knowledge, and output knowledge. For example, in the gradient-based attack, we keep the assumption unchanged that the adversary has access to the intermediate computations of the target model.

4.3.2 Attack Methodology

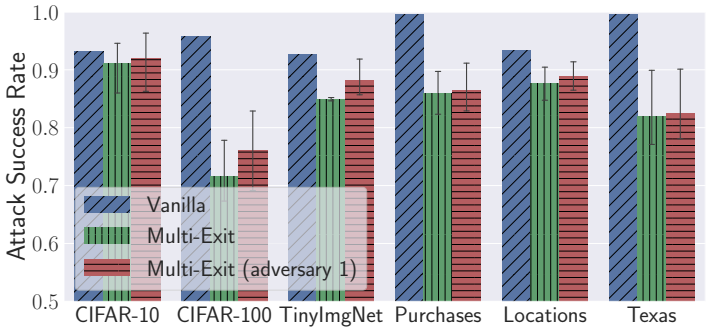
The attack methodology is organized into two stages: hyperparameter stealing and enhanced membership inference.

Hyperparameter Stealing. The adversary first queries the target model using a large number of data samples, which can come from the shadow dataset or random data samples collected from the Internet. They then count all exit indexes and sort them from smallest to largest. Thus, the largest index implies the number of exits attached to the backbone model.

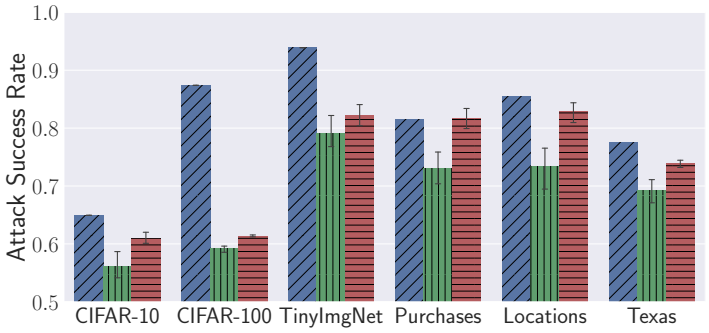
Enhanced Membership Inference. According to the two different types of attack models used in existing attacks, we propose different methods for each attack model to exploit the exit information.

- **MLP Attack Model.** In gradient-based and score-based attacks using MLP as the attack model, given the exit information (number and depth), the adversary first converts it to a one-hot encoding, which is the same as the one-hot encoding of the true label used in the gradient-based attack. They then provide the one-hot encoding of the exit information and other existing information to the attack model.

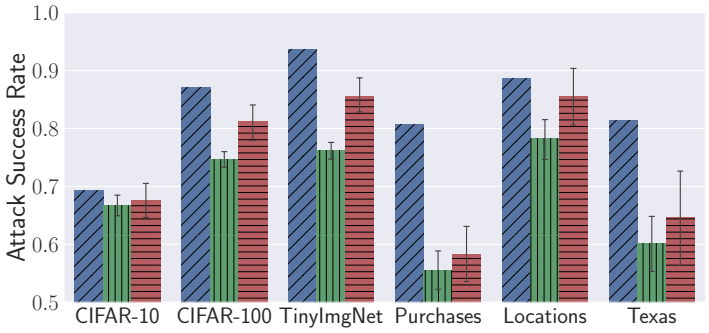
- **Decision Function.** In the original label-only attack, the adversary measures the magnitude of the perturbation and treats the data samples as members if their magnitude is larger than a predefined threshold. Here, instead of performing the above operation directly on all data samples, the adversary first separates the data samples according to their exit depths and then performs the above operation to distinguish members and non-members of each exit depth. The thresholds are also derived in this way on the shadow model. Note that label-only attack proceeds directly to the second stage without the hyperparameter stealing stage, because



(a) Gradient-based



(b) Score-based

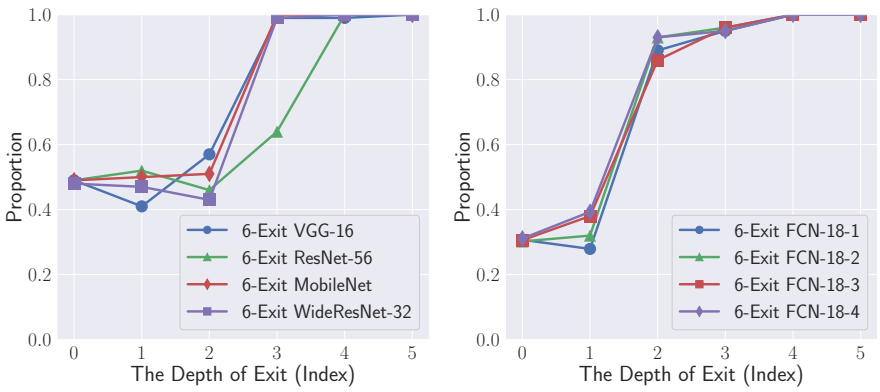


(c) Label-only

Figure 4.7: The attack performance of different membership inference attacks on all datasets. The blue and green bars indicate the original attack on the vanilla and multi-exit models, while the red bar indicates our hybrid attack on the multi-exit model. Computer vision tasks are on VGG-16, and non-computer vision tasks are on FCN-18-1.

the adversary does not need to generate one-hot encoding based on the exit depth and number.

In addition, all other attack steps are the same as those used in original attacks, such as shadow model training and attack training dataset building.



(a) CIFAR-10 **(b)** Purchases

Figure 4.8: Proportion of non-members in all samples leaving at each exit.

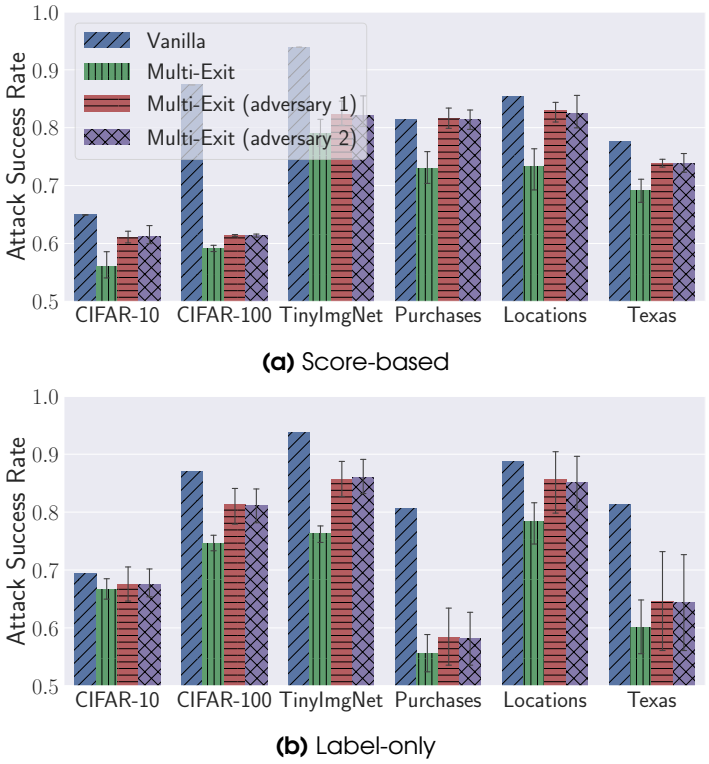
4.3.3 Evaluation

Experimental Setup. The adversary has six inputs and two inputs for the attack models in the gradient-based and score-based attacks, respectively, where the extra one is the one-hot encoding of the exit depths. Thus the new attack model has one more input component. For the evaluation metric, we again use the attack success rate (denoted as ASR). Note that in label-only attacks, we average ASR scores across all exit depths, as ASR is independent of exit depths. Besides that, we use the same experimental setup as presented in Section 4.2.3, such as the datasets, multi-exit model structures, and training settings.

Results. Figure 4.7 depicts the original and hybrid attacks' performance. See more results in our peer-review publication [P2]. Note that we also average the performance of multi-exit models with the number of exits varying from 2 to 6 and report the mean and standard deviation. Encouragingly, we can observe that hybrid attacks achieve clearly higher ASR scores than original attacks, regardless of datasets, architectures, and attack types. These results convincingly demonstrate that extra exit information of the multi-exit model leaks more membership information about the training set than the original information.

More interestingly, we also find that compared with gradient-based attacks, the extra exit information used in score-based and label-only attacks can significantly improve the performance of the original attacks. Recall that gradient-based attacks are applicable in white-box scenarios. This indicates that the original gradient-based attack has already exploited almost all the information and thus can achieve the attack performance close to the upper bound. Therefore, in gradient-based attacks, extra exit information can not lead to much higher attack performance gains. In contrast, we can observe that in label-only attacks, extra exit information leads to much higher attack performance gains.

The above results fully demonstrate the efficacy of our hybrid attack. Here, we delve more deeply into the reasons for the success. Our insight is that the exit depth is a critical indicator for membership inference. Figure 4.8 shows the proportion of



(a) Score-based

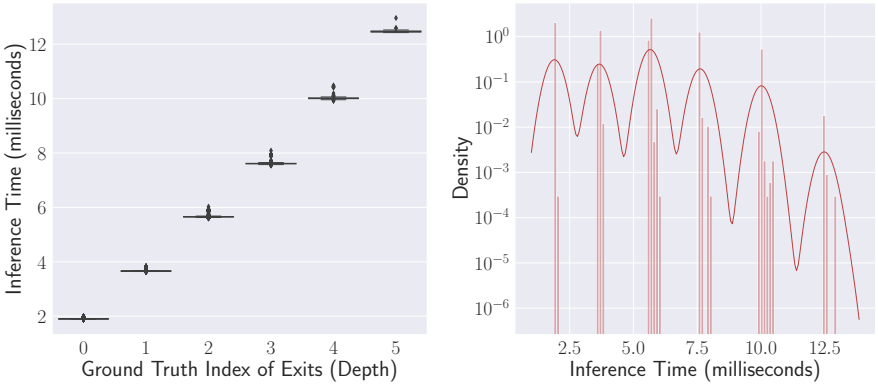
(b) Label-only

Figure 4.9: The attack performance of different membership inference attacks against vanilla and multi-exit models. The blue and green bars indicate the original attack on the vanilla and multi-exit models, while the red and purple bars indicate our hybrid attack on the multi-exit model. Computer vision tasks are on VGG-16, and non-computer vision tasks are on FCN-18-1.

non-members in all samples leaving at each exit. We can see that members tend to exit early, while non-members tend to exit late. In other words, the later exit depth itself indicates that the samples leaving here are likely to be non-members, in contrast to early exits where the samples are likely to be members. Recall that Figure 4.6 shows that the JS divergence of late exits is much larger than early exits, which further contributes to our hybrid attack. Such separability of members/non-members in terms of exit depth guarantees the efficacy of our hybrid attack.

4.4 Hybrid Membership Inference Attack without Exit Information (Adversary 2)

In this section, we relax the assumption that the adversary has direct access to exit information. We start by explaining the threat model, then describe the adversary's attack methodology. In the end, we present a comprehensive experimental evaluation.



(a) 6-Exit ResNet-56 **(b)** 6-Exit ResNet-56

Figure 4.10: The inference time with respect to the ground truth index of exit (a), and the density estimation by KDE based on inference time (b). They are both obtained from the same model, i.e., 6-exit ResNet-56 trained on CIFAR-100.

Table 4.1: The prediction accuracy of exit depths when we run 4 models simultaneously, each on a single GPU. We average the performance with the number of exits varying from 2 to 6 and report the mean and stand deviations.

Target Model	CIFAR-10	CIFAR-100	TinyImageNet
VGG	0.9998±2e-4	0.9999±1e-5	0.9999±1e-4
ResNet	0.9999±2e-5	1.0±0.0	1.0±0.0
MobileNet	0.9998±8e-5	1.0±0.0	0.9998±2e-4
WideResNet	0.9996±2e-7	0.9999±1e-5	0.9999±1e-5

4.4.1 Threat Model

Different from the threat model in Section 4.3, we remove the assumption that the adversary has direct access to exit information, i.e., exit depth. This largely reduces the attack capabilities of the adversary. Given a data sample, the multi-exit model gives a prediction that includes only the score or label and does not include any exit information. This is a more realistic but also more challenging scenario. Note that we only focus on score-based and label-only attacks, as in this scenario it is unlikely the adversary can obtain gradients or features of target models.

4.4.2 Attack Methodology

Recall that the goal of multi-exit models is to reduce computational costs by allowing data samples to be predicted and to exit at an early layer. Therefore, the inference time for data samples inevitably varies with the depth of the exit, i.e., data samples leaving deeper exit points imply longer inference times, as shown in Figure 4.10a. This renders us a new perspective to determine the exit information, i.e., the magnitude of inference time actually represents the different exit depths. We refer to this method as *time-based hyperparameter stealing*.

The adversary first queries the target multi-exit model using a large number of data

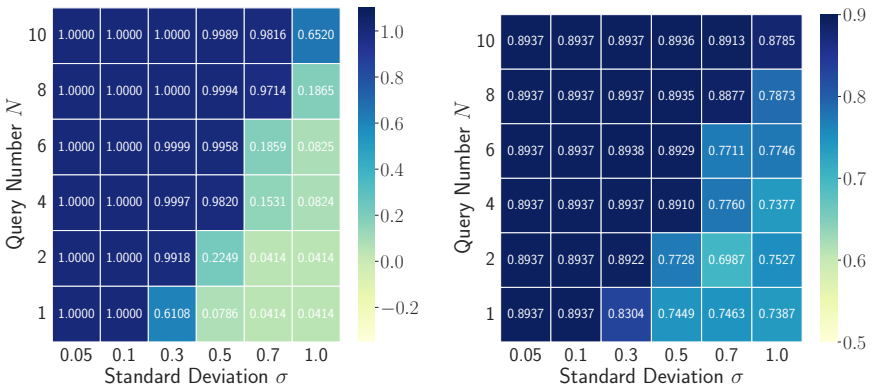
samples and records the inference time of these samples. These query samples can come from the shadow dataset, or random data collected from the Internet or any source. The adversary then sorts all recorded inference time as a one-dimensional array. Note that a longer inference time indicates a deeper exit point. Thus the adversary can partition this one-dimensional array into different clusters. Here we leverage Kernel Density Estimation (*KDE*)[111], an unsupervised statistical method for clustering one-dimensional data. Figure 4.10b shows a set of records of inference time, and we can see that *KDE* fits these records with a smoothed line. Then, several minima of the smoothed line can be used to partition them into different clusters. Thus the number of clusters means the number of exits attached to the target model, and the index of each cluster means the exit depth. The reason why we adopt *KDE* is that we want to cluster one-dimensional arrays (i.e., recorded time), for which *KDE* is well suited, while other popular techniques such as K-means [84], kNN [91] and DBSCAN [41] are multidimensional clustering algorithms.

4.4.3 Evaluation

Experimental Setup. We use all the same setups as presented in Section 4.2.3, such as the attack model design and evaluation metric. All experiments are conducted on an NVIDIA HGX-A100 server with 4-GPU deployed. We run 4 models simultaneously at a time, each on a single GPU. Practically, in order to get a stable inference time, we calculate the inference time by averaging the time of each sample 10 times.

Results. First, we report the prediction accuracy of exit depth overall datasets and model architectures in Table 4.1. As we can see, our proposed *time-based hyperparameter stealing* can achieve almost 100% accuracy. This indicates that the magnitude of inference time indeed can represent the exit depth, i.e., a longer inference time represents a deeper exit point and vice versa. Consequently, we can observe that our two adversaries achieve very similar performance for all datasets and model architectures in Figure 4.9. These results clearly demonstrate our hybrid attacks are very broadly applicable.

Next, we focus on the practicality of our hybrid attack against remotely deployed models, i.e., Machine Learning as a Service (MLaaS). This is a more challenging scenario where the communication channel can be very noisy. To simulate the complex communication channel, we assume that the noise z in the channel follows Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. More specifically, we first measure the clean inference time t, then sample the noise z from $\mathcal{N}(\mu, \sigma^2)$, and finally obtain the noisy inference time $t' = t + z(z > 0)$. Here, the $z > 0$ is to ensure that the noisy inference time t' is larger than the clean inference time t. To obtain a stable inference time, we propose a simple method that computes the inference time by averaging the noisy inference time 10 or more times, i.e., querying the remote model multiple times for each sample. Figure 4.11 shows the prediction and attack performance under the effect of variance σ of noise and query numbers for each sample. We can see that the highest prediction accuracy and ASR scores can be achieved if the number of queries is large enough, i.e., multiple queries can indeed eliminate the effect of noise. Furthermore, as shown in Figure 4.11a, we can find that even if the prediction accuracy of the exit depth drops by more than 30%, it still leads to high attack ASR scores.



(a) Exit Prediction Accuracy **(b)** ASR Score

Figure 4.11: The exit prediction and attack performance under the effect of query numbers N and standard deviation σ. The model is WideResNet-32 trained on TinyImageNet.

Furthermore, we delve more deeply into the lower bound of query numbers that can guarantee high attack performance. Consider the noise follows $\mathcal{N}(\mu, \sigma^2)$, and two clean inference time t_1 and t_2 from two adjacent exits *exit #1* and *exit #2*, respectively. Thus, the noisy inference time t' actually follows $\mathcal{N}(t+\mu, \sigma^2)$. The research question now is how many query numbers can guarantee the averaged noisy inference time $\bar{t'_1}$ and $\bar{t'_2}$ can be distinguished with high confidence. Here, we leverage Z-Test [9], a statistical technique, to determine whether two population means $\bar{t'_1}$ and $\bar{t'_2}$ are significantly different. To this end, we first query the target model with one certain sample many times (typically more than 100 times) to estimate the standard deviation σ. Then we calculate the Z-Score by the following formula:

$$Z = \frac{\bar{t'_1} - \bar{t'_2}}{\sqrt{\frac{\sigma^2}{n_1} + \frac{\sigma^2}{n_2}}} = \frac{(t_1 + \mu) - (t_2 + \mu)}{\sqrt{\frac{\sigma^2}{N} + \frac{\sigma^2}{N}}} = \frac{(t_1 - t_2)}{\sigma\sqrt{\frac{2}{N}}} \tag{4.1}$$

where n_1 and n_2 represent the query numbers for $\bar{t'_1}$ and $\bar{t'_2}$, and we consider the same query numbers N for all samples, i.e., $n_1 = n_2 = N$. Besides, as Figure 4.10 shows, we consider the minimal time difference (ms) between two adjacent exit $|t_1 - t_2| \in \{3, 5, 7, 9, 11\}$. To satisfy $p \leqslant 0.05$, i.e., the average noise inference time $\bar{t'_1}$ and $\bar{t'_2}$ can be distinguished with more than 95% confidence, we should ensure that $|Z| \geqslant 1.96$.[1] Thus we can derive the relationship between N and σ as shown in Figure 4.12. Given $|t_1 - t_2|$ and σ, the corresponding N denotes the lower bound of query numbers that can guarantee to divide two adjacent exits with more than 95% confidence. Recall that as shown in Figure 4.11a, even if the prediction accuracy of the exit depth drops by more than 30%, it still leads to a high attack ASR score, so the lower bound on the number of queries can also lead to high attack ASR score.

[1] https://pro.arcgis.com/en/pro-app/2.8/tool-reference/spatial-statistics/what-is-a-z-score-what-is-a-p-value.htm

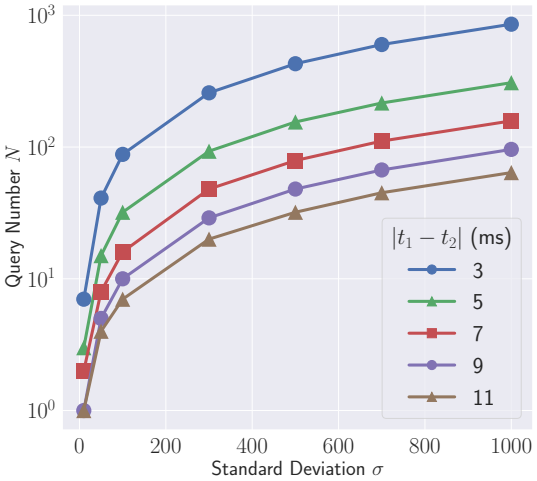


Figure 4.12: The relationship between query numbers N and standard deviation σ. The y-axis N denotes the lower bound of query numbers that can guarantee to divide two adjacent exits t_1 and t_2 with more than 95% confidence.

4.5 Model and Dataset Independent Hybrid Membership Inference Attack without Exit Information (Adversary 3)

Previous work [116, 124, P1, 128, 78, 52] has focused on the setup where the adversary trains a shadow model with the same architecture as the target model. Here, we have to ask *Does the same exit placement and the same architectural model lead to attack performance gains?*. Therefore, here we investigate whether the exit information still leaks more membership information when we relax this assumption. In addition, we also investigate the effect of the shadow dataset when we relax the assumption that the shadow dataset and target dataset are identically distributed. In the following, we start with the threat model description. Then, we list the attack methodology. In the end, we present the evaluation results.

4.5.1 Threat Model

To challenge our hybrid attack, we remove the assumption that the adversary can build a shadow model with the same architecture and exit placement as the target model, which largely reduces the attack capabilities of the adversary. In addition, we perform the evaluation of the gain of the exit to attack performance by relaxing the assumption that the shadow and target datasets are identically distributed.

4.5.2 Attack Methodology

The strategy of the third adversary is very similar to the second adversary. The only difference is that the third adversary uses a shadow model with a different architecture from the target model, which further inevitably leads to a different exit placement between shadow and target models. For example, given a target model ResNet-56 with

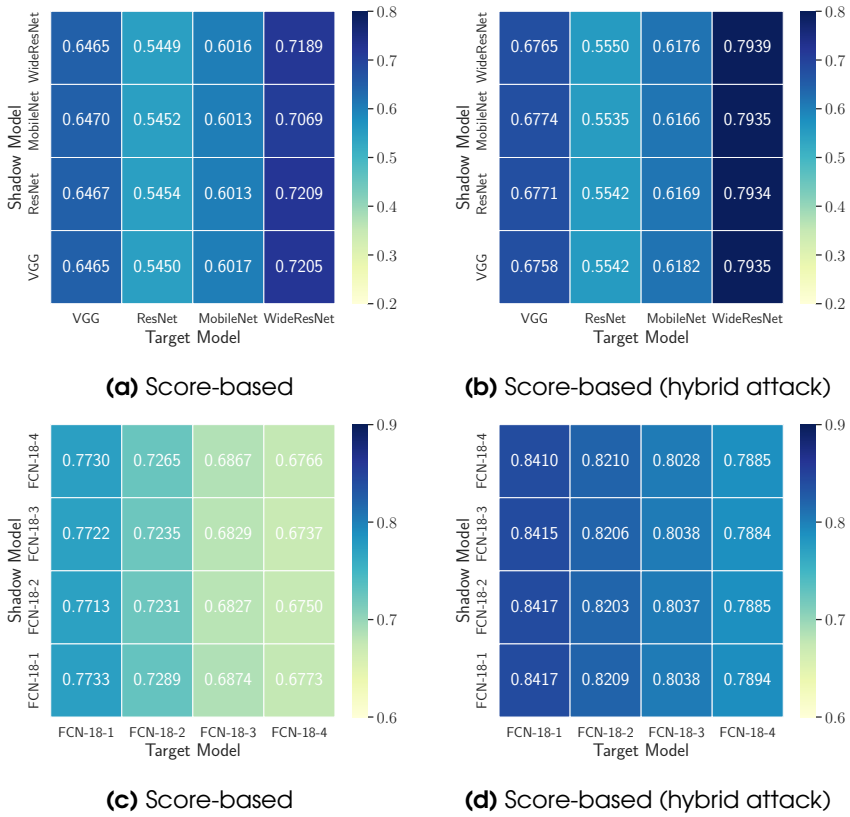


Figure 4.13: The attack performance when the shadow model has different architecture compared to the target model. These computer vision models (a and b) are trained on CIFAR-100, and these non-computer vision models (c and d) are trained on Purchases.

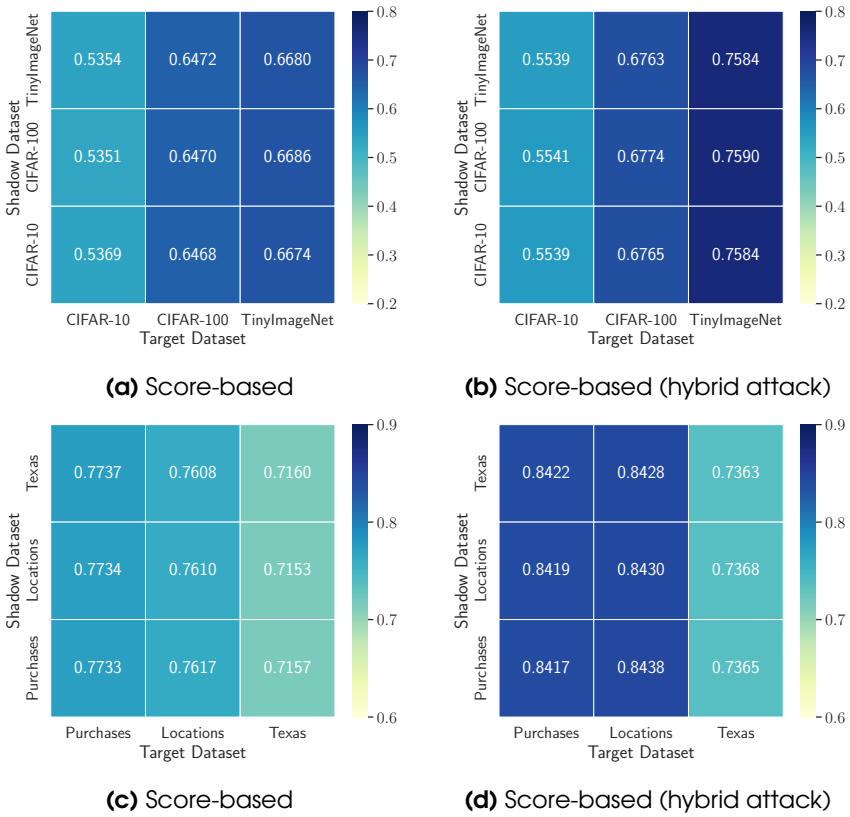
6 exits, the adversary can only train a different model, like VGG-16 with 6 exits, to perform membership inference. In this case, the placement of these 6 exits attached to the backbone model is different between ResNet-56 and VGG-16.

To relax the assumption on the same distribution between the shadow and target datasets, we use different datasets, e.g., CIFAR-10 as the target dataset and TinyImageNet as the shadow dataset, to launch our hybrid attack.

4.5.3 Evaluation

Experimental Setup. We use the same settings as described in Section 4.4.

Results. Figure 4.13 shows the attack performance when the shadow models are constructed by different architectures as the target models. First, we observe that the attack performance remains almost the same in both the original attack and hybrid attack, respectively. More encouragingly, we can also find that the attack performance of our hybrid attack is clearly higher than that of the original attack. For instance, when the target model is WideResNet-32 and the shadow model is VGG-16, the ASR score of our hybrid attack is 0.7935, while that of the original attack is only 0.7205.



(a) Score-based **(b)** Score-based (hybrid attack)

(c) Score-based **(d)** Score-based (hybrid attack)

Figure 4.14: The attack performance when the shadow dataset comes from different distributions of the target dataset. The computer vision model (a and b) we used is MobileNet, and the non-computer vision model (c and d) we used is FCN-18-1.

Such observation indicates that we can relax the assumption that the shadow model has the same model architecture and exit placement as the target model.

Furthermore, we also investigate whether we can relax another assumption of the same distribution between the shadow dataset and the target dataset. Figure 4.14 shows the attack performance when the shadow dataset is distributed differently from the target dataset. We observe that the performance of our hybrid attack is still better than the original attack even when the target and shadow datasets are different. Such observation hints that we can also relax the assumption of a same-distribution shadow dataset.

In conclusion, we show that adversary 3 can free the attacker from knowing the target model (especially exit placements) and target dataset, which further enlarges the scope of the hybrid attack. These results convincingly show that the corresponding risks are much more severe under the threats caused by our hybrid attack. Furthermore, the fact that privacy risks are much more severe shown by our hybrid attacks would hinder the process of green AI that aims at fast inference and energy-efficient computing.

4.6 Defense

In this section, we explore the possible defense and empirically conduct the evaluation. Recall that the adversary determines the exit depths by observing the different magnitude of inference time, thus the intuition of our defense is to hide the difference in inference time for different exit points. We name our defense *TimeGuard*.

TimeGuard. The key point is that the multi-exit networks delays giving predictions, rather than giving them immediately. One simple but naive defense mechanism is delaying giving predictions to the maximum inference time, i.e., a sample passes forward through all the layers of the model, acting like a vanilla model without any exit inserted. This behavior will make it impossible for the adversary to determine the exit information by observing inference time. However, this mechanism preserves privacy perfectly but is less efficient because it destroys one of the core ideas of multi-exit networks, which is to reduce the inference time for certain samples.

To achieve a better trade-off between privacy and efficiency, here we propose a novel mechanism for *TimeGuard* with high efficiency. See Figure 4.15 for an illustration of *TimeGuard* working on a 3-exit network. More concretely, consider the clean inference time t of one certain exit, thus the delay inference times of all the samples leave at this exit follow the right part of Gaussian distribution $\mathcal{N}(t, \sigma^2)$. See algorithm of *TimeGuard* in Algorithm 3.

Algorithm 3: *TimeGuard* with high efficiency.

Input: a data sample x, standard deviation σ, multi-exit model $\mathcal{M}$, a secret global seed S.

Output: delay time t' for x.

1 calculate hash h by Hash(x); `/* Hash(x) is Sha512 or ImageHash for non-images or images. */`

2 set random seed by random.seed(HKDF(h, S)); `/* the seed of Gaussian noise is secret */`

3 sample Gaussian noise I by random.normal(t, σ^2, size=1); `/* I is unique and repetitive for x */`

4 observe the exit depth where x leaves by feeding x to $\mathcal{M}$;

5 obtain clean inference time t of the exit;

6 calculate delay time $t' = t + |t - I|$;

7 return delay time t';

Here, we leverage ImageHash [10] or Sha512 [35] to calculate the unique hash h, and leverage HKDF [75] to generate the secret seed for Gaussian noise. In other words, we can obtain a fixed delay inference time t' for x since the hash h is unique (line 1) regardless of the number of queries. Thus, multiple queries on a data sample always give us the same delayed inference time, which is different from the scenario of noise variance reduction.

To further investigate the trade-off between privacy and efficiency under the influence of the standard deviation, we report attack performance and averaged inference time of each sample by varying the standard deviation in Figure 4.16. We can observe that as

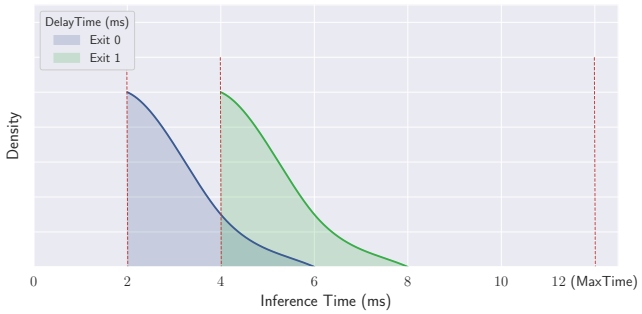
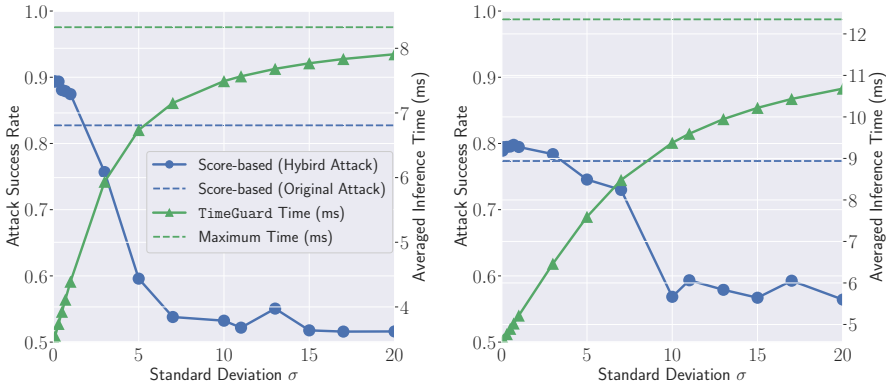


Figure 4.15: An illustration of how *TimeGuard* works on a multi-exit network with 3 exits. The y-axis represents the density of samples leaving at a certain delaytime among all samples at the same exit. These delaytimes follow the right part of Gaussian distribution $\mathcal{N}(t, \sigma^2)$.



(a) TinyImageNet, WRNet-32 **(b)** Purchases, FCN-18-1

Figure 4.16: The attack performance and *TimeGuard*'s efficiency under the effect of standard deviation used in *TimeGuard*. Here, WRNet means WideResNet.

the standard deviation increases, the ASR score decreases, while the averaged inference time increases. Since the ASR score of the original attack is the lower bound of the attack performance in both the original and hybrid attacks, the intersection of the two blue lines shown in Figure 4.16 is the best defense scenario for the model. In other words, the corresponding standard deviation is the optimal setting for *TimeGuard*, which not only reduces the ASR score to the lower bound but also maintains fast inference.

4.7 Conclusion

In this work, we take the first step to audit the privacy risk of multi-exit networks through the lens of membership inference. We conduct extensive experiments and find that multi-exit networks are less susceptible to membership leakage and that exits (number and depth) are highly correlated with attack performance. We further propose a hybrid attack to improve the performance of existing membership inference attacks

by using exit information as new adversary knowledge. We investigate three different adversarial settings for different adversary knowledge and end up with a model-free and data-free adversary, which shows that our hybrid attack is broadly applicable and thus the corresponding risk is much more severe than that shown by existing attacks. Finally, we present a simple but effective defense mechanism called *TimeGuard* and empirically evaluate its effectiveness.

5

Defending Against GAN-based Face Manipulation

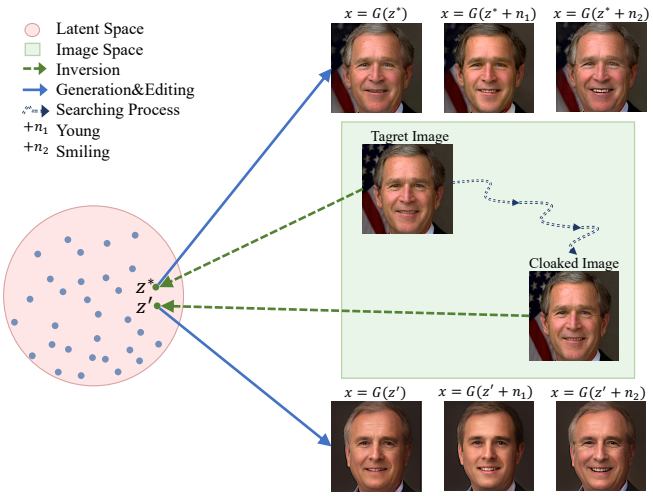


Figure 5.1: An illustration of GAN inversion and latent code manipulation, as well as a high-level overview of `UnGANable`.

5.1 Introduction

The rapid development and widespread use of machine learning have greatly contributed to the growth of artificial intelligence generated content (AIGC) [45, 71, 103, 105, 109]. By enabling the production of multimedia content of superior quality, AIGC has opened new avenues for innovative applications in various fields, such as the creative arts, advertising, filmmaking, and video games.

While AIGC is promising, it is also vulnerable to being exploited for data privacy violations [141, 37, 160, 122, 63, 48, 123, 149, 99, 31, 44, 32]. Adversaries can abuse AIGC to compromise data privacy, and the most representative privacy risk is visual misinformation through deepfake technology, which is based on machine learning and especially, generative models such as Generative Adversarial Networks (GANs) [45]. For instance, malicious editing of face images based on GAN-based face manipulation [141, 160, 122, 63, 48] can create false impressions, deceive people, or even trick biometric systems. Therefore, heavy concerns about such privacy risks are raised, and we believe that individuals need tools to protect their facial images from being abused by malicious manipulators.

To leverage GANs to manipulate facial images, the manipulator/adversary needs to perform a two-step operation. The first step is *GAN inversion* [159, 14, 15, 158, 21, 139] which inverts a victim's facial image to a latent code. The second step is *latent code manipulation* [141, 160, 122, 63, 48, 123, 149, 99, 31, 44] which manipulates the latent code to get the modified image, such as adding a pair of glasses on the victim's face. See Figure 5.1 for an illustration of the two-step operation.

5.1.1 Contributions

In this chapter, we propose the first defense system, namely `UnGANable`, against GANs-inversion-based face manipulation. In particular, `UnGANable` focuses on defending

against GAN inversion. Once an image is successfully inverted to its accurate latent code, it is extremely hard (if not possible) to defend the following manipulation step as the adversary can perform any operation on the latent code. Therefore, we believe the most effective defense is to reduce the performance of GAN inversion - the adversary can only obtain an inaccurate latent code that is far from the accurate one, thus the following latent code manipulation step will not achieve the ideal result. See Figure 5.1 for an illustration of our defense.

`UnGANable` searches for cloaked images in the image space which are indistinguishable from the target images but can cause the adversary's GAN inversion to obtain an inaccurate latent code. In this way, any individual can use `UnGANable` to protect their images by sharing only the cloaked images online. Further, we focus on two state-of-the-art GAN inversion techniques, i.e., optimization-based inversion [14, 15] and hybrid inversion [159, 158, 139], and consider five scenarios to characterize the defender's background knowledge along multiple dimensions. By considering what knowledge the defender has, we obtain a taxonomy of five different types of methods (called "cloaks" throughout the chapter) to disenable GAN inversion. More concretely, two cloaks are designed against optimization-based inversion, while the other three cloaks are designed against hybrid inversion.

We evaluate all our five cloaks on four popular GAN models that are constructed on two benchmark face datasets of different sizes and complexity. Extensive experiments show that `UnGANable` in general achieves remarkable performance with respect to both effectiveness and utility. We also conduct a comparison of our `UnGANable` with thirteen baseline image distortion methods. The results show that our defenses can outperform all these methods. We also explore four adaptive adversaries to bypass `UnGANable` and conduct sophisticated studies. Empirical results show that Spatial Smoothing [11] and more iterations of inversion are slightly effective.

In summary, we make the following contributions.

- We take the first step towards defending against malicious face manipulation by proposing `UnGANable`, a system that can jeopardize the process of GAN inversion.
- We consider five scenarios to comprehensively characterize a defender's background knowledge along multiple dimensions, and propose five different defenses for each scenario. Extensive evaluations on four popular GAN models show that `UnGANable` can achieve remarkable performance with respect to both effectiveness and utility.
- We conduct a comparison of our defenses with thirteen baseline image distortion methods. The results show that our defenses can outperform all these methods.
- We further explore four adaptive adversaries to bypass `UnGANable` and show that some of them are slightly effective.

5.1.2 Organization

The rest of this chapter is organized as follows. In Section 5.2, we present the overview of our defense. We present the defense methodologies and evaluations for optimization-

based inversion and hybrid inversion in Section 5.3 and Section 5.4, respectively. In Section 5.4, Next, we present the evaluation of our defenses on real images in Section 5.5 and the possible adaptive adversaries in Section 5.6. Finally, we discuss the limitation in Section 5.7 and conclude the chapter in Section 5.8.

5.2 Overview of Defense

In this section, we provide an overview of `UnGANable`.

5.2.1 Intuition

We derive the intuition behind our `UnGANable` from the basic pipeline of how inversion works. Since the optimization-based inversion is part of the hybrid inversion, here we focus only on the former. As described in Section 2.3, the inversion employs a loss function that is a weighted combination of the perceptual loss [68] and the pixel-wise MSE loss, to guide the optimization into the correct region of the latent space. This methodology leads to the following observations.

- The pixel-wise MSE loss works in the pixel space, i.e., the image space.
- The perceptual loss measures the similarity of features extracted from different images using a pretrained model, which works in the feature space.
- The optimization aims to search for the optimal latent code, which works in the latent space.

Thus, GAN inversion actually works in at least three spaces, i.e., the image space, the feature space, and the latent space. These observations motivate our `UnGANable`, which aims to maximize deviations in both latent and feature spaces with the cloaked images, meanwhile maintaining the image indistinguishable in the image space.

5.2.2 Threat Model

The goal of the face manipulator (i.e., adversary) is to manipulate the face without any authorization from the owner of the face image to serve its purposes, such as violating individual privacy or even misleading political opinions, The face manipulator could be a commercial company or even an individual. We assume the face manipulator has access to advanced GANs (e.g., via GitHub) and can apply two advanced GAN inversion techniques, namely optimization-based inversion and hybrid inversion, to invert the images into the latent space. These two inversion methods are shown in Figure 2.2.

5.2.3 System Model

Any user (also called defender) can use `UnGANable` to search for cloaked images, which are around the target images in the image space. The design goals for these cloaks are:

- cloaked images should be indistinguishable from the target images;

Table 5.1: An overview of assumptions. "✓" means the defender needs the knowledge and "-" indicates the knowledge is not necessary. "Target" means the adversary-controlled entities, and "Shadow" means the defender-built entities locally.

Inversion Category	Cloaks	Target Generator	Shadow Generator	Target Encoder	Shadow Encoder	Feature Extractor	Inversion Technique
Optimization-based	White-box	✓	-	-	✓	✓	✓
	Black-box	-	-	-	-	✓	-
Hybrid	White-box	-	-	✓	-	✓	-
	Gray-box	-	✓	-	✓	✓	-
	Black-box	-	-	-	-	✓	-

- when inverting the cloaked image, the adversary can only get a misleading latent code, which is far from its accurate one in the latent space (see Equation 5.1).

Generally, `UnGANable` aims to maximize the deviations in the latent space and feature space, while keeping the images indistinguishable in image space. Therefore, the challenge for `UnGANable` is to obtain the representation in each space. To this end, we make different assumptions for `UnGANable` in different scenarios where `UnGANable` can use different methods to search for invisible images. The overview of background knowledge is introduced in Table 5.1

5.3 Defending Against Optimization-based Inversion

In this section, we present `UnGANable` against the first type of GAN inversion, i.e., optimization-based inversion.

5.3.1 Defender's Knowledge

For optimization-based inversion, we consider two different scenarios to characterize a defender's background knowledge.

White-Box (Cloak v0). To maximize the deviation in the latent space, a defender has white-box access to the target generator G_{t}, and knows the adversary's inversion techniques I_{o}, thus he/she can obtain the accurate latent code of the original image. Besides, the defender trains a shadow encoder E_{s} to embed interim cloaked images to obtain the cloaked latent code. Then, the adversary can maximize the deviation between them. To maximize deviation in the feature space, we further assume that the defender has access to a feature extractor F, which can map both the original image and the cloaked image to feature space. Here, the feature extractor can be different from the feature extractor used in the perceptual loss.

Black-Box (Cloak v1). In this scenario, we assume the defender has no knowledge of the target generator or inversion techniques. Here, the defender only has access to a feature extractor F.

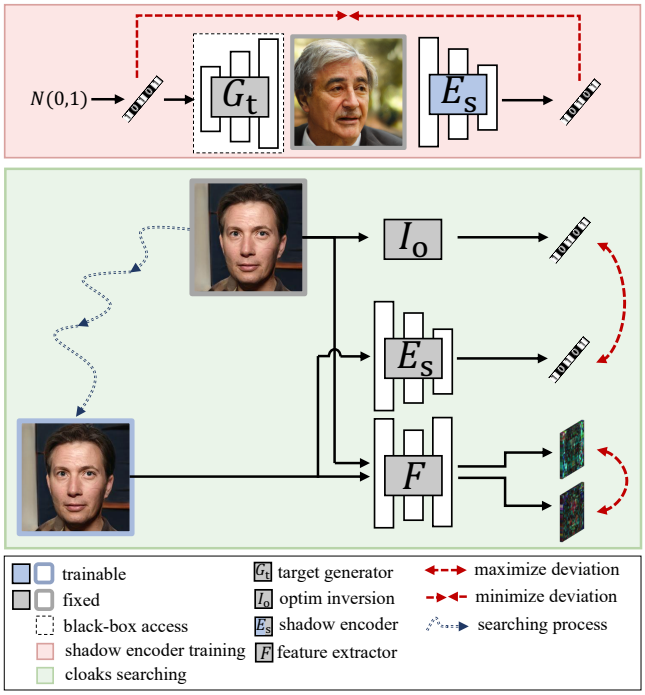


Figure 5.2: An illustration of white-box (Cloak v0) and black-box (Cloak v1) defenses against optimization-based inversion.

5.3.2 Methodologies

From a high-level overview, the defense can be divided into three simultaneous components, namely maximizing latent deviation, maximizing feature deviation, and searching for cloaked images in the image space. The algorithms are in Appendix Algorithm 4 and Algorithm 7.

White-Box (Cloak v0). The defender first leverages optimization-based inversion I_o to invert a target image $\mathbf{x}$ to obtain its exact latent code $I_o(\mathbf{x})$.[1] For maximizing latent deviation, the defender needs to build an end-to-end model, namely shadow encoder E_s, to invert the cloaked image $\hat{\mathbf{x}}$ of each step to obtain its latent code.[2] To train E_s, as shown in the pink part of Figure 5.2, the defender leverages the target generator G_t to create a dataset of generated images $G_t(\mathbf{z})$ and their latent codes $\mathbf{z}$, then minimize a similarity reconstruction loss $\mathcal{L}_{rec}$ between these latent codes $E_s(G_t(\mathbf{z}))$ and $\mathbf{z}$.

$$\mathcal{L}_{rec} = -\mathcal{L}_{cos}\Big(E_s(G_t(\mathbf{z})), \mathbf{z}\Big) + \mathcal{L}_{mse}\Big(E_s(G_t(\mathbf{z})), \mathbf{z}\Big) \tag{5.1}$$

where both $\mathcal{L}_{cos}$ and $\mathcal{L}_{mse}$ measure the element-wise similarity of latent codes. Here, $\mathcal{L}_{cos}$ is cosine similarity loss, and $\mathcal{L}_{mse}$ is MSE similarity loss.

[1] This process requires white-box access to the target generator G_t, as shown in Figure 2.2.

[2] The reason is that when iteratively searching in the image space, the defender needs to compute the cloaked image's gradient of each step with respect to the latent deviation by backpropagation, which is intractable through optimization-based inversion. The optimization-based inversion is just an inverted process, not an end-to-end model.

Table 5.2: Target GANs, datasets, and resolutions used to evaluate defense performance.

Model Zoo	Z dims	Dataset	Resolution
DCGAN (2016)[104]	100	CelebA [83]	64×64
WGAN (2017)[46]	128	CelebA [83]	128×128
StyleGANv1 (2019)[70]	512	FFHQ [70, 71]	256×256
StyleGANv2 (2020)[71]	512	FFHQ [70, 71]	256×256

For maximizing feature deviation, the defender uses a third-party pre-trained model (e.g., via GitHub) as the feature extractor F to obtain the features $F(\mathbf{x})$ and $F(\hat{\mathbf{x}})$. Once the defender obtains $I_{\text{o}}(\mathbf{x})$, E_{s} and F, the defender iteratively searches for $\hat{\mathbf{x}}$ in the image space by modifying $\mathbf{x}$, to maximize the latent and feature deviations between $\mathbf{x}$ and $\hat{\mathbf{x}}$.

$$\begin{gathered}\max_{\hat{\mathbf{x}}} \kappa\Big(\mathcal{L}_{\text{rec}}\big(E_{\text{s}}(\hat{\mathbf{x}}), I_{\text{o}}(\mathbf{x})\big)\Big) + (1-\kappa)\Big(\mathcal{L}_{\text{rec}}\big(F(\hat{\mathbf{x}}), F(\mathbf{x})\big)\Big) \\ \text{s.t. } |\hat{\mathbf{x}} - \mathbf{x}|_\infty < \epsilon \\ \kappa \in [0,1]\end{gathered}$$

where $\mathcal{L}_{\text{rec}}(.)$ introduced in Equation 5.1 measures the element-wise similarity of two feature vectors or latent vectors, $|\hat{\mathbf{x}} - \mathbf{x}|_\infty$ measures the distance between $\hat{\mathbf{x}}$ and $\mathbf{x}$, ε is the distance budget in image space, and κ is a trade-off hyper-parameter between latent and feature spaces.

Black-Box (Cloak v1). The defender can only produce significant alterations to images' feature space, i.e., searching for $\hat{\mathbf{x}}$ in the image space by modifying $\mathbf{x}$, to maximize the feature deviation between $\hat{\mathbf{x}}$ and $\mathbf{x}$.

$$\begin{gathered}\max_{\hat{\mathbf{x}}} \mathcal{L}_{\text{rec}}(F(\hat{\mathbf{x}}), F(\mathbf{x})) \\ \text{s.t. } |\hat{\mathbf{x}} - \mathbf{x}|_\infty < \epsilon\end{gathered}$$

5.3.3 Experimental Setup

GAN Models and Datasets. Without losing representativeness, we focus on four generative applications in recent years - DCGAN [104], WGAN [46], StyleGANv1 [70], and StyleGANv2 [71]. These GAN models are built with different architectures, losses and training schemes. Each generation application benchmarks its own dataset. As summarized in Table 5.2, we considered two benchmark datasets of different sizes and complexities, including CelebA [83] and FFHQ [70, 71], to construct different GAN models. Details of datasets can be found in Section 2.4.

Manipulator/Adversary. For face manipulator/adversary, we follow the original configurations of optimization-based inversion (Image2StyleGAN [14]). More specifically, we set up 500 iterations for the optimization step of inversion. In addition, we use perceptual loss and pixel-level MSE loss to reconstruct the target image in the optimization step. Though StyleGANv1 [70] and StyleGANv2 [71] also work on $\mathbf{w}$ space that is converted from $\mathbf{z}$ space, $\mathbf{z}$ space is applicable to all GAN models, thus we only consider $\mathbf{z}$ space in this work.

Defender. For the defender, we use a random initialized ResNet-18 [50] as the shadow encoder E_s in the white-box scenario (Cloak v0). Besides, for both white- and black-box scenarios (Cloak v0/v1), we adopt the easy-to-download, widely-used, and pre-trained ResNet-18 as the feature extractor. Further, we set up 500 iterations to iteratively search for the cloaked image in the image space by modifying the target image.

Target Samples. We first evaluate `UnGANable` on generated images from each GAN model. The reason is that, as stated in previous works [14, 15, 158], and also shown in our experimental results, the generated images are more easily inverted into accurate latent codes. In other words, in the competition between attackers and defenders, we actually make a very strong advantageous assumption for the former. We investigate whether `UnGANable` can achieve acceptable or superior performance in such a worst-case scenario. Thus, for each GAN model, we evaluate the performance of `UnGANable` on 500 randomly selected generated images that can be successfully reconstructed.

Evaluation Metrics. For evaluation metrics, we consider two perspectives: effectiveness and utility. Effectiveness measures the extent to which `UnGANable` jeopardizes the GAN inversion process. Given a target image, the sign of successful defense is a change in the identity of the reconstructed image, as shown in Figure 5.1. The reason is that once the identity of the reconstructed image changes, the defender no longer cares about the manipulation of the reconstructed image because the reconstructed image does not belong to the defender. To this end, we use *Matching Rate* to evaluate effectiveness:

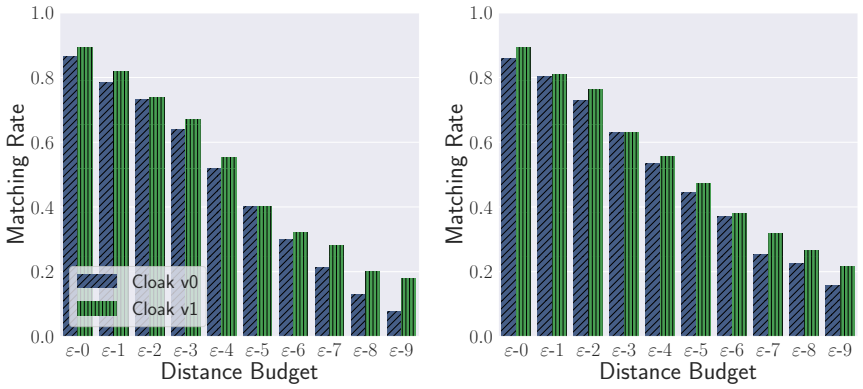
$$\textit{Matching Rate} = \frac{\#\text{successful reconstructed images}}{\#\text{total images}}$$

Therefore, the lower the matching rate is, which means the more reconstructed images with changed identity, the better effectiveness `UnGANable` achieves. In our implementation, we utilize a popular open-source face verification/comparison tool FaceNet [118] to compute the defense success rate. Given the embedding distance of a pair of two face images, a pre-calibrated threshold is used to determine the classification of *same* and *different*, i.e., the two face images belong to the same person if the embedding distance is less than the threshold, otherwise different person. See more details on threshold selection in our technical report [81].

Utility measures whether the cloaked images searched by `UnGANable` is indistinguishable from the target images. To measure the utility, we use a variety of most widely-used similarity metrics, including mean squared error (MSE), structural similarity (SSIM) [138], and peak signal-to-noise ratio (PSNR). Here, the lower the MSE is, the higher the SSIM and PSNR are, then the better utility `UnGANable` achieves. More details about these metrics are presented in our technical report [81].

5.3.4 Evaluation

Effectiveness Performance. In our `UnGANable`, we adopt a budget ε to limit distance between the cloaked and target image, aiming to ensure that the cloaked image is indistinguishable from the target image. Here, we first investigate the effectiveness



(a) StyleGANv1 **(b)** StyleGANv2

Figure 5.3: The effectiveness performance of Cloak v0/v1.

Table 5.3: Some visual examples of reconstructed images based on StyleGANv2. The defense method is Cloak v1.

Target Image	No cloak	ε-1	ε-3	ε-5	ε-7	ε-9
						
						

of `UnGANable` by reporting matching rate under the effects of the distance budget ε. More concretely, we set 10 different distance budgets ε-0, ε-1, ... , ε-9 (uniformly ranging from 0.01 to 0.07 for DCGAN and WGAN, and from 0.01 to 0.1 for StyleGANv1 and StyleGANv2.[3]). Under each distance budgets, we perform grid search to find the optimum trade-off hyper-parameter κ. The exact settings for ε and κ can be found in our technical report [81].

Figure 5.3 depicts the effectiveness performance of Cloak v0 and Cloak v1 (More results on DCGAN and WGAN in our technical report [81]). As we can see, with the increase of the budget ε, both Cloak v0 and Cloak v1 can significantly reduce the matching rate. For example, in Figure 5.3 (Cloak v0, StyleGANv2), the matching rate of ε-0 is 0.86, and that of ε-9 is 0.156, which drops sharply. These results imply that if we set a relatively high distance budget, `UnGANable` can achieve significant effectiveness against optimization-based inversion.

Besides the above quantitative results, we further provide random qualitative examples to demonstrate the effectiveness of `UnGANable` performed on StyleGANv2. As

[3]We conducted a pre-experiment and showed that only a small distance can jeopardize DCGAN and WGAN inversions, so we set the maximum magnitude of the distance budget to 0.07 for DCGAN and WGAN, and 0.1 for StyleGANv1/v2.

Table 5.4: Some visual examples of cloaked images searched by Cloak v1 performed on StyleGANv2 under different perturbation budgets.

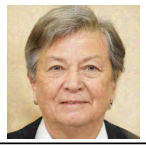
Target Image	ε-1	ε-5	ε-9
			
			

Table 5.5: The utility performance of `UnGANable` against optimization-based inversion.

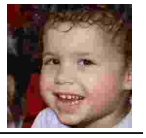
Budget	Metric	Cloak v0	Cloak v1	Budget	Metric	Cloak v0	Cloak v1
ε-1	MSE	7.3e-05	7.2e-05	ε-7	MSE	0.0010	0.0014
	SSIM	0.9889	0.9891		SSIM	0.8802	0.8431
	PSNR	41.376	41.408		PSNR	30.118	28.532
ε-3	MSE	0.0003	0.0003	ε-9	MSE	0.0014	0.0022
	SSIM	0.9612	0.962		SSIM	0.8347	0.7820
	PSNR	35.684	35.716		PSNR	28.423	26.637
ε-5	MSE	0.0006	0.0006				
	SSIM	0.9228	0.9245				
	PSNR	32.419	32.455				

shown in Table 5.3, we can observe that as ε increases, more and more facial attributes cannot be successfully reconstructed. The difference between the reconstructed image and the target image becomes more extensive, which implies the effectiveness is getting better.

Utility Performance. To evaluate the utility performance, we first quantitatively report a variety of similarity metrics (MSE/SSIM/PSNR) in Table 5.5. Typically, a SSIM value greater than 0.9 or a PSNR greater than 35 means a good quality of cloaked images. To elaborate more on utility performance, we show in Table 5.4 some qualitative samples of cloaked images searched by `UnGANable` performed on StyleGANv2. We can observe that when distance budget is set as ε-1 (0.02) and ε-3 (0.04), which represents a completely imperceptible perturbation, `UnGANable` can achieve acceptable effectiveness performance (see qualitative reconstructed examples in Table 5.3). In addition, we acknowledge that some perturbations are perceptible to our naked eye when the distance budget is set to ε-7 (0.08) or ε-9 (0.1). But note that these visual results are performed on the images generated by their corresponding GAN models. In the following Section 5.5, we further conduct experiments on real images. It is encouraging that `UnGANable` can apply a much lower distance budget to obtain excellent effectiveness performance while guaranteeing the visual quality of the cloaked image.

The Effect of Latent/Feature Deviation. We further investigate the effect of latent/feature deviation on the performance of `UnGANable`. In the white-box scenario

Table 5.6: Visual examples of different baseline distortion methods.

Target Image	ShearX	ShearY	TranslateX	TranslateY	Rotate	Brightness
						
Color	**Contrast**	**Solarize**	**CenterCrop**	**GaussianBlur**	**GaussianNoise**	**Compress**
						

(Cloak v0), `UnGANable` searches for the cloaked images which can maximize both latent and feature deviations, while in the black-box scenario (Cloak v1) only feature deviations are maximized. As shown in Figure 5.3, we can obverse that Cloak v0 achieves better effectiveness performance than Cloak v1 under each distance budget. However, we cannot prematurely claim that Cloak v0 is better because we need to consider whether Cloak v0 is at least as good as Cloak v1 in terms of utility performance. Table 5.5 reports the utility performance of `UnGANable` on the StyleGANv2. First, we can observe that Cloak v0 performs at least on par with Cloak v1 under budget ε-1, ε-3, and ε-5. More encouragingly, under budget ε-7 and ε-9, Cloak v0 achieves better utility performance than Cloak v1. These results show that Cloak v0 outperforms Cloak v1 in terms of both effectiveness and utility, and convincingly demonstrate that the additional latent deviation we introduce for Cloak v0 does improve performance.

Comparison with Baselines. To elaborate on `UnGANable`'s performance in a more convincing manner, we compare `UnGANable` extensively with thirteen baseline distortion methods, as shown in Table 5.6. For each baseline method, we evaluate both effectiveness and utility performance with a wide variety of different magnitudes of the budget. More detailed descriptions of each method are presented in our technical report [81]. Figure 5.4 displays the relationship between each baseline method's matching rate and MSE/SSIM/PSNR score (see more results in our technical report [81]). Thus, we can make the following observations.

First, as the budget increases (i.e., MSE becomes larger and SSIM/PSNR becomes smaller), all baseline methods can significantly reduce the matching rate, meaning that baseline methods that work only in image space can also achieve good effectiveness performance.

More encouragingly, the plot also clearly indicates the benefits of latent and feature deviations: among baseline methods with similar utility performance levels (similar MSE/SSIM/PSNR), our Cloak v0 and Cloak v1 consistently achieve better effectiveness (lower matching rate), as they benefit from maximizing latent and feature deviations. In other words, searching for cloaked images to maximize latent and feature deviations can further disenable GAN inversions at nearly no cost in utility. Another interesting finding is that when `UnGANable` is not an option, *GaussianNoise*, *GaussianBulr*, and *JPEGCompression* appear to perform better.

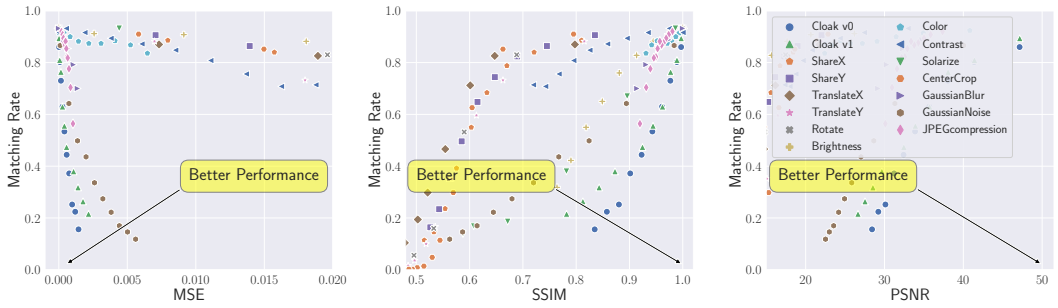


Figure 5.4: Comparison between all baseline methods and Cloak v0/v1 on generated images and StyleGANv2. The different points of each method represent different budgets.

5.4 Defending Against Hybrid Inversion

We now present `UnGANable` against the second GAN inversion technique, i.e., hybrid inversion.

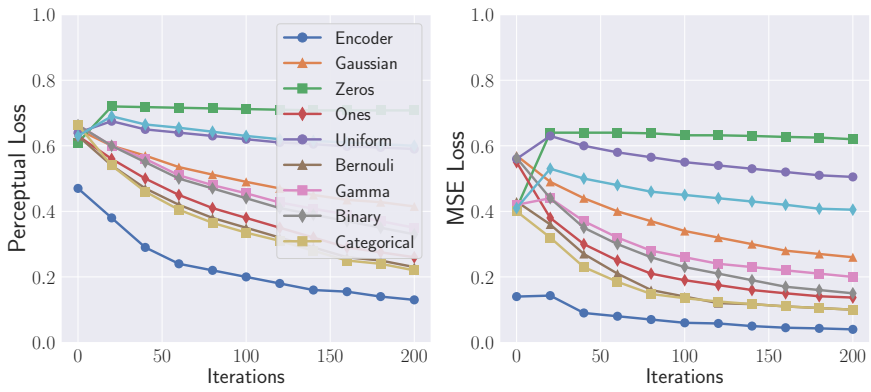
5.4.1 Defender's Knowledge

For hybrid inversion, we consider three different scenarios to characterize a defender's background knowledge. The algorithms are in Appendix Algorithm 5, Algorithm 6 and Algorithm 7.

White-Box (Cloak v2). Hybrid inversion adopts an encoder to provide a better initialization $\mathbf{z}$ for the following optimization step. Here, we assume that a defender has complete knowledge of the target encoder E_{t} to mislead the encoder, i.e., provide a worse initialization latent code $\mathbf{z}$ for the optimization. We give a quantitative illustration of this intuition in Section 5.4.2. Besides that, we also assume that the defender has access to a feature extractor F. Note that the defender does not need to have white-box access to the target generator G_{t} due to the design of this defense (see Section 5.4.2 for more details).

Grey-Box (Cloak v3). Here, we relax the assumption that the defender has complete knowledge of the target encoder E_{t}. In particular, we assume that the defender can send many queries to the target encoder E_{t} and train a shadow encoder E_{s} to mimic the behavior of the target encoder E_{t}, and relies on the shadow encoder to act as the target encoder. Besides that, we assume that the defender has access to a feature extractor F for feature deviation.

Black-Box (Cloak v4). Here, we assume the defender has no knowledge of the adversary's generator or encoder. Here, the defender only has access to a feature extractor F.



(a) Perceptual Loss **(b)** MSE Loss

Figure 5.5: The loss trend under the effect of different initialization for optimization.

5.4.2 Methodologies

Here the defenses are also divided into three simultaneous components, namely maximizing latent deviation, maximizing feature deviation, and searching for cloaked images in the image space. In particular, we introduce a new novel method to maximize the latent deviation.

New Perspective of Latent Deviation. As aforementioned in Section 2.3, an important issue for optimization-based inversion is initialization. Recent research [69, 24, 70, 104] shows that using different initializations leads to a significant perceptual difference in generated images. Here, we conduct a pre-experiment on using different initializations to perform the optimization-based inversion, including Gaussian, zeros, etc (see [12] for each distribution). In particular, hybrid inversion adopts an encoder to provide initialization for optimization.

Figure 5.5 shows the trend of perceptual and MSE loss, respectively. First, the encoder indeed provides better initialization, which leads to better initial and final performance. Second, the trend of loss remains constant when the initialization is set to zero, which means it is quite difficult to invert the target image into the latent space. This observation suggests a new perspective on the latent deviation – misleading the encoder to provide zero initialization, or close to zero. In other words, our defense's goal against hybrid inversion should be to force the output of the encoder to zero. This is actually a special case of maximizing latent deviation, which provides the movement direction of the cloaked image in the latent space, i.e., towards zero.

White-Box (Cloak v2). In this scenario, we assume that the defender has full knowledge of the target encoder E_t, as well as an additional feature extractor F. As shown in the green part of Figure 5.6, the defender iteratively searches for $\hat{\mathbf{x}}$ in the image space by modifying $\mathbf{x}$, in order to minimize the deviation between $E_t(\hat{\mathbf{x}})$ and zero,

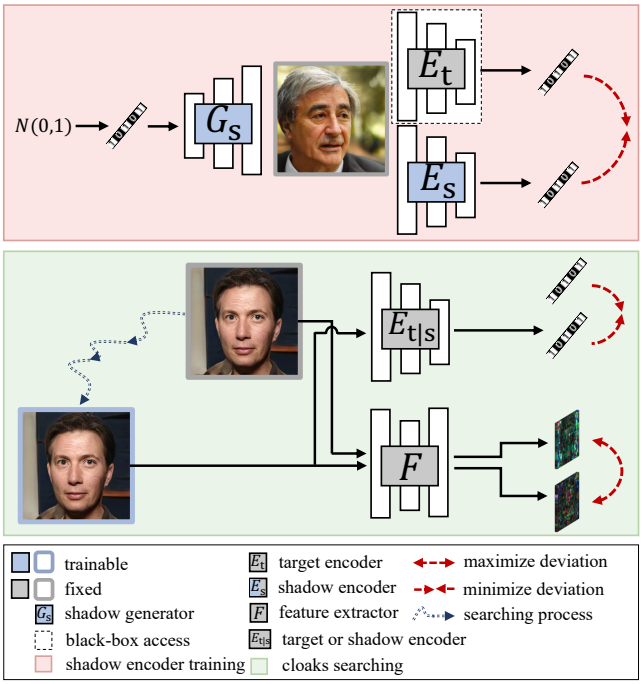


Figure 5.6: An illustration of white-box (Cloak v2), grey-box (Cloak v3) and black-box (Cloak v4) `UnGANable` against hybrid inversion.

and maximize the deviation between $F(\hat{\mathbf{x}})$ and $F(\mathbf{x})$.

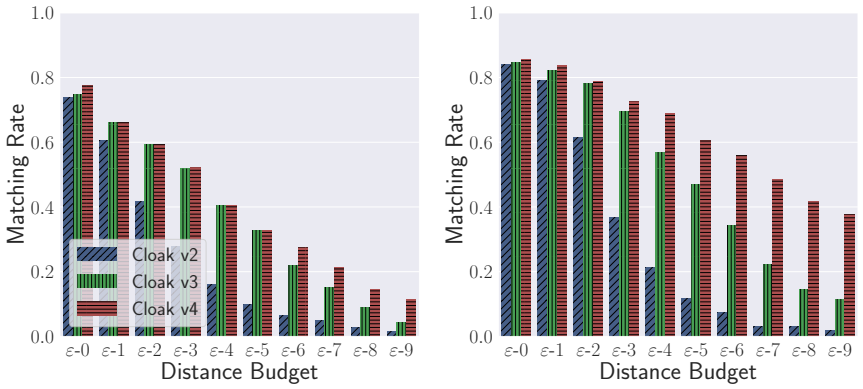
$$\begin{gathered}\max_{\hat{\mathbf{x}}} \kappa\Big(-\mathcal{L}_{\text{rec}}(E_{\text{t}}(\hat{\mathbf{x}}), 0)\Big) + (1-\kappa)\Big(\mathcal{L}_{\text{rec}}(F(\hat{\mathbf{x}}), F(\mathbf{x}))\Big) \\ \text{s.t. } |\hat{\mathbf{x}} - \mathbf{x}|_\infty < \epsilon \\ \kappa \in [0,1]\end{gathered}$$

Grey-Box (Cloak v3). Here, we relax the assumption that the defender has complete knowledge of the target encoder E_{t}. The defender needs to build a shadow encoder E_{s} to match the predictions of E_{t}, i.e., find the shadow encoder's parameters that minimize the probability of errors between the shadow and target predictions.

As shown in the pink part of Figure 5.6, the defender builds a shadow generator G_{s} which is responsible for crafting some input images, and E_{s} serves as a discriminator while being trained to match the target encoder's predictions on these images. In this setting, the two adversaries are E_{s} and G_{s}, which try to minimize and maximize the deviation between E_{s} and E_{t} respectively. Then, shadow encoder E_{s} becomes a functionally equivalent copy of target encoder E_{t}.

Finally, the defender iteratively searches for $\hat{\mathbf{x}}$ in the image space by modifying $\mathbf{x}$, in order to minimize the deviation between $E_{\text{s}}(\hat{\mathbf{x}})$ and zero and maximize the deviation between $F(\hat{\mathbf{x}})$ and $F(\mathbf{x})$.

$$\begin{gathered}\max_{\hat{\mathbf{x}}} \kappa\Big(-\mathcal{L}_{\text{rec}}(E_{\text{s}}(\hat{\mathbf{x}}), 0)\Big) + (1-\kappa)\Big(\mathcal{L}_{\text{rec}}(F(\hat{\mathbf{x}}), F(\mathbf{x}))\Big) \\ \text{s.t. } |\hat{\mathbf{x}} - \mathbf{x}|_\infty < \epsilon \\ \kappa \in [0,1]\end{gathered}$$



(a) StyleGANv1 **(b)** StyleGANv2

Figure 5.7: The effectiveness performance of Cloak v2, Cloak v3 and Cloak v4.

Black-Box (Cloak v4). In this scenario, the defender has no knowledge of the target generator or target encoder or inversion techniques. The defender can only search for $\hat{\mathbf{x}}$ in the image space by modifying $\mathbf{x}$, to maximize the feature deviation between $\hat{\mathbf{x}}$ and $\mathbf{x}$.

$$\begin{aligned} &\max_{\hat{\mathbf{x}}} \mathcal{L}_{\text{rec}}(F(\hat{\mathbf{x}}), F(\mathbf{x})) \\ &\text{s.t. } |\hat{\mathbf{x}} - \mathbf{x}|_\infty < \epsilon \end{aligned}$$

5.4.3 Experimental Setup

For the manipulator/adversary, we follow the configurations of hybrid inversion (Zhu et al. [158]). Here, we again only consider the $\mathbf{z}$ space for all GAN models. We set up 100 iterations for the optimization step of inversion, and use perceptual loss and pixel-level MSE loss to reconstruct the target image in the optimization step.

As a defender, for Cloak v3, we build the shadow generator by using 1 linear layer to accept Gaussian noise, followed by five convolutional layers and five Batch Normalization [61] layers. Furthermore, we again use a random initialized ResNet-18 as the shadow encoder. For all Cloaks (v2/v3/v4), we again use a pretrained ResNet-18 [50] as the feature extractor. Besides, we fix the number of iterations as 500, to search for cloaked images. In addition, all other experimental settings are the same as described in Section 5.3.3.

5.4.4 Evaluation

Effectiveness Performance. To evaluate the effectiveness performance quantitatively, we use the same evaluation setup as presented in Section 5.3.4. Figure 5.7 depicts the effectiveness performance of Cloak v2/v3/v4. First, we again observe that as the budget increases, all Cloak v2/v3/v4 can significantly reduce the matching rate. These results indeed imply that UnGANable can achieve significant effectiveness against hybrid inversion. For qualitative results, the same perturbation budget will lead to similar reconstructed results, as shown in Table 5.3.

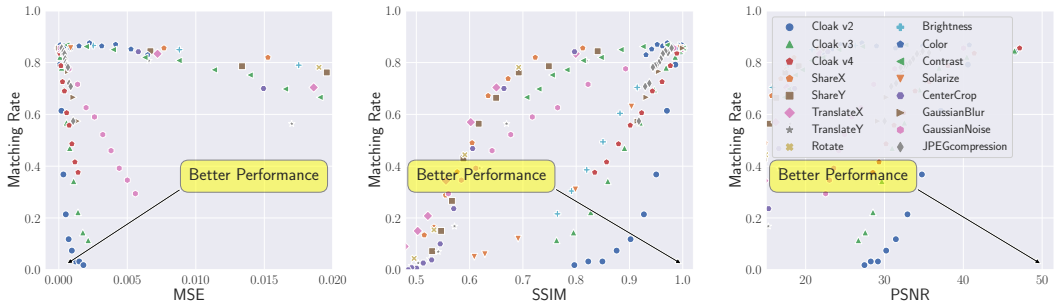


Figure 5.8: Comparison between all baseline methods and Cloak v2/v3/v4 on generated images and StyleGANv2. The different points of each method represent different budgets.

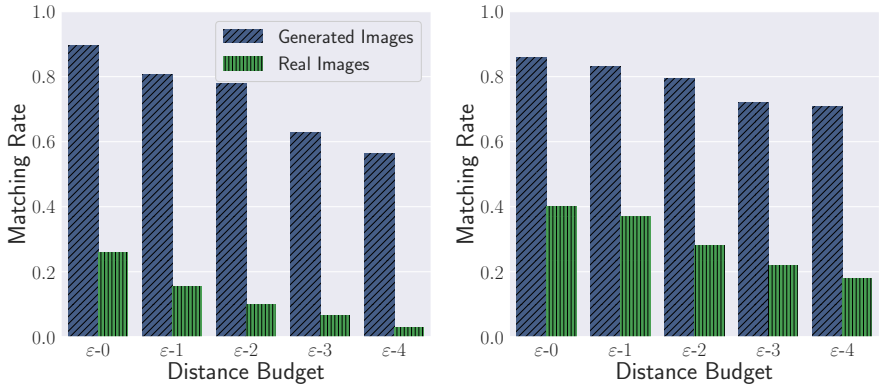
Utility Performance. Similarly, since we set the same distance budgets as adopted against optimization-based inversion, thus for the same perturbation budget will lead to similar quantitative and qualitative utility performance, as shown in Table 5.3 and Table 5.5.

The Effect of Latent/Feature Deviation. In Figure 5.7a and Figure 5.7b, we can observe that searching for cloaked images to mislead the target encoder controlled by the adversary (Cloak v2) leads to much better effectiveness performance. Furthermore, the larger the distance budget, the larger the gap between Cloak v2 and both Cloak v3 and Cloak v4, reflecting the fact that zero initialization can significantly jeopardize the process of GAN inversion. This convincingly verifies our new perspective of latent deviation–misleading the adversary's encoder to provide zero initialization, or close to zero.

Comparison with Baselines. We compare `UnGANable` extensively with thirteen baseline methods, as shown in Table 5.6. We use the same experimental setup as described in Section 5.3.4, such as the perturbation budget setting strategy and the result reporting metrics. We report comparisons between baseline methods and `UnGANable` in Figure 5.8, and we can make similar observations as mentioned in Section 5.3.4. Here, we again emphasize that Cloak v2/v3/v4 achieves consistently better effectiveness (lower matching rate) and utility (lower MSE, higher SSIM and PSNR) performance than all baselines.

5.5 Evaluation on Real Images

To elaborate on `UnGANable`'s performance, here we investigate the performance of `UnGANable` on real facial images. Concretely, we consider the strictest setting in which the defender has no knowledge of the adversary-controlled entities. Thus, we only consider the black-box scenario against optimization-based and hybrid inversion, i.e., Cloak v1 and Cloak v4. In addition, the adversary-controlled GAN model is the state-of-the-art deepfake generative model StyleGANv2. We collect 200 real images from the FFHQ dataset, and these images are the most successfully inverted into the latent space among the whole FFHQ dataset.



(a) Optimization-based **(b)** Hybrid

Figure 5.9: The effectiveness performance of Cloak v1/v4 on generated and real images, respectively.

Effectiveness Performance. We first present the effectiveness performance of `UnGANable`. We use the same evaluation setup as presented in Section 5.3.4. We set 5 different distance budgets ε-0/1/2/3/4, the same as adopted in previous evaluations. Figure 5.9 depicts the effectiveness performance of Cloak v1 and Cloak v4. First, we again observe that as the budget ε increases, both Cloak v1 and Cloak v4 can significantly reduce the matching rate. Then we can see that the matching rate of Cloak v4 is clearly higher than that of Cloak v1, which verifies that the encoder of hybrid inversion indeed leads to better reconstruction performance.

What is more encouraging is that `UnGANable` can achieve better effectiveness performance compared to that on generated images. For example, when the distance budget is set as ε-4 (0.05), the matching rate of Cloak v1/v4 on the real image is about 0.072/0.191, while that on the generated image is about 0.474/0.606. The results clearly show that `UnGANable` can apply a much lower perturbation budget to obtain better effectiveness performance, and this lower distance budget further benefits utility performance.

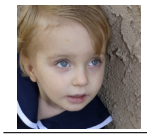
Utility Performance. For utility performance, we conduct the evaluations both quantitatively and qualitatively. We first quantitatively report various similarity metrics (MSE/SSIM/PSNR) in Table 5.7. Generally, SSIM values of 0.97, 0.98, and 0.99 imply the excellent visual quality of the cloaked images. We then show in Table 5.8 some qualitative samples of cloaked images. We can observe that when the distance budget is set as ε-4 (0.05), which represents a completely imperceptible perturbation, `UnGANable` can achieve remarkable effectiveness performance (see matching rate in Figure 5.9b). Therefore, we claim that `UnGANable` provides acceptable protection for real images by much lower distance budgets and still yields good effectiveness and utility performance.

Comparison with Baselines. We then compare `UnGANable` extensively with thirteen baseline distortion methods, as shown in Table 5.6. For each baseline method, we evaluate both effectiveness and utility performance with a wide variety of different magnitudes of the budget. Figure 5.11 displays the compassion between baseline

Table 5.7: The quantitative utility performance of `UnGANable` under Cloak v1/v4.

Budget	Metric	Cloak v1	Cloak v4	Budget	Metric	Cloak v1	Cloak v4
ε-0	MSE	1.9e-05	1.9e-05	ε-3	MSE	0.0003	0.0002
	SSIM	0.9968	0.9969		SSIM	0.9606	0.967
	PSNR	47.205	47.210		PSNR	35.783	35.783
ε-1	MSE	7.1e-05	7.2e-05	ε-4	MSE	0.0004	0.0004
	SSIM	0.9887	0.9887		SSIM	0.9422	0.9423
	PSNR	41.473	41.473		PSNR	33.983	33.982
ε-2	MSE	0.0002	0.0002				
	SSIM	0.9764	0.9764				
	PSNR	38.144	38.145				

Table 5.8: Some visual examples of cloaked real images searched by Cloak v4 performed on StyleGANv2.

Target Image	ε-0	ε-2	ε-4
			
			

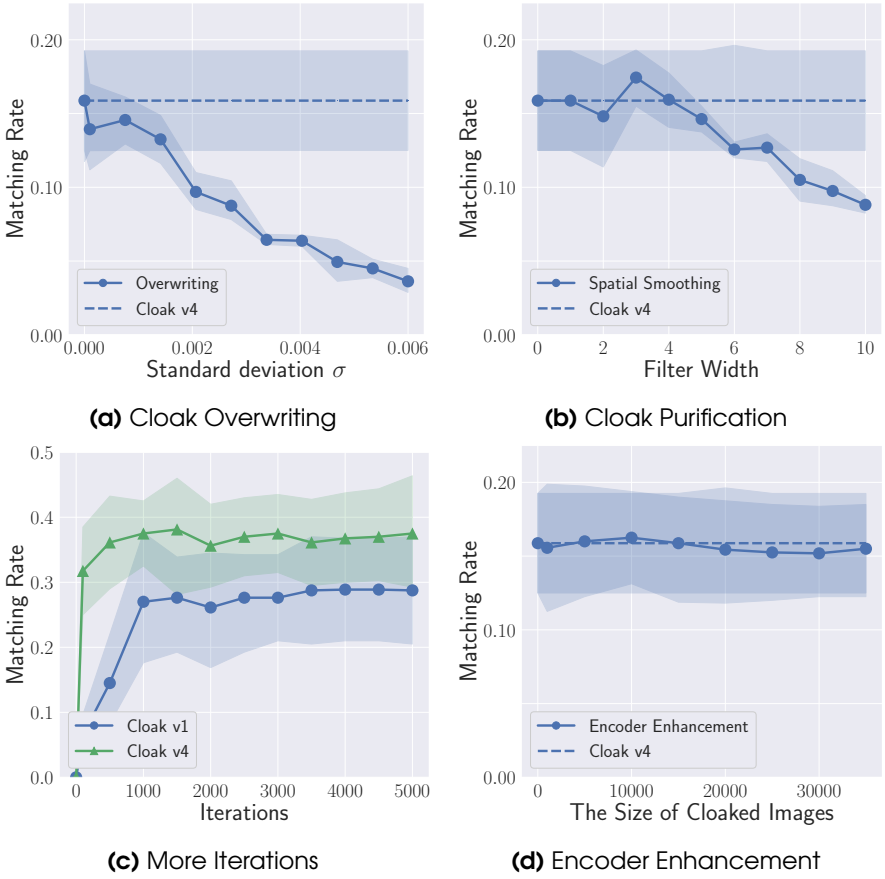
methods and Cloak v1/v4, respectively (see more results of MSE/SSIM in our technical report [81]). Thus, we can make the same observations as `UnGANable` on generated images, i.e., our Cloak v1/v4 of `UnGANable` achieves consistently better effectiveness (lower matching rate) and utility (lower MSE, higher SSIM, and PSNR) performance compared to all baseline methods.

5.6 Possible Adaptive Adversary

Here, we explore four possible adaptive adversaries and empirically evaluate the performance of `UnGANable` on real facial images. We conduct extensive experiments under the black-box scenario against optimization-based and hybrid inversion, i.e., Cloak v1 and Cloak v4. Note that for the purpose of straightforward comparisons, we average the performance of `UnGANable` with a varying number of distance budgets, i.e., ε-0/1/2/3.

Cloak Overwriting. This adaptive adversary aims to disturb the cloaks, i.e., the imperceptible perturbation searched by `UnGANable`. The adversary samples random noise from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ to overwrite the cloaks.

We report the matching rate by varying the standard deviation σ (set μ as 0 for simplicity) in Figure 5.10a (see more results of Cloak v1 in our technical report [81]). We can observe that as the standard deviation increases, the matching rate of cloak overwriting is significantly reduced. The reason is that the cloak overwriting actually introduces more noise in the image space on top of the imperceptible noise searched by



(a) Cloak Overwriting **(b)** Cloak Purification

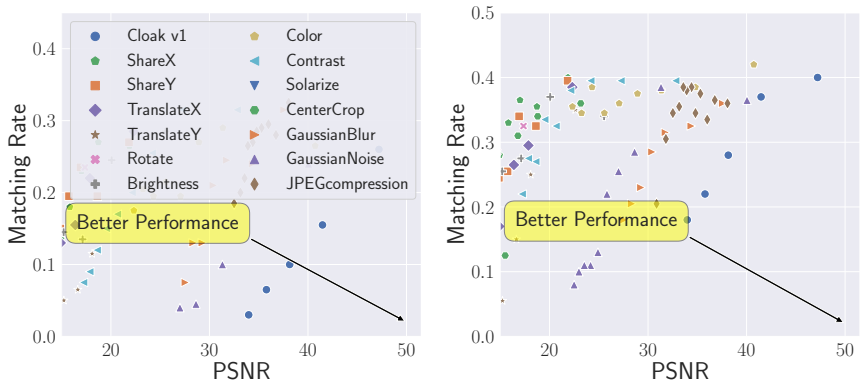
(c) More Iterations **(d)** Encoder Enhancement

Figure 5.10: The effectiveness performance of `UnGANable` (Cloak v4) on real images under the effect of four possible adaptive adversaries.

the `UnGANable`, which further jeopardizes the GAN inversion process. These results indicate that cloak overwriting is not an applicable adaptive strategy for adversaries.

Cloak Purification. This adaptive adversary aims to remove or purify the cloaks searched by `UnGANable`. As aforementioned, these cloaks actually are the imperceptible noise added to the images. Thus, we consider one of the most wide-used and easy-to-apply image noise reduction mechanisms, i.e., Spatial Smoothing [11]. Spatial Smoothing means that pixel values are averaged with their neighboring pixel values with a low-pass filter, leading to the sharp "edges" of the image becoming blurred and the spatial correlation within the data becoming more apparent.

We report the matching rate by varying the filter widths of Spatial Spatial in Figure 5.10b (see more results of Cloak v1 in our technical report [81]). We can clearly observe that the matching rate increases at first and then decreases. These results indicate that Spatial Smoothing indeed can purify the imperceptible noise added by `UnGANable` to some extent. We should also note that even the optimal setting for Spatial Smoothing can only lead to a slightly increased matching rate, and they all drop further sharply when the filter width is very large, as the Spatial Smoothing destroys



(a) Optimization-based **(b)** Hybrid

Figure 5.11: Comparison between all baseline methods and Cloak v1/v4 on real images. The different points of each method represent different budgets.

the pixel space of the original image. This observation implies that Spatial Smoothing is only a slightly effective adaptive strategy to reduce the jeopardy of `UnGANable` to GAN inversions.

More Iterations of Inversion. This adaptive adversary has significant computational resources to perform many optimization iterations to increase the matching rate. More specifically, we vary the number of optimization iterations from 0 to 5000 for both optimization-based and hybrid inversions. Note that the default settings for the number of iterations are 500 and 100 for optimization-based inversion and hybrid inversion, respectively.

Figure 5.10c shows the matching rate of `UnGANable` under the effect of numbers of iterations. As expected, we can find that the matching rate increases with the number of optimization iterations. Specifically, the matching rate increases sharply up to 1000/100 iterations and continues to increase slowly afterward. These results clearly demonstrate that more iterations of inversion indeed can reduce the jeopardy of `UnGANable` to GAN inversions. We should also note that a larger number of iterations (even up to 5000) does not lead to great effects but is a huge cost in terms of resource usage.

Encoder Enhancement. We further consider another adaptive adversary where the adversary retrains the encoder to be more robust to imperceptible noise searched by `UnGANable`. More concretely, we assume that the adversary can collect a large number of cloaked images from crawler-accessible websites or social media. We consider various numbers of cloaked images from 5k to 35k that an adversary can collect. Note that the number of images in the full FFHQ dataset used to train StyleGANv2 is only 70k. Then the adversary retrains the encoder with a mixed set of original clean images and collected cloaked images.

Since the encoder is only employed for hybrid inversion, we only consider here Cloak v4, the black-box setting against hybrid inversion, for evaluation. Figure 5.10d reports the matching rate under the effect of the different numbers of cloaked images collected by the adversary. We can observe that the matching rate decreases slightly

with increasing cloaked images, which means that retraining the encoder increases the jeopardy of UnGANable to GAN inversion. In a nutshell, encoder enhancement is not an applicable adaptive strategy for adversaries to reduce the jeopardy of UnGANable to GAN inversions.

5.7 Limitation

There are two major paradigms for image manipulation: GAN-inversion-based and image-translation-based. The latter, represented by StarGANv2 [33] and AttGAN [53], transforms an image from the source domain to the target domain without the GAN-inversion process. Therefore, our proposed UnGANable is not applicable to image-translation-based manipulation, as the key idea of UnGANable is to jeopardize the process of GAN inversion. Moreover, we emphasize here that GAN-inversion-based and image-translation-based are two orthogonal image manipulation techniques. Considering that the defense against the latter has been well studied [113, 145, 59, 80], the defense against GAN-inversion-based is still an open research problem. Our work is therefore well-motivated to complete this puzzle map.

Moreover, except for $\mathbf{z}$ space we consider in this work, recent works [158, 14, 15, 133, 71, 100, 18, 136, 39] also works on $\mathbf{w}$ space, which is transformed from $\mathbf{z}$ space, leading to a better inversion performance. We leave the in-depth exploration of a more efficient UnGANable against $\mathbf{w}$ space for future work.

5.8 Conclusion

In this chapter, we take the first step towards defending against GAN-based face manipulation by proposing UnGANable, a system that can jeopardize the process of GAN inversion. We consider two advanced GAN inversions: optimization-based and hybrid inversions, as well as five scenarios to comprehensively characterize the defender's background knowledge in multiple dimensions. We extensively evaluate UnGANable on four popular GAN models built on two benchmark face datasets of different sizes and complexity. The results show that UnGANable can achieve remarkable performance with respect to both effectiveness and utility. We further conduct a comparison of UnGANable with thirteen image distortion methods as well as Fawkes, and the results show that UnGANable generally outperforms all these methods. In addition, we explore four possible adaptive adversaries against UnGANable, and empirical evaluation shows that Spatial Smoothing and more inversion iterations are slightly effective.

6
Related Work

In this chapter, we survey the areas most relevant to our work. We first review the vulnerability of ML models to privacy risks. Then, we present the abuse of ML models leading to privacy risks. Finally, we briefly describe some privacy-preserving mechanisms for ML models.

6.1 Privacy Risks of Machine Learning Models

Privacy risks of ML models refer to the vulnerability of ML models to privacy risks. Specifically, they refer to inferring/stealing/reconstructing sensitive information from machine learning models. We now briefly review some well-known privacy risks inherent in machine learning models.

6.1.1 Membership Inference

Currently, membership inference is one of the major methods to evaluate privacy risks of machine learning models [124, 146, 49, 116, 94, 129, 76, 51]. Shokri et al. [124] propose the first membership inference attack against ML models. They build a series of attack models on a dataset that comprises various shadow models' outputs. These attack models take the target model's posterior as input and estimate whether it is a member or not. Then, by gradually loosening the assumption established by Shokri et al. [124], Salem et al. [116] propose a more general method, i.e., model and data-independent membership inference attack. Nasr et al. [94] later emphasize the privacy issue in centralized and federated learning situations and conduct comprehensive experiments in black-and-white-box scenarios. Song et al. [129] investigate the correlation between adversarial examples and the privacy risk of membership inference attacks. Li and Zhang [P1] and Choquette-Choo et al. [34] propose the label-only membership inference attack by changing the target model's predicted labels, then measuring the magnitude of the noise added on the input. They consider samples as members if the noise is larger than the predefined threshold.

6.1.2 Attribute Inference

Attribute inference (also called property inference) is another major type of privacy attack against ML models. In this attack, the goal of the adversary is to determine some private attributes of a given data sample by observing the representation generated by the target model [89, 127]. Melis et al. [89] present the first attribute inference attack against machine learning with a focus on federated learning. Language models, according to Song and Raghunathan [126], are also vulnerable to attribute inference assaults. Song and Shmatikov [127] confirm that attribute inference attacks are effective against another training paradigm, called model partitioning. Furthermore, they show evidence that the reason behind the success of attribute inference attacks is due to the overbearing behavior of ML models.

6.1.3 Model Inversion

Model inversion attack is another major privacy attack in which an adversary aims to reconstruct the data sample used to train an ML model [43, 42, 56, 26, 153]. This attack can cause serious privacy risks as it allows the adversary to extract private information about individuals, such as medical records or financial transactions. Model inversion attacks have received significant attention in recent years, and a large body of studies has shown that even highly complex ML models are vulnerable to these attacks. As a result, defending against model inversion attacks has become an important research topic in privacy-preserving machine learning.

6.2 Privacy Risks by Machine Learning Models

Privacy risks by ML models refer to those caused by abusing ML models. We here review two representative cases of abuse of ML models that lead to privacy risks: unauthorized collection of individual data and unauthorized manipulation of individual data.

6.2.1 Unauthorized Collection of Individual Data

The success of ML models is typically dependent on large training datasets. Companies and organizations frequently gather vast data from many sources, such as social media, to train accurate ML models. Yet, the extensive collection and use of personal data without the required authorization or agreement could result in major privacy concerns [16].

6.2.2 Unauthorized Manipulation of Individual Data

As machine learning techniques are increasingly used in various applications, there is growing concerned about the potential for unauthorized manipulation of individual data. In particular, the emergence of deepfake technology has raised concerns about the potential for unauthorized manipulation of individual data. Deepfakes that leverage ML models to produce real and convincing synthetic media content, e.g., videos and images, reflecting the events or individuals that have never occurred. The technology, however, is shown to be used for malicious purposes, such as spreading false information or creating false images of individuals for their malicious purposes.

In chapter 5, we investigate how to defend a representative deepfake application, Facial Manipulation, which modifies the facial attributes of a victim in an image, e.g., changing her age or hair color. Typically, there are two types of deepfake face manipulation techniques, one is GAN-inversion-based, and another is image-translation-based. In the former, to leverage GANs to manipulate facial images, the manipulator/adversary needs to perform a two-step operation. The first step is *GAN inversion* [159, 14, 15, 158, 21, 139] which inverts a victim's facial image to a latent code. The second step is *latent code manipulation* [141, 160, 122, 63, 48, 123, 149, 99, 31, 44] which manipulates the latent code to get the modified image, such as adding a pair of glasses on the victim's face. Image-to-Image Translations (I2I), represented by StarGANv2 [33] and AttGAN [53], have received increasing attention in recent years. More concretely, I2I

builds an end-to-end neural network as the backbone to translate source images into the target domain with many aligned image pairs for training. I2I uses the backbone network to accept the target image and output a new style without the GAN-inversion process when editing images.

6.3 Privacy-Preserving Machine Learning Models

To mitigate the threat of privacy risks, a large body of defense mechanisms has been proposed in the literature.

6.3.1 Privacy-Preserving Techniques for the Vulnerability of ML Models

Researchers have proposed to improve privacy against membership inference via different types of generalization enhancement. For example, Shokri et al. [124] adopted L2 regularization with a polynomial in the model's loss function to penalize large parameters. Salem et al. [116] demonstrated two effective methods of defending MI attacks: dropout and model stacking. Nasr et al. [93] introduced a defensive confidence score membership classifier in a min-max game mechanism to train models with membership privacy, namely adversarial regularization. Other existing generalization enhancement methods can be used to mitigate membership leakages, such as L1 regularization and data augmentation. Another direction is privacy enhancement. Many differential privacy-based defenses [28, 40, 62] involve clipping and adding noise to instance-level gradients and are designed to train a model to prevent it from memorizing training data or being susceptible to membership leakage. Shokri et al. [124] designed a differential privacy method for collaborative learning of DNNs. As for confidence score alteration, Jia et al. [66] introduce MemGuard, the first defense with formal utility-loss guarantees against membership inference. The basic idea behind this work is to add carefully crafted noise to the confidence scores of an ML model to mislead the membership classifier. Yang et al. [143] also propose a similar defense in this direction.

Raval et al. [108] propose Olympus as a defense against attribute inference. Olympus uses an adversarial classifier to infer sensitive attributes. Then it uses adversarial training to optimize the model against the adversarial classifier to maintain the model's utility while safeguarding the sensitive attributes of the sample. Jia and Gong [65] later propose AttriGuard.

This defense uses modified evasive attack techniques to provide an adversarial example for each possible value of the sensitive characteristic. The new representation is then selected after a sensitive attribute value is sampled using a probability distribution. The related adversarial example found in the first phase is then used as the basis for the sample. Song and Shmatikov [127] designed a joint training defense that iteratively trains the model and adversarial classifier to obstruct the embedding of sensitive information.

To defend against model inversion attacks, The Mutual Information Regularization-based Defense (MID), proposed by Wang et al. [137], presents a theoretical perspective on the finite efficacy of differential privacy approaches. There are also many works that concentrate on decreasing the link between input and output by purifying the output confidence scores to defend against the black box setup [115, 43, 143].

6.3.2 Privacy-Preserving Techniques Against the Abuse of ML Models

There are multiple works against privacy risks caused by the abuse of ML models. To prevent the unauthorized collection of individual data, Shan et al. [120] proposed Fawkes, one of the most representative efforts to protect individual privacy. By purposefully introducing undetectable alterations to the images, it promises to safeguard users' photos from illegal access and exploitation. Besides, other researchers [142, 107] have also proposed different techniques to protect data privacy and prevent misuse, highlighting the growing concern over these issues.

Since unauthorized manipulation of individual data poses a significant threat to personal privacy and even political security, it is critical to develop countermeasures against it. Many defenses have been proposed to mitigate this risk, and these defenses can be broadly divided into two categories: detection [79, 112, 17, 156, 88, 95] and disrupting I2I [113, 145, 59, 80]. However, the former defense is designed passively to detect whether face images have been tampered with after wide propagation. The latter defense can only mitigate image-translation-based face manipulation by spoofing the backbone network. However, there is still no approach to defend against GAN-inversion-based face manipulation in a proactive manner. In chapter 5, we propose `UnGANable` of initiative defense to degrade the performance of GAN inversion, which is an essential step for subsequent face manipulation.

7

Summary and Conclusion

Over the past decade, machine learning (ML) has made remarkable advancements and has been utilized extensively in various domains. However, despite their widespread use and popularity, machine learning models are also prone to pose various privacy risks.

In this dissertation, we investigate the privacy risks of machine learning models from two perspectives. The first one is the vulnerability of ML models to privacy risks, and the second one is the abuse of ML models that leads to privacy risks. To study the former, we conduct two works focusing on one of the most severe privacy attacks against ML models, i.e., membership inference attacks. More concretely, chapter 3 proposes label-only membership inference attacks and demonstrates that machine learning models are vulnerable to membership leakage in the label-only scenario. In chapter 4, we perform the first privacy analysis of multi-exit networks through the lens of membership leakages, i.e., revealing that multi-exit networks are less vulnerable to membership leakage. We further propose a more powerful membership inference attack called hybrid attack. To study the latter, chapter 5 proposes the first defense system, namely `UnGANable`, against GAN-based face manipulation by adding imperceptible perturbation on the face images.

Our works presented in this dissertation led to three peer-reviewed publications [P1, P2, P3], each investigating one perspective of privacy risks of machine learning models.

Our first work [P1] explores membership inference attacks against machine learning models. The success of existing membership inference attacks is that they rely on the confidence scores returned by the target machine learning models. However, these score-based attacks can be trivially mitigated if the model only exposes the predicted label instead of confidence scores. In chapter 3, we focus on a new category of membership inference attacks that have so far received fairly little attention, namely label-only membership inference attacks. We relax the assumption that the adversary has access to the target model's confidence score, but can solely rely on the final decision of the target model, i.e., the top-1 predicted label. We design two different label-only membership inference attacks under different scenarios, namely transfer-based attack and boundary-based attack. In the transfer-based attack, the adversary first builds a local model to mimic the target model by querying the target model in a manner analog to a cryptographic oracle and then launches existing membership inference attacks against the local model. In the boundary-based attack, The adversary queries the target model on a data sample and perturbs it to change the model's predicted labels. Then, the adversary measures the magnitude of the perturbation and considers the data samples as members if their magnitude is greater than a predefined threshold. Empirical evaluation shows that our label-only attacks can achieve remarkable performance and even outperform the previous score-based attacks in some cases. Further, we evaluate multiple defense mechanisms against our label-only attacks and show that our two attacks can bypass most defenses.

Our second work [P2] explores the vulnerability of multi-exit networks to membership inference attacks. The key design of multi-exit networks is endowing a backbone model with early exits, allowing the predictions to exit from the intermediate layers of the model. In chapter 4, we first leverage the existing attack methodologies, namely gradient-based, score-based, and label-only attacks, to audit the membership leakage risks of

multi-exit networks. Our extensive experiments demonstrate that multi-exit networks are less vulnerable to membership leakage than normal models. Furthermore, our study shows that exit information of multi-exit networks has a strong correlation with attack performance. Based on this observation, we propose hybrid attack that exploits exit information as new adversary knowledge to improve the performance of existing member inference attacks. We evaluate our hybrid attack in three different adversarial settings, yielding a model-free and data-free adversary, demonstrating the broad applicability and severe risk compared to existing attacks. Finally, we introduce a simple but effective defense mechanism called *TimeGuard* and evaluate its effectiveness through empirical experiments.

Different the above two works explore the privacy risks of machine learning models, i.e., models leak sensitive information itself. In our final work [P3], we focus on the privacy risks caused by abusing machine learning models, namely GAN-based Facial Manipulation. This technique is one of the most representative deepfake applications that modify the facial attributes of a victim in an image, e.g., changing her age or hair color. Specially, we focus on reducing the performance of GAN inversion and propose the first defense mechanism, called `UnGANable`, which is targeted at jeopardizing GAN inversion. We consider five scenarios to comprehensively characterize a defender's background knowledge along multiple dimensions and propose five different defenses for each scenario. Comprehensive analyses of four well-known GAN models reveal that the `UnGANable` achieves outstanding performance in terms of both efficacy and utility. We conduct a comparison and show that our defense can outperform all thirteen baseline image distortion methods. We further explore four adaptive adversaries to bypass `UnGANable` and show that some of them are slightly effective.

Future Research Directions. This dissertation provides some insights into the privacy risks of machine learning models from two perspectives: the vulnerability of ML models and the abuse of ML models. We now discuss possible directions for future work.

In chapter 3, we propose label-only membership inference attacks against machine learning models. As a new type of attack, one possible direction is to evaluate their practicality. For example, in our boundary-based label-only membership inference attack, we utilize adversarial attacks, e.g., HopSkipJump and QEBA, to add the adversarial noise on the input sample to mislead the target model. However, these adversarial noises typically require many times of queries, as shown in Figure 3.9. In the real world, the adversary typically cannot query the target model many times which would raise suspicion of the victim model owner. So, how to launch a successful attack with an acceptable number of queries? We believe this would be an interesting and challenging research question. One possible solution is that we can add noise to mislead the target model using different image distortion methods such as GaussianBlur, GaussianNoise, JPEGCompression, etc., which may provide new insights into this research direction.

In chapter 4, we take the first step to audit the privacy risks of multi-exit networks through the lens of membership inference attacks. We reveal that the multi-exit networks are less susceptible to membership leakage. Further, we further propose a hybrid attack to improve the performance of membership inference attacks by using exit information as new adversary knowledge. Inspired by these observations, another possible research

direction that is worthwhile to study is "How vulnerable are other new design forms of neural networks to privacy risks (e.g., membership inference attacks)?" For instance, many new forms of neural networks have been proposed in recent years, such as masked image modeling [38, 140, 157, 57] and prompt-based learning [155, 144, 67]. Current various designs of new forms of neural networks are only taken into consideration to obtain the greatest performance of their primary purpose, the privacy risks resulting from them have never been examined. We, therefore, believe that this is a research direction worth investigating.

In chapter 5, we take the first step towards defending against malicious face manipulation by reducing the performance of GAN inversion - the adversary can only obtain an inaccurate latent code that is far from the accurate one, thus the following latent code manipulation step will not achieve the ideal result. As we mentioned before, all GAN models considered in this dissertation work on latent code $\mathbf{z}$ space, thus we only consider how to mislead the GAN model in the $\mathbf{z}$ space. Recent works of face manipulation [158, 14, 15, 133, 71, 100, 18, 136, 39] also works on $\mathbf{w}$ space, which is transformed from $\mathbf{z}$ space, leading to a better inversion performance. Thus, we think this is another challenging and meaningful research direction. One possible solution is to apply the defense methodologies proposed in this chapter, as there is no fundamental difference between $\mathbf{z}$ space and $\mathbf{w}$ space. For more efficient defense against $\mathbf{w}$ space, we leave the in-depth exploration for future work.

A
Appendix

Table A.1: Dataset splitting strategy. $\mathcal{D}_{train}$ is used to train the target model and serves as the members, while the other $\mathcal{D}_{test}$ serves as the non-members. $\mathcal{D}_{shadow}$ is used to train the shadow model after relabelled by the target model.

Target Model	CIFAR10		CIFAR100		GTSRB		Face	
	$\mathcal{D}_{train}$	$\mathcal{D}_{test}$	$\mathcal{D}_{train}$	$\mathcal{D}_{test}$	$\mathcal{D}_{train}$	$\mathcal{D}_{test}$	$\mathcal{D}_{train}$	$\mathcal{D}_{test}$
$\mathcal{M}$-0	3000	1000	7000	1000	600	500	350	100
$\mathcal{M}$-1	2000	1000	6000	1000	500	500	300	100
$\mathcal{M}$-2	1500	1000	5000	1000	400	500	250	100
$\mathcal{M}$-3	1000	1000	4000	1000	300	500	200	100
$\mathcal{M}$-4	500	1000	3000	1000	200	500	150	100
$\mathcal{M}$-5	100	1000	2000	1000	100	500	100	100
Shadow Model	$\mathcal{D}_{shadow}$							
	46000		42000		38109		1417	

Table A.2: The threshold τ set for computer vision tasks.

Dataset	Exit Number	Model Architecture			
		VGG	ResNet	MobileNet	WideResNet
CIFAR-10	2	0.9	0.7	0.6	0.85
	3	0.9	0.7	0.6	0.85
	4	0.9	0.7	0.6	0.85
	5	0.9	0.7	0.6	0.85
	6	0.9	0.7	0.6	0.85
CIFAR-100	2	0.2	0.3	0.4	0.8
	3	0.2	0.3	0.4	0.8
	4	0.2	0.3	0.4	0.8
	5	0.2	0.3	0.4	0.8
	6	0.2	0.3	0.4	0.8
TinyImageNet	2	0.4	0.25	0.55	0.85
	3	0.4	0.25	0.55	0.85
	4	0.4	0.25	0.55	0.85
	5	0.4	0.25	0.55	0.85
	6	0.4	0.25	0.55	0.85

Table A.3: The threshold τ set for non-computer vision tasks.

Dataset	Exit Number	Model Architecture			
		FCN-18-1	FCN-18-2	FCN-18-3	FCN-18-4
Purchases	2/3/4/5/6	0.7	0.7	0.7	0.7
Locations	2/3/4/5/6	0.5	0.5	0.5	0.5
Texas	2/3/4/5/6	0.7	0.7	0.7	0.7

Algorithm 4: Cloaking facial image of Cloak-0

Input: A target image $\mathbf{x}$ to cloak; a pre-trained target generator $G_{\mathrm{t}}(\cdot)$; a shadow encoder $E_{\mathrm{s}}(\cdot)$; a pre-trained ResNet feature extractor F; cosine similarity $\mathcal{L}_{\cos}(\cdot,\cdot)$; MSE similarity $\mathcal{L}_{\mathrm{mse}}(\cdot,\cdot)$; minibatch m; perturbation budget ε; trade-off κ.

Output: The trained shadow encoder E_{s} and the cloaked image $\hat{\mathbf{x}}$.

Initialize $\mathcal{L}_{\mathrm{rec}}(\cdot,\cdot) = -\mathcal{L}_{\cos}(\cdot,\cdot) + \mathcal{L}_{\mathrm{mse}}(\cdot,\cdot)$;
for *number of training iterations* **do**
  sample a minibatch of latent codes $\mathbf{z}^{'} \in \mathcal{N}(0,1)$;
  $\min_{\Theta_{E_{\mathrm{s}}}} \mathcal{L}_{\mathrm{rec}}\Big(E_{\mathrm{s}}(G_{\mathrm{t}}(\mathbf{z}^{'}))), \mathbf{z}^{'}\Big)$
end
Initialize $\mathbf{x}_{\mathrm{t}}$ = optimization-based inversion($\mathbf{x}$);
Initialize $\delta \in \mathcal{N}(0,1)$ and $|\delta|_{\infty} < \epsilon$;
Initialize κ;
for *number of optimized iterations* **do**
  $\max_{\delta} \kappa\Big(\mathcal{L}_{\mathrm{rec}}(E_{\mathrm{s}}(\mathbf{x}+\delta), \mathbf{x}_{\mathrm{t}})\Big) + (1-\kappa)\Big(\mathcal{L}_{\mathrm{rec}}(F(\mathbf{x}+\delta), F(\mathbf{x}))\Big)$;
  clip δ for $|\delta|_{\infty} < \epsilon$;
  clip $\mathbf{x}+\delta$ for $\mathbf{x}+\delta \in [0,1]$;
end
$\hat{\mathbf{x}} = \mathbf{x} + \delta$;
return $E_{\mathrm{s}}, \hat{\mathbf{x}}$

Algorithm 5: Cloaking facial image of Cloak-2

Input: A target image $\mathbf{x}$ to cloak; a pre-trained target encoder $E_{\mathrm{t}}(\cdot)$; a pre-trained ResNet feature extractor F; cosine similarity $\mathcal{L}_{\cos}(\cdot,\cdot)$; MSE similarity $\mathcal{L}_{\mathrm{mse}}(\cdot,\cdot)$; perturbation budget ε; trade-off κ.

Output: The cloaked image $\hat{\mathbf{x}}$.

Initialize $\mathcal{L}_{\mathrm{rec}}(\cdot,\cdot) = -\mathcal{L}_{\cos}(\cdot,\cdot) + \mathcal{L}_{\mathrm{mse}}(\cdot,\cdot)$;
Initialize $\delta \in \mathcal{N}(0,1)$ and $|\delta|_{\infty} < \epsilon$;
Initialize κ;
for *number of optimized iterations* **do**
  $\max_{\delta} \kappa\Big(-\mathcal{L}_{\mathrm{rec}}(E_{\mathrm{t}}(\mathbf{x}+\delta), 0)\Big) + (1-\kappa)\Big(\mathcal{L}_{\mathrm{rec}}(F(\mathbf{x}+\delta), F(\mathbf{x}))\Big)$;
  clip δ for $|\delta|_{\infty} < \epsilon$;
  clip $\mathbf{x}+\delta$ for $\mathbf{x}+\delta \in [0,1]$;
end
$\hat{\mathbf{x}} = \mathbf{x} + \delta$;
return $\hat{\mathbf{x}}$

Algorithm 6: Cloaking facial image of Cloak-3

Input: A target image $\mathbf{x}$ to cloak; a pre-trained target encoder $E_{\mathrm{t}}(\cdot)$; a shadow encoder E_{s}; a shadow generator G_{s}; a pre-trained ResNet feature extractor F; cosine similarity $\mathcal{L}_{\cos}(\cdot,\cdot)$; MSE similarity $\mathcal{L}_{\mathrm{mse}}(\cdot,\cdot)$; perturbation budget ε; trade-off κ.

Output: The trained shadow encoder E_{s}, the trained shadow generator G_{s} and the cloaked image $\hat{\mathbf{x}}$.

Initialize $\mathcal{L}_{\mathrm{rec}}(\cdot,\cdot) = -\mathcal{L}_{\cos}(\cdot,\cdot) + \mathcal{L}_{\mathrm{mse}}(\cdot,\cdot)$;
for *number of training iterations* **do**
  sample a minibatch of latent codes $\mathbf{z}^{'} \in \mathcal{N}(0,1)$;
  $\min_{\Theta_{E_{\mathrm{s}}}} \mathcal{L}_{\mathrm{rec}}\Big(E_{\mathrm{s}}(G_{\mathrm{s}}(\mathbf{z}^{'}))), \mathbf{z}^{'}\Big)$;
  $\max_{\Theta_{G_{\mathrm{s}}}} \mathcal{L}_{\mathrm{rec}}\Big(E_{\mathrm{s}}(G_{\mathrm{s}}(\mathbf{z}^{'}))), \mathbf{z}^{'}\Big)$;
end
Initialize $\delta \in \mathcal{N}(0,1)$ and $|\delta|_{\infty} < \epsilon$;
Initialize κ;
for *number of optimized iterations* **do**
  $\max_{\delta} \kappa\Big(- \mathcal{L}_{\mathrm{rec}}\big(E_{\mathrm{s}}(\mathbf{x}+\delta), 0\big)\Big) + (1-\kappa)\Big(\mathcal{L}_{\mathrm{rec}}\big(F(\mathbf{x}+\delta), F(\mathbf{x})\big)\Big)$;
  clip δ for $|\delta|_{\infty} < \epsilon$;
  clip $\mathbf{x}+\delta$ for $\mathbf{x}+\delta \in [0,1]$;
end
$\hat{\mathbf{x}} = \mathbf{x} + \delta$;
return $E_{\mathrm{s}}, G_{\mathrm{s}}, \hat{\mathbf{x}}$

Algorithm 7: Cloaking facial image of Cloak-1/4

Input: A target image $\mathbf{x}$ to cloak; a pre-trained ResNet feature extractor F; cosine similarity $\mathcal{L}_{\cos}(\cdot,\cdot)$; MSE similarity $\mathcal{L}_{\mathrm{mse}}(\cdot,\cdot)$; perturbation budget ε.

Output: The cloaked image $\hat{\mathbf{x}}$.

Initialize $\mathcal{L}_{\mathrm{rec}}(\cdot,\cdot) = -\mathcal{L}_{\cos}(\cdot,\cdot) + \mathcal{L}_{\mathrm{mse}}(\cdot,\cdot)$;
Initialize $\delta \in \mathcal{N}(0,1)$ and $|\delta|_{\infty} < \epsilon$;
for *number of optimized iterations* **do**
  $\max_{\delta} \mathcal{L}_{\mathrm{rec}}\big(F(\mathbf{x}+\delta), F(\mathbf{x})\big)$;
  clip δ for $|\delta|_{\infty} < \epsilon$;
  clip $\mathbf{x}+\delta$ for $\mathbf{x}+\delta \in [0,1]$;
end
$\hat{\mathbf{x}} = \mathbf{x} + \delta$;
return $\hat{\mathbf{x}}$

Bibliography

Author's Papers for this Thesis

[P1] Li, Z. and Zhang, Y. Membership Leakage in Label-Only Exposures. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2021, 880–895.

[P2] Li, Z., Liu, Y., He, X., Yu, N., Backes, M., and Zhang, Y. Auditing Membership Leakages of Multi-Exit Networks. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2022, 1917–1931.

[P3] Li, Z., Yu, N., Salem, A., Backes, M., Fritz, M., and Zhang, Y. UnGANable: Defending Against GAN-based Face Manipulation. In: *USENIX Security Symposium (USENIX Security)*. USENIX, 2023.

Other Published Papers of the Author

[S1] Liu, Y., Li, Z., Backes, M., Shen, Y., and Zhang, Y. Backdoor Attacks Against Dataset Distillation. In: *Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2023.

Other Technical Reports of the Author

[T1] He, X., Li, Z., Xu, W., Cornelius, C., and Zhang, Y. Membership-Doctor: Comprehensive Assessment of Membership Inference Against Machine Learning Models. *CoRR abs/2208.10445* (2022).

[T2] Sha, Z., Li, Z., Yu, N., and Zhang, Y. DE-FAKE: Detection and Attribution of Fake Images Generated by Text-to-Image Generation Models. *CoRR abs/2210.06998* (2022).

[T3] Shen, X., He, X., Li, Z., Shen, Y., Backes, M., and Zhang, Y. Backdoor Attacks in the Supply Chain of Masked Image Modeling. *CoRR abs/2210.01632* (2022).

[T4] Wu, Y., Yu, N., Li, Z., Backes, M., and Zhang, Y. Membership Inference Attacks Against Text-to-image Generation Models. *CoRR abs/2210.00968* (2022).

[T5] Yang, Z., He, X., Li, Z., Backes, M., Humbert, M., Berrang, P., and Zhang, Y. Data Poisoning Attacks Against Multimodal Encoders. *CoRR abs/2209.15266* (2022).

Other references

[1] https://www.cs.toronto.edu/~kriz/cifar.html.

[2] http://benchmark.ini.rub.de/?section=gtsrb.

[3] http://vis-www.cs.umass.edu/lfw/.

[4] https://www.kaggle.com/c/tiny-imagenet.

[5] https://github.com/DmitryUlyanov/Multicore-TSNE.

[6] https://www.kaggle.com/c/acquire-valued-shoppers-challenge/data.

[7] https://sites.google.com/site/yangdingqi/home/foursquare-dataset.

[8] https://www.dshs.texas.gov/thcic/hospitals/Inpatientpudf.shtm.

[9] https://www.statisticshowto.com/probability-and-statistics/hypothesis-testing/z-test/.

[10] https://github.com/JohannesBuchner/imagehash.

[11] https://support.brainvoyager.com/brainvoyager/functional-analysis-preparation/29-pre-processing/86-spatial-smoothing.

[12] https://github.com/graykode/distribution-is-all-you-need.

[13] Abadi, M., Chu, A., Goodfellow, I., McMahan, B., Mironov, I., Talwar, K., and Zhang, L. Deep Learning with Differential Privacy. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2016, 308–318.

[14] Abdal, R., Qin, Y., and Wonka, P. Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space? In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2019, 4431–4440.

[15] Abdal, R., Qin, Y., and Wonka, P. Image2StyleGAN++: How to Edit the Embedded Images? In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, 8293–8302.

[16] Acquisti, A., Brandimarte, L., and Loewenstein, G. Privacy and Human Behavior in the Age of Information. *Science* (2015).

[17] Afchar, D., Nozick, V., Yamagishi, J., and Echizen, I. MesoNet: a Compact Facial Video Forgery Detection Network. In: *IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2018, 1–7.

[18] Alaluf, Y., Tov, O., Mokady, R., Gal, R., and Bermano, A. H. HyperStyle: StyleGAN Inversion with HyperNetworks for Real Image Editing. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, 18511–18521.

[19] Backes, M., Berrang, P., Humbert, M., and Manoharan, P. Membership Privacy in MicroRNA-based Studies. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2016, 319–330.

[20] Backes, M., Humbert, M., Pang, J., and Zhang, Y. walk2friends: Inferring Social Links from Mobility Profiles. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2017, 1943–1957.

[21] Bartz, C., Bethge, J., Yang, H., and Meinel, C. One Model to Reconstruct Them All: A Novel Way to Use the Stochastic Noise in StyleGAN. *CoRR abs/2010.11113* (2020).

[22] Bicer, Y., Alizadeh, A., Ure, N. K., Erdogan, A., and Kizilirmak, O. Sample Efficient Interactive End-to-End Deep Learning for Self-Driving Cars with Selective Multi-Class Safe Dataset Aggregation. *CoRR abs/2007.14671* (2020).

[23] Biggio, B., Corona, I., Maiorca, D., Nelson, B., Srndic, N., Laskov, P., Giacinto, G., and Roli, F. Evasion Attacks against Machine Learning at Test Time. In: *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*. Springer, 2013, 387–402.

[24] Brock, A., Donahue, J., and Simonyan, K. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In: *International Conference on Learning Representations (ICLR)*. 2019.

[25] Burlina, P., Freund, D. E., Dupas, B., and Bressler, N. M. Automatic Screening of Age-related Macular Degeneration and Retinal Abnormalities. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2011, 3962–3966.

[26] Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In: *USENIX Security Symposium (USENIX Security)*. USENIX, 2019, 267–284.

[27] Carlini, N. and Wagner, D. Towards Evaluating the Robustness of Neural Networks. In: *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2017, 39–57.

[28] Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. Differentially Private Empirical Risk Minimization. *Journal of Machine Learning Research* (2011).

[29] Chen, J., Jordan, M. I., and Wainwright, M. J. HopSkipJumpAttack: A Query-Efficient Decision-Based Attack. In: *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2020, 1277–1294.

[30] Chen, Y., Wang, S., She, D., and Jana, S. On Training Robust PDF Malware Classifiers. In: *USENIX Security Symposium (USENIX Security)*. USENIX, 2020, 2343–2360.

[31] Cherepkov, A., Voynov, A., and Babenko, A. Navigating the GAN Parameter Space for Semantic Image Editing. *CoRR abs/2011.13786* (2020).

[32] Choi, Y., Choi, M., Kim, M., Ha, J., Kim, S., and Choo, J. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, 8789–8797.

[33] Choi, Y., Uh, Y., Yoo, J., and Ha, J. StarGAN v2: Diverse Image Synthesis for Multiple Domains. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, 8185–8194.

[34] Choo, C. A. C., Tramèr, F., Carlini, N., and Papernot, N. Label-Only Membership Inference Attacks. In: *International Conference on Machine Learning (ICML)*. PMLR, 2021, 1964–1974.

[35] Dang, Q. H. Secure Hash Standard. *Federal Information Processing Standard* (2015).

[36] Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., Nita-Rotaru, C., and Roli, F. Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks. In: *USENIX Security Symposium (USENIX Security)*. USENIX, 2019, 321–338.

[37] Denton, E., Hutchinson, B., Mitchell, M., and Gebru, T. Detecting Bias with Generative Counterfactual Face Attribute Augmentation. *CoRR abs/1906.06439* (2019).

[38] Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. ACL, 2019, 4171–4186.

[39] Dinh, T. M., Tran, A. T., Nguyen, R., and Hua, B. HyperInverter: Improving StyleGAN Inversion via Hypernetwork. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, 11389–11398.

[40] Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In: *Theory of Cryptography Conference (TCC)*. Springer, 2006, 265–284.

[41] Ester, M., Kriegel, H., Sander, J., and Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: *International Conference on Knowledge Discovery and Data Mining (KDD)*. AAAI, 1996, 226–231.

[42] Fredrikson, M., Jha, S., and Ristenpart, T. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2015, 1322–1333.

[43] Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., and Ristenpart, T. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. In: *USENIX Security Symposium (USENIX Security)*. USENIX, 2014, 17–32.

[44] Goetschalckx, L., Andonian, A., Oliva, A., and Isola, P. GANalyze: Toward Visual Definitions of Cognitive Image Properties. In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2019, 5743–5752.

[45] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative Adversarial Nets. In: *Annual Conference on Neural Information Processing Systems (NIPS)*. NIPS, 2014, 2672–2680.

[46] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved Training of Wasserstein GANs. In: *Annual Conference on Neural Information Processing Systems (NIPS)*. NIPS, 2017, 5767–5777.

[47] Hagestedt, I., Zhang, Y., Humbert, M., Berrang, P., Tang, H., Wang, X., and Backes, M. MBeacon: Privacy-Preserving Beacons for DNA Methylation Data. In: *Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2019.

[48] Härkönen, E., Hertzmann, A., Lehtinen, J., and Paris, S. GANSpace: Discovering Interpretable GAN Controls. In: *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2020.

[49] Hayes, J., Melis, L., Danezis, G., and Cristofaro, E. D. LOGAN: Evaluating Privacy Leakage of Generative Models Using Generative Adversarial Networks. *Privacy Enhancing Technologies Symposium* (2019).

[50] He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, 770–778.

[51] He, X., Wen, R., Wu, Y., Backes, M., Shen, Y., and Zhang, Y. Node-Level Membership Inference Attacks Against Graph Neural Networks. *CoRR abs/2102.05429* (2021).

[52] He, X. and Zhang, Y. Quantifying and Mitigating Privacy Risks of Contrastive Learning. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2021, 845–863.

[53] He, Z., Zuo, W., Kan, M., Shan, S., and Chen, X. AttGAN: Facial Attribute Editing by Only Changing What You Want. *IEEE Transactions on Image Process* (2019).

[54] Hilprecht, B., Härterich, M., and Bernau, D. Monte Carlo and Reconstruction Membership Inference Attacks against Generative Models. *Privacy Enhancing Technologies Symposium* (2019).

[55] Hinton, G. E., Vinyals, O., and Dean, J. Distilling the Knowledge in a Neural Network. *CoRR abs/1503.02531* (2015).

[56] Hitaj, B., Ateniese, G., and Perez-Cruz, F. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2017, 603–618.

[57] Hou, L., Huang, Z., Shang, L., Jiang, X., Chen, X., and Liu, Q. DynaBERT: Dynamic BERT with Adaptive Width and Depth. In: *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2020.

[58] Huang, G., Chen, D., Li, T., Wu, F., Maaten, L. van der, and Weinberger, K. Q. Multi-Scale Dense Networks for Resource Efficient Image Classification. In: *International Conference on Learning Representations (ICLR)*. 2018.

[59] Huang, Q., Zhang, J., Zhou, W., Zhang, W., and Yu, N. Initiative Defense against Facial Manipulation. In: *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2021, 1619–1627.

[60] Hui, B., Yang, Y., Yuan, H., Burlina, P., Gong, N. Z., and Cao, Y. Practical Blind Membership Inference Attack via Differential Comparisons. In: *Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2021.

[61] Ioffe, S. and Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: *International Conference on Machine Learning (ICML)*. PMLR, 2015, 448–456.

[62] Iyengar, R., Near, J. P., Song, D. X., Thakkar, O. D., Thakurta, A., and Wang, L. Towards Practical Differentially Private Convex Optimization. In: *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2019, 299–316.

[63] Jahanian, A., Chai, L., and Isola, P. On the "Steerability" of Generative Adversarial Networks. In: *International Conference on Learning Representations (ICLR)*. 2020.

[64] Jayaraman, B. and Evans, D. Evaluating Differentially Private Machine Learning in Practice. In: *USENIX Security Symposium (USENIX Security)*. USENIX, 2019, 1895–1912.

[65] Jia, J. and Gong, N. Z. AttriGuard: A Practical Defense Against Attribute Inference Attacks via Adversarial Machine Learning. In: *USENIX Security Symposium (USENIX Security)*. USENIX, 2018, 513–529.

[66] Jia, J., Salem, A., Backes, M., Zhang, Y., and Gong, N. Z. MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2019, 259–274.

[67] Jin, W., Cheng, Y., Shen, Y., Chen, W., and Ren, X. A Good Prompt Is Worth Millions of Parameters: Low-resource Prompt-based Learning for Vision-Language Models. In: *Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2022, 2763–2775.

[68] Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In: *European Conference on Computer Vision (ECCV)*. Springer, 2016, 694–711.

[69] Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In: *International Conference on Learning Representations (ICLR)*. 2018.

[70] Karras, T., Laine, S., and Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, 4401–4410.

[71] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and Improving the Image Quality of StyleGAN. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, 8107–8116.

[72] Kaya, Y., Hong, S., and Dumitras, T. Shallow-Deep Networks: Understanding and Mitigating Network Overthinking. In: *International Conference on Machine Learning (ICML)*. PMLR, 2019, 3301–3310.

[73] Kemelmacher-Shlizerman, I., Seitz, S. M., Miller, D., and Brossard, E. The MegaFace Benchmark: 1 Million Faces for Recognition at Scale. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, 4873–4882.

[74] Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., and Fotiadis, D. I. Machine Learning Applications in Cancer Prognosis and Prediction. *Computational and Structural Biotechnology Journal* (2015).

[75] Krawczyk, H. and Eronen, P. HMAC-based Extract-and-Expand Key Derivation Function (HKDF). *Request for Comments* (2010).

[76] Leino, K. and Fredrikson, M. Stolen Memories: Leveraging Model Memorization for Calibrated White-Box Membership Inference. In: *USENIX Security Symposium (USENIX Security)*. USENIX, 2020, 1605–1622.

[77] Li, H., Xu, X., Zhang, X., Yang, S., and Li, B. QEBA: Query-Efficient Boundary-Based Blackbox Attack. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, 1218–1227.

[78] Li, J., Li, N., and Ribeiro, B. Membership Inference Attacks and Defenses in Classification Models. In: *ACM Conference on Data and Application Security and Privacy (CODASPY)*. ACM, 2021, 5–16.

[79] Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., and Guo, B. Face X-Ray for More General Face Forgery Detection. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, 5000–5009.

[80] Li, Y., Yang, X., Wu, B., and Lyu, S. Hiding Faces in Plain Sight: Disrupting AI Face Synthesis with Adversarial Perturbations. *CoRR abs/1906.09288* (2019).

[81] Li, Z., Yu, N., Salem, A., Backes, M., Fritz, M., and Zhang, Y. UnGANable: Defending Against GAN-based Face Manipulation. *CoRR abs/2210.00957* (2022).

[82] Liu, Y., Chen, X., Liu, C., and Song, D. Delving into Transferable Adversarial Examples and Black-box Attacks. *CoRR abs/1611.02770* (2016).

[83] Liu, Z., Luo, P., Wang, X., and Tang, X. Deep Learning Face Attributes in the Wild. In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, 3730–3738.

[84] Lloyd, S. P. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory* (1982).

[85] Long, Y., Bindschaedler, V., and Gunter, C. A. Towards Measuring Membership Privacy. *CoRR abs/1712.09136* (2017).

[86] Long, Y., Bindschaedler, V., Wang, L., Bu, D., Wang, X., Tang, H., Gunter, C. A., and Chen, K. Understanding Membership Inferences on Well-Generalized Learning Models. *CoRR abs/1802.04889* (2018).

[87] Lu, E. and Zhang, L. Machine Learning Methods for Smartphone Application Prediction. *International Symposium on Industrial Electronics* (2022).

[88] Matern, F., Riess, C., and Stamminger, M. Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations. In: *Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, 83–92.

[89] Melis, L., Song, C., Cristofaro, E. D., and Shmatikov, V. Exploiting Unintended Feature Leakage in Collaborative Learning. In: *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2019, 497–512.

[90] Messaoud, S., Bradai, A., Bukhari, S. H. R., Quang, P. T. A., Ahmed, O. B., and Atri, M. A survey on machine learning in Internet of Things: Algorithms, strategies, and applications. *Internet Things* (2020).

[91] Nandy, S. C., Das, S., and Goswami, P. P. An Efficient K Nearest Neighbors Searching Algorithm for A Query Line. *Theoretical Computer Science* (2003).

[92] Naseer, M., Khan, S. H., Khan, M. H., Khan, F. S., and Porikli, F. Cross-Domain Transferability of Adversarial Perturbations. In: *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2019, 12885–12895.

[93] Nasr, M., Shokri, R., and Houmansadr, A. Machine Learning with Membership Privacy using Adversarial Regularization. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2018, 634–646.

[94] Nasr, M., Shokri, R., and Houmansadr, A. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In: *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2019, 1021–1035.

[95] Nguyen, H. H., Yamagishi, J., and Echizen, I. Use of a Capsule Network to Detect Fake Images and Videos. *CoRR abs/1910.12467* (2019).

[96] Papernot, N., McDaniel, P., and Goodfellow, I. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. *CoRR abs/1605.07277* (2016).

[97] Papernot, N., McDaniel, P. D., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical Black-Box Attacks Against Machine Learning. In: *ACM Asia Conference on Computer and Communications Security (ASIACCS)*. ACM, 2017, 506–519.

[98] Papernot, N., McDaniel, P. D., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The Limitations of Deep Learning in Adversarial Settings. In: *IEEE European Symposium on Security and Privacy (Euro S&P)*. IEEE, 2016, 372–387.

[99] Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., and Lischinski, D. StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. *CoRR abs/2103.17249* (2021).

[100] Pavlopoulos, J., Laugier, L., Xenos, A., Sorensen, J., and Androutsopoulos, I. From the Detection of Toxic Spans in Online Discussions to the Analysis of Toxic-to-Civil Transfer. In: *Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2022, 3721–3734.

[101] Phuong, M. and Lampert, C. Distillation-Based Training for Multi-Exit Architectures. In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2019, 1355–1364.

[102] Pyrgelis, A., Troncoso, C., and Cristofaro, E. D. Knock Knock, Who's There? Membership Inference on Aggregate Location Data. In: *Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2018.

[103] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning Transferable Visual Models From Natural Language Supervision. In: *International Conference on Machine Learning (ICML)*. PMLR, 2021, 8748–8763.

[104] Radford, A., Metz, L., and Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In: *International Conference on Learning Representations (ICLR)*. 2016.

[105] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language Models are Unsupervised Multitask Learners. *OpenAI blog* (2019).

[106] Rahimian, S., Orekondy, T., and Fritz, M. Differential Privacy Defenses and Sampling Attacks for Membership Inference. In: *PriML Workshop (PriML)*. NeurIPS, 2020.

[107] Rajabi, A., Bobba, R. B., Rosulek, M., Wright, C. V., and Feng, W. On the (Im)Practicality of Adversarial Perturbation for Image Privacy. *Proceedings on Privacy Enhancing Technologies* (2021).

[108] Raval, N., Machanavajjhala, A., and Pan, J. Olympus: Sensor Privacy through Utility Aware Obfuscation. *Privacy Enhancing Technologies Symposium* (2019).

[109] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, 10684–10695.

[110] Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. FitNets: Hints for Thin Deep Nets. In: *International Conference on Learning Representations (ICLR)*. 2015.

[111] Rosenblatt, M. Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics* (1956).

[112] Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., and Nießner, M. FaceForensics++: Learning to Detect Manipulated Facial Images. In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2019, 1–11.

[113] Ruiz, N., Bargal, S. A., and Sclaroff, S. Disrupting Deepfakes: Adversarial Attacks Against Conditional Image Translation Networks and Facial Manipulation Systems. In: *European Conference on Computer Vision (ECCV)*. Springer, 2020, 236–251.

[114] Sablayrolles, A., Douze, M., Schmid, C., Ollivier, Y., and Jégou, H. White-box vs Black-box: Bayes Optimal Strategies for Membership Inference. In: *International Conference on Machine Learning (ICML)*. PMLR, 2019, 5558–5567.

[115] Salem, A., Bhattacharya, A., Backes, M., Fritz, M., and Zhang, Y. Updates-Leak: Data Set Inference and Reconstruction Attacks in Online Learning. In: *USENIX Security Symposium (USENIX Security)*. USENIX, 2020, 1291–1308.

[116] Salem, A., Zhang, Y., Humbert, M., Berrang, P., Fritz, M., and Backes, M. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In: *Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2019.

[117] Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., and Chen, L. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, 4510–4520.

[118] Schroff, F., Kalenichenko, D., and Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, 815–823.

[119] Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks. In: *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2018, 6103–6113.

[120] Shan, S., Wenger, E., Zhang, J., Li, H., Zheng, H., and Zhao, B. Y. Fawkes: Protecting Privacy against Unauthorized Deep Learning Models. In: *USENIX Security Symposium (USENIX Security)*. USENIX, 2020, 1589–1604.

[121] Shejwalkar, V. and Houmansadr, A. Membership Privacy for Machine Learning Models Through Knowledge Transfer. In: *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2021, 9549–9557.

[122] Shen, Y., Gu, J., Tang, X., and Zhou, B. Interpreting the Latent Space of GANs for Semantic Face Editing. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, 9240–9249.

[123] Shen, Y. and Zhou, B. Closed-Form Factorization of Latent Semantics in GANs. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, 1532–1540.

[124] Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership Inference Attacks Against Machine Learning Models. In: *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2017, 3–18.

[125] Simonyan, K. and Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *International Conference on Learning Representations (ICLR)*. 2015.

[126] Song, C. and Raghunathan, A. Information Leakage in Embedding Models. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2020, 377–390.

[127] Song, C. and Shmatikov, V. Overlearning Reveals Sensitive Attributes. In: *International Conference on Learning Representations (ICLR)*. 2020.

[128] Song, L. and Mittal, P. Systematic Evaluation of Privacy Risks of Machine Learning Models. In: *USENIX Security Symposium (USENIX Security)*. USENIX, 2021.

[129] Song, L., Shokri, R., and Mittal, P. Privacy Risks of Securing Machine Learning Models against Adversarial Examples. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2019, 241–257.

[130] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* (2014).

[131] Stanfill, M. H., Williams, M., Fenton, S. H., Jenders, R. A., and Hersh, W. R. A Systematic Literature Review of Automated Clinical Coding and Classification Systems. *J. Am. Medical Informatics Assoc.* (2010).

[132] Teerapittayanon, S., McDanel, B., and Kung, H. T. BranchyNet: Fast Inference via Early Exiting from Deep Neural Networks. In: *International Conference on Pattern Recognition (ICPR)*. 2016, 2464–2469.

[133] Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., and Cohen-Or, D. Designing an encoder for StyleGAN image manipulation. *ACM Transactions on Graphics* (2021).

[134] Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. Ensemble Adversarial Training: Attacks and Defenses. In: *International Conference on Learning Representations (ICLR)*. 2017.

[135] Truex, S., Liu, L., Gursoy, M. E., Yu, L., and Wei, W. Towards Demystifying Membership Inference Attacks. *CoRR abs/1807.09173* (2018).

[136] Wang, T., Zhang, Y., Fan, Y., Wang, J., and Chen, Q. High-Fidelity GAN Inversion for Image Attribute Editing. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, 11379–11388.

[137] Wang, T., Zhang, Y., and Jia, R. Improving robustness to model inversion attacks via mutual information regularization. In: *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2021, 11666–11673.

[138] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image Quality Assessment: from Error Visibility to Structural Similarity. *IEEE Transactions on Image Process* (2004).

[139] Wei, T., Chen, D., Zhou, W., Liao, J., Zhang, W., Yuan, L., Hua, G., and Yu, N. A Simple Baseline for StyleGAN Inversion. *CoRR abs/2104.07661* (2021).

[140] Xin, J., Tang, R., Lee, J., Yu, Y., and Lin, J. DeeBERT: Dynamic Early Exiting for Accelerating BERT Inference. In: *Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2020, 2246–2251.

[141] Xu, Z., Yu, X., Hong, Z., Zhu, Z., Han, J., Liu, J., Ding, E., and Bai, X. FaceController: Controllable Attribute Editing for Face in the Wild. In: *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2021, 3083–3091.

[142] Xue, M., Sun, S., Wu, Z., He, C., Wang, J., and Liu, W. SocialGuard: An adversarial example based privacy-preserving technique for social images. *Journal of Information Security and Applications* (2021).

[143] Yang, Z., Shao, B., Xuan, B., Chang, E.-C., and Zhang, F. Defending Model Inversion and Membership Inference Attacks via Prediction Purification. *CoRR abs/2005.03915* (2020).

[144] Yao, Y., Zhang, A., Zhang, Z., Liu, Z., Chua, T., and Sun, M. CPT: Colorful Prompt Tuning for Pre-trained Vision-Language Models. *CoRR abs/2109.11797* (2021).

[145] Yeh, C., Chen, H., Tsai, S., and Wang, S. Disrupting Image-Translation-Based DeepFake Algorithms with Adversarial Attacks. In: *Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2020, 53–62.

[146] Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In: *IEEE Computer Security Foundations Symposium (CSF)*. IEEE, 2018, 268–282.

[147] Yu, L., Zhang, W., Wang, J., and Yu, Y. Seqgan: sequence generative adversarial nets with policy gradient. In: *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2017, 2852–2858.

[148] Yuan, X., Chen, K., Zhang, J., Zhang, W., Yu, N., and Zhang, Y. Pseudo Label-Guided Model Inversion Attack via Conditional Generative Adversarial Network. In: *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2023.

[149] Yüksel, O. K., Simsar, E., Er, E. G., and Yanardag, P. LatentCLR: A Contrastive Learning Approach for Unsupervised Discovery of Interpretable Directions. *CoRR abs/2104.00820* (2021).

[150] Yurtsever, E., Lambert, J., Carballo, A., and Takeda, K. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access* (2020).

[151] Zagoruyko, S. and Komodakis, N. Wide Residual Networks. In: *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 2016.

[152] Zhai, R., Dan, C., He, D., Zhang, H., Gong, B., Ravikumar, P., Hsieh, C.-J., and Wang, L. MACER: Attack-free and Scalable Robust Training via Maximizing Certified Radius. In: *International Conference on Learning Representations (ICLR)*. 2020.

[153] Zhang, Y., Jia, R., Pei, H., Wang, W., Li, B., and Song, D. The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, 250–258.

[154] Zheng, T., Deng, W., and Hu, J. Cross-Age LFW: A Database for Studying Cross-Age Face Recognition in Unconstrained Environments. *CoRR abs/1708.08197* (2017).

[155] Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Learning to Prompt for Vision-Language Models. *International Journal of Computer Vision* (2022).

[156] Zhou, P., Han, X., Morariu, V. I., and Davis, L. S. Two-Stream Neural Networks for Tampered Face Detection. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, 1831–1839.

[157] Zhou, W., Xu, C., Ge, T., McAuley, J. J., Xu, K., and Wei, F. BERT Loses Patience: Fast and Robust Inference with Early Exit. In: *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2020.

[158] Zhu, J., Shen, Y., Zhao, D., and Zhou, B. In-Domain GAN Inversion for Real Image Editing. In: *European Conference on Computer Vision (ECCV)*. Springer, 2020, 592–608.

[159] Zhu, J., Krähenbühl, P., Shechtman, E., and Efros, A. A. Generative Visual Manipulation on the Natural Image Manifold. In: *European Conference on Computer Vision (ECCV)*. Springer, 2016, 597–613.

[160] Zhuang, P., Koyejo, O., and Schwing, A. G. Enjoy Your Editing: Controllable GANs for Image Editing via Latent Space Navigation. In: *International Conference on Learning Representations (ICLR)*. 2021.