# Quantifying Privacy Risks of Prompts in Visual Prompt Learning

Yixin Wu[1] Rui Wen[1] Michael Backes[1] Pascal Berrang[2] Mathias Humbert[3] Yun Shen[4] Yang Zhang[1]

[1]*CISPA Helmholtz Center for Information Security*
[2]*University of Birmingham* [3]*University of Lausanne* [4]*Netapp*

## Abstract

Large-scale pre-trained models are increasingly adapted to downstream tasks through a new paradigm called prompt learning. In contrast to fine-tuning, prompt learning does not update the pre-trained model's parameters. Instead, it only learns an input perturbation, namely prompt, to be added to the downstream task data for predictions. Given the fast development of prompt learning, a well-generalized prompt inevitably becomes a valuable asset as significant effort and proprietary data are used to create it. This naturally raises the question of whether a prompt may leak the proprietary information of its training data. In this paper, we perform the first comprehensive privacy assessment of prompts learned by visual prompt learning through the lens of property inference and membership inference attacks. Our empirical evaluation shows that the prompts are vulnerable to both attacks. We also demonstrate that the adversary can mount a successful property inference attack with limited cost. Moreover, we show that membership inference attacks against prompts can be successful with relaxed adversarial assumptions. We further make some initial investigations on the defenses and observe that our method can mitigate the membership inference attacks with a decent utility-defense trade-off but fails to defend against property inference attacks. We hope our results can shed light on the privacy risks of the popular prompt learning paradigm. To facilitate the research in this direction, we will share our code and models with the community.[1]

## 1 Introduction

Recent research has provided ample evidence that increasing the size of machine learning (ML) models, i.e., the number of parameters, is a pivotal factor in enhancing their overall performance [6, 11, 40]. One of the commonly employed strategies for adapting such large-scale pre-trained ML models to downstream tasks is fine-tuning [59], which updates model parameters for specific downstream tasks via back-propagation. Fine-tuning, however, suffers from two main drawbacks. First, it leads to high computational costs because all model parameters need to be updated. In addition, it is storage inefficient since a separate copy of the fine-tuned model needs to be stored for each downstream task.

In order to address these limitations, researchers have proposed prompt learning as an alternative to fine-tuning [4, 5, 20, 25, 26, 30, 31, 41, 56]. Prompt learning involves learning an input perturbation, referred to as a *prompt*, that enables shifting downstream task data to the original data distribution. The pre-trained model generates a task-specific output based on this prompt. It is important to note that, during prompt learning, the pre-trained model remains frozen, leading to a significant decrease in the number of learned parameters compared to fine-tuning (see Section 2). In recent years, prompt learning has been extensively validated and shown to be effective in the domains of computer vision (CV) [4, 5, 20, 31, 56] and natural language processing (NLP) [25, 26, 30, 41]. It is expected that prompt as a service (PaaS) will gain popularity.[2] In this scenario, a user can request a prompt for a downstream task from the PaaS provider without the need for arduous fine-tuning. The user then combines their data with the prompt and inputs them into the pre-trained model to obtain the predictions, as depicted in Figure 1. In this way, the user can run the pre-trained model and keep their data on-premise, while the PaaS provider can reuse a single pre-trained model to support multiple downstream tasks. These benefits differentiate PaaS from machine learning as a service (MLaaS) [42]. As a result, a well-generalized prompt becomes a valuable asset for PaaS providers, as they invest significant efforts and use proprietary data to develop it.

Previous research has demonstrated that ML models are vulnerable to various privacy attacks, such as property inference attacks [14, 55] and membership inference attacks [28, 44, 46], which can disclose sensitive information about the training data used to create the models. Such data leakage can

[1] https://github.com/yxoh/prompt_leak_usenix2024/.

[2] https://twitter.com/AndrewYNg/status/1650938079027548160.

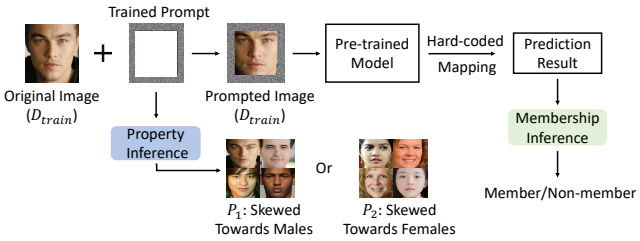



Figure 1: Overview of prompt usage and inference attacks. The prompt is a pixel patch. The prompted image is an original image with an added prompt. Property inference infers sensitive properties of the target prompt's training dataset that the PaaS provider does not intend to disclose. Membership inference infers whether a given sample was in the target prompt's training dataset.

severely damage the provider's privacy as well as intellectual property. However, to the best of our knowledge, previous research about such privacy risks has focused on ML models at the model level and has not yet been explored on prompts at the input level. As the number of learned parameters is significantly reduced in prompt learning, it is natural to assume that this paradigm would compress the proprietary information of its training data, leading to less effective privacy attacks (see Section 2.2). This motivates us to investigate whether a prompt also leaks the proprietary information of its training data that the PaaS provider does not intend to disclose, especially when such prompts are generated from images containing sensitive private information.

**Contributions.** In this paper, we conduct the first privacy risk assessment of prompts learned by prompt learning. We focus on prompt learning for image classification tasks [4], which represents one of the most promising directions in computer vision research [4,5,8,20,22,31,49,51,56]. Our primary objective is to determine *to what extent a visual prompt possesses the potential to disclose confidential information.* Specifically, we perform property inference and membership inference, two dominant privacy attacks against ML models [7,14,36,46], where the former aims to deduce sensitive properties of the dataset used to train the target prompt, and the latter determines if a given data sample is part of the target prompt's training dataset. We adopt the existing attack methodologies for property inference [3,14] and membership inference (neural network-based attacks [44,46], metric-based attacks [47], and gradient-based attacks [24,38]). Note that our goal is not to develop new property inference attacks or membership inference attacks against prompts. Instead, we aim to use existing methods with well-established threat models to systematically assess the privacy risks of prompt learning. The overview of our study is depicted in Figure 1.

The empirical evaluation shows prompts are susceptible to property inference attacks across multiple datasets and pre-trained models. For example, we can achieve at least 81% accuracy in inferring different target properties from prompts learned for CelebA [34]. Moreover, when inferring the training dataset size of the prompt, we can achieve 100% test accuracy in all cases. We also conduct a cost analysis to show that the adversary can either train the shadow prompts for fewer epochs or use fewer shadow prompts to minimize their cost while maintaining decent attack performance.

Our study also provides empirical evidence that membership inference poses a practical threat to prompts. The experimental results demonstrate that existing attack methodologies are effective across a range of datasets and pre-trained models. In particular, the metric-based attack with modified prediction entropy is the most effective one, e.g., achieving 93% membership inference accuracy on the AFAD dataset [39]. The gradient-based attacks follow closely behind and outperform the neural network-based (NN-based) attacks. We further investigate factors that may affect membership inference from both the victim's and the adversary's perspectives. Specifically, from the victim's side, we conduct a detailed analysis of the relationship between the overfitting levels of prompts and attack success [50]. The results indicate that the attack success is positively correlated with the overfitting level. Moreover, excessive training epochs and inadequate training data increase overfitting levels, exacerbating the privacy threat posed by membership inference attacks. From an adversarial perspective, we demonstrate that the adversary can relax the assumption that the shadow dataset has the same distribution as the target prompt's training dataset. This finding further exemplifies the membership privacy risks of prompts learned by prompt learning.

We also conduct preliminary investigations into mitigating privacy risks associated with prompt learning. In particular, we explore the effectiveness of adding Gaussian noise to prompts, as proposed in prior research [18,54,57]. Our experiments demonstrate that there exists a decent utility-defense trade-off when mitigating both naive and adaptive membership inference attacks. However, when defending against property inference attacks, we need higher Gaussian noise to reduce the attack performance, leading to unacceptable utility deterioration. Our findings indicate that the statistical information of the training dataset in the target prompts is harder to hide than individual information, i.e., membership. Our results highlight the need for further research into more effective defense mechanisms for mitigating property inference attacks in prompt learning.

**Impact.** This study presents an exploration of the privacy risks associated with prompt learning, an emerging machine-learning paradigm. Our investigation represents the first of its kind in this area. Our findings indicate that prompts learned through prompt learning are susceptible to privacy breaches. We hope our study will increase the awareness of the stakeholders when deploying prompt learning in real-world applications. Moreover, to facilitate research in the field, we will share our code and models.

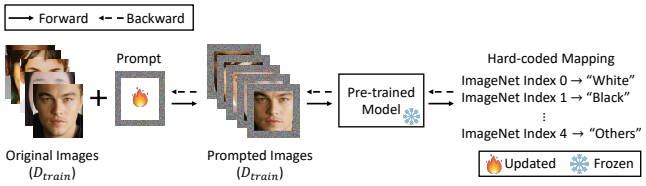



Figure 2: Overview of visual prompt learning (VPL). We learn an input prompt via back-propagation [4] at the input transformation stage. We apply hard-coded mapping [13] to map the pre-trained model's outputs into the target labels at the output transformation stage.

# 2 Preliminaries

## 2.1 Prompt Learning

**Overview.** Prompt learning is a new machine-learning paradigm introduced to address the limitations of fine-tuning [4, 5, 20, 25, 26, 30, 31, 41, 56]. It aims at learning a task-specific prompt that can be added to the input data while keeping the pre-trained model's parameters frozen. With this new paradigm, the service provider can share the same pre-trained model across various downstream tasks with different prompts in a space- and computation-efficient manner. In this paper, we focus on prompt learning in computer vision, i.e., *visual prompt learning (VPL)* [4]. It is generally composed of two stages: input transformation and output transformation.

**Input Transformation.** As shown in Figure 2, the goal of input transformation is to learn an input prompt $\delta$ in the pixel space, *i.e., in the form of a single image*, via back-propagation. Given a dataset $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$, a pre-trained model $\mathcal{M}$ parameterized by $\omega$, and a prompt $\delta$ parameterized by $\theta$, the prompt generation process $q(\mathcal{D}, \mathcal{M})$ uses Equation 1 to maximize the likelihood of $\mathcal{Y}$:

$$\max_{\theta} P_{\theta;\omega}(\mathcal{Y}|\mathcal{X} + \delta_\theta), \tag{1}$$

where the prompt parameters $\theta$ are learned via back-propagation and the model parameters $\omega$ are frozen. Note that the prompt can be any visual template chosen by the users, e.g., padding [4]. At inference time, the learned prompt $\delta$ is added to each test image $x$ to specify the task.

**Output Transformation.** Usually, the pre-trained model has a different number of classes from the downstream tasks. To accomplish the downstream task, the prompt owner supplies a label mapping scheme $\tau$ to map the model's outputs into the target labels. As shown in Figure 2, a commonly used scheme is hard-coded mapping [13]. It consists of mapping the first $n$ pre-trained model class indices to the downstream class indices, where $n$ is the number of classes in the downstream task. The unassigned pre-trained classes are left out for the loss computation. We rely on hard-coded mapping due to its simplicity and proven effectiveness [4].

## 2.2 Prompt Learning vs. Fine-Tuning

**Training Time.** The fine-tuning paradigm updates all parameters of the pre-trained model via back-propagation. However, as shown in Figure 2, VPL learns a visual prompt, *i.e., in the form of a single image*, on the training dataset $\mathcal{D}_{train} = (\mathcal{X}, \mathcal{Y})$. During the back-propagation, the pre-trained model is frozen, and only the parameters of the visual prompt are updated. In this way, prompt learning dramatically lowers the bar for users adapting large-scale vision models for real-world applications. Prompt learning saves significant training resources and storage space, especially when a pre-trained model serves multiple downstream tasks. For example, the Vision Transformer (ViT-B) [23] we use in later experiments has 86,567,656 parameters, and the visual prompt, i.e., a padding template with a prompt size of 30, has 69,840 parameters. For each downstream task, the fine-tuning paradigm updates the entire model (86,567,656 parameters), whereas, in prompt learning, a single prompt, i.e., a single image with only 69,840 parameters, is updated. The number of parameters updated by prompt learning is only 0.08% of those of fine-tuning, so it is natural to assume that prompt learning would heavily compress the training dataset information, leading to less effective privacy attacks. However, we show that the prompts are still susceptible to two privacy attacks in later experiments.

**Inference Time.** As shown in Figure 1, both the trained prompt and pre-trained models are involved in the inference process. Given a test image $x$, the user gets the prompted image, i.e., adding the trained prompt $\delta$ to $x$, and then feeds the prompted image into the pre-trained model $\mathcal{M}$ to get the prediction result. In the fine-tuning approach, the user directly feeds the given test image $x$ into the fine-tuned model to get the prediction result.

## 2.3 Application Scenario

Taking a medical researcher as an example, they aim to classify CT images for COVID-19 diagnosis. Instead of hiring a computer vision expert to fine-tune a model, the researcher can request a prompt from a PaaS provider. They can either opt for a publicly available pre-trained model or allow the PaaS provider to select a suitable one for them. The provider uses their proprietary data, e.g., CT images with explicit consent, to learn a customized prompt and return it to them. At inference time, the researcher simply combines their testing data with the prompt and feeds them to the pre-trained model to get the predictions. In this way, users minimize their effort in developing a well-generalized prompt and keep their data on-premise, while the PaaS provider can reuse a single pre-trained model to support multiple downstream tasks. Meanwhile, the user can adapt to different tasks, e.g., clinical decision support, by trivially switching to different prompts. These benefits differentiate PaaS from machine learning as a service (MLaaS) [42].

## 3 Property Inference Attacks

We first measure the privacy risks of prompt learning through the lens of property inference attacks. Our objective here is not to devise novel attacks for prompts but rather to leverage well-established threat models and existing techniques to gauge the privacy implications of prompts.

### 3.1 Threat Model

**Attack Scenario.** The PaaS provider is a resourceful entity that uses a pre-trained model $\mathcal{M}$ and their private dataset $\mathcal{D}_{target}$ to create well-generalized prompts $\Delta$ for downstream tasks. The adversary can be any legitimate user of this PaaS provider and can obtain a prompt $\delta$ for a target downstream task together with the white-box access to $\mathcal{M}$. Note that a pre-defined label mapping $\tau$ is also provided by the PaaS provider (see Section 2.1). The adversary runs the target downstream task locally and does not interact with the PaaS provider.

**Adversary's Goal.** Given a target prompt $\delta_{target}$, the goal of the adversary is to infer confidential macro-level properties of the training dataset $\mathcal{D}_{target}$, which the PaaS provider does not intend to share. Taking a prompt $\delta_{target}$ for facial recognition as an example, the adversary may intend to infer the confidential properties of the private dataset $\mathcal{D}_{target}$, such as the proportion of males and the proportion of youth. The adversary considers such confidential properties as targets, causing real-world harm to the PasS provider, e.g., reputation damages, if the adversary can infer that certain classes of people, such as minorities, are underrepresented in the training data [14]. For simplicity, we focus on binary properties, such as if the proportion of males in the training dataset is 30% or 70%, in most of our experiments, following previous work [14]. We later show that our attack can be generalized to properties with multiple choices (see Section 3.4).

**Adversary's Knowledge and Capability.** We assume that the adversary has white-box access to the pre-trained model $\mathcal{M}$ and the label mapping $\tau$. Note that the white-box access in the paper is more restricted than conventional white-box access, as the latter can retrieve all information about the model, such as model parameters and intermediate outputs. In this paper, the adversary only needs to know the architecture and version of the pre-trained model from the PaaS provider, and such knowledge is often disclosed by the PaaS provider for marketing purposes. We also assume that the adversary has a shadow dataset $\mathcal{D}_{shadow}$ of similar distribution as $\mathcal{D}_{target}$. For instance, in our evaluation (see Section 3.3), we select both $\mathcal{D}_{shadow}$ and $\mathcal{D}_{target}$ from the same dataset CelebA [34]. These two subsets are disjoint and may have different statistical properties, such as gender/race/age ratios. We emphasize that previous property inference attacks also make the same assumption [14, 55].

### 3.2 Measurement Methodology

**Shadow Prompt Generation.** Given a shadow dataset $\mathcal{D}_{shadow}$ and associated data properties $\mathcal{P} = \{p^1, ..., p^k\}$, the adversary uses Equation 2 to generate the shadow prompts:

$$\Delta_{shadow} = \{q(s_{\mathcal{P}}(\mathcal{D}_{shadow}, \Phi_i, N_i), \mathcal{M})\}_{i=1}^{m}, \tag{2}$$

where $s_{\mathcal{P}}$ denotes a sampling function that samples $N_i$ data points from $\mathcal{D}_{shadow}$ without replacement and the distribution of sampled data with properties $\mathcal{P}$ satisfying the conditions $\Phi_i$. Note that, $\Phi_i = \{\phi_i^1, \phi_i^2, ..., \phi_i^k\}$, $\phi_i^k$ is the actual value of $p^k$ in round $i$, $m$ denotes the number of shadow prompts, and $N_i$ denotes the size of the sampled dataset from $\mathcal{D}_{shadow}$ in round $i$. In previous work [14, 36, 55], apart from the targeted property, say $p^1$ (and associated $\phi^1$), they tend to use a fixed set of other conditions, i.e., $\{\phi^2, ..., \phi^k\}$, and $N$. For example, the target property is the proportion of males. They tend to keep the training data size the same when training all target prompts and shadow prompts in their evaluation. However, the training data size of target prompts and shadow prompts is likely to be different in a realistic scenario. If the target prompt is trained on 500 samples with 70% males, but shadow prompts are trained on 2000 samples with 70% males. Such discrepancies in training dataset sizes, e.g., 500 and 2000, may influence the attack performance. In contrast to those approaches, we consider a *mixed setting* by design. As we can see in Equation 2, in every round $i$, we generate a prompt $\delta_i$ from a subset sampled by $s_{\mathcal{P}}(\mathcal{D}_{shadow}, \Phi_i, N_i)$ with properties $\mathcal{P}$ satisfying different $\Phi_i$. For instance, given $\mathcal{P} = \{youth, male\}$, $\Phi = \{70\%, 70\%\}$, and $N = 2000$, $s_{\mathcal{P}}(\mathcal{D}, \Phi, N)$ samples 2000 data points from $\mathcal{D}$ to train the prompt. Among them, 980 data points are *young males*, 420 data points are *young females*, 420 data points are *old males*, and 180 data points are *old females*. As such, our approach can guarantee a more realistic shadow prompt generation and fairer evaluation.

**Attack Model Training.** After obtaining the shadow prompts $\Delta_{shadow}$, we can build the attack model for each property $p^k$:

$$\mathcal{A} : \Delta_{shadow} \rightarrow y^k. \tag{3}$$

We train the attack model $\mathcal{A}$ by optimizing the following loss function:

$$\mathcal{L}[\mathcal{A}(\Delta_{shadow}), y^k], \tag{4}$$

where $\mathcal{L}$ is a cross-entropy loss function in this paper. Concretely, the attack model takes $\delta_i \in \Delta_{shadow}$ as input. To incorporate the input size of the attack model, we use zero value to pad it to an image of size $224 \times 224$, with RGB channels. This means the attack model is an image classifier. The adversary then treats the corresponding condition value $\phi_i^k$ of the target property $p^k$ as the class labels $y_i^k$. To infer the target property $p^k$ of $\mathcal{D}_{target}$, the adversary queries the attack model $\mathcal{A}$ with $\delta_{target}$ and obtains the corresponding prediction result, i.e., the exact condition value of $p^k$.

Table 1: Experimental settings of the property inference attacks with the corresponding attack performance.

| Inference Task | Dataset | Downstream Task | Target Property | Inference Labels | Test Accuracy RN18 | BiT-M | ViT-B |
|---|---|---|---|---|---|---|---|
| $T_1$ | CIFAR10 | Image Classification | Size ($T_1^{size}$) | {500, 2000} | 100.00 | 100.00 | 100.00 |
| $T_2$ | CelebA | Multi-Atrribute Classification | Size ($T_2^{size}$) | {500, 2000} | 100.00 | 100.00 | 100.00 |
| | | | Proportion of Males ($T_2^{male}$) | {30%, 70%} | 99.75 | 99.25 | 93.00 |
| | | | Proportion of Youth ($T_2^{youth}$) | {30%, 70%} | 93.00 | 90.75 | 81.00 |
| $T_3$ | UTKFace | Race Classification | Size ($T_3^{size}$) | {500, 2000} | 100.00 | 100.00 | 100.00 |
| | | | Proportion of Males ($T_3^{male}$) | {30%, 70%} | 80.50 | 80.50 | 82.00 |
| | | | Proportion of Youth ($T_3^{youth}$) | {30%, 70%} | 81.75 | 87.50 | 84.00 |
| $T_4$ | AFAD | Age Classification | Size ($T_4^{size}$) | {500, 2000} | 100.00 | 100.00 | 100.00 |
| | | | Proportion of Males ($T_4^{male}$) | {30%, 70%} | 80.75 | 78.00 | 72.25 |

### 3.3 Measurement Settings

**Datasets and Downstream Tasks.** We use four datasets in our study, including CIFAR10 [1], CelebA [34], UTKFace [52], and AFAD [39]. These datasets contain sensitive properties (the proportion of males, the proportion of youth, etc.) and are widely used to evaluate the performance of property inference attacks [14, 55]. The introduction of these datasets and corresponding downstream tasks are as follows.

- **CIFAR10** is a benchmark dataset for image classification that contains 60K images in 10 classes. In this paper, the downstream task is a 10-class image classification.
- **CelebA** is a large-scale facial attribute dataset containing more than 200K facial images with 40 binary attributes. We pick three attributes, including *MouthSlightlyOpen*, *Attractive*, and *WearingLipstick*, and use their combinations to create an 8-class attribute classification as the downstream task.
- **UTKFace** has about 23K facial images. Each image has three attributes: *gender*, *race*, and *age*. We consider race classification, i.e., White, Black, Asian, Indian, and Others, as the downstream task.
- **AFAD** is short for Asian Face Age Dataset. It contains more than 160K facial images, each with *age* and *gender* attributes. In this paper, we consider age classification as the downstream task. Specifically, we divide the values of *age* attribute into five bins: $15 \leq age < 20$, $20 \leq age < 25$, $25 \leq age < 30$, $30 \leq age < 35$, and $35 \leq age < 40$, leading to a 5-class image classification.

**Property Inference Task Configurations.** For each task, we split the dataset into three disjoint subsets $\mathcal{D}_{target}$, $\mathcal{D}_{shadow}$, and $\mathcal{D}_{validation}$ in the ratio of $0.475 : 0.475 : 0.05$. $\mathcal{D}_{target}$ and $\mathcal{D}_{shadow}$ are used to develop the target prompt set $\Delta_{target}$ and shadow prompt set $\Delta_{shadow}$, respectively. We evaluate the utility of all prompts on $\mathcal{D}_{validation}$. We train 2000 shadow prompts to construct the attack training dataset and 400 target prompts to build the attack testing dataset in our experiments. Our property inference targets include training dataset size, proportion of males, and proportion of youth. Note that recent research demonstrates that the size of the training dataset significantly affects the performance of the model, necessitating substantial efforts to identify the optimal values [35, 58]. Consequently, we also view the training dataset size as confidential information and as one of our inference objectives. We outline the details of all inference tasks below and summarize them in Table 1.

- **Inference Task on CIFAR10 ($T_1$).** For CIFAR10, we only consider the size of the prompt training dataset $N$ as the property inference target ($T_1^{size}$). We focus on two training data sizes, i.e., $y^1 \in \{500, 2000\}$, and run the sampling function (see Equation 2) 1000 times on $\mathcal{D}_{shadow}$ to generate 1000 shadow prompts for each training data size. Meanwhile, we generate 200 target prompts in the same manner for each training data size.
- **Inference Task on CelebA ($T_2$).** For CelebA, we consider the size of the prompt training dataset ($T_2^{size}$), the proportion of males ($T_2^{male}$), and the proportion of youth ($T_2^{youth}$) of the data samples used to train the target prompts as the property inference targets. $T_2^{male}$ is based on the *male* attribute, and $T_2^{youth}$ is based on the *young* attribute. Both attributes are binary. The inference labels of each property are: $y^1 \in \{500, 2000\}$, $y^2 \in \{30\%, 70\%\}$, and $y^3 \in \{30\%, 70\%\}$. Recall that we consider a mixed data sample strategy. Given these three properties, we end up with eight sampling functions in total. We run each sampling function 250 times on $\mathcal{D}_{shadow}$ and 50 times on $\mathcal{D}_{target}$ to generate the shadow prompt set $\Delta_{shadow}$ and target prompt set $\Delta_{target}$, respectively.
- **Inference Task on UTKFace ($T_3$).** For UTKFace, we also consider the size of the prompt training dataset ($T_3^{size}$), the proportion of males ($T_3^{male}$), and the proportion of youth ($T_3^{youth}$) as the property inference targets. Note that $T_3^{male}$ is based on the *gender* attribute, and $T_3^{youth}$ is based on the *age* attribute. Specifically, we use the median of *age* values from all images, i.e., 30, as the threshold. We then label samples with $0 \leq age \leq 30$ as Young and $30 \leq age \leq 116$ as Old. The inference labels

of each property are the same as those of CelebA. Thus, we follow the same sampling settings as those of $T_2$ to generate the shadow and target prompts.

- **Inference Task on AFAD ($T_4$).** For AFAD, we consider the size of the prompt training dataset ($T_4^{size}$) and the proportion of males ($T_4^{male}$) as the property inference targets. $T_4^{male}$ is based on the *gender* attribute. The inference labels of each property are: $y^1 \in \{500, 2000\}$ and $y^2 \in \{30\%, 70\%\}$. We use four sampling functions to generate the shadow and target prompts. We run each sampling function 500 times on $\mathcal{D}_{shadow}$ and 100 times on $\mathcal{D}_{target}$ to generate the shadow prompt set $\Delta_{shadow}$ and target prompt set $\Delta_{target}$, respectively.

**Metric.** As the attack training/testing dataset is balanced in terms of class distribution, we use test accuracy as the main metric to evaluate the prompt utility and the property inference attacks.

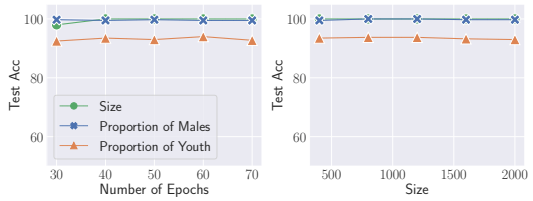
**Pre-trained Models.** We select three representative vision models in our experiments, including ResNet-18 (RN18) [15], Big Transfer (BiT-M) [23], and Vision Transformer (ViT-B) [12]. More details can be found in Appendix A.

**Prompts.** We follow the default training settings [4] to train prompts on the above vision models. Specifically, we choose the padding template with a prompt size of 30. The number of parameters for each prompt is calculated as $2 \times C \times p \times (H + W - 2p)$, where $p$, $C$, $H$, and $W$ are prompt size, image channels, height, and width, respectively. All images are resized to 224 $\times$ 224 to match the input of the pre-trained models. The number of parameters for each prompt is 69,840. We leverage the same hard-coded mapping method [4] to map the first $n$ indices of the pre-trained model's outputs to the target labels, where $n$ is the number of target classes. We adopt cross-entropy as the loss function and SGD as the optimizer with a learning rate of 40 and the cosine scheduler. In our property inference attacks, we train all prompts for 50 epochs for efficiency.

**Attack Models.** We leverage the pre-trained RN18 [15] as the backbone of the attack model $\mathcal{A}$. We fit a linear classifier on top of the pre-trained RN18 to infer the property labels. We employ cross-entropy as the loss function and Adam as the optimizer with a learning rate of 1e-5. The attack model is trained on the shadow prompt set $\Delta_{shadow}$ for 100 epochs.

## 3.4 Measurement Results

**Property Inference Privacy Risks.** We report the main results on four datasets and three pre-trained models in Table 1. In general, we observe that proposed attacks achieve good performance across different pre-trained models and datasets. For example, on CelebA and RN18, we achieve at least 93.00% accuracy in inferring target properties. Furthermore, we achieve maximum performance (100.00%) on all datasets, considering the size of the prompt training dataset as the target property.



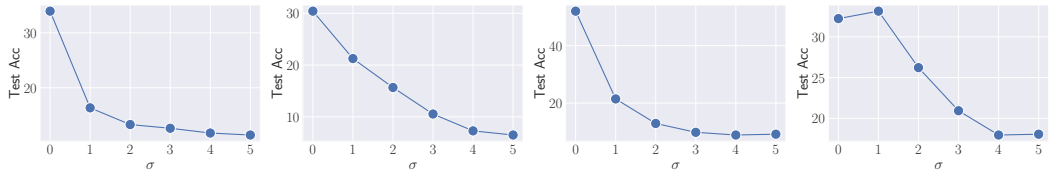

(a) Epoch Effect (b) Size Effect

Figure 3: Attack performance of the proposed property inference attacks on CelebA with (a) different numbers of epochs for training shadow prompts and (b) different sizes of the attack training dataset, using RN18 as the pre-trained model.

Additionally, we observe that the pre-trained model has a moderate influence on the attack performance. Specifically, when inferring the proportion of youth on UTKFace ($T_3^{youth}$ in Table 1), the test accuracy is 81.75% on RN18, 87.50% on BiT-M, and 84.00% on ViT-B. Although the test accuracy varies across pre-trained models and datasets, the proposed attacks are generally effective, indicating prompts are indeed vulnerable to property inference attacks.

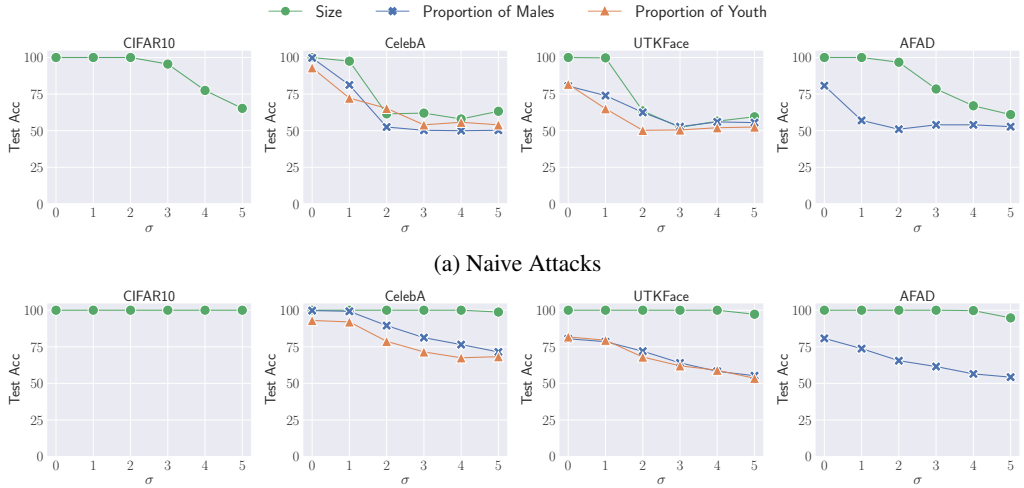
**Extension to Multi-Class Property Inference.** In the above experiments, we treat property inference as a binary classification task. Here, we extend it to multi-class classification and explore if the adversary can infer finer granularity information from the prompts. To this end, we adjust the condition values of the proportion of males in CelebA to {10%, 30%, 50%, 70%, 90%} and use RN18 as the pre-trained model. Note that the condition values of the training dataset size and the proportion of youth remain the same. In turn, we have 20 sampling functions in total. We keep the sizes of $\Delta_{shadow}$ and $\Delta_{target}$ unchanged and run each sampling function 100 times on $\mathcal{D}_{shadow}$ and 20 times on $\mathcal{D}_{target}$ to generate the shadow prompt set $\Delta_{shadow}$ and target prompt set $\Delta_{target}$, respectively. We further adjust the condition values of the training dataset size in CIFAR10 to {500, 1000, 1500, 1750, 1800, 2000} and use RN18 as the pre-trained model to explore the performance of the property inference attack when the range of options for the dataset size is closer together. To this end, we have 6 sampling functions in total and run each sampling function 400 times on $\mathcal{D}_{shadow}$ and 80 times on $\mathcal{D}_{target}$ to generate the shadow prompt set $\Delta_{shadow}$ and target prompt set $\Delta_{target}$, respectively. The test accuracy for the proportion of males is 90.25%, while for training dataset size is 95.40%, demonstrating that property inference attacks can successfully infer finer granularity information from prompts.

**Takeaways.** We show that the property inference attacks achieve remarkable performance on diverse datasets and pre-trained models. Moreover, the proposed attacks can be extended to multi-class classification, providing evidence that the adversary can infer fine granularity property information from prompts.




(a) CIFAR10 (b) CelebA (c) UTKFace (d) AFAD

Figure 4: Target performance on four datasets. The x-axis denotes the magnitude of the Gaussian noise, from 0 to 5, where 0 means the proposed defense mechanism is not implemented. The y-axis represents the target performance on the downstream tasks with respect to the average test accuracy of all target prompts in the attack testing dataset.




(a) Naive Attacks

(b) Adaptive Attacks

Figure 5: Attack performance of (a) naive attacks where the adversary is not aware of the proposed defense and (b) adaptive attacks on four datasets. The x-axis denotes the magnitude of Gaussian noise, from 0 to 5, where 0 means the proposed defense mechanism is not implemented. The y-axis represents the attack performance with respect to the test accuracy.

## 3.5 Factors Affecting Property Inference

We conduct an empirical analysis to investigate the factors that may influence the performance and cost of property inference attacks on prompts.

**Number of Epochs.** Previously, we train both shadow prompts and target prompts for 50 epochs. However, it is likely that the adversary has no knowledge about the number of epochs for target prompts. Next, we investigate whether the number of epochs in the training process of shadow prompts must match that of the target prompts in order to maintain a strong attack performance. Concretely, we vary the number of epochs for shadow prompts from 30 to 70 while fixing the number of epochs for target prompts to 50. The minimum number of epochs is 30 because the prompt starts to outperform the pre-trained model solely on the downstream task at this point. We show the attack performance on CelebA in Figure 3a. In general, the proposed attacks work similarly well when the number of epochs for target and shadow prompts do not match. The results also show that the proposed attacks can achieve comparable performance even with fewer epochs, e.g., 30 epochs, to train shadow prompts. In addition, increasing the number of epochs for shadow prompts does not improve the attack performance. For example, the test accuracy for inferring the proportion of youth is between 92.50% and 94.00% depending on the number of epochs. This implies that the proposed attacks are robust to variations in the number of epochs for training shadow prompts, making them more practical and efficient.

**Attack Training Dataset Size.** So far, we have assumed the adversary can rely on an attack training dataset containing 2000 shadow prompts. However, creating such a dataset costs considerable computational resources. Hence, we investigate the influence of the attack training dataset size on the attack performance. Specifically, we randomly sample balanced subsets from the original attack training dataset on CelebA with different sizes {400, 800, 1200, 1600, 2000}. The size of the attack testing dataset remains the same as for the previous experiments, i.e., 400 target prompts. As shown in Figure 3b, the size of the training dataset only has negligible influence on the

attack performance, indicating that a relatively small number of training samples, e.g., 400 shadow prompts, are sufficient to launch the property inference attacks against prompts. This finding implies that the cost of the proposed attack can be further reduced.

**Takeaways.** We demonstrate that the proposed attacks can be performed with cost-efficiency by training the shadow prompts with fewer epochs or a smaller number of shadow prompts. We further show that to achieve a good attack performance, the adversary must have a shadow dataset of similar distribution as the target dataset and must have access to the same pre-trained model. The results are displayed in Appendix B.

### 3.6 Defense

**Gaussian Noise as Defense.** We propose a defense mechanism [18, 54, 57]. Specifically, the PaaS provider adds Gaussian noise $\mathcal{N}(0,\sigma^2 I)$ to the released prompts, resulting in noised target prompts $\Delta'_{target} = \{\delta_i + \varepsilon_i \mid \forall \delta_i \in \Delta_{target}, \varepsilon_i \sim \mathcal{N}(0,\sigma^2 I)\}$. The magnitude of the noise is controlled by the value of $\sigma$, with larger values corresponding to higher noise. We examine the effectiveness of the proposed defense with $\sigma \in \{1,2,3,4,5\}$. We first report the target performance, i.e., prompt utility, on all datasets in Figure 4. The evaluation metric is the average test accuracy of all target prompts in the attack testing dataset on specific downstream tasks. In general, we observe that the target performance decreases on all datasets by a large margin with the increase of $\sigma$. For example, the prompt utility decreases from 33.95% to 10.55%, which is even lower than random guess (12.50%), meaning the prompt is no longer usable. We present the attack performance in Figure 5a. We can observe that the effectiveness of the proposed attack significantly declines with the increase of $\sigma$. The test accuracy on CelebA and UTKFace drops to almost random guess when $\sigma \geq 2$. The attacks on CIFAR10 are more robust to the defense, but the performance still starts decreasing when $\sigma = 3$.

**Adaptive Attacks.** We further consider an adaptive adversary [21] who is aware of the defense mechanism, i.e., that Gaussian noise has been added to the target prompts. They can construct their attack training dataset with noised shadow prompts. Similarly, we set $\sigma \in \{1,2,3,4,5\}$ for both shadow and target prompts. We report the performance of adaptive attacks on all datasets in Figure 5b. The results show that the attack performance declines less and more slowly. For example, the attack performance barely decreases when $\sigma = 1$ on all datasets. In addition, when considering the size of the prompt training dataset as the target property, the attack performance only has negligible degradation with the growth of Gaussian noise. For example, the attack performance has almost no drop even with $\sigma = 5$ on all datasets.

**Takeaways.** These findings indicate that adding Gaussian noise as a defense mechanism can ostensibly decrease the attack performance. But the defender suffers from unacceptable prompt utility degradation. Moreover, this defense can be bypassed by the adaptive attack. We leave it as future work to investigate more effective defenses. We later show that the proposed defense can achieve a decent utility-defense trade-off by using a smaller $\sigma$, e.g., $\sigma = 0.6$, indicating that the statistical information of the training dataset in the target prompts is harder to hide than individual information, i.e., membership (see Section 4.7).

## 4 Membership Inference Attacks

In this section, we leverage the membership inference attacks to quantify the privacy risks of prompts.

### 4.1 Threat Model

**Adversary's Goal.** In membership inference, the goal of the adversary is to infer whether a given data sample $x$ is in the training dataset of the target prompt $\delta_{target}$.

**Adversary's Knowledge and Capability.** Similar to the property inference attack, the adversary can query the PaaS service to get $\delta_{target}$ and has white-box access to the pre-trained model $\mathcal{M}$. The adversary has a shadow dataset $\mathcal{D}_{shadow}$ that is from the same distribution as $\mathcal{D}_{target}$ to train the shadow prompt $\delta_{shadow}$. We later demonstrate that the adversary can operate in a data-free manner, i.e., leveraging $\mathcal{D}_{shadow}$ that comes from a different distribution than $\mathcal{D}_{target}$.

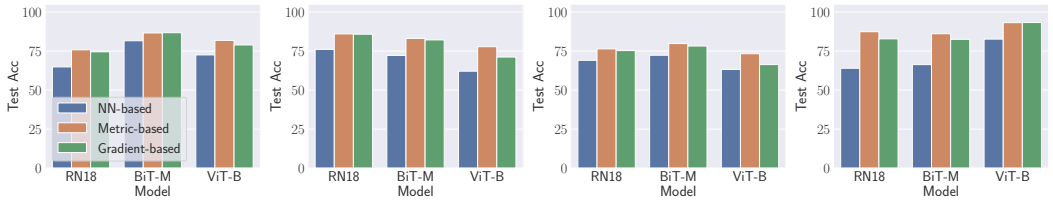
### 4.2 Measurement Methodology

**Attack Setup.** The adversary first divides the shadow dataset into two disjoint subsets: $\mathcal{D}^{train}_{shadow}$, referred to as the member split, and $\mathcal{D}^{test}_{shadow}$, referred to as the non-member split. The member split is then utilized for training the shadow prompt $\delta_{shadow}$, which mimics the behavior of $\delta_{target}$.

**Attack Descriptions.** We adopt three types of membership inference attacks, i.e., neural network-based (NN-based) attacks [44], metric-based attacks [47], and gradient-based attacks [24, 38]. We outline their technical details below.

*NN-based Attacks [44].* The adversary constructs the attack training dataset on $\mathcal{D}_{shadow}$. Specifically, they combine each sample in $\mathcal{D}_{shadow}$ with the shadow prompt trained on $\mathcal{D}^{train}_{shadow}$ and query the corresponding pre-trained model to get the top-5 posteriors as attack input features. Then, for each sample in the member split, the adversary labels the corresponding top-5 posteriors as "member." For samples that belong to the non-member split, their top-5 posteriors are labeled as "non-member." At inference time, the adversary queries the pre-trained model with the given data sample $x$ and $\delta_{target}$ to obtain the top-5 posteriors and feeds them to the attack model to obtain its membership prediction.

*Metric-based Attacks [47].* Song and Mittal propose metric-based attacks using four metrics, i.e., prediction correctness

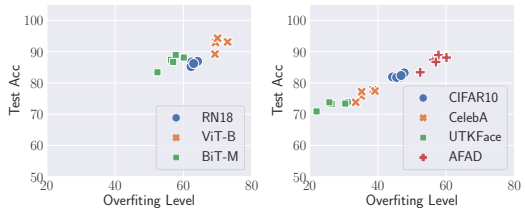



(a) CIFAR10 (b) CelebA (c) UTKFace (d) AFAD

Figure 6: Attack performance of three membership inference attacks on four datasets.

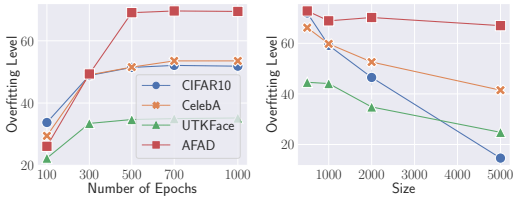
(metric-corr), prediction confidence (metric-conf), prediction entropy (metric-ent), and modified prediction entropy (metric-ment). Unlike NN-based attacks where a neural network is trained to make membership predictions, metric-based attacks first calculate class-wise thresholds over $\delta_{shadow}$. Then, at inference time, the adversary calculates the metric values and compares them with the pre-calculated thresholds to determine the membership status for given data samples. It is worth noting that in scenarios where the adversary possesses data from a different distribution than the target dataset, we calculate an overall threshold for all classes. This is because certain classes present in the target dataset may not be represented in the shadow dataset, so class-specific thresholds would not be applicable.

*Gradient-based Attacks [38].* Nasr et al. propose gradient-based attacks on the basis of the NN-based attacks by incorporating augmented input information. Specifically, the adversary has white-box access to the pre-trained model and target prompt with its intermediate computations, e.g., gradients. They combine each sample $x$ with the prompt and input resulting data into the pre-trained model to obtain top-5 posteriors, the loss incurred during the forward pass, the gradient of the prompt during the backward pass, and an indicator that denotes the correctness of the prediction. These obtained data are treated as the attack input for the attack model.




(a) Pre-trained Models (b) Datasets

Figure 7: Overfitting levels of target prompts across (a) different pre-trained models on AFAD and (b) different datasets using BiT-M as the pre-trained model. Different points with the same marker denote different runs of the same pre-trained model/dataset using different random seeds.




(a) Epoch Effect (b) Size Effect

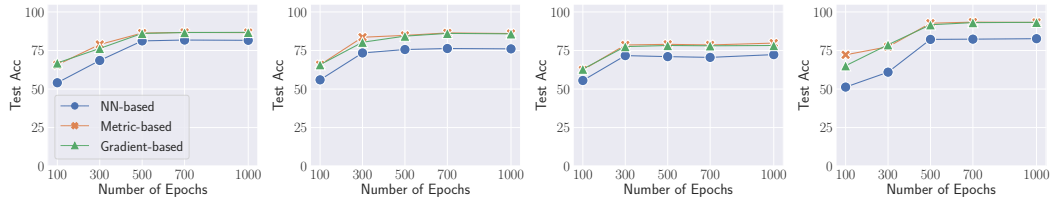
Figure 8: Overfitting levels of target prompts with (a) different numbers of epochs and (b) different sizes of the training dataset, using ViT-B as the pre-trained model.

## 4.3 Measurement Settings

**Datasets and Downstream Tasks.** We reuse CIFAR10, CelebA, UTKFace, and AFAD to evaluate membership inference attacks. The downstream tasks for all datasets are the same as those for property inference attacks (see Section 3.3). We randomly sample 8000 data samples for each dataset in the main experiments and then evenly split each dataset into four disjoint sets, i.e., $\mathcal{D}^{train}_{target}$, $\mathcal{D}^{test}_{target}$, $\mathcal{D}^{train}_{shadow}$, and $\mathcal{D}^{test}_{shadow}$. $\mathcal{D}^{train}_{target}$ is used to develop the target prompt $\delta_{target}$, and $\mathcal{D}^{test}_{target}$ is the evaluation set. $\mathcal{D}^{train}_{shadow}$ and $\mathcal{D}^{test}_{shadow}$ are used to build the attack model as discussed in Section 4.2.

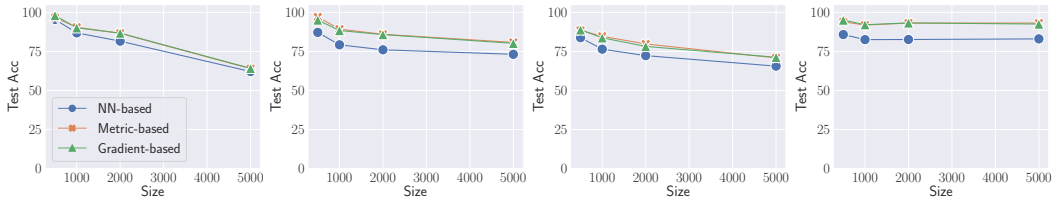
**Attack Configurations.** All experimental settings of the pre-trained models and target prompts are the same as those for property inference attacks except for the number of epochs. We follow the default setting to train all prompts for 1000 epochs. For attacks that leverage neural networks as the attack model, we employ the cross-entropy loss function and optimize it using Adam optimizer. We conduct a grid search on {1e-2, 1e-3, 1e-4, 1e-5} to determine the optimal learning rate for each attack, and all attack models are trained for 100 epochs. For the NN-based attacks, we use a 2-layer MLP as the attack model and set the size of the hidden layer to 32. For the gradient-based attacks, we utilize an attack model composed of four sub-networks, each corresponding to one attack information (gradient, top-5 posteriors, loss, and indicator), and the outputs of these sub-networks are concatenated to form the final input of a 2-layer MLP.

**Metric.** Following the convention [17, 46], we use test accuracy as the main metric to evaluate the attack performance.




(a) CIFAR10 (b) CelebA (c) UTKFace (d) AFAD

Figure 9: Attack performance of three membership attacks with varying number of epochs, using ViT-B as the pre-trained model.




(a) CIFAR10 (b) CelebA (c) UTKFace (d) AFAD

Figure 10: Attack performance of three membership attacks with different sizes of $\mathcal{D}_{target}^{train}$, using ViT-B as the pre-trained model.

## 4.4 Measurement Results

**Membership Inference Privacy Risks.** We report the performance of three membership inference attacks in Figure 6. We conduct three separate runs of each attack experiment and report the average values as the final results. We observe that metric-based attacks achieve the best performance in most cases, e.g., 93.20% membership inference attack accuracy on AFAD. Unless otherwise specified, we use metric-ment attacks as the representative of the metric-based attacks, as they consistently achieve the best performance across all datasets (see Appendix C). The gradient-based attacks also exhibit strong performance, with results that are closely comparable to those of the metric-based attacks. Song and Mittal [47] also report that the metric-based attacks can outperform NN-based attacks and have similar performance as gradient-based attacks. NN-based attacks perform worse than gradient-based attacks. This is expected since the adversary leverages less information from the target prompt.

**Analysis.** Figure 6 shows that the performance of three attacks varies on different pre-trained models and different datasets. We hypothesize that the different overfitting levels may affect the attack performance. Following previous work [17, 43], we calculate the difference between train accuracy and test accuracy to measure the overfitting level of a given target prompt. We train five target prompts with different random seeds for each experimental setting. The relationship between overfitting levels and attack performance is illustrated in Figure 7. We observe that different pre-trained models and different datasets have different overfitting levels. Meanwhile, our results demonstrate that overfitting does have a significant effect on membership inference attacks. The overall trend is that the higher the overfitting level, the better the attack performance. To quantify this correlation, we calculate the Pearson correlation scores between the overfitting level and attack performance. The result is 0.89. Our finding is in line with previous analyses [33, 45].

**Takeaways.** We show that prompts can leak sensitive membership information of their training dataset. Similar to previous analyses, overfitting is strongly correlated with the membership inference performance.

## 4.5 Factors Affecting Membership Inference From the Victim's Side

In this section, we measure the factors that may affect the membership inference privacy risks from the perspective of the victim. As shown in Figure 8, the number of epochs used to train the target prompt and its training dataset size ($\mathcal{D}_{target}^{train}$) are closely related to the overfitting level of the target prompt. A larger number of epochs results in larger overfitting levels. On the contrary, a larger training dataset size results in reduced overfitting levels. Therefore, we explore how these two factors affect the attack performance.

**Number of Epochs.** We set the number of epochs for the target prompt to {200, 400, 600, 800, 1000} and use the same number of epochs for the shadow prompt in each experiment. The results are shown in Figure 9. We observe that, in general, more epochs lead to better attack performance, hence greater membership inference privacy risks. The attack performance becomes steady after 500 epochs, while the overfitting level also becomes stable simultaneously in Figure 8a.

**Prompt Training Dataset Size.** We investigate the effect of the training dataset size on the attack performance by varying

the size from 500 to 5000. To control the variables, we always fix the other three sets ($\mathcal{D}^{test}_{target}$, $\mathcal{D}^{train}_{shadow}$, and $\mathcal{D}^{test}_{shadow}$) to the same size as $\mathcal{D}^{train}_{target}$. As illustrated in Figure 10, the attack performance decreases as the dataset size grows. The general trend of the attack performance is also consistent with the findings in Figure 8b. That is, more training data reduces the overfitting level in most cases, leading to a decrease in the attack performance. There is a significant drop in the overfitting level on CIFAR10 when increasing the size of $\mathcal{D}^{train}_{target}$ from 2000 to 5000. Therefore, the test accuracy of metric-based attacks drops from 86.60% to 64.00%.

**Takeaways.** We perform an analysis of the relation between overfitting levels and attack performance. Our results show that more epochs and fewer training data can aggravate overfitting and pose a more severe threat to membership privacy.

## 4.6 Factors Affecting Membership Inference From the Adversary's Side

We evaluate the factors that may affect the membership inference privacy risks from the perspective of the adversary. In previous experiments, we made two assumptions: 1) the adversary has a dataset $\mathcal{D}_{shadow}$ that comes from the same distribution as $\mathcal{D}_{target}$, and 2) the PaaS provider offers users the target prompt with white-box access to the pre-trained model. Here, we evaluate if these two assumptions are needed to mount a successful membership inference attack.

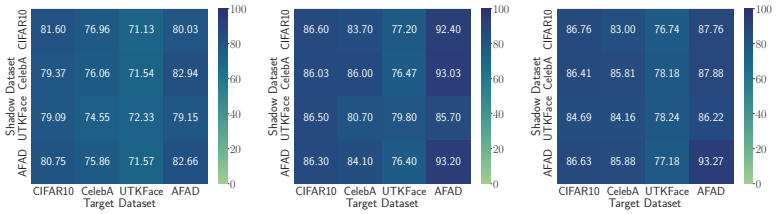
**Dataset Assumption.** We relax the same distribution assumption by leveraging a shadow dataset that comes from a different distribution than $\mathcal{D}_{target}$ to train the shadow prompt; the results with three attack methodologies are shown in Figure 11. In the diagonal of the heatmaps, we show the results of the adversary having access to $\mathcal{D}_{shadow}$ that comes from the same distribution as $\mathcal{D}_{target}$. We observe that the performance of NN-based, metric-based, and gradient-based attacks is slightly reduced but remains effective. For instance, as shown in Figure 11b, using any one of the four datasets as the shadow dataset to launch the metric-based attack can achieve a test accuracy of around 86.00%, when the target dataset is CIFAR10. Interestingly, CIFAR10 contains images of 10 classes such as cars and trucks, but the other three datasets only include facial images. This supports the findings of Salem et al. [44] and Li et al. [27], which also report the effectiveness of membership inference using shadow datasets from different domains.

Moreover, we present the average test accuracy and the average drop in accuracy of three attacks on different pre-trained models in Appendix D. The results show that the metric-based and gradient-based attacks achieve the best attack performance on average, while the NN-based and gradient-based attacks, in general, are more robust than metric-based attacks. Hence, we conclude that the gradient-based attacks exhibit superior performance in terms of both utility and robustness after relaxing the dataset assumption. However, it should be noted that the gradient-based attacks come at the cost of high computational resources and a significant amount of information needed. Overall, our findings suggest that we can relax the assumption of the same-distribution shadow dataset, implying greater membership inference privacy risks of prompts.

**Pre-trained Model Assumption.** In the previous evaluation, we assume the adversary has white-box access to the pre-trained model. However, the PaaS provider may only allow users to submit prompted images and receive corresponding results, thus limiting access to the pre-trained model. The adversary has to develop their pre-trained models, which may be different from the pre-trained models used to train the target prompts (abbreviated as the target model). We, therefore, measure the impact of the discrepancy in architecture between the pre-trained model used to train the shadow prompts (abbreviated as shadow model) and the target model on the attack performance. The results of three attacks are shown in Figure 12.

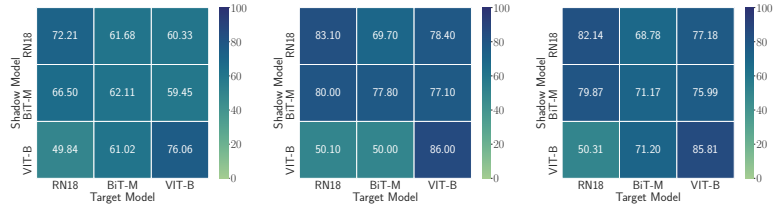
In the diagonal of the heatmaps, we show the results of the adversary having white-box access to the same pre-trained model used to train the target prompt. We observe that, in some cases, the attack performance decreases noticeably but remains effective. For example, when the pre-trained model of the target prompt is ViT-B on CelebA, the performance of metric-based attacks drops from 86.00% to 78.40% (77.10%) when using RN18 (BiT-M) as the pre-trained model for the shadow prompt. However, in certain cases, all three attacks fail completely, i.e., they become random guesses. For instance, when the adversary uses ViT-B to attack the target prompt trained on RN18, these three methodologies become random guesses. We also present the average test accuracy and the average drop in accuracy of three attacks on different datasets in Appendix D. The gradient-based and metric-based attacks achieve the best attack performance, and the gradient-based attacks are more robust than the metric-based attacks. However, the average drop in accuracy of all attacks after relaxing the pre-trained model assumption, in general, is larger than that of relaxing the dataset assumption.

**Discussion.** We have shown that all methodologies only have slight performance degradation after relaxing the dataset assumption. Meanwhile, after relaxing the pre-trained model assumption, these attack methodologies are effective in some cases but fail to maintain high robustness, i.e., they fail in other cases. Previous work [27, 44] on membership inference against traditional ML classifiers has shown that having shadow models with different architectures than the target models does not have a strong impact on the attack performance. However, we do not observe the same in VPL. One possible explanation is that a prompt is specific to the machine learning model it is trained on. In other words, prompts from different models share less similarity, which makes the membership inference knowledge hard to transfer among them. As illustrated in Figure 13, we find that metric-corr attacks have no performance deterioration after relaxing these assumptions, as they do not rely on the shadow technique. Thus, the




(a) NN-based (b) Metric-based (c) Gradient-based

Figure 11: Attack performance of three attacks after relaxing the dataset assumption, using ViT-B as the pre-trained model.




(a) NN-based (b) Metric-based (c) Gradient-based

Figure 12: Attack performance of three attacks after relaxing the pre-trained model assumption on CelebA.

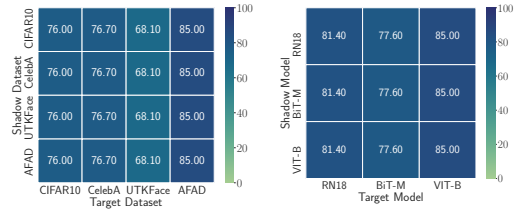



Figure 13: Attack performance of metric-corr attacks after relaxing (a) dataset assumption using ViT-B as the pre-trained model and (b) pre-trained model assumption on AFAD.

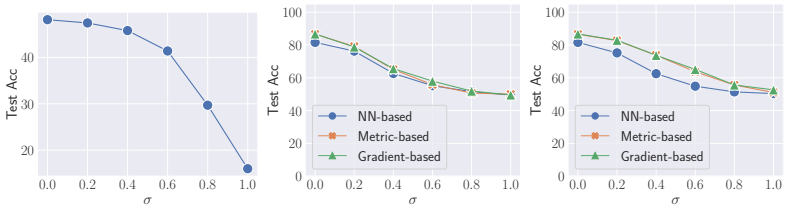
adversary can leverage the metric-corr attacks when relaxing the data assumption and the pre-trained model assumption.

**Takeaways.** Our results show that the adversary can be data-free, as the attack performance only has a slight deterioration and remains effective. The results also indicate that the adversary has some dependency on the knowledge of pre-trained models to steal private information, as not all attack methodologies can be successfully launched after relaxing the pre-trained model assumption. However, we show that the adversary can still leverage the metric-corr attacks to obtain decent attack performance with high robustness, as they do not rely on the shadow technique.

## 4.7 Defense

**Gaussian Noise as Defense.** We have demonstrated that the prompts are also vulnerable to membership inference attacks. Meanwhile, in the above experiments, we observe that the performance of membership inference is heavily related to the overfitting level of the target prompt. Potentially, a defender can decrease the threat to membership privacy by reducing the overfitting level. As shown in Figure 9 and Figure 10, leveraging fewer epochs and more data to train the target prompt can decrease the attack performance to some extent. However, using these methods comes at the cost of either the utility of the target prompt or the resource to collect and process data. We also apply the widely adopted Differential Privacy-Stochastic Gradient Descent (DP-SGD) [2], which involves adding noise to clipped gradients, as a defense mechanism. However, the experimental results show that it is hard to maintain the prompt utility even with a larger privacy budget, e.g., $\varepsilon = 20$. We hypothesize that DP-SGD may work on large datasets, but not on the data for prompt learning since it is relatively small. Hence, we investigate if the defense mechanism used in Section 3.6, i.e., adding Gaussian noise to the prompts, can reduce the risks of membership leakage. We set $\sigma \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. We first report the target performance on CIFAR10 in Figure 14a. The evaluation metric is the test accuracy of the target prompt. We observe that the target performance, i.e., prompt utility, only decreases heavily when $\sigma \geq 0.6$. For example, the prompt utility remains above 41.40% when $\sigma \leq 0.6$ and then decreases heavily from 41.40% to 29.70% on CIFAR10 when increasing $\sigma$ from 0.6 to 0.8. We then present the attack performance where the adversary is unaware of the defense mechanism in Figure 14b. We can observe that all attacks are close to random guesses when $\sigma \geq 0.6$, showing that there is a practical utility-defense trade-off when $\sigma = 0.6$.

**Adaptive Attacks.** We further consider an adaptive adversary [21] who is aware of the defense mechanism. Hence,




(a) Prompt Utility (b) Naive Attacks (c) Adaptive Attacks

Figure 14: Prompt utility and attack performance using the proposed defense on CIFAR10.

the adversary can craft their attack training datasets using the shadow prompt with Gaussian noise. We set $\sigma \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ for both shadow and target prompts. We report the performance of adaptive attacks on CIFAR10 in Figure 14c. The results show that the attack performance is still close to random guess when $\sigma \geq 0.6$.

**Takeaways.** When defending against membership inference attacks, the proposed defense mechanism can achieve a decent utility-defense trade-off when setting $\sigma = 0.6$. A similar conclusion can be drawn from the other three datasets.

## 5 Related Work

**Property Inference Attacks.** Property inference [3, 7, 14, 36, 53, 55] aims to extract sensitive global properties of the training data distribution from an ML model that the model owner does not want to share. It is an important privacy attack against ML models, as it can violate prompt owner's privacy, i.e., proprietary information about the dataset, and enable attackers to perform tailored attacks, e.g., enhancing membership inference attacks [55]. The main approach for launching these attacks is building a meta-classifier on a large number of shadow models [3]. Existing work focuses on deep neural networks, including fully connected neural networks [14], generative adversarial networks (GANs) [55], and graph neural networks (GNNs) [53].

**Membership Inference Attacks.** Membership inference [24, 28, 29, 38, 44, 46, 47] is another important type of privacy attack against ML models, where the adversary aims to infer whether the given data sample was involved in a target model's training dataset. Shokri et al. [46] propose the first membership inference attack which depends on training multiple shadow models for developing their attack models. Salem et al. [44] then relax assumptions proposed by Shokri et al. [46]. Yeom et al. [50] attribute the vulnerability of membership inference to the overfitting of ML models. Song and Mittal [47] propose metric-based attacks that rely on pre-calculated thresholds over shadow models to determine the membership status. Nasr et al. [38] perform a thorough investigation of membership privacy in both black-box and white-box settings for both centralized and federated learning scenarios. More recently, Liu et al. [32] leverage the loss trajectory to further enhance the attack performance. Most recent work focuses on deep neural networks, including GNNs [16, 48], multi-modal models [19], and multi-exit networks [27].

Previous work has demonstrated that the fine-tuning paradigm is vulnerable to these privacy attacks [10, 37]. The privacy risk in the fine-tuning paradigm resides at the model level, as the privacy information is leaked through fine-tuned models. This differs from the privacy risk associated with the prompt learning paradigm, where the risk lies at the input level, as the prompt exists in the pixel space.

## 6 Limitation and Future Work

**Efficacy of VPL.** VPL is an emerging ML paradigm. Although its current performance cannot rival that of a fine-tuned model, an increasing number of studies are attempting to enhance its performance through various approaches, e.g., label mapping [9] and better data homogeneity [20]. Since we are the first to explore the vulnerabilities of the visual prompt, we have focused on the widely recognized VPL paradigm and followed their default training settings [4]. We anticipate that as VPL with enhanced performance are introduced in the future, it will be straightforward for us to extend our measurement, and thus we recognize this as a promising avenue for future research.

**Defense.** In the evaluation, we show that adding Gaussian noise to the prompt can mitigate the membership inference attacks with a decent utility-defense trade-off but fails to defend against property inference attacks. DP-SGD fails to preserve the original prompt utility. Since the privacy risk in the prompt learning paradigm is at the input level, devising diverse defense mechanisms for it is more challenging compared to addressing the privacy risk at the model level. We leave it as a future work to explore effective defenses against property inference attacks.

**NLP Prompt Learning.** Another interesting future work is to apply the two proposed privacy attacks along with their motivation to prompt learning in the NLP domain [25, 26], as the NLP prompt is essentially a (soft) token that can be added to the text input, operating at the input level.

## 7 Conclusion

In this paper, we conduct the first privacy assessment of prompts learned by VPL through the lens of property inference attacks and membership inference attacks. Our empirical evaluation shows that prompts are vulnerable to both of these attacks. Moreover, we have discovered that an adversary can successfully mount the property inference attacks by training only a few shadow prompts. They can also relax the dataset assumption to achieve effective membership inference attacks. We further make some initial investigations on possible defenses. Experiments show that our method, i.e., adding Gaussian noise to prompts, can mitigate the membership inference attacks with a decent utility-defense trade-off but fails to defend against property inference attacks. We hope our results can raise the awareness of the stakeholders when deploying prompt learning in real-world applications. Moreover, we will share our code and models to facilitate research in this field.

**Acknowledgements.** We thank all anonymous reviewers for their constructive comments. This work is partially funded by the European Health and Digital Executive Agency (HADEA) within the project "Understanding the individual host response against Hepatitis D Virus to develop a personalized approach for the management of hepatitis D" (D-Solve) (grant agreement number 101057917).

## References

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

[2] Martin Abadi, Andy Chu, Ian Goodfellow, Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 308–318. ACM, 2016.

[3] 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.

[4] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring Visual Prompts for Adapting Large-scale Models. *CoRR abs/2203.17274*, 2022.

[5] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei A. Efros. Visual Prompting via Image Inpainting. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 25005–25017. NeurIPS, 2022.

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

[7] Harsh Chaudhari, John Abascal, Alina Oprea, Matthew Jagielski, Florian Tramèr, and Jonathan R. Ullman. SNAP: Efficient Extraction of Private Properties with Poisoning. In *IEEE Symposium on Security and Privacy (S&P)*, pages 1935–1952. IEEE, 2023.

[8] Aochuan Chen, Peter Lorenz, Yuguang Yao, Pin-Yu Chen, and Sijia Liu. Visual Prompting for Adversarial Robustness. In *NeurIPS Workshop on Trustworthy and Socially Responsible Machine Learning (TSRML)*. NeurIPS, 2022.

[9] Aochuan Chen, Yuguang Yao, Pin-Yu Chen, Yihua Zhang, and Sijia Liu. Understanding and Improving Visual Prompting: A Label-Mapping Perspective. *CoRR abs/2211.11635*, 2022.

[10] Christopher A. Choquette Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. Label-Only Membership Inference Attacks. In *International Conference on Machine Learning (ICML)*, pages 1964–1974. PMLR, 2021.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186. ACL, 2019.

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

[13] Gamaleldin F. Elsayed, Ian J. Goodfellow, and Jascha Sohl-Dickstein. Adversarial Reprogramming of Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[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 *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 619–633. ACM, 2018.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.

[16] Xinlei He, Rui Wen, Yixin Wu, Michael Backes, Yun Shen, and Yang Zhang. Node-Level Membership Inference Attacks Against Graph Neural Networks. *CoRR abs/2102.05429*, 2021.

[17] Xinlei He and Yang Zhang. Quantifying and Mitigating Privacy Risks of Contrastive Learning. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 845–863. ACM, 2021.

[18] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric Noise Injection: Trainable Randomness to Improve Deep Neural Network Robustness Against Adversarial Attack. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 588–597. IEEE, 2019.

[19] Pingyi Hu, Zihan Wang, Ruoxi Sun, Hu Wang, and Minhui Xue. $M^4I$: Multi-modal Models Membership Inference. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2022.

[20] Qidong Huang, Xiaoyi Dong, Dongdong Chen, Weiming Zhang, Feifei Wang, Gang Hua, and Nenghai Yu. Diversity-Aware Meta Visual Prompting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023.

[21] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 259–274. ACM, 2019.

[22] Minsu Kim, Hyung-Il Kim, and Yong Man Ro. Prompt Tuning of Deep Neural Networks for Speaker-adaptive Visual Speech Recognition. *CoRR abs/2302.08102*, 2023.

[23] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big Transfer (BiT): General Visual Representation Learning. In *European Conference on Computer Vision (ECCV)*, pages 491–507. Springer, 2020.

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

[25] Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3045–3059. ACL, 2021.

[26] Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL/IJCNLP)*, pages 4582–4597. ACL, 2021.

[27] Zheng Li, Yiyong Liu, Xinlei He, Ning Yu, Michael Backes, and Yang Zhang. Auditing Membership Leakages of Multi-Exit Networks. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1917–1931. ACM, 2022.

[28] Zheng Li and Yang Zhang. Membership Leakage in Label-Only Exposures. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 880–895. ACM, 2021.

[29] Hongbin Liu, Jinyuan Jia, Wenjie Qu, and Neil Zhenqiang Gong. EncoderMI: Membership Inference against Pre-trained Encoders in Contrastive Learning. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2021.

[30] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Computing Surveys*, 2023.

[31] Weihuang Liu, Xi Shen, Chi-Man Pun, and Xiaodong Cun. Explicit Visual Prompting for Low-Level Structure Segmentations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023.

[32] Yiyong Liu, Zhengyu Zhao, Michael Backes, and Yang Zhang. Membership Inference Attacks by Exploiting Loss Trajectory. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 2085–2098. ACM, 2022.

[33] Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario

Fritz, and Yang Zhang. ML-Doctor: Holistic Risk Assessment of Inference Attacks Against Machine Learning Models. In *USENIX Security Symposium (USENIX Security)*, pages 4525–4542. USENIX, 2022.

[34] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738. IEEE, 2015.

[35] Yajuan Lü, Jin Huang, and Qun Liu. Improving Statistical Machine Translation Performance by Training Data Selection and Optimization. In *Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 343–350. ACL, 2007.

[36] Saeed Mahloujifar, Esha Ghosh, and Melissa Chase. Property Inference from Poisoning. In *IEEE Symposium on Security and Privacy (S&P)*, pages 1120–1137. IEEE, 2022.

[37] Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David Evans, and Taylor Berg-Kirkpatrick. An Empirical Analysis of Memorization in Fine-tuned Autoregressive Language Models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1816–1826. ACL, 2022.

[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 *IEEE Symposium on Security and Privacy (S&P)*, pages 1021–1035. IEEE, 2019.

[39] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal Regression with Multiple Output CNN for Age Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4920–4928. IEEE, 2016.

[40] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 2020.

[41] Laria Reynolds and Kyle McDonell. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. In *Annual ACM Conference on Human Factors in Computing Systems (CHI)*, pages 314:1–314:7. ACM, 2021.

[42] Mauro Ribeiro, Katarina Grolinger, and Miriam A. M. Capretz. MLaaS: Machine Learning as a Service. In *International Conference on Machine Learning and Applications (ICMLA)*, pages 896–902. IEEE, 2015.

[43] Itay Safran and Ohad Shamir. Depth-Width Tradeoffs in Approximating Natural Functions with Neural Networks. In *International Conference on Machine Learning (ICML)*, pages 2979–2987. PMLR, 2017.

[44] 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 *Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2019.

[45] Virat Shejwalkar and Amir Houmansadr. Membership Privacy for Machine Learning Models Through Knowledge Transfer. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 9549–9557. AAAI, 2021.

[46] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *IEEE Symposium on Security and Privacy (S&P)*, pages 3–18. IEEE, 2017.

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

[48] Bang Wu, Xiangwen Yang, Shirui Pan, and Xingliang Yuan. Adapting Membership Inference Attacks to GNN for Graph Classification: Approaches and Implications. In *International Conference on Data Mining (ICDM)*. IEEE, 2021.

[49] Yinghui Xing, Qirui Wu, De Cheng, Shizhou Zhang, Guoqiang Liang, Peng Wang, and Yanning Zhang. Dual Modality Prompt Tuning for Vision-Language Pre-Trained Model. *CoRR abs/2208.08340*, 2023.

[50] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In *IEEE Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

[51] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Unified Vision and Language Prompt Learning. *CoRR abs/2210.07225*, 2022.

[52] Zhifei Zhang, Yang Song, and Hairong Qi. Age Progression/Regression by Conditional Adversarial Autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4352–4360. IEEE, 2017.

[53] Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. Inference Attacks Against Graph Neural Networks. In *USENIX Security Symposium (USENIX Security)*, pages 4543–4560. USENIX, 2022.

[54] Yuan Zhao, Bo Liu, Ming Ding, Baoping Liu, Tianqing Zhu, and Xin Yu. Proactive Deepfake Defence via Identity Watermarking. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 4602–4611. IEEE, 2023.

[55] Junhao Zhou, Yufei Chen, Chao Shen, and Yang Zhang. Property Inference Attacks Against GANs. In *Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2022.

[56] Jiawen Zhu, Simiao Lai, Xin Chen, Dong Wang, and Huchuan Lu. Visual Prompt Multi-Modal Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023.

[57] Ligeng Zhu, Zhijian Liu, and Song Han. Deep Leakage from Gradients. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 14747–14756. NeurIPS, 2019.

[58] Xiangxin Zhu, Carl Vondrick, Charless C. Fowlkes, and Deva Ramanan. Do We Need More Training Data? *International Journal of Computer Vision*, 2016.

[59] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A Comprehensive Survey on Transfer Learning. *CoRR abs/1911.02685*, 2019.

# Appendix

## A Details of Pre-trained Models

Table 2 shows details about the architectures and pre-trained datasets of the pre-trained models used in the paper.

Table 2: Overview of the pre-trained models.

| Model | Architecture | Pre-trained Dataset | # Parameters |
|---|---|---|---|
| RN18 | ResNet-18 | 1.2M ImageNet-1k | 11,173,962 |
| BiT-M | ResNet-50 | 14M ImageNet-21k | 23,520,842 |
| ViT-B | ViT-B/16 | 14M ImageNet-21k | 86,567,656 |

## B Relax Assumptions of Property Inference Attacks

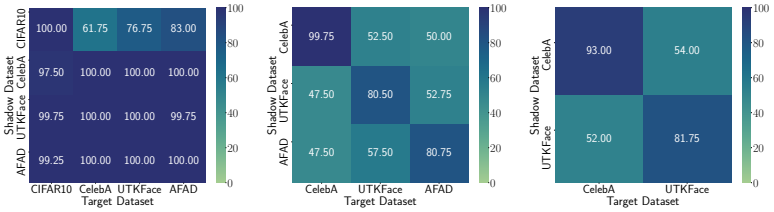
We first relax the assumption that the adversary has a shadow dataset of similar distribution as the target dataset, i.e., the dataset assumption. As shown in Figure 15, we observe that the performance of the property inference attack is close to a random guess, considering the proportion of males and youth as the target properties. When inferring the prompt training dataset size, the property inference attacks maintain high accuracy in most cases. However, the attack performance decreases significantly when leveraging CIFAR10 as the shadow dataset. We reason this is because the CIFAR10 datasets contain objections such as cars and birds, whereas other datasets only contain facial images. We then relax the assumption that the pre-trained model used to train the shadow prompts and target prompts are the same, i.e., the pre-trained model assumption. As illustrated in Figure 16, we notice that the attack performance has a significant degradation when inferring the proportion of males and youth. When inferring the prompt training dataset size, the property inference attacks are effective in some cases. However, they are not robust, as they become random guesses in certain cases. For example, when the target prompt is trained on ViT-B and the shadow prompt is trained on RN18, the property inference attacks fail. Thus, we conclude that it is necessary to leverage a shadow dataset of similar distribution as the target dataset and the same pre-trained model to train the shadow prompt.

## C Performance of Metric-based Attacks With Different Metrics

As shown in Figure 17, metric-conf and metric-ment attacks achieve the best performance in all cases. The reason why they work better than the other two metrics is that they take both prediction correctness and confidence into account, while the other two metrics only consider prediction correctness.

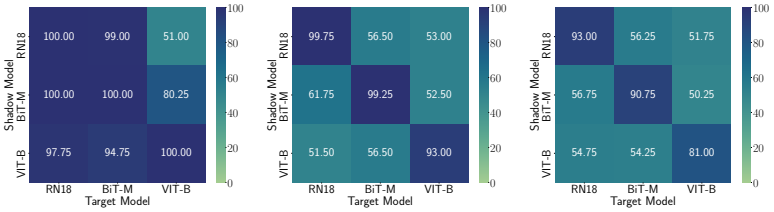
## D Average Test Accuracy and Drop in Accuracy

We present the average test accuracy and drop in accuracy of three attacks on different pre-trained models in Figure 18. We calculate these values based on the heatmap in Figure 11. Specifically, for each heatmap, we take the average of all values as the average test accuracy for each attack methodology on a pre-trained model. We calculate the difference between each cell value and the diagonal value in the corresponding column and take the average as the average drop in accuracy. Basically, a lower drop accuracy value means a more robust attack when relaxing the dataset assumption. As illustrated in Figure 18a, metric-based and gradient-based attacks achieve the best attack performance on average. Meanwhile, as shown in Figure 18b, we observe that the average drop in accuracy is smaller than 5.00% in most cases. The NN-based and gradient-based attacks, in general, are more robust than metric-based attacks. We also present the average test accuracy and drop in accuracy of three attacks on different datasets in Figure 19. The gradient-based and metric-based attacks achieve the best attack performance, and the gradient-based attacks are more robust than the metric-based attacks.



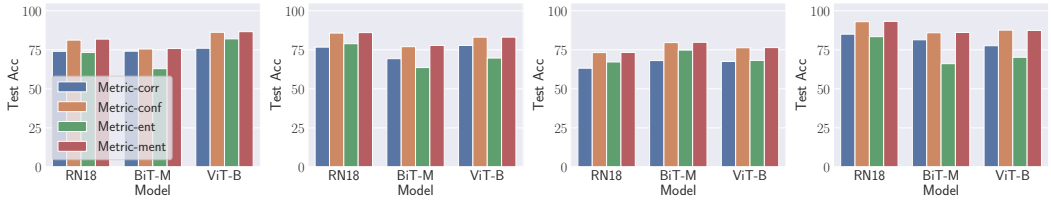

(a) Size (b) Proportion of Males (c) Proportion of Youth

Figure 15: Attack performance of property inference attacks after relaxing the dataset assumption, using RN18 as the pre-trained model.



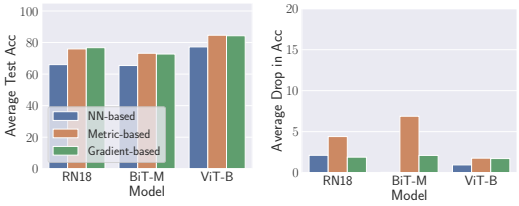

(a) Size (b) Proportion of Males (c) Proportion of Youth

Figure 16: Attack performance of property inference attacks after relaxing the pre-trained model assumption on CelebA.



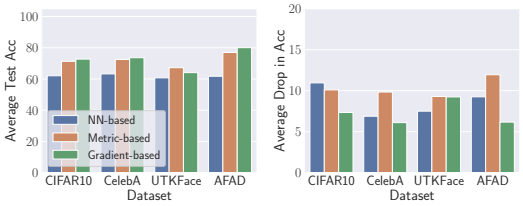

(a) CIFAR10 (b) CelebA (c) UTKFace (d) AFAD

Figure 17: Attack performance of four metric-based attacks on four datasets.




(a) Test Accuracy (b) Drop in Accuracy

Figure 18: Average test accuracy and drop in accuracy of three attacks after relaxing the dataset assumption.




(a) Test Accuracy (b) Drop in Accuracy

Figure 19: Average test accuracy and drop in accuracy of three attacks after relaxing the pre-trained model assumption.