ML-DOCTOR: Holistic Risk Assessment of Inference Attacks Against Machine Learning Models

Yugeng Liu[1]* Rui Wen[1]* Xinlei He[1] Ahmed Salem[1] Zhikun Zhang[1]
Michael Backes[1] Emiliano De Cristofaro[2] Mario Fritz[1] Yang Zhang[1]

[1]*CISPA Helmholtz Center for Information Security* [2]*UCL & Alan Turing Institute*

Abstract

Inference attacks against Machine Learning (ML) models allow adversaries to learn sensitive information about training data, model parameters, etc. While researchers have studied, in depth, several kinds of attacks, they have done so in isolation. As a result, we lack a comprehensive picture of the risks caused by the attacks, e.g., the different scenarios they can be applied to, the common factors that influence their performance, the relationship among them, or the effectiveness of possible defenses. In this paper, we fill this gap by presenting a first-of-its-kind holistic risk assessment of different inference attacks against machine learning models. We concentrate on four attacks – namely, membership inference, model inversion, attribute inference, and model stealing – and establish a threat model taxonomy.

Our extensive experimental evaluation, run on five model architectures and four image datasets, shows that the complexity of the training dataset plays an important role with respect to the attack's performance, while the effectiveness of model stealing and membership inference attacks are negatively correlated. We also show that defenses like DP-SGD and Knowledge Distillation can only mitigate *some* of the inference attacks. Our analysis relies on a modular re-usable software, ML-DOCTOR, which enables ML model owners to assess the risks of deploying their models, and equally serves as a benchmark tool for researchers and practitioners.[1]

1 Introduction

Over the last decade, research in Machine Learning (ML), and in particular Deep Learning, has made tremendous progress. However, the deployment and success of these technologies might be affected by attacks against ML models that prompt serious security and privacy risks. In particular, inference attacks [12, 18, 35, 41, 45, 50, 51, 53, 57, 59, 65] allow adversaries to infer information from a target ML model, e.g., about the training data, the model's parameters, and so on.

In this paper, we focus on four representative attacks: membership inference [53], model inversion [12], attribute inference [35], and model stealing [59]. The first three target a model's *training dataset*, aiming to, respectively, determine whether or not an exact data sample belongs to it, recover (part of) it, or predict properties that are not related to the model's original task. Model stealing involves reconstructing the target model's (non-public) parameters. Inference attacks can lead to severe consequences, including violating individuals' privacy, as ML models are often trained on sensitive data or compromising the model owner's intellectual property [10].

Overall, existing inference attacks have been studied under different threat models and experimental settings, albeit in isolation. This prompts the need for a holistic understanding of the risks caused by these attacks, such as the scenarios different inference attacks can be applied to, the common factors that influence these attacks' performance, and the relations among the attacks, as well as the overall effectiveness of defense mechanisms. To fill this gap, we perform a first-of-its-kind holistic security and privacy risk assessment of ML models, vis-à-vis four representative inference attacks.

Threat Model Taxonomy. Our work starts with a systematical categorization of the knowledge that an adversary might have to launch the attacks, along two dimensions: 1) access to a target model (white-box or black-box), 2) availability of an auxiliary dataset (partial training dataset, shadow dataset, or no dataset). We consider four types of state-of-the-art inference attacks and describe under which threat models they can be applied. This provides us with a comprehensive spectrum of the inference attack surface for ML models.

Experimental Evaluation. We perform an extensive measurement study of the attacks, jointly, over five different ML model architectures (AlexNet [27], ResNet18 [17], VGG19 [54], Xception [8], and SimpleCNN) and four image datasets (CelebA [34], Fashion-MNIST (FMNIST) [61], STL10 [9], and UTKFace [66]). Our analysis aims to answer three research questions: 1) What is the impact of dataset complexity on different attacks? 2) What is the impact of

*The first two authors made equal contributions.
[1]See https://github.com/liuyugeng/ML-Doctor.

overfitting on different attacks? 3) What is the relationship among different attacks?

Main Findings. Our analysis shows that the complexity of the target model's training dataset plays a major role in the accuracy of membership inference, model inversion, and model stealing. In particular, membership inference is much more effective on complex datasets, while the other two exhibit the opposite trend. For instance, for membership inference (with black-box access to the target model and a shadow dataset) against ResNet18, there is a 68.4% increase when going from a simple dataset (FMNIST) to a complex one (STL10).[2] On the other hand, model stealing achieves 0.524 agreement (the standard metric for this attack) on ResNet18 trained on STL10 but a much higher 0.927 on FMNIST. This stems from ML models being more prone to overfitting on complex datasets, which leads to better membership inference, whereas when an ML model is trained on a complex dataset, it is harder for an adversary to obtain a dataset with similar complexity (by querying the target model) to train their stolen model.

We also find that the performance of membership inference and model stealing are negatively correlated ($r = -0.821$), i.e., a target model with higher membership risks is less vulnerable to model stealing. This is due to the opposite effect of overfitting on these two attacks. Moreover, access to a partial training dataset does not significantly improve attack performance for membership inference, attribute inference, and model stealing. E.g., the accuracy for attribute inference (on ResNet18/CelebA) is 0.719 with a partial training dataset and 0.726 with a shadow dataset.

Defenses. We then evaluate two defense mechanisms, i.e., DP-SGD [1] and Knowledge Distillation (KD) [21], against all the inference attacks. Empirical results show that DP-SGD can mitigate membership inference attacks in general without damaging target models' utility significantly. Meanwhile, KD also reduces membership inference risks, but generally, to a lesser extent compared to DP-SGD. However, neither of them is effective against other inference attacks. This highlights the lack of a general, effective defense mechanism, and our work sheds light as to what extent/why.

ML-DOCTOR. To support the comprehensive evaluation of a wide range of inference attacks and defenses (current and future), we introduce a framework called ML-DOCTOR. This can be used by multiple entities and for multiple purposes. For instance, model owners can use it to seamlessly and meaningfully assess potential security and privacy risks before deploying their model. Also, as we make source code publicly available, researchers will be able to re-use ML-DOCTOR to benchmark new inference attacks and defense mechanisms. ML-DOCTOR follows a modular design, which easily supports the integration of additional inference attacks and defenses, as well as plugging in datasets, models, etc.

[2] We refer to both sample complexity and class diversity (see Section 6.3).

2 Threat Modeling

In this work, we focus on image classification, one of the most popular ML applications. In general, the goal of an ML classifier is to map a data sample to a label/class. The input to an ML model is a data sample, and the output is a vector of probabilities, or posteriors, with each element representing the likelihood of the sample belonging to a class.

We categorize the threat models for all the inference attacks considered in this paper along two dimensions, i.e., 1) *access to the target model* and 2) *auxiliary dataset*. In total, we consider five different scenarios.

Access to the Target Model. We consider two access settings: *white-box* and *black-box*. The former, denoted with $\mathcal{M}^{\mathsf{W}}$, means that an adversary has full information about the target model, including its parameters and architecture. In black-box attacks, denoted with $\mathcal{M}^{\mathsf{B}}$, the adversary can only access the target model in an API-like manner, e.g., they can query the target model and get the model's output. However, most of the existing black-box literature [14, 53, 62] also assumes that the adversary knows the target model's architecture which they use to build shadow models (see Section 3).

Overall, the white-box model captures scenarios where the target model's parameters are leaked, e.g., following a data breach or through reverse engineering, e.g., from pre-trained models deployed to mobile devices [21]. The black-box model encapsulates API access akin to features provided by Machine Learning as a Service (MLaaS) platforms.

Auxiliary Dataset. The adversary needs an auxiliary dataset in order to train their attack model. We consider three scenarios, in decreasing order of adversarial "strength": 1) *partial training dataset* ($\mathcal{D}^{\mathsf{P}}_{\mathsf{aux}}$), 2) *shadow dataset* ($\mathcal{D}^{\mathsf{S}}_{\mathsf{aux}}$), and 3) *no dataset* ($\mathcal{D}^{\mathsf{N}}_{\mathsf{aux}}$). In the first scenario, the adversary obtains parts of the actual training data from the target model (e.g., it is public knowledge), while in the last one, they have no information at all. In between is the $\mathcal{D}^{\mathsf{S}}_{\mathsf{aux}}$ setting, where the adversary gets a "shadow" dataset from the same distribution as the target model's training data (see Section V-C in [53] for a discussion on how to generate such data, using, e.g., through model-based or statistics-based synthesis, or noisy real data).

Considered Settings. Overall, the two different types of model access and the three types of auxiliary dataset availability lead to six threat models. In the rest of the paper, we consider five of them: $\langle \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$, $\langle \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$, $\langle \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$, $\langle \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$, and $\langle \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{N}}_{\mathsf{aux}} \rangle$. We do not experiment with black-box access and no auxiliary dataset, as this is unlikely to yield successful attacks.

3 Inference Attacks

In this section, we present the four inference attacks measured in this paper. Specifically, we consider membership inference (MemInf), model inversion (ModInv), attribute inference (AttrInf), and model stealing (ModSteal). The first three are designed to infer information about a target ML

Auxiliary Dataset	Model Access	
	Black-Box ($\mathcal{M}^{\mathsf{B}}$)	White-Box ($\mathcal{M}^{\mathsf{W}}$)
Partial ($\mathcal{D}^{\mathsf{P}}_{\mathsf{aux}}$)	MemInf, ModSteal	MemInf, AttrInf
Shadow ($\mathcal{D}^{\mathsf{S}}_{\mathsf{aux}}$)	MemInf, ModSteal	MemInf, AttrInf, ModInv
No ($\mathcal{D}^{\mathsf{N}}_{\mathsf{aux}}$)	-	ModInv

Table 1: Different attacks under different threat models.

model's training data, while the last one aims to steal the target model's parameters.

Different attacks can be applied to different threat models; see Table 1. For each attack and each threat model, we concentrate on one representative state-of-the-art method.

3.1 Membership Inference

Membership Inference (MemInf) [53] against ML models involves an adversary aiming to determine whether or not a target data sample was used to train a target ML model. More formally, given a target data sample x_{target}, (the access to) a target model $\mathcal{M}$, and an auxiliary dataset $\mathcal{D}_{\mathsf{aux}}$, a membership inference attack can be defined as:

$$\mathsf{MemInf} : x_{\mathsf{target}}, \mathcal{M}, \mathcal{D}_{\mathsf{aux}} \rightarrow \{\textit{member}, \textit{non-member}\}$$

where $\mathcal{M} \in \{\mathcal{M}^{\mathsf{B}}, \mathcal{M}^{\mathsf{W}}\}$ and $\mathcal{D}_{\mathsf{aux}} \in \{\mathcal{D}^{\mathsf{P}}_{\mathsf{aux}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}}\}$.

Membership inference has been extensively studied in literature [6, 7, 23, 29, 31, 37, 49, 51, 53]. Inferring membership of a target sample prompts severe privacy threats; for instance, if an ML model for drug dose prediction is trained using data from patients with a certain disease, then inclusion in the training set inherently leaks the individuals' health status. Overall, MemInf is also often a signal that a target model is "leaky" and can be a gateway to additional attacks [10].

In the following, we illustrate how to implement membership inference (MemInf) under different threat models.

Black-Box/Shadow $\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$ **[51].** We start with the most common and difficult setting for the attack [51, 53], whereby the adversary has black-box access ($\mathcal{M}^{\mathsf{B}}$) to the target model and a shadow auxiliary dataset ($\mathcal{D}^{\mathsf{S}}_{\mathsf{aux}}$).

The adversary first splits the shadow dataset into two parts and uses one to train a shadow model on the same task. Next, the adversary uses the entire shadow dataset to query the shadow model. For each querying sample, the shadow model returns its posteriors and the predicted label: if the sample is part of the shadow model's training set, the adversary labels it as a member and as a non-member otherwise. With this labeled dataset, the adversary trains an attack model, which is a binary membership classifier. Finally, to determine whether a data sample is a member of the target model's training dataset, the sample is fed to the target model, and the posteriors and the predicted label (transformed to a binary indicator on whether the prediction is correct) are fed to the attack model.

Black-Box/Partial $\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$ **[51].** If the adversary has black-box access to the target model and a partial training dataset, the attack method is very similar to that for $\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$. However, the adversary does not need to train a shadow model; rather, they use the partial training dataset as the ground truth for membership and directly train their attack model.

White-Box/Shadow $\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$ **[38].** Nasr et al. [38] introduce an attack in the white-box setting with either a shadow or a partial training dataset as the auxiliary dataset.[3] In the former, similar to $\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$, the adversary uses $\mathcal{D}^{\mathsf{S}}_{\mathsf{aux}}$ to train a shadow model to mimic the behavior of the target model and to generate data to train their attack model. As the adversary has white-box access to the target model, they can also exploit the target sample's gradients with respect to the model parameters, embeddings from different intermediate layers, classification loss, and prediction posteriors (and label).

White-Box/Partial $\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$ **[38].** The attack methodology here is almost identical to the black-box counterpart. The only difference is that the adversary can use the same set of features as the attack model for $\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$.

3.2 Model Inversion

Model inversion attacks (ModInv) [12] aim to reconstruct data samples from a target ML model, i.e., they allow an adversary to directly learn information about the training dataset.

For instance, in a facial recognition system, a ModInv adversary tries to learn the facial data of a victim whose data is used to train the model. Model inversion requires the adversary to have white-box access to the target model; this is due to the fact that the attack needs to perform back-propagation over the target model's parameters (detailed below).

Formally, we define model inversion as:

$$\mathsf{ModInv} : \mathcal{M}^{\mathsf{W}}, \mathcal{D}_{\mathsf{aux}} \rightarrow \{\textit{training samples}\}$$

where $\mathcal{D}_{\mathsf{aux}} \in \{\mathcal{D}^{\mathsf{N}}_{\mathsf{aux}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}}\}$.

We consider two types of model inversion attacks: the one proposed by Fredrikson et al. [12], which aims to reconstruct a representative sample for each class of the target model, and that by Zhang et al. [65], which aims to synthesize the training dataset. These two attacks follow different threat models, which we discuss next.

White-Box/No Auxiliary $\langle \mathsf{ModInv}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{N}}_{\mathsf{aux}} \rangle$ **[12].** The method by Fredrikson et al. [12] assumes the adversary has white-box access to the target model[4] and does not need any auxiliary dataset. For each class of the target model, the adversary first creates a noise sample, feeds this sample to the model, and gets the posteriors. The adversary then uses back-propagation over the target model's parameters to optimize the input sample so that the corresponding posterior of the class can exceed a pre-set threshold. Once the threshold is reached, the optimized sample is the representative sample of that class, i.e., the attack output.

[3] The attack by Nasr et al. [38] was originally designed for the partial training dataset setting, but it can be adapted to the shadow dataset setting.

[4] Fredrikson et al. [12] also introduce a model inversion attack where the adversary only has black-box access to the target model; however, its performance is not as good and therefore we do not consider it.

White-Box/Shadow $\langle \mathsf{ModInv}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$ **[65].** The attack by Zhang et al. [65] uses a shadow dataset to enhance the quality of the reconstructed samples by training a generative adversarial network (GAN) [15]. First, the adversary trains a GAN with a shadow dataset. Next, the adversary optimizes the inputs to the GAN's generator, i.e., the noise, to find those GAN-generated samples that can achieve high posteriors on the target model. These samples are the attack's final outputs. In other words, this attack performs optimization on the inputs to the GAN instead of the samples to the target model directly [12]. Since the GAN is capable of generating high-quality samples, the attack's final outputs will be more realistic. Moreover, due to the fact that GAN can generate multiple samples, this attack is able to generate multiple samples for each class of the target model as well.

3.3 Attribute Inference

An ML model may learn extra information about the training data that is not related to its original task; e.g., a model predicting age from profile photos can also learn to predict race [35, 57]. Attribute inference (AttrInf) aims to exploit such unintended information leakage.

State-of-the-art attacks usually rely on the embeddings of a target sample (x_{target}) obtained from the target model to predict the sample's target attributes. Thus, the adversary is assumed to have white-box access to the target model. Formally, attribute inference is defined as:

$$\mathsf{AttrInf} : x_{\mathsf{target}}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}_{\mathsf{aux}} \rightarrow \{\textit{target attributes}\}$$

where $\mathcal{D}_{\mathsf{aux}} \in \{\mathcal{D}^{\mathsf{P}}_{\mathsf{aux}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}}\}$ can either be a partial training dataset or a shadow dataset.

White-Box/Shadow and Partial [35, 57]. Both attacks, i.e., $\langle \mathsf{AttrInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$ [35] and $\langle \mathsf{AttrInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$ [57], follow a similar attack methodology. The only difference lies in the dataset used to train the attack model. In both cases, the adversary is assumed to know the target attributes of the auxiliary dataset. Then, they use the embeddings, and the target attributes to train a classifier to mount the attack.

3.4 Model Stealing

The goal of model stealing attacks (ModSteal) [41, 59], aka model extraction, is to extract the parameters from a target model. Ideally, an adversary will be able to obtain a model (the "stolen" model) with very similar performance as the target model. More formally:

$$\mathsf{ModSteal} : \mathcal{M}^{\mathsf{B}}, \mathcal{D}_{\mathsf{aux}} \rightarrow \mathcal{M}^{\mathsf{C}}$$

where $\mathcal{M}^{\mathsf{C}}$ is the stolen model and $\mathcal{D}_{\mathsf{aux}} \in \{\mathcal{D}^{\mathsf{P}}_{\mathsf{aux}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}}\}$.

Model stealing prompts severe security risks. For instance, as it is often difficult to train an advanced ML model (e.g., due to the lack of data or computing resources), stealing a trained model inherently constitutes intellectual property theft. Also, as many other attacks, such as adversarial examples [46], require white-box access to the target ML model, model stealing can be a stepping stone to perform these attacks.

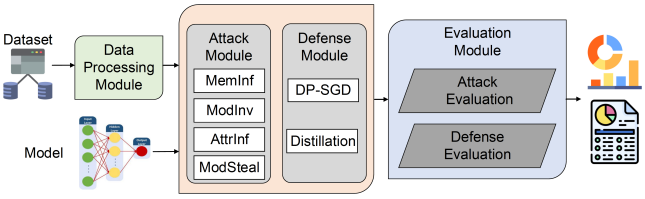

Figure 1: Overview of ML-DOCTOR's modules.

Black-Box/Partial and Shadow [59]. In this paper, we concentrate on the attacks proposed by Tramèr et al. [59], for $\langle \mathsf{ModSteal}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$ and $\langle \mathsf{ModSteal}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$. The adversary is assumed to have knowledge of the target model's architecture, and both attacks follow a similar methodology. Specifically, the adversary uses data samples from their auxiliary dataset ($\mathcal{D}^{\mathsf{P}}_{\mathsf{aux}}$ or $\mathcal{D}^{\mathsf{S}}_{\mathsf{aux}}$) to query the target model and get the corresponding posteriors. Then, they use them to train the stolen model, with the posteriors as the ground truth.

4 ML-DOCTOR

In this section, we introduce ML-DOCTOR, a modular framework geared to evaluate the four inference attacks, as well as the two defenses (see Section 7), considered in this paper.

Prior work has proposed software tools to evaluate attacks against ML models, such as adversarial examples [32, 44], backdoor attacks [42], and membership inference [36]. To the best of our knowledge, ML-DOCTOR is the first framework that jointly considers different types of inference attacks.

Modules. In Figure 1, we report the four different modules of ML-DOCTOR:

1. **Data Processing.** This module processes the datasets to mount different attacks. It also involves data preprocessing methods, e.g., normalization.
2. **Attack.** This module performs the actual inference attacks. At the moment, it supports ten different attacks belonging to four different attack types (see Section 3).
3. **Defense.** We currently support two representative mitigation techniques for inference attacks against ML models, as discussed later in Section 7.
4. **Evaluation.** This module is used to evaluate the performance of attacks and defenses.

The modular design of ML-DOCTOR allows to easily integrate additional attacks and defense mechanisms, as well as plugging in any dataset or model.

Using ML-DOCTOR. A user needs to input their target model and its training dataset to use ML-DOCTOR. This is to achieve a full-fledged privacy risk assessment. We envision ML-DOCTOR to be used for the following purposes:

- As it supports a systematic taxonomy of different threat models for inference attacks, ML-DOCTOR enables model owners to obtain an overview of the threats their model may face when deployed in the real world.

- ML-DOCTOR provides a holistic assessment of different attacks, as well as the effectiveness of possible defenses. To our best knowledge, this is the first tool to provide such a comprehensive analysis of inference attacks.
- Researchers can re-use ML-DOCTOR as a benchmark tool to experiment with new inference attacks and defenses in the future. ML-DOCTOR's data processing and evaluation modules can be seamlessly re-used by other attacks or defenses. Moreover, the maturity of the topic, as demonstrated by the state of the art [12, 26, 51, 53, 56, 59, 65], suggests that new attacks against ML models are very likely to fall into one of the threat models summarized in our taxonomy (Section 2). This means that new attacks/defenses can be easily implemented within ML-DOCTOR's attack and defense modules.
- Since ML-DOCTOR follows a modular design, the communication between different modules is implemented using an API-based approach. To include a new attack/defense, one only needs to specify the attack/defense models' architecture in the attack/defense module, which can be easily done with the support of the current deep learning libraries. To further extend ML-DOCTOR into different domains like text or audio, users can re-implement the processing function and attack methods in the corresponding modules, and reuse other modules directly.

5 Experimental Settings

5.1 Experimental Protocol

We first select four benchmark datasets (see Section 5.2) and five state-of-the-art ML models (see Section 5.3) to train a total of 20 target models. These are used to evaluate different attacks (see Section 3) and defenses (see Section 7). For each dataset, we partition it into four parts (see Section 5.2), including target training dataset, target testing dataset, shadow training dataset, and shadow testing dataset, to comply with the different threat models discussed in Section 2.

We then submit each target model and the corresponding dataset partition to ML-DOCTOR, running the attacks (see Section 5.4) and applying the defenses (see Section 7). Finally, we use the evaluation module of ML-DOCTOR to summarize the results and perform a comprehensive analysis to answer the research questions listed in Section 6.1.

5.2 Datasets

For the sake of this paper, we experiment with four datasets:

- **CelebA [34]** contains 202,599 face images, each is associated with 40 binary attributes. We select and combine 3 attributes out of 40, including *HeavyMakeup*, *MouthSlightlyOpen*, and *Smiling* to form our target models' classes/labels, leading to 8-class classification.
- **FMNIST (Fashion-MNIST) [61]** is also an image dataset containing 70,000 gray-scale images equally distributed among 10 different classes, including T-shirt, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot.
- **STL10 [9]** is a 10-class image dataset, each contains 1,300 images. The classes include airplane, bird, car, cat, deer, dog, horse, monkey, ship, and truck.
- **UTKFace [66]** has 23,000 face images associated with age, gender, and race. We consider the images from the largest four races (White, Black, Asian, and Indian) in the dataset and use race as the label for the corresponding target models. This leaves us with 22,012 images.

Note that all the samples in the datasets are re-sized to 32×32 pixels. This is common practice in ML and ensures that the comparison among different datasets is fair. We randomly split each dataset into four equal disjoint parts:

1. **Target Training Dataset** is used to train all the target models and to evaluate the performance of all membership inference attacks and model inversion attacks. For attacks that require a partial training dataset ($\mathcal{D}^{\mathsf{P}}_{\mathsf{aux}}$), i.e., $\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$, $\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$, $\langle \mathsf{AttrInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$, $\langle \mathsf{AttrInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$, and $\langle \mathsf{ModSteal}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$, we randomly select 70% samples from the target training dataset.
2. **Target Testing Dataset** is used to evaluate the performance of the target model. It is also used to evaluate the performance of all membership inference, attribute inference, and model stealing attacks.
3. **Shadow Training Dataset** is used to train all the attack models that require a shadow auxiliary dataset.
4. **Shadow Testing Dataset** is used to train two membership inference attack models, i.e., $\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$ and $\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$, that require a shadow dataset as the auxiliary dataset.

In a nutshell, all the datasets we choose in this paper are benchmark datasets for evaluating inference attacks against ML models [19, 22, 25, 33, 56]. These datasets have different numbers of classes and cover a variety of objects, e.g., human faces, transportation tools, and animals. Also, the images are ranging from gray-scale to colored in different datasets. Note that ML-DOCTOR is not bounded by certain types of datasets. We plan to extend ML-DOCTOR to other security-related datasets such as network scans, malware traces, etc.

5.3 Target Models

We focus on five model architectures, including AlexNet [27], ResNet18 [17], VGG19 [54], Xception [8], and SimpleCNN (containing 2 convolutional layers and 2 fully connected layers) for all the four datasets introduced above. In total, we train 20 different target models.

For training, we set the mini-batch size to 64 and use cross-entropy as the loss function. We use stochastic gradient descent (SGD) as the optimizer with a weight decay of 5e-4 and momentum of 0.9. Each target model is trained for 300 epochs.

The learning rate is 1e-2 before 50 epochs, 1e-3 from 50-100 epochs, and 1e-4 until the end. All target models' training and testing accuracy are shown in Table 2. Note that for shadow models used in the membership inference attacks, we train them following the same process as the target models.

5.4 Attack Models

Membership Inference. Recall that there are four different scenarios for MemInf; we establish two types of attack models: one for the black-box and the other for the white-box setting. For black-box, our attack model has two inputs; the target sample's ranked posteriors and a binary indicator on whether the target sample being predicted correctly. Each input is first fed into a different 2-layer MLP (Multilayer Perceptron), then the two obtained embeddings are concatenated together and fed into a 4-layer MLP. For white-box, we have four inputs for this attack model, like the one used by Nasr et al. [38], including the target sample's ranked posteriors, classification loss, gradients of the parameters of the target model's last layer, and one-hot encoding of its true label. Each input is fed into a different neural network, and the resulted embeddings are concatenated together as input to a 4-layer MLP. We use ReLU as the activation function for the attack models. The mini-batch size is set to 64, and cross-entropy is the loss function. We use Adam as the optimizer, with learning rate of 1e-5. The attack model is trained for 50 epochs. We adopt *accuracy, F1 score, and AUC (area under the ROC curve) score* as the evaluation metrics.

Model Inversion. For $\langle \mathsf{ModInv}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{N}}_{\mathsf{aux}} \rangle$, following the attack settings in [12], we set the threshold to 0.999, learning rate to 1e-2, maximum iteration to 3,000, and early stop criteria to 100. This attack can only generate one representative sample for each class of the target model. To evaluate the quality of the reconstructed sample, we first obtain an average sample from all samples of each target class, then calculate the mean squared error (MSE) between this average sample and the reconstructed sample. Finally, we use the average of the MSE values for all target classes as the evaluation metric. Note that smaller MSE within the same dataset indicates better attack performance. However, for different datasets, different MSE can be caused by the different normalization effects. For example, FMNIST has the highest MSE; this is due to the characteristic of FMNIST: most of the pixels are normalized to -1 or 1.

For $\langle \mathsf{ModInv}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$ [65], we first use the shadow training dataset to train a DCGAN [48] with the generator's noise dimension setting to 100. For the attack, we set the learning rate to 1e-3, momentum to 0.9, loss ratio λ to 100, iteration round to 1,500, and clip range to 1. To evaluate the effectiveness of this attack, we use the same approach by Zhang et al. [65], i.e., we train an evaluation classifier on the identical task of the target model and use this evaluation classifier to check whether the reconstructed samples can be recognized correctly. We use *accuracy and macro-F1 score* of this evaluation classifier on reconstructed samples as the performance metrics.

	CelebA	FMNIST	STL10	UTKFace
AlexNet	1.000 / 0.680	1.000 / 0.884	1.000 / 0.522	1.000 / 0.792
ResNet18	1.000 / 0.742	1.000 / 0.909	1.000 / 0.524	1.000 / 0.852
VGG19	1.000 / 0.734	1.000 / 0.905	1.000 / 0.587	1.000 / 0.834
Xception	1.000 / 0.735	1.000 / 0.916	1.000 / 0.574	1.000 / 0.846
SimpleCNN	1.000 / 0.707	1.000 / 0.903	1.000 / 0.517	1.000 / 0.818

Table 2: Performance of target models, namely, training/testing accuracy for each setting.

Attribute Inference. We only use two datasets, namely, CelebA and UTKFace, to evaluate this attack as both of them have extra attributes that can be used as the target attributes. For the former, we utilize *Male*/*Female* and *Young*/*Old* as the target attribute, resulting in a combination of four target attribute values. For the latter, we choose *Male*/*Female* as the target attribute.

Our attack model is a 2-layer MLP; its input is the target sample's embeddings from the second-to-last layer of the target model. We use cross-entropy as the loss function and Adam as the optimizer with learning rate of 1e-3. The attack model is trained for 50 epochs. For the evaluation metrics, we use *accuracy* and *F1 score* (*macro-F1 score* for CelebA as the attack has four target attributes).

Model Stealing. We evaluate the model stealing attack over all the 20 target models. For the stolen model, we use the same architecture as the target model [59]. Each stolen model is trained using the MSE loss and SGD as the optimizer (momentum 0.9 and learning rate 1e-2) for 50 epochs. *Accuracy and agreement* are used to assess the success of the attack, where agreement represents the proportion of samples in the target testing dataset on which the target and the stolen models make the same prediction.

6 Experimental Evaluation

In this section, we build on ML-DOCTOR to provide a holistic assessment of inference attacks against ML models. Experiments are performed on an NVIDIA DGX-A100 server with Ubuntu 18.04 operating system. We run all the experiments 10 times, reporting mean and standard deviation values.

6.1 Research Questions

We start by assessing the overall performance of the four attacks. We then analyze the impact of dataset and overfitting on the attack performance, as well as the relationship among different attacks. Concretely, we aim to answer the following key research questions:

- *RQ1:* What is the impact of dataset complexity on different attacks?
- *RQ2:* What is the impact of overfitting on different attacks?
- *RQ3:* What is the relationship among different attacks?

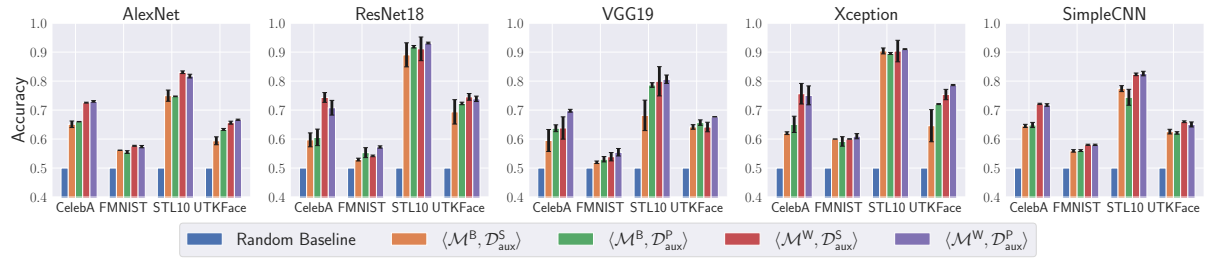

Figure 2: Accuracy of membership inference attacks (MemInf) under different threat models, datasets, and target model architectures.

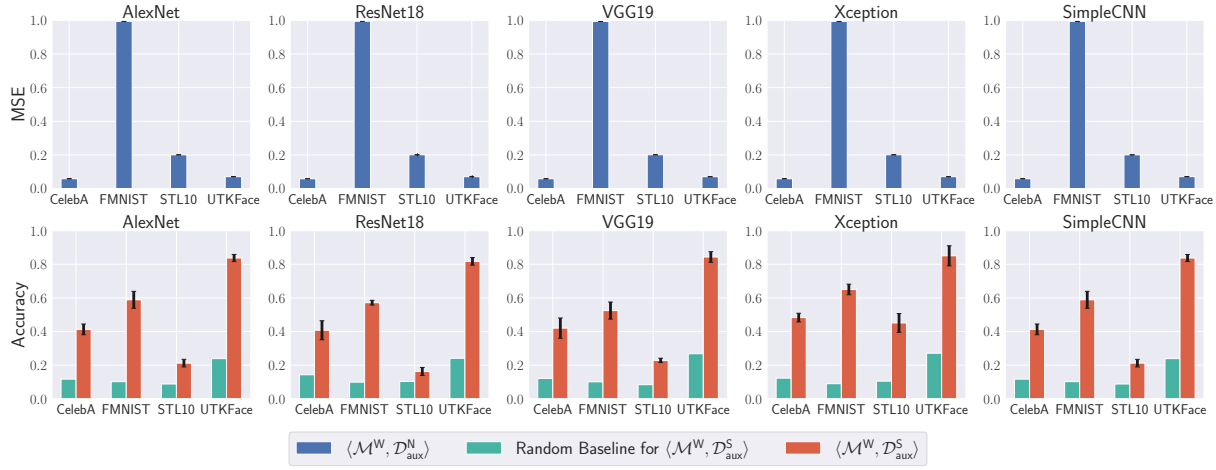

Figure 3: MSE ($\langle \text{ModInv}, \mathcal{M}^{W}, \mathcal{D}^{N}_{aux} \rangle$) and accuracy ($\langle \text{ModInv}, \mathcal{M}^{W}, \mathcal{D}^{S}_{aux} \rangle$) of model inversion attacks (ModInv) under different threat models, datasets, and target model architectures.

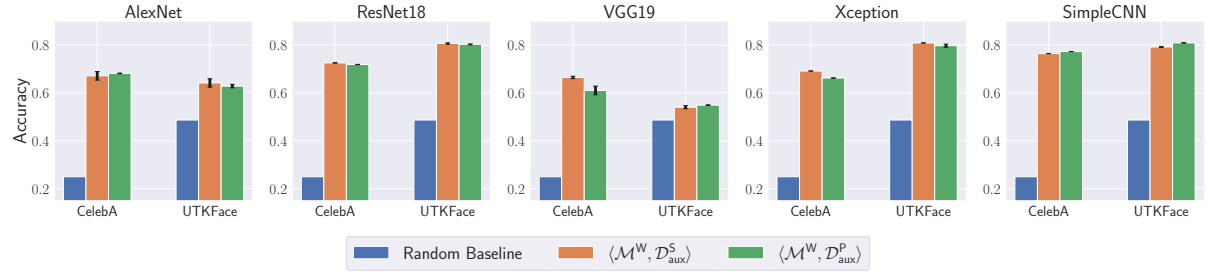

Figure 4: Accuracy of attribute inference attacks (AttrInf) under different threat models, datasets, and target model architectures.

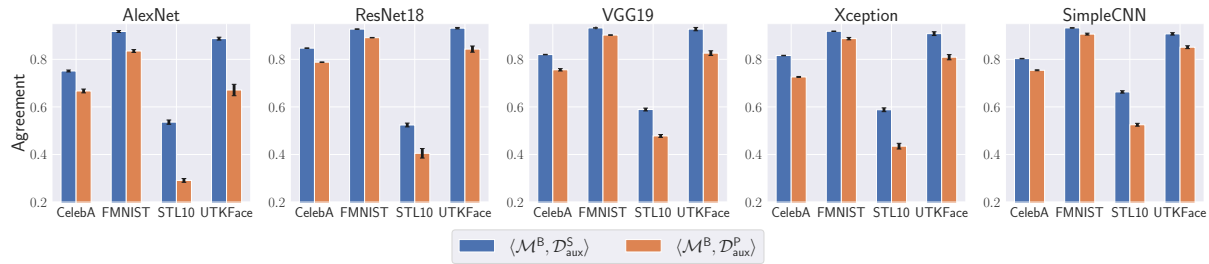

Figure 5: Agreement of model stealing attacks (ModSteal) under different threat models, datasets, and target model architectures.

6.2 Attack Performance

Membership Inference. In Figure 2, we report the accuracy of MemInf. We observe that the attack achieves high accuracy on CelebA, STL10, and UTKFace. For instance, the attack accuracy of $\langle \text{MemInf}, \mathcal{M}^{W}, \mathcal{D}^{S}_{aux} \rangle$ on ResNet18 trained on the STL10 dataset is 0.911. On the other hand, the performance on FMNIST is not strong, as models trained on FMNIST generalize well on non-member data samples—in other words, there is less overfitting [53] (see Section 6.4). We also report the F1 score and AUC score in Appendix A (Figure 11 and Figure 12), and the corresponding ROC curves are depicted in Appendix A (Figure 13, Figure 14, Figure 15, and Figure 16).

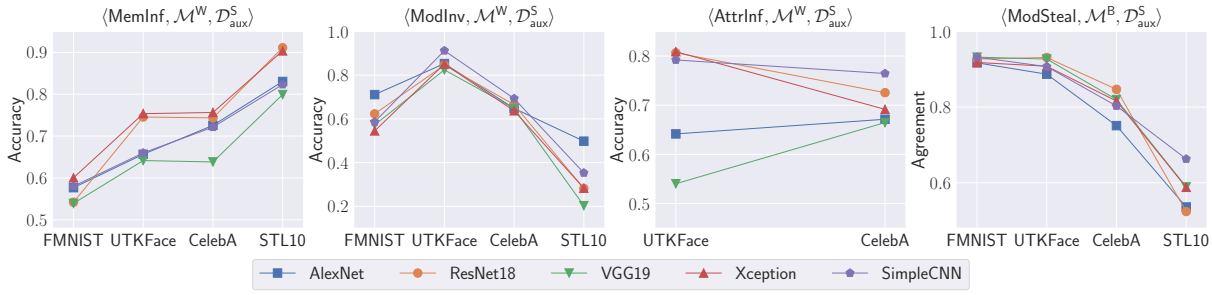

Figure 6: The relation between dataset complexity and attack performance. For MemInf, ModInv, and AttrInf. we use accuracy, for ModSteal, agreement.

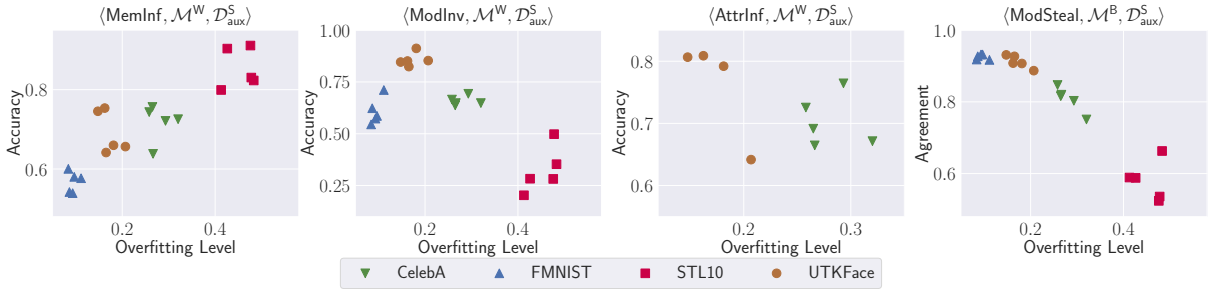

Figure 7: The relation between overfitting level (on target models) and attack performance. For MemInf, ModInv, and AttrInf, we use accuracy, for ModSteal, agreement.

An adversary with white-box access to the target model generally achieves better performance than the one with black-box access. For instance, the accuracy of $\langle \mathsf{MemInf}, \mathcal{M}^W, \mathcal{D}^S_{aux} \rangle$ is higher than that of $\langle \mathsf{MemInf}, \mathcal{M}^B, \mathcal{D}^S_{aux} \rangle$, except for Xception on STL10 and FMNIST and VGG19 on UTKFace (see Figure 2). A similar observation can be drawn from $\langle \mathsf{MemInf}, \mathcal{M}^W, \mathcal{D}^P_{aux} \rangle$ and $\langle \mathsf{MemInf}, \mathcal{M}^B, \mathcal{D}^P_{aux} \rangle$; this is expected as the adversary can exploit more information in the white-box setting. In particular, we find that the classification loss possesses the strongest signal among others for the attack [38]. Meanwhile, partial training dataset also leads to better membership inference performance than the shadow dataset; however, the effect is less pronounced.

Model Inversion. Next, we measure the performance of model inversion (see Figure 3). As discussed in Section 5.4, we use different metrics to evaluate these two attacks, i.e., MSE for $\langle \mathsf{ModInv}, \mathcal{M}^W, \mathcal{D}^N_{aux} \rangle$ and accuracy for $\langle \mathsf{ModInv}, \mathcal{M}^W, \mathcal{D}^S_{aux} \rangle$, due to their different design. Thus, we cannot directly compare them. Rather, we evaluate their attack performance qualitatively (see Figure 10 in Appendix A for two examples) and discover that the images generated by $\langle \mathsf{ModInv}, \mathcal{M}^W, \mathcal{D}^S_{aux} \rangle$ are more realistic than those by $\langle \mathsf{ModInv}, \mathcal{M}^W, \mathcal{D}^N_{aux} \rangle$. This is due to the capability of GAN for generating high-quality samples. For $\langle \mathsf{ModInv}, \mathcal{M}^W, \mathcal{D}^S_{aux} \rangle$, we also use macro-F1 as the metric; see Figure 17 in Appendix A.

Attribute Inference. The accuracy of the attribute inference attacks is shown in Figure 4. We also report macro-F1 score for CelebA and F1 score for UTKFace in Figure 18 (see Appendix A). The corresponding ROC curves are reported in Figure 19 (see Appendix A). In general, the attacks work quite well. For instance, both $\langle \mathsf{AttrInf}, \mathcal{M}^W, \mathcal{D}^S_{aux} \rangle$ and $\langle \mathsf{AttrInf}, \mathcal{M}^W, \mathcal{D}^P_{aux} \rangle$ reach around 0.800 accuracy for ResNet18 trained on UTKFace. The F1 scores with respect to these two models are both about 0.779. We also notice that using a partial training dataset does not provide the adversary with many advantages compared to using a shadow dataset. In some cases, partial training dataset even yields worse performance, as in the case of VGG19 trained on CelebA.

Model Stealing. We report the agreement (Figure 5) and accuracy (Figure 20 in Appendix A) to evaluate model stealing attacks. Overall, ModSteal has strong performance. For instance, $\langle \mathsf{ModSteal}, \mathcal{M}^B, \mathcal{D}^S_{aux} \rangle$ for ResNet18 trained on FMNIST achieves an agreement of 0.927. Similar to attribute inference, we observe that using a partial training dataset as the auxiliary dataset has a lower performance than the shadow dataset for model stealing. One reason might be that using a partial training dataset querying the target model results in more confident posteriors (low entropy), which contain less information for the adversary to exploit.

6.3 The Role of the Datasets

To answer the first research question, we plot the relationship between dataset complexity and attack performance in Figure 6. (The x-axis represents the datasets and the y-axis shows the attack performance, and each node corresponds to one attack against one target model). Due to space limitations, we only show one plot for one threat model for each attack.

Dataset Complexity. As mentioned before, all the samples in the four datasets are re-sized to 32×32 pixels. FMNIST is the simplest dataset as it only contains gray-scale images,

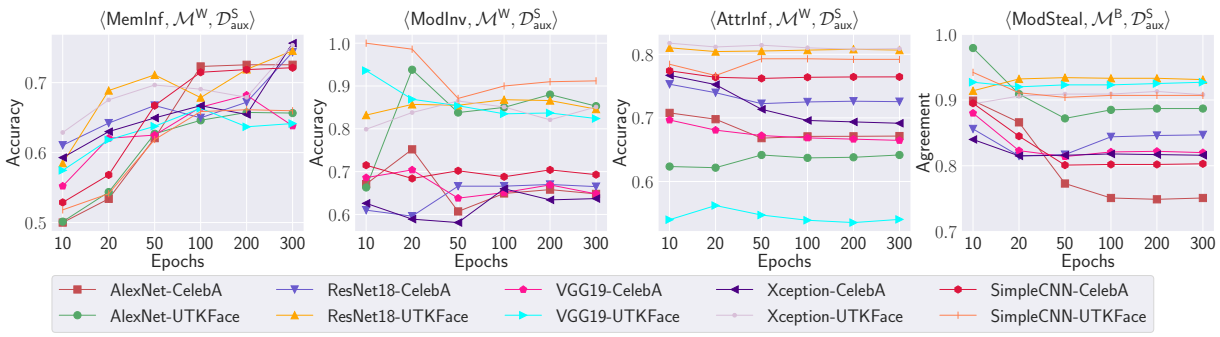

Figure 8: The relation between the number of epochs and attack performance. For MemInf, ModInv, and AttrInf, we use accuracy, for ModSteal, agreement.

followed by UTKFace, which consists of (full-color) human faces, and CelebA, which has 10 times more images than UTKFace. The most complex dataset is STL10, as it contains images of 10 diverse classes, ranging from cat to ship.

Results. Overall, the complexity of the dataset does have a significant effect on MemInf and ModSteal. More precisely, more complex datasets lead to better membership inference but worse model stealing performance. Ostensibly, this is due to the fact that a complex dataset is harder for a model to generalize on, and thus more prone to overfitting, which results in better membership inference attack [53], whereas when a model is trained on a complex dataset, it is harder for an adversary to obtain a dataset with similar complexity (by querying the target model) to train their stolen model.

We also observe that model inversion is less effective on STL10 than on UTKFace and CelebA, whereas there is no strong influence of dataset complexity on attribute inference; this might be due to the different target classes of our attacks on these two datasets (see Section 5.4). Note that we also investigate the complexity of the target model structure on the attack performance but do not observe any clear relation.

6.4 The Effect of Overfitting

To answer the second research question, we analyze the effect of target models' overfitting on inference attacks' performance. Concretely, we adopt two metrics to quantify overfitting in each target model: 1) the difference between the training accuracy and the testing accuracy of the target model, referred to as the *overfitting level*, and 2) the number of epochs used to train the target model [51].

Overfitting Level. The relation between overfitting level and attack performance is shown in Figure 7. First, we observe that different datasets have different overfitting levels, and this correlates well with the dataset complexity (see Section 6.3). Specifically, the largest (smallest) overfitting level happens on the most (least) complex dataset in our experiments, i.e., STL10 (FMNIST).

Overall, overfitting does have a significant impact on MemInf ($\langle \text{MemInf}, \mathcal{M}^{W}, \mathcal{D}^{S}_{aux} \rangle$). That is, a higher overfitting level leads to better membership inference. This is in line with previous analysis [51, 53], and is expected, as an overfitted model provides more confident predictions on its member samples (reflected on the posteriors) than on non-member samples, which can be exploited by the attack model to effectively differentiate them.

Meanwhile, model stealing displays a completely opposite trend, i.e., it is more difficult to steal a highly overfitted model. This can be explained by the fact that an overfitted model memorizes its training dataset to a large extent, and an adversary usually does not have the ability to get the exact training dataset; thus, the stolen model is likely to be dissimilar to the target model. Also, model inversion tends to have better performance on less overfitted models, except for FMNIST. We believe this is due to the quality of the GAN employed in the attack. For attribute inference, we do not observe a clear relationship between attack performance and overfitting level.

Number of Epochs. The relation between the number of epochs and attack performance (on UTKFace and CelebA) is shown in Figure 8. First, we find that all attacks' performance becomes steady after 100 epochs; this is reasonable since 100 epochs are usually enough to train good target models, and further training does not cause an obvious effect on overfitting. Second, the performance of membership inference increases from 10 epochs until 100 epochs, while model stealing shows the opposite trend. This observation echoes our previous argument that a highly overfitted model is easier to be attacked by membership inference but harder to be stolen. For model inversion and attribute inference, the attack performance only has slight fluctuations with a different number of epochs.

6.5 Relation Among Different Attacks

Next, we analyze the relationship between different attacks under the same threat model, which corresponds to our third research question. In total, we consider all the six pairs of attacks (as depicted in Table 1) including MemInf and ModSteal under $\langle \mathcal{M}^{B}, \mathcal{D}^{S}_{aux} \rangle$, MemInf and ModSteal under $\langle \mathcal{M}^{B}, \mathcal{D}^{P}_{aux} \rangle$, MemInf and AttrInf under $\langle \mathcal{M}^{W}, \mathcal{D}^{S}_{aux} \rangle$, MemInf and AttrInf under $\langle \mathcal{M}^{W}, \mathcal{D}^{P}_{aux} \rangle$, MemInf and ModInv under $\langle \mathcal{M}^{W}, \mathcal{D}^{S}_{aux} \rangle$, and AttrInf and ModInv under $\langle \mathcal{M}^{W}, \mathcal{D}^{S}_{aux} \rangle$.

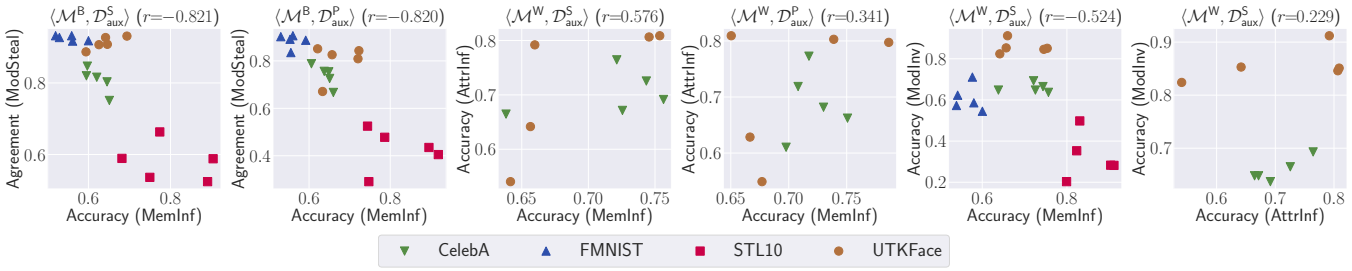

Figure 9: The relation between different attacks under the same threat model. For MemInf, ModInv, and AttrInf, we use accuracy, for ModSteal, agreement.

Figure 9 shows that there is a strong negative correlation between membership inference and model stealing ($\langle \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$) with respect to their accuracy ($r = -0.821$). Specifically, worse membership inference corresponds to better model stealing. This follows from the discussion around overfitting (Section 6.4). The correlation with respect to other evaluation metrics is reported in Figure 21 in Appendix A. We also observe a strong negative correlation between membership inference and model inversion, except for model inversion performing worse on FMNIST than on CelebA and UTKFace. We speculate this is due to the capability of the DCGAN used in the model inversion attack. FMNIST contains gray-scale images, while CelebA and UTKFace are both face datasets. DCGAN, in general, is more effective in generating human faces [48], which results in model inversion's better performance on CelebA and UTKFace than on FMNIST. In the future, we plan to extend the model inversion attack by using more advanced GANs, such as StyleGAN2 [24].

On the other hand, there does not seem to be any clear relation between attribute inference and model inversion, as well as between membership inference and attribute inference.

7 Defenses

We now evaluate two representative defense mechanisms, namely Differential Privacy (DP) and Knowledge Distillation (KD), and investigate whether or not, and how effectively, they can be used to mitigate these attacks. To the best of our knowledge, there is no one general defense against all the inference attacks. The reason we choose these two mechanisms is that they have been proposed to defend more diverse types of attacks compared to others. DP is used to defend several attacks, e.g., adversarial examples [28], membership inference [56], model stealing [26], and model inversion [65]. Shejwalkar and Houmansadr [52] use KD to defend some inference attacks like membership inference. Papernot et al. [47] also introduce KD to reduce the effectiveness of adversarial examples on ML models. Other common techniques cannot defend against all attacks simultaneously. For instance, regularization can reduce the performance of membership inference, but regularization also leads to better model stealing and model inversion as shown in Section 6.4.

7.1 Techniques

7.1.1 Differential Privacy (DP)

Differential Privacy (DP) [11, 30] guarantees that any single data sample in a dataset has a limited impact on the output.

Definition 1 ((ε,δ)-DP). *A randomization algorithm $\mathcal{A}$ satisfies (ε,δ)-differential privacy, with $\varepsilon > 0$ and $0 \leq \delta < 1$, if and only if for any two neighboring datasets D and D' that differ in one record, we have:*

$$\forall T \subseteq Range(\mathcal{A}) : \Pr[\mathcal{A}(D) \in T] \leq e^{\varepsilon} \Pr\left[\mathcal{A}(D') \in T\right] + \delta,$$

where $Range(\mathcal{A})$ denotes the set of all possible outputs of the algorithm $\mathcal{A}$, δ can be interpreted as the probability that the mechanism fails to satisfy ε-DP.

Gaussian Mechanism. There are several approaches to design mechanisms satisfying (ε,δ)-differential privacy. The Gaussian mechanism is arguably the most widely used one in the ML context. Essentially, it computes a function f on dataset D by adding random (Gaussian) noise to $f(D)$. The magnitude of the noise depends on Δ_f, i.e., the *global sensitivity* of f (also referred to as the ℓ_2 sensitivity). More formally, we define the $\mathcal{A}$ mechanism as

$$\mathcal{A}(D) = f(D) + \mathcal{N}\left(0, \Delta_f^2 \sigma^2 \mathbf{I}\right)$$

where $\Delta_f = \max_{(D,D'):D \simeq D'} ||f(D) - f(D')||_2$.

Here, $\mathcal{N}(0, \Delta_f^2\sigma^2\mathbf{I})$ denotes a multi-dimensional random variable sampled from the normal distribution with mean 0 and standard deviation $\Delta_f\sigma$, and $\sigma = \sqrt{2\ln\frac{1.25}{\delta}}/\varepsilon$.

DP-SGD. We experiment with Differentially-Private Stochastic Gradient Descent (DP-SGD) [1], the most representative DP mechanism for protecting machine learning models. In general, DP-SGD adds Gaussian noise to gradient g during the target ML model's training process, i.e., $\tilde{g} = g + \mathcal{N}\left(0, \Delta_g^2\sigma^2\mathbf{I}\right)$. Note that there is no prior knowledge to determine the influence of a single training sample on the gradient g; thus, the sensitivity of g cannot be directly computed. To address this problem, DP-SGD proposes to bound the ℓ_2 norm of the gradient to C by clipping g to $g/\max\{1, ||g||_2/C\}$. This clipping ensures that if $||g||_2 \leq C$, g is preserved; otherwise, it gets scaled down to be norm of C. As such, the sensitivity of g is bounded by C.

	CelebA			FMNIST			STL10			UTKFace		
	Original	ε_1=5.139	ε_2=6.574	Original	ε_1=8.408	ε_2=9.355	Original	ε_1=8.834	ε_2=9.604	Original	ε_1=9.578	ε_2=7.762
$\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$	0.645±0.005	0.500±0.000	0.500±0.000	0.559±0.004	0.500±0.000	0.500±0.000	0.774±0.010	0.501±0.005	0.500±0.000	0.626±0.008	0.500±0.001	0.499±0.001
$\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$	0.649±0.009	0.500±0.000	0.500±0.000	0.560±0.003	0.498±0.003	0.500±0.000	0.744±0.028	0.502±0.010	0.498±0.006	0.621±0.004	0.499±0.007	0.500±0.000
$\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$	0.717±0.004	0.500±0.000	0.501±0.001	0.580±0.002	0.505±0.001	0.500±0.000	0.826±0.008	0.511±0.005	0.541±0.002	0.650±0.009	0.504±0.000	0.505±0.006
$\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$	0.721±0.002	0.500±0.000	0.500±0.000	0.580±0.002	0.500±0.000	0.500±0.000	0.823±0.005	0.538±0.002	0.543±0.006	0.660±0.003	0.504±0.000	0.501±0.001
$\langle \mathsf{ModInv}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$	0.693±0.024	0.640±0.053	0.686±0.053	0.586±0.022	0.520±0.034	0.570±0.032	0.353±0.008	0.209±0.032	0.227±0.038	0.912±0.011	0.814±0.020	0.744±0.041
$\langle \mathsf{ModInv}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{N}}_{\mathsf{aux}} \rangle$	0.058±0.000	0.059±0.000	0.059±0.000	0.991±0.000	0.993±0.000	0.993±0.000	0.201±0.000	0.201±0.000	0.201±0.000	0.070±0.000	0.071±0.000	0.070±0.000
$\langle \mathsf{AttrInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$	0.764±0.000	0.732±0.001	0.701±0.002	-	-	-	-	-	-	0.792±0.002	0.782±0.006	0.724±0.022
$\langle \mathsf{AttrInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$	0.773±0.000	0.732±0.004	0.707±0.002	-	-	-	-	-	-	0.809±0.001	0.768±0.000	0.740±0.001
$\langle \mathsf{ModSteal}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$	0.803±0.001	0.903±0.001	0.896±0.001	0.932±0.001	0.928±0.001	0.925±0.001	0.663±0.005	0.501±0.010	0.483±0.007	0.907±0.005	0.845±0.005	0.795±0.006
$\langle \mathsf{ModSteal}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$	0.754±0.002	0.903±0.001	0.895±0.001	0.906±0.004	0.927±0.001	0.924±0.001	0.525±0.006	0.477±0.006	0.466±0.006	0.851±0.006	0.838±0.007	0.785±0.004

Table 3: Attack performance under different threat models and datasets, on SimpleCNN, using DP-SGD. For MemInf, ModInv ($\langle \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$), and AttrInf, we use accuracy, for ModInv ($\langle \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{N}}_{\mathsf{aux}} \rangle$), MSE, and for ModSteal, agreement.

Composition. Note that we need to calculate the gradient multiple times when training an ML model. Each calculation requires access to the training data and thus consumes a portion of the privacy budget. We use the notion of zCDP [3] to calculate the total privacy budget consumption. The general idea of zCDP is to connect (ε, δ)-DP to Rényi divergence and use the properties of Rényi divergence to achieve tighter composition property. That is, for a given privacy budget (ε, δ) and the number of gradient calculation T, zCDP adds less Gaussian noise to the gradient than the naïve composition. For instance, when ε=1, δ=1e-5, T=1,000, and C=1, the standard deviation of Gaussian noise calculated by zCDP is 155, while that of naïve composition is 1,414.

7.1.2 Knowledge Distillation (KD)

Another defense mechanism we consider is Knowledge Distillation (KD) [21, 52]. Generally, KD is proposed to transfer the generalization ability (knowledge) from a larger model (original model) to a smaller model (distilled model) without utility degradation. Once the distilled model is trained, it can replace the original model in many scenarios as it is more computationally efficient and less dependent on resources.

A simple way to transfer the knowledge from the original model to the distilled model is to use the posteriors generated by the original model as a "soft label" to guide the training of the distilled model. Compared to the original labels (one-hot), the posteriors have higher entropy. It contains more information for each training sample and has less variance for the gradient among different training samples, which can speed up the training process of the distilled model [21]. To train the distilled model, we combine two loss terms, i.e., the soft target loss and the hard target loss. The first one is the Kullback-Leibler divergence loss between the output of the original model and the distilled model. The second one is the cross-entropy loss between the original label and the output of the distilled model. As suggested by Hinton et al. [21], we use a higher temperature value in the softmax function of the first loss for better performance.

KD transfers knowledge from the original model to a distilled model. Compared to the original model, the distilled model has a lower capacity. Intuitively, it should remember less information of the original model with respect to both its training dataset and parameters. Thus, we believe KD can serve as a general defense for inference attacks. Papernot et al. [47] show that KD can reduce the risks of adversarial examples against machine learning models. Shejwalkar and Houmansadr [52] also show that KD can mitigate membership inference attacks. Here, we take a broader view investigating whether or not KD is effective to defend against other inference attacks.

7.2 Experimental Setup

Both DP-SGD and KD are applied in the training process of target models. Due to space limitations, we only apply DP-SGD to SimpleCNN and KD to VGG19.

DP-SGD Target Model. We use the Opacus library[5] to implement DP-SGD. This library allows a user to configure the clip bound C, the standard deviation of the Gaussian noise σ, and the failure probability δ, then the library can automatically calculate the total privacy budget ε using zCDP. A larger number of epochs implies higher ε. Our target model is trained for 300 epochs; thus, we fix δ=1e-5 and choose two sets of C and σ such that ε is smaller than 10. We list these settings in the second row of Table 3. All the other hyperparameters are the same as presented in Section 5.3.

Distillation Target Model. We distill the model knowledge of VGG19 (16 convolution layers and 3 fully connected layers) to a smaller model, i.e., VGG11 [54] (8 convolution layers and 3 fully connected layers). We use Kullback-Leibler divergence as the soft target loss with the temperature setting to 20. For the hard target loss, we use cross-entropy. We set α to 0.7 for the ratio of the soft target loss. Other settings are the same as the target model's training phase in Section 5.3.

7.3 Results

DP-SGD. Table 3 reports the performance of inference attacks against target models protected by DP-SGD. For MemInf, DP-SGD is effective in almost all cases. For instance, for $\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$ on the CelebA dataset, the accuracy drops from 0.721 to 0.500, which is a random guess. This is expected as DP, by definition, can mitigate membership inference. For ModInv and AttrInf, DP-SGD can only

[5] https://github.com/pytorch/opacus

	CelebA		FMNIST		STL10		UTKFace	
	Original	Distilled	Original	Distilled	Original	Distilled	Original	Distilled
$\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$	0.595 ± 0.038	0.500 ± 0.000	0.520 ± 0.004	0.515 ± 0.005	0.682 ± 0.053	0.616 ± 0.069	0.642 ± 0.008	0.581 ± 0.024
$\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$	0.637 ± 0.012	0.572 ± 0.042	0.530 ± 0.009	0.538 ± 0.008	0.786 ± 0.008	0.703 ± 0.005	0.657 ± 0.009	0.596 ± 0.001
$\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$	0.698 ± 0.005	0.691 ± 0.002	0.555 ± 0.013	0.568 ± 0.001	0.806 ± 0.015	0.773 ± 0.006	0.677 ± 0.000	0.611 ± 0.002
$\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$	0.638 ± 0.038	0.559 ± 0.059	0.539 ± 0.014	0.530 ± 0.004	0.799 ± 0.050	0.677 ± 0.073	0.642 ± 0.017	0.620 ± 0.024
$\langle \mathsf{ModInv}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$	0.648 ± 0.038	0.650 ± 0.030	0.573 ± 0.021	0.447 ± 0.037	0.203 ± 0.020	0.244 ± 0.031	0.824 ± 0.021	0.815 ± 0.036
$\langle \mathsf{ModInv}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{N}}_{\mathsf{aux}} \rangle$	0.058 ± 0.000	0.058 ± 0.000	0.993 ± 0.000	0.992 ± 0.000	0.201 ± 0.000	0.201 ± 0.000	0.070 ± 0.000	0.070 ± 0.000
$\langle \mathsf{AttrInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$	0.665 ± 0.005	0.669 ± 0.019	-	-	-	-	0.540 ± 0.006	0.554 ± 0.030
$\langle \mathsf{AttrInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$	0.610 ± 0.018	0.660 ± 0.004	-	-	-	-	0.549 ± 0.001	0.584 ± 0.004
$\langle \mathsf{ModSteal}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$	0.820 ± 0.001	0.788 ± 0.003	0.932 ± 0.000	0.940 ± 0.001	0.589 ± 0.006	0.618 ± 0.003	0.927 ± 0.007	0.918 ± 0.013
$\langle \mathsf{ModSteal}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{P}}_{\mathsf{aux}} \rangle$	0.756 ± 0.005	0.741 ± 0.003	0.902 ± 0.001	0.914 ± 0.001	0.478 ± 0.006	0.510 ± 0.003	0.826 ± 0.010	0.836 ± 0.018

Table 4: Attack performance under different threat models and datasets, on VGG19, using Knowledge Distillation (KD). For MemInf, ModInv ($\langle \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$), and AttrInf, we use accuracy, for ModInv ($\langle \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{N}}_{\mathsf{aux}} \rangle$), MSE, and for ModSteal, agreement.

Experiment	Model	CelebA	FMNIST	STL10	UTKFace
DP-SGD (ε_1)	SimpleCNN	0.654	0.830	0.347	0.698
DP-SGD (ε_2)	SimpleCNN	0.675	0.836	0.313	0.680
KD	VGG19	0.713	0.919	0.588	0.823
No Defense	SimpleCNN	0.706	0.903	0.516	0.818
No Defense	VGG19	0.733	0.905	0.587	0.834

Table 5: Accuracy of target models protected by DP-SGD and KD.

reduce the attack accuracy to a small extent. However, the MSE loss for $\langle \mathsf{ModInv}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{N}}_{\mathsf{aux}} \rangle$ remains stable.

DP-SGD indeed reduces the risks of model stealing on STL10 and UTKFace under different threat models. For instance, for $\langle \mathsf{ModSteal}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$ on STL10, the agreement for the two different εs are 0.501 and 0.483, while the agreement for the original model is 0.663. Meanwhile, DP-SGD does not influence model stealing on FMNIST. Interestingly, it actually enhances the performance of model stealing on CelebA. Overall, DP-SGD can effectively defend against membership inference attacks, but not the others.

KD. In Table 4, we report the effectiveness of KD as a general defense mechanism. We do not observe any significant decrease in attack performance for model inversion, attribute inference, and model stealing on original vs. the distilled models. Specifically, the attack performance difference, in most cases, is less than 5%. In certain cases, KD is effective against membership inference, but to a lesser extent compared to DP-SGD. For instance, the accuracy of $\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{B}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$ on the STL10 dataset drops from 0.682 to 0.616.

Utility and Defense Effectiveness Trade-off. We observe that both DP-SGD and KD can defend some of the inference attacks. However, it comes at the cost of utility dropping (see Table 5). Compared to DP-SGD, KD preserves the target model's utility better. For instance, on the STL10 dataset, the target testing accuracy drops from 0.818 to 0.698 (ε_1) and 0.680 (ε_2) for DP-SGD, while the corresponding performance only drops from 0.834 to 0.823 for KD. Meanwhile, DP-SGD has better defense performance than KD. In particular, for $\langle \mathsf{MemInf}, \mathcal{M}^{\mathsf{W}}, \mathcal{D}^{\mathsf{S}}_{\mathsf{aux}} \rangle$ on the UTKFace dataset, SimpleCNN defended by DP-SGD reduces the attack accuracy significantly from 0.660 to 0.504 (ε_1) and 0.501 (ε_2), respectively. The VGG19 model defended by KD only reduces the attack accuracy to a smaller extends from 0.642 to 0.621. Also, as discussed above, both DP-SGD and KD are not general defenses against all the inference attacks, which inspires future research to better defend different inference attacks while maintaining the target model's utility.

8 Related Work

We now review relevant related work on inference attacks and defenses, as well as software dedicated to evaluating them.

Membership Inference Attacks. Shokri et al. [53] propose the first membership inference attack against black-box ML models: they train multiple shadow models to simulate the target model and use multiple attack models to conduct the inference. Salem et al. [51] later relax several key assumptions from [53]; namely, using multiple shadow models, the knowledge of the target model structure, and having a dataset from the same distribution as the target model's. Yeom et al. [63] assume that the adversary knows the target model's training dataset's distribution and size, and they collude with the training algorithm. Both [51] and [63] are close in performance to Shokri et al.'s attacks [53]. In this paper, we implement the attack proposed by Salem et al. [51], i.e., one shadow model, one attack model, and a shadow dataset. More recently, researchers have studied membership inference in other settings, including natural language processing [5, 56], generative models [6, 16, 20], recommender systems [64], and federated learning [35, 38]. Also, Song and Mittal have performed a systematic evaluation on membership inference [58]. Previous work [51, 53] also shows that overfitting is the major factor causing membership inference. To the best of our knowledge, however, no one has investigated other factors studied in our paper, such as the influence of dataset complexity or the relationship among different inference attacks.

Attribute Inference. Prior research [2, 14] has studied macro-level attribute inference attacks against ML models, whereby the adversary aims to infer some general properties of the training dataset. Melis et al. [35] propose the first sample-level attribute inference attack against federated machine learning systems. Song and Shmatikov [57] reveal that the risks of

attribute inference are caused by the intrinsic overlearning characteristics of machine learning models.

Model Inversion. Model inversion is first proposed by Fredrikson et al. [13] in the setting of drug dose classification. Later, they extend model inversion to general ML settings relying on back-propagation over a target ML model's parameters [12]. More recently, Zhang et al. [65] develop a more advanced attack aiming to synthesize the training dataset relying on GANs. Finally, Carlini et al. [4] show that model inversion can be effectively performed against natural language processing models as well.

Model Stealing. Tramèr et al. [59] propose the first model stealing attack against black-box machine learning API. Orekondy et al. [41] develop a reinforcement learning-based framework to optimize both query time and effectiveness. Also, Wang and Gong [60] and Oh et al. [40] show that hyperparameters of a target model can be inferred as well.

Defense Mechanisms. A few defense mechanisms have been proposed to mitigate membership inference attacks [23, 37, 51]. However, these defenses are specifically designed for membership inference and cannot mitigate other inference attacks. For instance, Salem et al. [51] propose to reduce overfitting of the target model as a defense; however, as we show in our analysis (see Section 6.4), reducing overfitting will improve the performance of model stealing.

Differential Privacy (DP) [11, 30] guarantees that any single data sample in a dataset has a limited impact on the output of an algorithm. As such, it is an effective defense mechanism against inference attacks. Abadi et al. [1] introduce DP-SGD, which adds Gaussian noise to the gradients of the target model during the training process. Another DP method for protecting the privacy of ML models is PATE [43]: a set of teacher models is trained on a private dataset, which is used to label a public dataset in a differentially private manner. The final public dataset is then used to train a student model. Recently, Nasr et al. [39] instantiate a number of attacks against ML to evaluate the effectiveness of DP defenses and, in particular, how tight are theoretical DP bounds.

Another defense mechanism, as mentioned, is Knowledge Distillation (KD) [21]. Papernot et al. [47] propose a defensive distillation mechanism to effectively reduce the risks for target models with respect to adversarial examples. Shejwalkar and Houmansadr [52] reveal that distillation can reduce the gap between the posteriors of members and non-members, thus protecting membership privacy. In our experiments, we show that distillation is indeed effective against certain target models supported by ML-DOCTOR; however, it cannot defend against other types of inference attacks.

Risk Assessment Tools. Finally, researchers have recently developed a number of software tools to measure the potential security/privacy risks of ML models. Ling et al. [32] propose DEEPSEC, a security analysis system to evaluate different adversarial example attacks and defenses. Another system for adversarial examples is CleverHans [44]. Pang et al. [42] introduce TROJANZOO, which focuses on backdoor attacks.

Closer to our work is ML Privacy Meter [36], which jointly considers membership inference attacks in both black-box and white-box settings. Unlike ML Privacy Meter, which focuses on membership inference only, ML-DOCTOR considers four types of inference attacks simultaneously. In addition, we rely on ML-DOCTOR to perform a comprehensive analysis for all these inference attacks.

9 Conclusion

In this paper, we performed the first holistic analysis of privacy risks caused by inference attacks against machine learning models. We established a taxonomy of threat models for four types of inference attacks, including membership inference, model inversion, attribute inference, and model stealing. We conducted an extensive measurement study, over five model architectures, and four datasets, of both attacks and defenses. Among other things, we found that the complexity of the training dataset plays an important role in the attack's performance, while the effectiveness of model stealing and membership inference attacks are negatively correlated. We also showed that defenses such as DP-SGD and KD could only hope to mitigate *some* of the inference attacks.

We integrated all the attacks and defenses into a re-usable, modular software called ML-DOCTOR, which can be used in various scenarios. For instance, an ML model owner can use ML-DOCTOR to evaluate the model's inference risks before deploying it in the real world. We are also confident that ML-DOCTOR will serve as a benchmark tool to facilitate future research on inference attacks and defenses.

Currently, ML-DOCTOR concentrates on image classification models, as image classification is the most popular ML application. Researchers have demonstrated that inference attacks can be successfully launched against other types of ML models, such as language models [55, 56], generative models [6, 16], and graph-based models [18], as well as other training paradigms, such as federated learning [35]. We plan to extend ML-DOCTOR to support a broader range of ML application scenarios. In addition, we will explore other general defense mechanisms, such as training target models with noisy data or GAN-generated data.

Finally, while ML-DOCTOR is designed for inference attacks, we plan to integrate tools [32, 42, 44] geared to evaluate risks aimed to jeopardize models' functionality, e.g., adversarial examples, data poisoning, etc., thus providing a one-stop-shop toward enabling secure and trustworthy AI.

Acknowledgments. The authors wish to thank Luca Melis and Jamie Hayes for valuable discussions and feedback. This work is partially funded by the Helmholtz Association within the project "Trustworthy Federated Data Analytics" (TFDA) (funding number ZT-I-OO1 4).

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. In *CCS*, pages 308–318, 2016.

[2] Giuseppe Ateniese, Luigi V. Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *Int. J. Secur. Networks*, 2015.

[3] Mark Bun and Thomas Steinke. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In *TCC*, pages 635–658, 2016.

[4] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *USENIX Security*, pages 267–284, 2019.

[5] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting Training Data from Large Language Models. *CoRR abs/2012.07805*, 2020.

[6] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models. In *CCS*, pages 343–362, 2020.

[7] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. When Machine Unlearning Jeopardizes Privacy. In *CCS*, 2021.

[8] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *CVPR*, pages 1800–1807, 2017.

[9] Adam Coates, Andrew Y. Ng, and Honglak Lee. An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In *AISTATS*, pages 215–223, 2011.

[10] Emiliano De Cristofaro. An Overview of Privacy in Machine Learning. *CoRR abs/2005.08679*, 2020.

[11] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*. Now Publishers Inc., 2014.

[12] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *CCS*, pages 1322–1333, 2015.

[13] Matt Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. In *USENIX Security*, pages 17–32, 2014.

[14] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations. In *CCS*, pages 619–633, 2018.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *NIPS*, pages 2672–2680, 2014.

[16] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. LOGAN: Evaluating Privacy Leakage of Generative Models Using Generative Adversarial Networks. *Proceedings of Privacy Enhancing Technologies Symposium*, 2019.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pages 770–778, 2016.

[18] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. Stealing Links from Graph Neural Networks. In *USENIX Security*, pages 2669–2686, 2021.

[19] Xinlei He and Yang Zhang. Quantifying and Mitigating Privacy Risks of Contrastive Learning. In *CCS*, 2021.

[20] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte Carlo and Reconstruction Membership Inference Attacks against Generative Models. *Symposium on Privacy Enhancing Technologies Symposium*, 2019.

[21] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the Knowledge in a Neural Network. *CoRR abs/1503.02531*, 2015.

[22] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning. In *S&P*, 2022.

[23] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples. In *CCS*, pages 259–274, 2019.

[24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. In *CVPR*, 2020.

[25] Yigitcan Kaya and Tudor Dumitras. When Does Data Augmentation Help With Membership Inference Attacks? In *ICML*, pages 5345–5355, 2021.

[26] Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer. Thieves on Sesame Street! Model Extraction of BERT-based APIs. In *ICLR*, 2020.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, pages 1106–1114, 2012.

[28] Mathias Lécuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified Robustness to Adversarial Examples with Differential Privacy. In *S&P*, pages 656–672, 2019.

[29] Klas Leino and Matt Fredrikson. Stolen Memories: Leveraging Model Memorization for Calibrated White-Box Membership Inference. In *USENIX Security*, pages 1605–1622, 2020.

[30] Ninghui Li, Min Lyu, Dong Su, and Weining Yang. *Differential Privacy: From Theory to Practice*. Morgan & Claypool Publishers, 2016.

[31] Zheng Li and Yang Zhang. Membership Leakage in Label-Only Exposures. In *CCS*, 2021.

[32] Xiang Ling, Shouling Ji, Jiaxu Zou, Jiannan Wang, Chunming Wu, Bo Li, and Ting Wang. DEEPSEC: A Uniform Platform for Security Analysis of Deep Learning Model. In *S&P*, pages 673–690, 2019.

[33] Hongbin Liu, Jinyuan Jia, Wenjie Qu, and Neil Zhenqiang Gong. EncoderMI: Membership Inference against Pre-trained Encoders in Contrastive Learning. In *CCS*, 2021.

[34] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *ICCV*, 2015.

[35] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting Unintended Feature Leakage in Collaborative Learning. In *S&P*, pages 497–512, 2019.

[36] Sasi Kumar Murakonda and Reza Shokri. ML Privacy Meter: Aiding Regulatory Compliance by Quantifying the Privacy Risks of Machine Learning. *CoRR abs/2007.09339*, 2020.

[37] Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine Learning with Membership Privacy using Adversarial Regularization. In *CCS*, pages 634–646, 2018.

[38] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *S&P*, pages 1021–1035, 2019.

[39] Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlini. Adversary Instantiation: Lower Bounds for Differentially Private Machine Learning. In *S&P*, 2021.

[40] Seong Joon Oh, Max Augustin, Bernt Schiele, and Mario Fritz. Towards Reverse-Engineering Black-Box Neural Networks. In *ICLR*, 2018.

[41] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff Nets: Stealing Functionality of Black-Box Models. In *CVPR*, pages 4954–4963, 2019.

[42] Ren Pang, Zheng Zhang, Xiangshan Gao, Zhaohan Xi, Shouling Ji, Peng Cheng, and Ting Wang. TROJANZOO: Everything You Ever Wanted to Know about Neural Backdoors (But Were Afraid to Ask). *CoRR abs/2012.09302*, 2020.

[43] Nicolas Papernot, Martin Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *ICLR*, 2017.

[44] Nicolas Papernot et al. Technical Report on the CleverHans v2.1.0 Adversarial Examples Library. *CoRR abs/1610.00768*, 2018.

[45] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. SoK: Towards the Science of Security and Privacy in Machine Learning. In *Euro S&P*, pages 399–414, 2018.

[46] Nicolas Papernot, Patrick D. McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical Black-Box Attacks Against Machine Learning. In *ASIACCS*, pages 506–519, 2017.

[47] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In *S&P*, 2016.

[48] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *CoRR abs/1511.06434*, 2015.

[49] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs Black-box: Bayes Optimal Strategies for Membership Inference. In *ICML*, pages 5558–5567, 2019.

[50] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. Updates-Leak: Data Set Inference and Reconstruction Attacks in Online Learning. In *USENIX Security*, pages 1291–1308, 2020.

[51] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *NDSS*, 2019.

[52] Virat Shejwalkar and Amir Houmansadr. Membership Privacy for Machine Learning Models Through Knowledge Transfer. In *AAAI*, 2021.

[53] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *S&P*, pages 3–18, 2017.

[54] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

[55] Congzheng Song and Ananth Raghunathan. Information Leakage in Embedding Models. In *CCS*, pages 377–390, 2020.

[56] Congzheng Song and Vitaly Shmatikov. Auditing Data Provenance in Text-Generation Models. In *KDD*, 2019.

[57] Congzheng Song and Vitaly Shmatikov. Overlearning Reveals Sensitive Attributes. In *ICLR*, 2020.

[58] Liwei Song and Prateek Mittal. Systematic Evaluation of Privacy Risks of Machine Learning Models. In *USENIX Security*, 2021.

[59] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing Machine Learning Models via Prediction APIs. In *USENIX Security*, pages 601–618, 2016.

[60] Binghui Wang and Neil Zhenqiang Gong. Stealing Hyperparameters in Machine Learning. In *S&P*, pages 36–52, 2018.

[61] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *CoRR abs/1708.07747*, 2017.

[62] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A. Gunter, and Bo Li. Detecting AI Trojans Using Meta Neural Analysis. In *S&P*, 2021.

[63] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In *CSF*, pages 268–282, 2018.

[64] Minxing Zhang, Zhaochun Ren, Zihan Wang, Pengjie Ren, Zhumin Chen, Pengfei Hu, and Yang Zhang. Membership Inference Attacks Against Recommender Systems. In *CCS*, 2021.

[65] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks. In *CVPR*, pages 250–258, 2020.

[66] Zhifei Zhang, Yang Song, and Hairong Qi. Age Progression/Regression by Conditional Adversarial Autoencoder. In *CVPR*, pages 4352–4360, 2017.

A Additional Experimental Results

In this appendix, we report plots for additional experiments as mentioned throughout the paper.

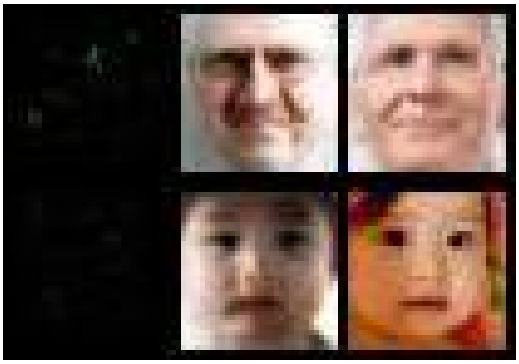

Figure 10: Visualization of model inversion (AlexNet trained on UTKFace). The left column depicts two samples reconstructed using [12], the middle one using [65], while the right column reports two samples from the target model's training dataset. The left column images are normalized, black indicating pixels' value in reconstructed images are close to 0. Note that similar results are shown by Zhang et al. [65].

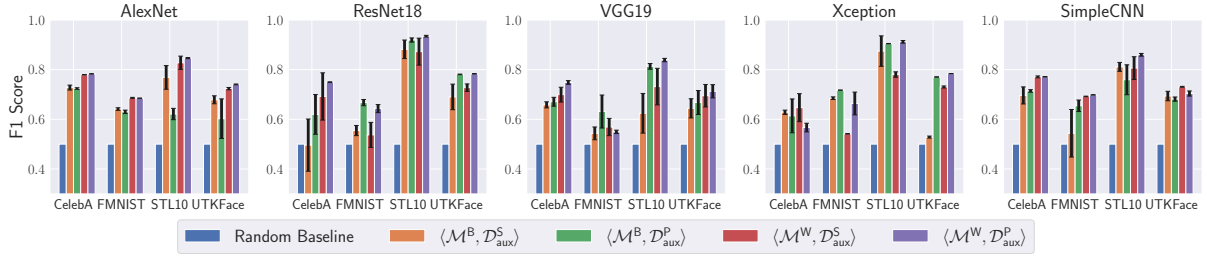

Figure 11: F1 score of membership inference attacks (MemInf) under different threat models, datasets, and target model architectures.

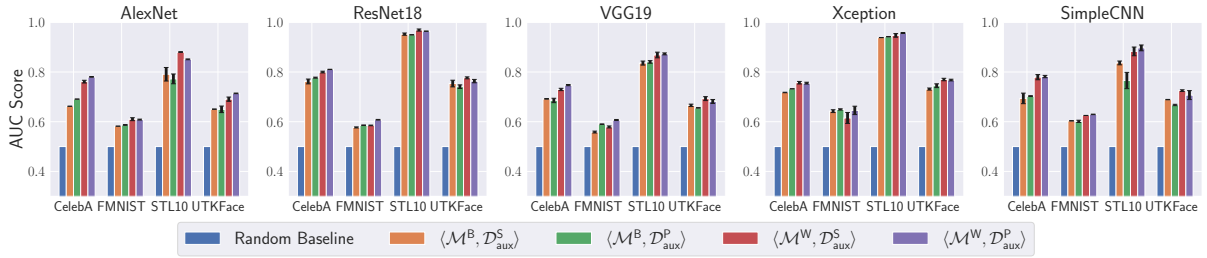

Figure 12: AUC score of membership inference attacks (MemInf) under different threat models, datasets, and target model architectures.

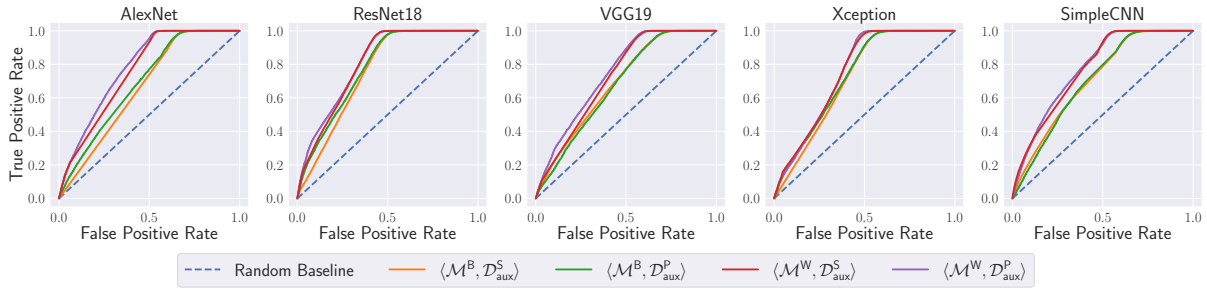

Figure 13: ROC curve of membership inference attacks (MemInf) under different threat models on CelebA.

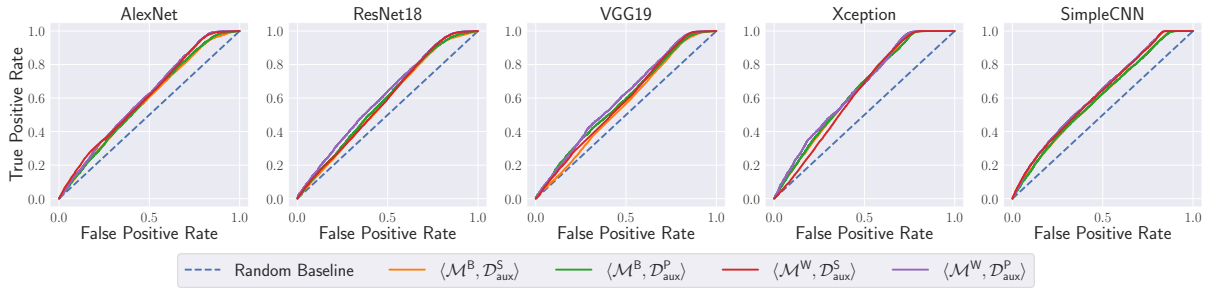

Figure 14: ROC curve of membership inference attacks (MemInf) under different threat models on FMNIST.

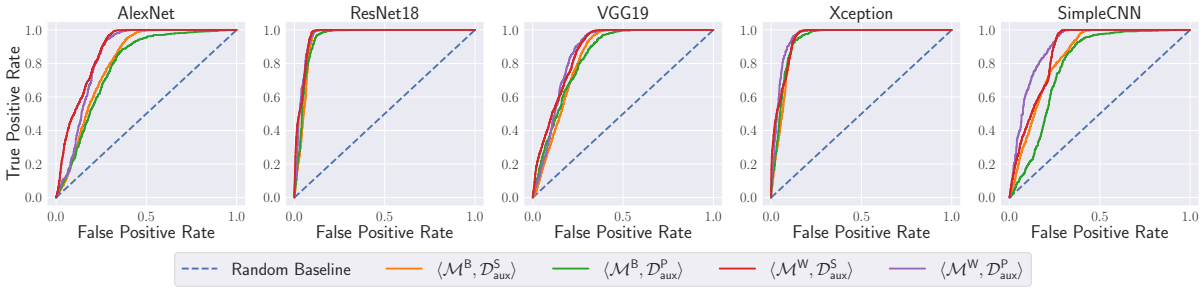

Figure 15: ROC curve of membership inference attacks (MemInf) under different threat models on STL10.

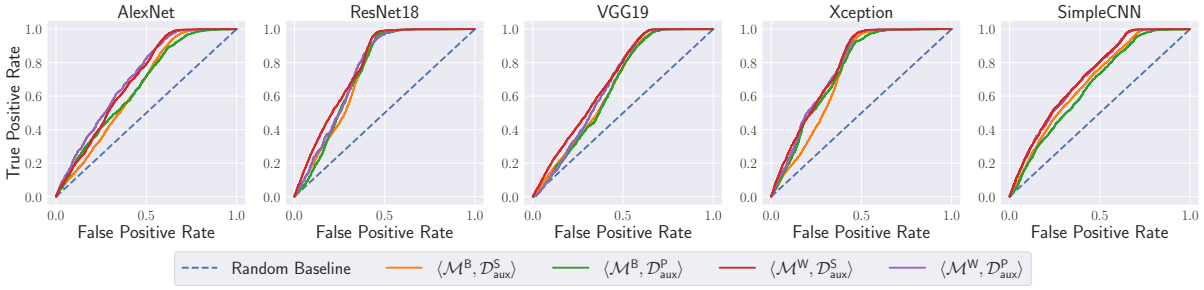

Figure 16: ROC curve of membership inference attacks (MemInf) under different threat models on UTKFace.

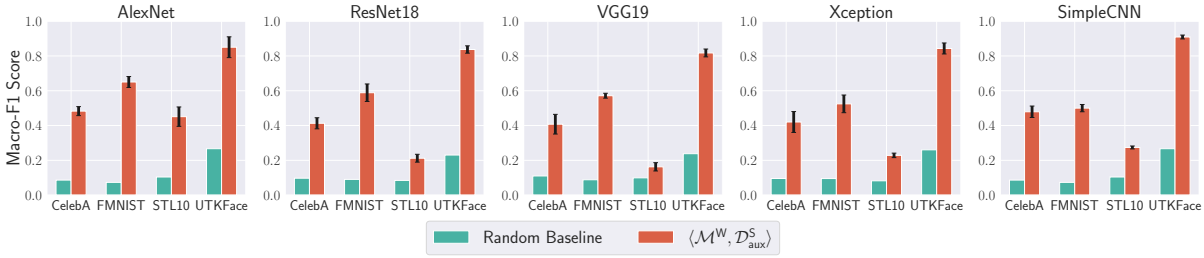

Figure 17: Macro-F1 score of model inversion attacks (ModInv) under different datasets and target model architectures.

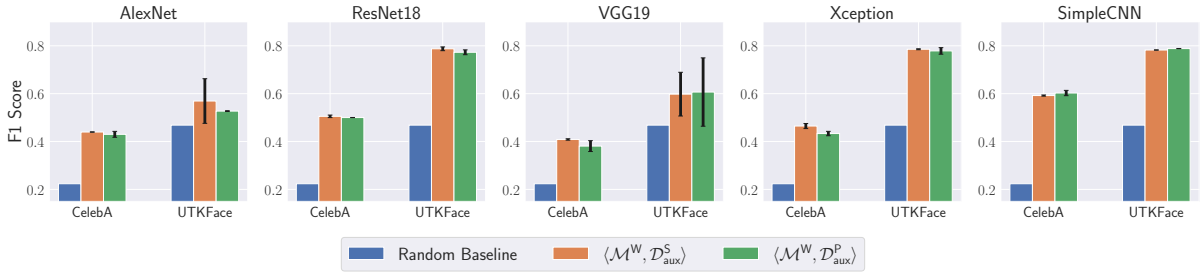

Figure 18: F1 score of attribute inference attacks (AttrInf) under different threat models, datasets, and target model architectures. Note that we report F1 score for UTKFace and macro-F1 score for CelebA.

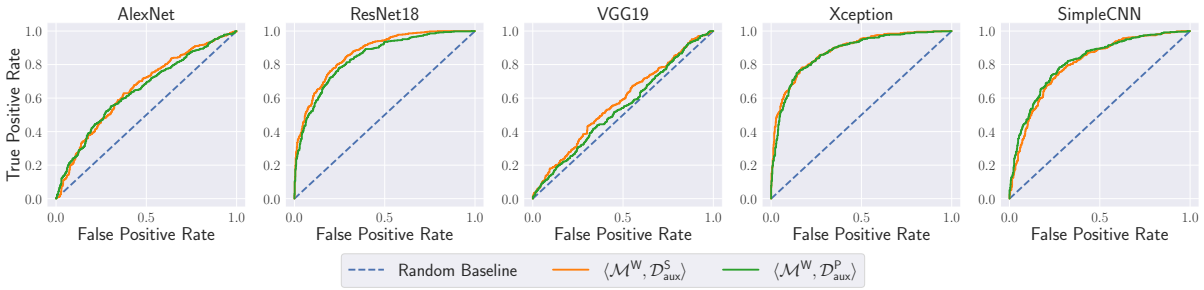

Figure 19: ROC curve of attribute inference attacks (AttrInf) on UTKFace under different threat models and target model architectures.

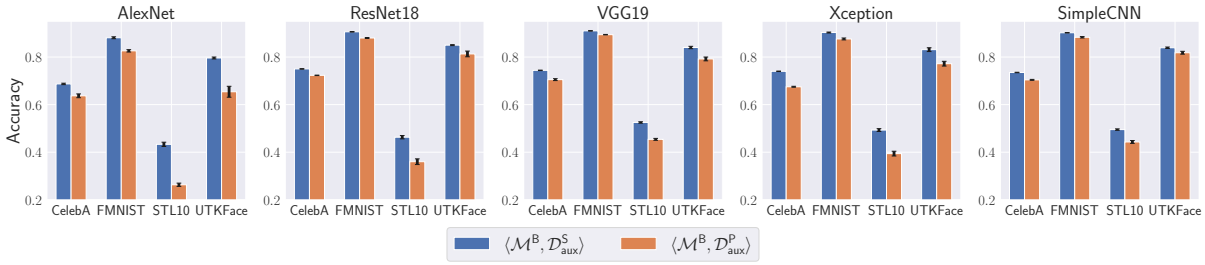

Figure 20: Accuracy of model stealing attacks (ModSteal) under different threat models, datasets, and target model architectures.

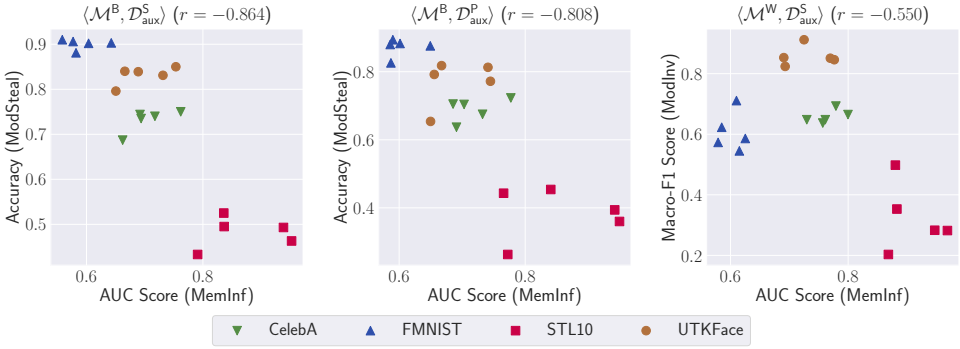

Figure 21: The relation between different attacks under the same threat model. For MemInf, we use AUC score, for ModInv, macro-F1, and for ModSteal, accuracy.