Property Existence Inference against Generative Models

Lijin Wang, Jingjing Wang, Jie Wan, and Lin Long, *Zhejiang University;*
Ziqi Yang and Zhan Qin, *Zhejiang University, ZJU-Hangzhou Global Scientific and Technological Innovation Center*

https://www.usenix.org/conference/usenixsecurity24/presentation/wang-lijin

This paper is included in the Proceedings of the 33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

Open access to the Proceedings of the 33rd USENIX Security Symposium is sponsored by USENIX.

Property Existence Inference against Generative Models

Lijin Wang[1], Jingjing Wang[1], Jie Wan[1], Lin Long[1], Ziqi Yang[1,2*], and Zhan Qin[1,2]

[1] *Zhejiang University*
[2] *ZJU-Hangzhou Global Scientific and Technological Innovation Center*
{wanglijin, wjjxjj, wanjie, llong, yangziqi, qinzhan}@zju.edu.cn

Abstract

Generative models have served as the backbone of versatile tools with a wide range of applications across various fields in recent years. However, it has been demonstrated that privacy concerns, such as membership information leakage of the training dataset, exist for generative models. In this paper, we perform *property existence inference* against generative models as a new type of information leakage, which aims to infer whether any samples with a given property are contained in the training set. For example, to infer if any images (i.e., samples) of a specific brand of cars (i.e., property) are used to train the target model. We focus on the leakage of existence information of properties with very low proportions in the training set, which has been overlooked in previous works. We leverage the feature-level consistency of the generated data with the training data to launch inferences and validate the property existence information leakage across diverse architectures of generative models. We have examined various factors influencing the property existence inference and investigated how generated samples leak property existence information. In our conclusion, most generative models are vulnerable to property existence inferences. Additionally, we have validated our attack in Stable Diffusion which is a large-scale open-source generative model in real-world scenarios, and demonstrated its risk of property existence information leakage. The source code is available at https://github.com/wljLlla/PEI_Code.

1 Introduction

Remarkable progress has been made in generative models with numerous new models outperforming previous models in terms of fidelity and scalability. The development of generative models has driven their use in commercial applications. Generative models like Stable Diffusion [36] and Imagen [37] provide interfaces for receiving user-input captions (descriptions of the desired generated data), allowing users to generate data with a payment. Many previous studies [3,5,17,41,44,58] have indicated that generative models can leak sensitive private information of their training sets. Existing information leakages of the training sets include the membership information leakage in GANs [5, 14, 17] and Diffusion Models [58], the property information leakage in GANs [57], and the duplication phenomena between the generated images and the training images of Diffusion Models and GANs [3,44]. These leakages can be classified into two types based on the object of the information: leakage of the global information of the training set and the leakage of the information of specific samples in the training set.

Many inference attacks emerge to obtain sensitive information from the training set. Among these, the most popular inference attacks against generative models are the property inference and the membership inference. For the leakage of global information, the property inference is proposed to infer the proportion of global properties of the training set, such as the proportion of a gender in a facial generative model. For the leakage of the information of specific samples, the membership inference is proposed to infer whether a specific sample exists in the training set, such as the existence of a specific facial image in a facial generative model.

In this paper, we focus on an inference attack against generative models called property existence inference, which aims to infer whether any samples with the target property are used to train the target model. Compared to property inference, property existence inference focuses on the leakage of more personal and personalized privacy. For example, while property inference will choose property such as gender in the inference of facial generative models, property existence inference focuses on more personally sensitive properties, such as individual identities. Compared to membership inference, property existence inference is evaluated under a more practical setting where no samples in the training set can be obtained by the adversary. The property existence inference carries potential privacy risks. With the property existence

* Corresponding author

inference, personal and personalized information can be inferred from the generative model that trainers may not want to share, such as whether specific individuals' images have been used to train the target model. Property existence inference can also be used to determine the presence of unauthorized data in the training set. For instance, if a vehicle manufacturer does not want images of their cars to be used for generating similar-model images, they can perform property existence inference to safeguard copyright.

We perform the property existence inference under the most practical black-box setting and design the method with the motivation that generative models will generate data with features related to the properties existing in the training set. For example, an image generated by a facial generative model may not depict the face of a specific person from the training set, but it may be similar in features like eyes to individuals in the training set. Therefore, we perform property existence inference through similarity comparisons between the generated images and the images with the target property. Specifically, we do the similarity comparison in the embedding space based on a well-trained property extractor which learns to distinguish features from different properties. During this stage, we mitigate data-induced uncertainties in similarity scores. We finally train shadow models to simulate the behavior of the target model and obtain the similarity score distributions of properties existing and not existing in the training set, to select a threshold for making the final decision.

We conduct comprehensive evaluations and find that generative models used in the evaluations including Stable Diffusion [36], the largest open-source diffusion model, exhibit the potential risk of property existence information leakage. Our method effectively determines whether the target properties exist in the training set. For example, we obtain AUCs for determining the existence of the target properties above 0.81 and 0.95 in generative models trained on ImageNet and CompCars respectively. Based on our evaluations of state-of-the-art generative models, it can be concluded that most of the generative models we evaluated suffer from property existence information leakage. To gain a deeper understanding of property existence inference, we further investigate the elements that impact the effectiveness of the property existence inference such as the size of the training sets, the adversary knowledge and the granularity of properties. Our evaluations further reveal that there is an increased risk of property existence information leakage for a generated image that closely resembles a larger number of samples that share the same properties as those in the training set.

Contributions. We summarize our contributions in the following.

- To the best of our knowledge, we are the first to perform the property existence inference against generative models and emphasize that property existence inference should focus on properties with very low proportions.
- We propose a method to perform the property existence inference by exploiting the differences in similarities between the generated samples and samples with the target property, based on whether or not the target property is present in the training set.
- We have conducted comprehensive evaluations on the state-of-the-art generative models including large-scale models like Stable Diffusion to study the effectiveness of our method and explored which generated samples are most likely to be utilized to extract information.

Roadmap. In Section 2, we introduce the generative models and the property inferences. In Section 3, we formally define the property existence inference and clarify the threat model. In Section 4, we detail the method to perform the property existence inferences. In Section 5, we analyze the effectiveness of property existence inference under different generative models and the factors that affect the attack effectiveness. In Section 6, we discuss the related work. In Section 7, we discuss the cost and effectiveness of our method. In Section 8, we conclude our work.

2 Background

2.1 Generative Models

We focus on three types of generative models: generative adversarial networks (GANs) [12], Variational Auto-Encoder (VAEs) [24] and Diffusion Models [8, 18, 34–37, 47].

GANs. GAN [12] consists of a generator G and a discriminator D. The generator is trained to learn the underlying distribution of the real data p_{data} by minimizing the divergence between the distribution p_{data} and the distribution of the generated data p_g, while the discriminator is trained to distinguish the two distributions p_{data} and p_g. The generator and the discriminator are trained simultaneously in an adversarial manner using the objective function below:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

where z is the input of the generator sampled from $p_z(z)$.

VAEs. As another broadly used generative model, VAE [24] maximizes the lower bound of the log-likelihood $\log p(x)$ of all observed real data x. The lower bound is quantified as the evidence lower bound (ELBO) and can be derived as the left term of the following equation:

$$\mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{q_\phi(z\mid x)}\right] = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x\mid z)\right] - D_{\text{KL}}\left(q_\phi(z\mid x)\|p(z)\right) \quad (1)$$

where $q_\phi(z\mid x)$ is the approximated posterior parameterized by ϕ, and $\log p_\theta(x\mid z)$ denotes the deterministic func-

tion parameterized by θ that converts a given latent variables z into an observation x. As Equation 1 shows, ELBO can be derived as the difference between the reconstruction term $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)]$ and the prior KL divergence term $D_{\mathrm{KL}}\left(q_\phi(z \mid x) \| p(z)\right)$.

VAE uses an encoder parameterized by ϕ to approximate the posterior distribution and a decoder parameterized by θ to generate the data from latent variables z. Therefore, VAE maximizes ELBO by optimizing parameters ϕ and θ.

Diffusion Models. The training of diffusion models involves two Markovian procedures: the forward diffusion process and the backward denoising process. In the forward diffusion process, Gaussian noise is added to the input data x_0 in T steps until x_0 is transformed into a standard Gaussian noise x_T. Transitions $q(x_t|x_{t-1})$ of each step $t \in [1,T]$ can be seen as: $x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\varepsilon$. where α_t is used to control the diffusion velocity and $\varepsilon \sim \mathcal{N}(0,I)$. In the backward denoising process, the original data x_0 is obtained by sequentially removing the added noise from x_T. Transitions of removing steps is denoted as $p(x_{t-1} \mid x_t), t \in [1,T]$. $p_\theta(x_{t-1} \mid x_t)$ parameterized by θ is learned to approximate $p(x_{t-1} \mid x_t)$.

Based on these two Markovian procedures, diffusion models maximize the lower bound of the log-likelihood $\log p(x)$ which can be derived from the following inequality:

$$\begin{aligned}\log p(x) \geq & \mathbb{E}_{q(x_1|x_0)}[\log p_\theta(x_0 \mid x_1)] - D_{\mathrm{KL}}(q(x_T \mid x_0) \| p(x_T)) \\ & - \sum_{t=2}^{T} \mathbb{E}_{q(x_t|x_0)}[D_{\mathrm{KL}}(q(x_{t-1} \mid x_t, x_0) \| p_\theta(x_{t-1} \mid x_t))]\end{aligned} \tag{2}$$

The first term in the right equation $\mathbb{E}_{q(x_1|x_0)}[\log p_\theta(x_0 \mid x_1)]$ can be regarded as a reconstruction term. The second term $D_{\mathrm{KL}}(q(x_T \mid x_0) \| p(x_T))$ which equals to zero in assumption. The third term can be considered as the mean difference between the cumulative equivalent of noise at each step and the predicted noise [18]. During the generation phase, the diffusion model generates data by gradually removing the predicted noise from x_T.

2.2 Property Inference Attacks

Property inference attacks aim to extract the overall (global) information about the training set from the target model. Property inference attacks against machine learning models are first introduced by [1] and formally defined by [48]: given two hypotheses $(\mathcal{H}_0, \mathcal{H}_1)$ about the distribution of the training set, the adversary is asked to determine which hypothesis is more consistent with the distribution of the training set of the target model. The hypotheses are widely adopted in previous studies as the description of the proportion of the samples with target property $\mathbb{P}$ as follows:

- $\mathcal{H}_0$: The proportion of samples with target property $\mathbb{P}$ in the training set of the target model is t_0.
- $\mathcal{H}_1$: The proportion of samples with target property $\mathbb{P}$ in the training set of the target model is t_1.

Previous studies commonly use a binary meta-classifier trained under numerous shadow models of $\mathcal{H}_0$ and $\mathcal{H}_1$ to make the final decision [1, 11, 31, 56]. [4, 31] assume the adversary could poison the training set to perform property inference attacks. [57] first introduces the property inference attack into the GANs. Different from typical property inference attacks, it aims to directly infer the proportion of the samples with target property. In this paper, we focus on the property existence inference which can be regarded as an extension of the property inference to determine whether any samples with target property are used to train the target model.

3 Problem Statement and Threat Model

In this section, we present the formulation of property existence inference, the significance of investigating this problem, and the clarity of our threat model.

3.1 Property Existence Inference

We follow the points of [4] and formally define the property existence inference as follows:

Definition 1 (Property Existence Inference)**.** *given a target model and two hypotheses $(\mathcal{H}_0, \mathcal{H}_1)$ about the distribution of the training set, the adversary is asked to determine which hypothesis is true. Each pair of hypotheses is defined as:*

- *$\mathcal{H}_0$: The proportion of samples with target property $\mathbb{P}$ in the training set of the target model is 0.*
- *$\mathcal{H}_1$: The proportion of samples with target property $\mathbb{P}$ is larger than 0.*

According to the definition of the property inference in Section 2.2, if we set t_0 in $\mathcal{H}_0$ as 0 and change the condition of $\mathcal{H}_1$ to $t_1 > 0$, the property inference is turned into our property existence inference.

3.2 Significance of Property Existence Inference

The property existence inference considers more personal and personalized privacy such as personal unauthorized training data while proposing a more practical setting where the adversary cannot obtain any samples in the training set.

As an extension of the property inference, property existence inference is very similar to the property inference in their forms of definitions. However, the proportion of target properties differs depending on the target of the adversary. Previous studies of the property inference usually focus on properties (e.g., gender) that account for more than 10% of

the samples in the training set [1, 11, 56, 57]. Therefore, the property inference usually serves as a fairness auditor of the training set to assess the sensitive global properties by predicting the absolute proportion of the properties [57]. By setting one of the proportion of samples with target property in hypotheses to 0, the property existence inference does not focus on the absolute proportion of the target property. Whatever the proportion, it only infers the existence of the target property. It might be meaningless to infer the existence of certain properties that account for a large proportion of the training set, such as inferring whether there are women in the world. However, for those properties that account for only a small proportion (e.g. person's identities and styles of painting), the existence of that property can leak personal and personalized privacy, which has not ever been considered by the previous works of the property inference.

Another well-known attack to infer the existence of samples in the training set is the membership inference [2, 20, 21, 26, 29, 33, 46, 52, 54, 55]. Though sharing a similar purpose of existential judgment, the property existence inference differs from the membership inference. Membership inference focuses on samples identical to (e.g., same objects in the same environment) those in the training set and ignores samples that share the same property (e.g., the same type of objects in different environments) with the training set which is considered in our property existence inference. The adversary who performs property existence inference can hardly obtain the exact data used to train the target model. Therefore, in our evaluations, the property existence inference considers a more practical setting where samples exactly the same as the training set cannot be obtained, which has never been considered in the membership inference.

3.3 Threat Model

We consider an adversary $\mathcal{A}$ interacting with a generative model based on the neural network f_θ. The goal of the adversary $\mathcal{A}$ is to infer whether any samples of the given property $\mathbb{P}$ are used to train the target generative model. For the sake of expediency in exposition, if there are samples with this property present in the training set of the target model, we refer to this property as in-property; otherwise, we call it out-property.

Adversary's Knowledge. In this paper, we focus on the practically and commonly utilized black-box setting where the adversary can only query the target model with a certain input prompt and get its corresponding output images. For generative models, the adversary is limited to passively getting a generated dataset D_{gen} of the target model without any knowledge of its parameters and structures. In particular, the adversary can specify the additional information such as captions for conditional generative models which generate corresponding outputs by accepting specific inputs from the user. However, the adversary has no way of specifying them for unconditional generative models which only use random noise as input.

We follow the assumption widely adopted in the research of privacy-preserving machine learning that the adversary can access the overall dataset that is used to collect the training set of the target model. Therefore, the adversary can sample data from the overall dataset to assist in conducting the inference. It is further assumed that the overall dataset is sufficiently large, thereby rendering it impossible for the adversary to access the samples with the same properties as those in the training set. Under this assumption, the adversary is always able to collect a dataset D_{out} with completely different properties from the training set. For each target property $\mathbb{P}$, we allow the adversary to collect a dataset D_A, where each sample in this dataset contains the target property.

4 Attack Methodology

In this section, we start by providing the overall attack procedure and then detail how to conduct the property existence inference against generative models.

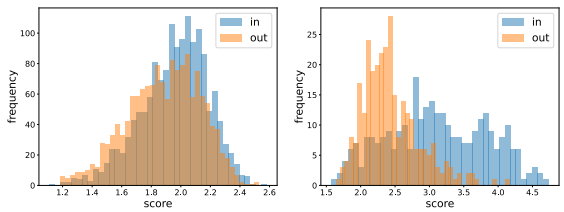
4.1 Overall Attack Procedure

The procedure of our property existence inference consists of three stages: property extractor training, similarity computation, and threshold selection. Firstly, we train a property extractor to map the data of D_A and D_{gen} into the embedding space based on different properties. Secondly, we assign the target property a score by computing the similarities between the embedding obtained from the data of D_A and D_{gen}. Finally, we train the shadow models to select a threshold to make the final decision.

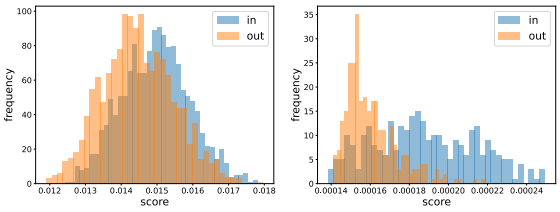
4.2 Property Extractor Training

Since we aim to utilize the influence of target properties on the generated images, we need to train a property extractor to distinguish features from different properties. We design the property extractor as a network trained with the triplet loss function [40]. It utilizes a triplet dataset containing three images for each data point: a base image with a specific property, a positive image with the same property, and a negative image with a different property. Samples used to train the target property extractor are labeled according to the same classification criteria as the target property. For instance, if we consider "identity = Alice" as the target property, then the properties we use to train the network are the identities of different people.

During the training stage, the siamese network architecture is employed with shared weights to embed these images into an embedding space. We train the network to minimize the cosine distance between the base and the positive embedding while maximizing the cosine distance between the base and the negative embedding. Evidently, the utilization of data with target property for training the network would yield



(a) CelebA-HQ wo. (b) Imagenet wo.



(c) CelebA-HQ w. (d) Imagenet w.

Figure 1: The frequency distribution of the similarity score on CelebA-HQ and Imagenet among all data generated by DDPMs. Scores computed with similarity score smoothing and likelihood calibration of properties absent in the training set (in orange) shown by the second row are further to those of properties present (in blue) compared with the pure cosine similarity scores in the first row.

better results. Nonetheless, our evaluations indicate that, even in cases where the target property is not explicitly a label in the training set of the property extractor, it can generalize its discriminating capabilities to the target property. This fact permits adversaries to leverage pre-trained models as property extractors for initiating attacks, thereby substantially reducing the associated attack costs while achieving a reasonable degree of attack effectiveness.

4.3 Similarity Computation

In this stage, we compute the similarity between the data with the target property and the generated data to determine whether the target property is used to train the target model. The process of computing similarity scores is detailed in Algorithm 1. Specifically, for each image with the target property collected by the adversary which is referred to as the anchor image, we compare its cosine similarity with all generated images after property extraction (Lines 12-14 in Algorithm 1). Subsequently, the summation of the K-largest computed similarity scores among all generated images is designated as the similarity score for that anchor image (Lines 11-23 in Algorithm 1). The chosen K-largest values (top K) represent generated images most likely to be influenced by the corresponding target property. Similarity scores of each anchor image in the entire dataset D_A collectively constitutes the

Algorithm 1: Compute Similarity Score.

Input: Generative model f, property extractor $\mathcal{T}$, anchor images set D_A, reference dataset D_{out}, hyperparameters α, K

Output: Similarity Score of the target property Λ

1 $D_{gen} \leftarrow f, h_A \leftarrow \mathcal{T}(D_A), h_{gen} \leftarrow \mathcal{T}(D_{gen}), h_{out} \leftarrow \mathcal{T}(D_{out})$
2 $\Lambda = 0, \mu = \{\}, \sigma^2 = \{\}$
3 **for** c_{gen} *in* h_{gen} **do**
4 $\quad h = \{\}$
5 $\quad$ **for** c_{out} *in* h_{out} **do**
6 $\quad\quad h = h \cup \text{Cosine}(c_{out},\ c_{gen})$
7 $\quad$ **end**
8 $\quad \mu = \mu \cup \text{mean}(h)$
9 $\quad \sigma^2 = \sigma^2 \cup \text{var}(h)$
10 **end**
11 **for** c_A *in* h_A **do**
12 $\quad$ **for** c_{gen} *in* h_{gen} **do**
13 $\quad\quad score = \{\}$
14 $\quad\quad score = score \cup \text{Cosine}(c_A,\ c_{gen})$
15 $\quad$ **end**
16 $\quad enhance\_score = \{\}$
17 $\quad$ **for** i *in* $|score|$ **do**
18 $\quad\quad enhance\_score = enhance\_score \cup -p\left(score[\text{i}] \mid \mathcal{N}\left(\mu[\text{i}], \sigma^2[\text{i}]\right)\right)$
19 $\quad$ **end**
20 $\quad score = \text{SoftMax}(score)$
21 $\quad enhance\_score = score + \alpha * enhance\_score$
22 $\quad \Lambda = \Lambda + \sum topK(enhance\_score)$
23 **end**
24 **return** Λ

derived similarity score for the given target property. During the computation of property similarity scores, we employ two operations, namely *similarity score smoothing* and *likelihood calibration*, to mitigate uncertainties arising from anchor images and generated images.

Similarity Score Smoothing. During our evaluations, we found that some anchor images tend to have fairly high (or low) similarities with all generated images. According to recent studies [3], memorized images tend to have abnormally high similarities among the whole similarity distribution. Inspired by this, we further adopt a smoothing method to polish our similarity scores (Line 20 in Algorithm 1). In contrast to a mere similarity comparison, a more effective indicator of the property's presence within a training set is the discernible pattern where a specific anchor image exhibits markedly higher similarity to a limited set of generated images as opposed to the rest. This indicates that the anchor image has evidently captured distinctive property characteristics within these conspicuously generated images with high similarity. Therefore, to mitigate the uncertainty of anchor images, we initially sub-

ject all resultant values to a softmax smoothing operation, followed by selecting the K-largest values.

Likelihood Calibration. Without a well-crafted input from the adversary to control the generating process, images randomly generated from a pre-defined latent space [12, 18] also exhibit comparable uncertainty of diverse similarity score distributions with images in the overall dataset. In order to mitigate this uncertainty, we introduce a likelihood test to calibrate the similarity score by approximating the similarity distribution. Rather than the whole similarity distribution of the overall dataset, we focus on the reference out-distribution which is the distribution of similarity scores between certain generated images and the D_{out} as mentioned in section 3.3.

The reference out-distribution indicates a general level of similarity for data with the out-properties and helps to mitigate the bias of the similarity introduced by the individual characteristics of generated images. For each generated image specially, we calculate the cosine similarity between the generated image and the data from the D_{out}. It is assumed that the reference out-distribution follows a Gaussian distribution and the data sampled from D_{out} is used to calculate its mean and variance (Lines 2-10 in Algorithm 1). For each anchor image, we calculate the corresponding probabilities under the reference out-distributions of all generated images (Lines 16-19 in Algorithm 1) and add them up when calculating the similarity score (Lines 21-22 in Algorithm 1). Aiming for a higher probability of the existing image with the target property present in the training set, we take the negative of the likelihood as likelihood calibration. We use a hyperparameter α to balance the weight of the original similarity and the likelihood calibration.

Figure 1 plots the frequency distribution histogram of the similarity scores of the properties present (blue) and absent (orange) in the training set of CelebA-HQ [22] and ImageNet [7], respectively. The first line illustrates the results obtained without utilizing similarity score smooth and likelihood calibration, whereas the second line showcases the results under these methods. It can be noticed that the gap between the two distributions of similarity scores in the second row is larger than that in the first row, indicating the effectiveness of reducing uncertainty.

4.4 Distinguishing Test

After obtaining the similarity scores for the target property, we select a threshold to decide the result. If the similarity score exceeds the threshold, we predict the samples with the target property as used to train the target model; otherwise, we predict it as not used. In this stage, we first train shadow models with the training set sampled from D_{out}. Next, we select the properties used for training and those not used to obtain their corresponding similarity scores. We model the scores of these two types of properties as Gaussian distributions $\mathcal{N}(\mu_0, \sigma_0^2)$ and $\mathcal{N}(\mu_1, \sigma_1^2)$. The adversary can choose thresholds based on different false positive rate requirements. In our evaluations, we adopt the method of [4] to set the threshold as one of the following to minimize both two test errors:

$$\mathrm{T} = \frac{(\mu_0\sigma_1^2 - \mu_1\sigma_0^2) \pm 2\sigma_1\sigma_0\sqrt{\left(\frac{\mu_1-\mu_0}{2}\right)^2 + (\sigma_0^2 - \sigma_1^2)\log\left(\frac{\sigma_0}{\sigma_1}\right)}}{\sigma_1^2 - \sigma_0^2} \tag{3}$$

We set the $\mathrm{T} = \frac{\mu_0+\mu_1}{2}$ when $\sigma_0 = \sigma_1$.

In this paper, we do not require the adversary to possess any prior knowledge about the model structure or training algorithm of the target model. In our evaluation, we note that the optimal T obtained on different models is similar and does not have a significant impact on the effectiveness. We keep the proportion of samples with different properties fixed at 0.05% when training the shadow models and we demonstrate that when the proportion of the target property is larger than we expected (even larger than 10%), the attack effect will not worsen. In fact, the higher the proportion of property, the better the resulting effect. We provide detailed proof and explanation in the Appendix A.2.

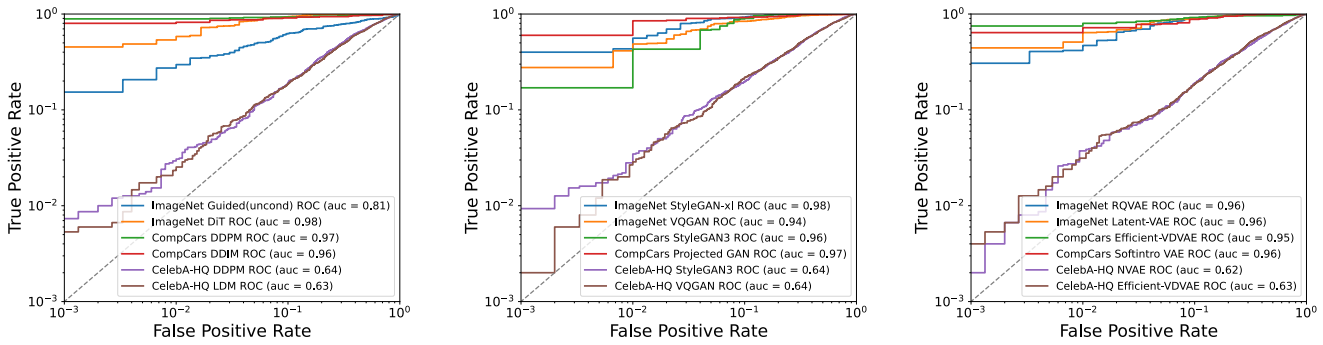
5 Evaluation

5.1 Evaluation on Generative Models

Datasets and Property Extractors. We adopt the following three datasets to investigate the property existence inference and clarify the property extractor used for each dataset.

CelebA-HQ [22] The CelebA-HQ dataset is a CelebFaces Attributes dataset which contains 30,000 face images. CelebA-HQ is the high-resolution version of the CelebA [28] which consists of more than 200,000 RGB face images. In our evaluations, we employ the CelebA-HQ dataset with a resolution of 256×256 to train the target generative models and use the identity as target properties. In our evaluations, we randomly select a group of 1,500 individuals from CelebA-HQ as in-properties and another set of 1,500 individuals from CelebA but not in CelebA-HQ as out-properties. For each property, we randomly sample three images with this property from CelebA as anchor images to create D_A. We also collect an extra 4,500 images from CelebA to form the D_{out}. Furthermore, we assume that the attacker could generate 10,000 images from target models trained on CelebA-HQ to conduct their attack. For all generated and original images, we employ pre-trained facenet [40] as our property extractor to generate embeddings.

ImageNet [7] ImageNet is a large-scale visual dataset consisting of 14,197,122 annotated images according to the WordNet hierarchy used for the Large Scale Visual Recognition Challenge (ILSVRC). We use the version of ImageNet2012 containing 1,000 classes to train target models and ImageNet2010 as the extended dataset to select out-properties. In



(a) Diffsuion Models (DMs). (b) GANs. (c) VAEs.

Figure 2: The ROC curve plot for DMs, GANs, and VAEs showcases a pair of models for both CompCars, ImageNet, and CelebA-HQ in each individual graph.

our evaluations, we randomly select 300 classes from ImageNet2012 to serve as in-properties and another 300 classes from the class difference between ImageNet2012 and ImageNet2010 as out-properties. We sample 5 images for each property to compose D_A. Moreover, we sample 46,000 images from the rest classes of ImageNet2010 to compose D_{out} and resize all the data of ImageNet to 256×256. To extract embeddings, we use the clip fine-tuned on Laion2B[1].

CompCars [53] CompCars is a widely utilized benchmark in computer vision research for tasks of fine-grained object recognition and retrieval. It comprises diverse images of cars captured from various angles, lighting conditions, and occlusions. The dataset includes annotations for car make, model, and year. In our evaluations, we adopt the attributes of models as target properties and select 50,000 images with the resolution of 256×256 to train the target generative models. For in/out-properties, we randomly select 200 models of cars each, ensuring that the data with any selected in-properties constitute no more than 0.1% of the training dataset. We sample 5 images to create D_A for each model of cars and sample 10,000 images from CompCars to obtain D_{out}. We train the property extractor with the network backbone of [19] under VGG16 [43] which has been demonstrated to perform well on fine-grained classification tasks.

Target Models. Among state-of-the-art generative models for each dataset, we select two models as target models for each of the three types of models, e.g., DMs, GANs, and VAEs. For DMs, we adopt the DDPM [18] and LDM [36] trained on CelebA-HQ, DIT [34] and Guided Diffusion [8] trained on ImageNet, DDPM [18] and DDIM [45] trained on CompCars as the target models. For GANs, we adopt the styleGAN3 [23] and VQGAN [10] trained on CelebA-HQ, styleGAN-xl [39] and VQGAN [10] trained on ImageNet, Projected Gan [38] and styleGan3 [23] trained on CompCars as the target models. For VAEs, we adopt the NVAE [51] and Efficient-VDVAE [15] trained on CelebA-HQ, Latent VAE [30] and RQVAE [25] trained on ImagNet, Efficient-VDVAE [15] and Softintro Vae [6] trained on CompCars as the target models. We use the pre-trained models mentioned above for CelebA-HQ and ImageNet, and we train the target models of CompCars by ourselves.

Metrics. The property existence inference can be regarded as a binary classification task to distinguish whether the target property is an in- or out-property. Therefore, we measure the property existence inference with the metrics of Area under the ROC Curve (AUC) and Accuracy (ACC), which are widely used for the evaluation of binary classification tasks. We also use the True Positive Rate (TPR) at 1% False Positive Rate (FPR) to focus on the in-properties that can be confidently inferred. The larger the metrics mentioned above, the better the property existence inference performs. To measure the quality of the generated images, we use the Fréchet Inception Distance (FID) metric [16], where a lower FID indicates better quality of the generated images.

Baseline. Since we consider property existence inference as an extension of property inference, we added the method of [57] as a baseline in our evaluations. [57] predicts the proportion of images with a target property in the training set by evaluating the proportion of generated images that have the target property. We slightly modified this method to align with the goals of property existence inference: for each generated image, we classify it into a specific property category. The number of images classified under the target property is considered as the score for that property. We use those scores to calculate the metrics.

Results. For various models of corresponding datasets under the property existence inference, we show metrics of AUC, ACC, and TPR achieved at 1% FPR in Table 1. We depict the ROC curves in Figure 2, from left to right denoting models of DMs, GANs, and VAEs. Horizontally, there is no particularly

[1] https://mmpretrain.readthedocs.io/en/latest/papers/clip.html

Table 1: The AUCs, ACCs and TPRs achieved at 1% FPR for the effectiveness of property existence inference obtained from our method (PEI) and the baseline method [57] (PIA).

Dataset		Model	FID	PEI (Ours)			PIA (Baseline)		
				AUC	ACC	TPR@1%FPR	AUC	ACC	TPR@1%FPR
ImageNet	DMs	DiT	2.27	0.98	0.92	53.7%	0.81	0.79	4.6%
		guided	4.59	0.81	0.78	27.3%	0.67	0.68	0%
	GANs	styleGAN-xl	2.30	0.98	0.92	43.3%	0.82	0.79	2.4%
		VQGAN	5.2	0.94	0.88	41.3%	0.76	0.73	3.0%
	VAEs	Latent VAE	9.34	0.96	0.91	51.0%	0.72	0.69	0.7%
		RQVAE	4.45	0.96	0.90	41.7%	0.78	0.79	1.7%
CompCars	DMs	DDPM	9.75	0.97	0.95	89.0%	0.87	0.86	64.7%
		DDIM	12.85	0.96	0.92	80.0%	0.81	0.77	19.0%
	GANs	StyleGAN3	28.87	0.96	0.92	17.0%	0.66	0.63	19.4%
		Projected GAN	8.47	0.97	0.94	60.0%	0.86	0.80	30.0%
	VAEs	Efficient-VDVAE	78.12	0.95	0.91	75.0%	0.72	0.71	34.7%
		Softintro VAE	75.81	0.96	0.90	64.0%	0.77	0.74	25.4%
CelebA-HQ	DMs	DDPM	20.25	0.64	0.61	2.9%	0.59	0.58	2.2%
		LDM	19.82	0.63	0.60	2.3%	0.54	0.54	3.2%
	GANs	StyleGAN3	15.68	0.64	0.60	2.8%	0.58	0.57	2.4%
		VQGAN	19.32	0.64	0.60	2.7%	0.59	0.57	2.4%
	VAEs	NVAE	44.31	0.62	0.59	3.7%	0.53	0.53	1.3%
		Efficient-VDVAE	23.55	0.63	0.60	3.1%	0.54	0.54	2.3%

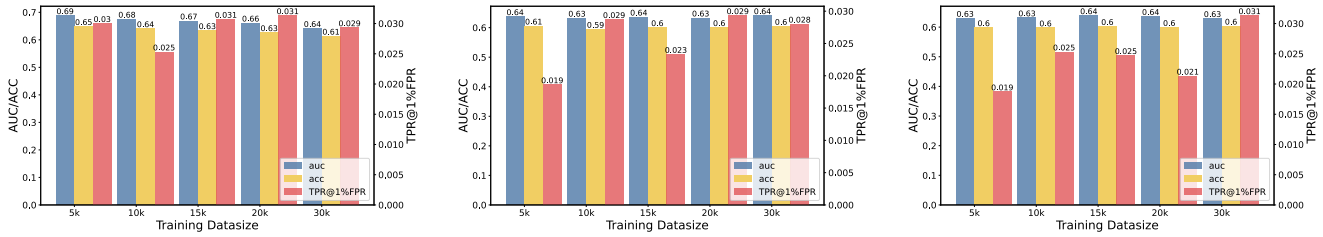
large difference in the performance of different generative models under inferences on the same dataset. In specific, the three aforementioned metrics are consistently close among all models such as for CelebA-HQ, with only a 0.01 fluctuation for AUC, and a 0.02 fluctuation for ACC and TPR achieved at 1% FPR. Vertically for a single model, the property existence inference can get advantages in the three datasets with ImageNet and CompCars considerably high compared to CelebA-HQ, which indicates that ImageNet and CompCars are more susceptible to the property existence inference. For instance, the ROC curves of ImageNet and CompCars with almost all AUCs around 0.95 located far above those of CelebA-HQ whose AUCs fall within a lower range of 0.63 to 0.64. Compared to the baseline, our method outperforms it in almost all metrics on each dataset, except for two values of metric TPR achieved at 1% FPR. Our method demonstrates significant improvement over the baseline in terms of AUC and ACC metrics. For example, our method improves the average AUC by 0.18, 0.18, and 0.07 on ImageNet, CompCars, and CelebA-HQ, while the average ACC improves by 0.14, 0.17, and 0.02, respectively.

Summary I: Most of the generative models we evaluated are vulnerable to the property existence inference and property existence inferences perform similarly against generative models trained on the same dataset.

5.2 Effect of Training Datasize

As mentioned in previous works, the size of the training set is a critical factor in determining the performance of inference attacks [5, 27, 41]. The size of a training set has a direct impact on how a model overfits it. In general, overfitting is reduced as the training set size increases. We also investigate effectiveness of property existence inference with different sizes of training sets.

Setup. We select DMs (DDPMs), GANs (styleGAN3s), and VAEs (Efficient-VDVAEs) as the target models to be trained with different sizes of training sets. We randomly selected 5k, 10k, 15k, 20k, and 30k samples from the CelebA-HQ dataset to train the target generative models, and subsequently conduct property existence inferences in those models. For each model, we randomly select 1,500 in-properties and 1,500 out-properties based on different identities and sample three images with each property to compose D_A. In addition, we generate 10,000 images from each target model to carry out the inferences. To ensure evaluation fairness, we train all



(a) DMs. (b) GANs. (c) VAEs.

Figure 3: The bar charts of AUC (blue), ACC (yellow), and TPR achieved at 1% FPR (red) for Diffusion models, GANs, and VAEs trained on the CelebA-HQ dataset with training set sizes of 5k, 10k, 15k, 20k, and 30k.

models within a certain type with similar FID values.

Results. Figure 3 shows the changes of AUC, ACC, and TPR achieved at 1% FPR across the different sizes of training sets in DMs, GANs, and VAEs respectively. With the exception of the attacks in DDPMs, which exhibit a slight decrease in AUC (i.e., from 0.69 to 0.64) when the training data size increases, attacks in GANs and VAEs do not show significant changes in AUC and ACC as the size of training set increases. In contrast to the general conclusion regarding MIAs [27,42,58], it is surprising that the effectiveness of property existence inferences seems to be insensitive to the change of sizes of the training set. When analyzing the uniqueness of DDPMs, we find that the increase in the replication behavior of DDPMs towards the training set, as the size of the training set decreases, results in an additional advantage for attacks.

Summary II: The primary reason for the effectiveness of the property existence inference is not overfitting, and increasing the size of the training set is not an effective way to prevent property existence inference.

5.3 Effect of Adversary Knowledge

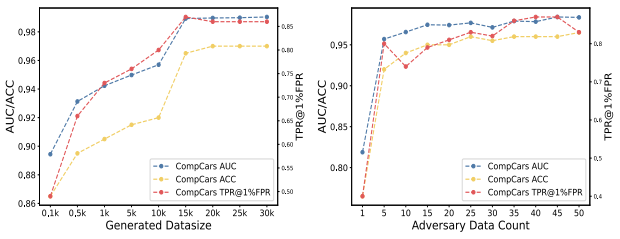


Figure 4: The changes in AUC (blue), ACC (yellow), and TPR achieved at 1% FPR (red) obtained from conducting property existence inferences on the DDIM vary with the increase in the number of the generated images (left), and the number of the data with target property (right).

In our attack scenario, we assume that the adversary can (1) generate a certain amount of images from the target model, (2) choose the target property and collect a specific number of images with the target property as anchor images to conduct property existence inferences. In this section, we evaluate the effect of the amount of both the generated images and the anchor images used by the adversary.

Setup. We investigate the effect of the adversary knowledge by performing property existence inferences against the diffusion model (DDIM) on the CompCars. In our basic settings, we randomly select 200 in-properties and 200 out-properties in CompCars based on the model of cars. To conduct the inference, the adversary samples 5 images for each property as anchor images and generates 10,000 images from the target model. We change these basic settings from two aspects while keeping other settings constant to evaluate the effect of the adversary knowledge. To evaluate the effect of the amount of the generated images, we let the adversary to generate images in the range of 0.1k to 30k from the target model. To evaluate the effect of the amount of the anchor images, we increase the number of the anchor images every 5 from 1 to 50.

Results. Figure 4 illustrates the changing trend of inference performance metrics on DDIM as the adversary knowledge changes. We show the trend of AUC, ACC, and TPR achieved at 1% FPR with a growing number of generated images and anchor images used by the adversary respectively on the left and right of Figure 4. It's clear that with a growing size of generated images and anchor images, the attack metrics will grow until they reach a threshold.

To further simulate real-world scenarios, we conducted the evaluations on CelebA-HQ by supplementing it with anchor images from social media, which are potential sources for adversaries to collect anchor images. The result is attached in the Appendix A.1. We get the same results as CompCars when the number of the generated images increases. However, we find a peak in metrics as the number of the anchor images increases. The peak in the figure signifies that as the number of the anchor images with target property increases, the gap in similarity scores between in-properties and out-properties

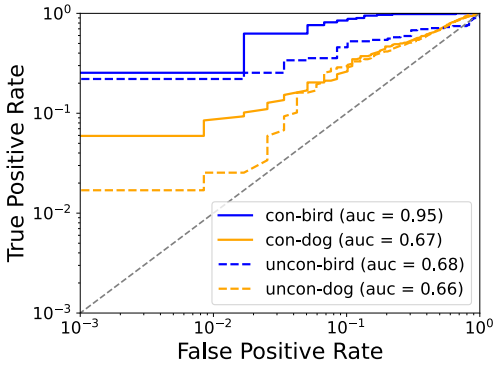


Figure 5: ROC curves for the fine-grained evaluations. “con” and “uncon” stands for the different types of the target models.

rapidly increases until reaching the optimum and finally decreases. Hence, we suggest the adversary train shadow models to select the appropriate number of anchor images for performing property existence inferences.

Summary III: 1) For the same training dataset, the adversary will get better performance of the property existence inference when the number of the generated images and anchor images increases. 2) For different training datasets, there exists an optimal quantity of anchor images that maximizes the attack performance.

5.4 Effect of Property Granularity

As we rely on manual labeling to determine the properties, samples can be classified into properties at different levels of granularity. For instance, an image of a canary may be classified as a fine-grained canary, a coarse-grained bird, or even a coarser-grained animal. In this section, we investigate the effectiveness of the property existence inference under different property granularities. It is worth mentioning that in the previous overall evaluations, we have adopted “identity”, “models of cars”, and the original label class in ImageNet as target properties, which may differ in granularities across different datasets. To better compare the effects of different granularities for the same dataset, we further conduct the property existence inference of ImageNet based on classification criteria under WordNet, a hierarchical tree-like structure with basic relations of synsets and hypernyms for fine-grained categorization. For instance, the synset "animal" is a hypernym of "dog" and the synset "dog" is further a hypernym of more fine-grained synsets such as "poodle", "labrador" and "bulldog", representing the "is-a" relationship.

Setup. To study the effectiveness of the property existence inference with different property granularities, we collect original synsets with hypernyms of birds and dogs from ImageNet2012 respectively as our in-properties and extract the other synsets from ImageNet as out-properties under the same hypernyms, ensuring that the in- and out-properties are not hypernyms of each other in the WordNet. In brief, we find out other children as out-properties sharing the same father as in-properties but not in one branch with in-properties. For birds and dogs, we use 100 and 300 classes of in-/out- properties respectively, and samples D_{out} from the rest classes. We use the Guided Diffusion [9] trained on ImageNet2012 as the target models and use the EfficientNet-b1 [50] pre-trained by torchvision as our property extractor.

Results. As shown in Figure 5, the ROC curves of the birds lie above those of the dogs under conditional and unconditional Guided Diffusions. In the conditional model, we get AUCs of 0.95 and 0.67 for birds and dogs respectively while in the unconditional model, we get AUCs of 0.68 and 0.66 respectively, which indicates that there still exists property existence information leakage in the fine-grained properties. Compared to attacks at a coarse level (e.g., ImageNet) in Figure 2 with AUCs of 0.81 and 0.98 at conditional and unconditional diffusion models, we can conclude that the finer the granularity of the properties, the harder it is to infer their existence information. Between the two fine-grained properties, it’s observed that the bird species are relatively easier to infer, due to the larger intra-class diversity of birds. Though the overall AUCs between conditional and unconditional models differ a lot, we observe close TPR at 1% FPR of 0.25 and 0.22 for birds evaluations. This indicates that there are indeed some properties whose features are more easily manifested in the generated images, regardless of whether it is in conditional or unconditional models.

Summary IV: Property existence inferences prove effective in the context of fine-grained datasets, and selecting properties at a finer granularity level results in higher inference difficulty.

5.5 Why Our Attacks Work

In previous sections, we present the effectiveness of our approach to perform the property existence inference under various settings from different generative models, different datasets to different granularities. To gain a deeper understanding of property existence inference, we further investigate which type of generated data contributes to the leakage of the property existence information.

How generated samples leak the property existence information. On left of the Figure 6, we display the performance of our attacks under DDPM trained on CelebA-HQ. According to Algorithm 1, we mainly focus on those generated samples most similar to each anchor image of target properties, since they are used to calculate the similarity scores during the attack. Specifically, we collect the top 20 generated samples with the highest similarity scores corresponding to each anchor image. In brief, we call these samples as chosen generated samples and the anchor image as their corresponding anchor image. Each chosen generated sample can have many corresponding anchor images and each anchor image

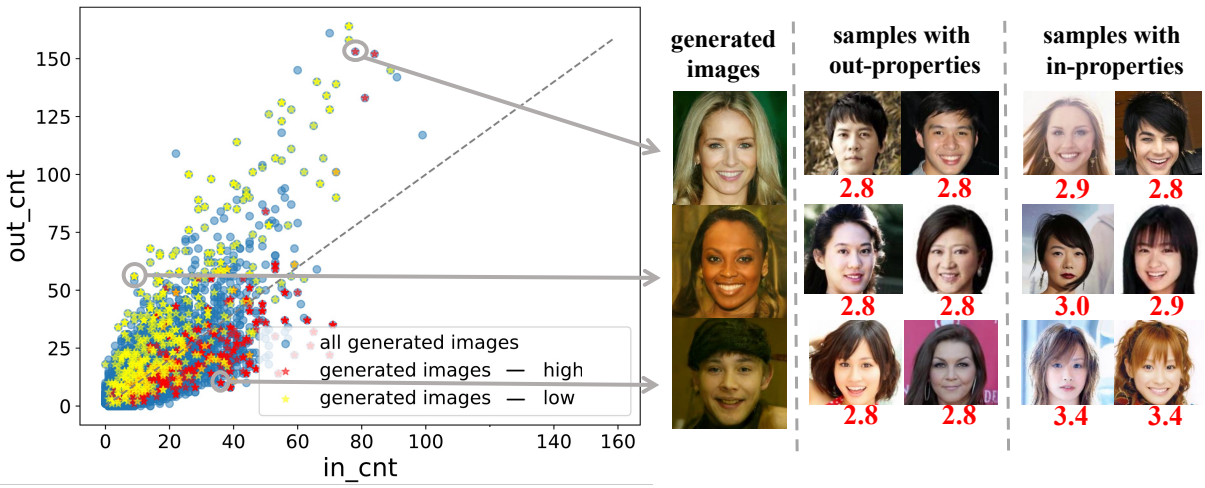


Figure 6: Figure in the left plots the top 20 generated samples collected with the highest similarity scores corresponding to each anchor image. The right side displays generated images of three chosen points representing generated data with high risks, low risks, and nearly no risks of property existence information leakage. For each chosen point, the corresponding anchor images with the highest similarity scores are plotted on the right. Below each anchor image marks their similarity scores ($\times 1,000$). For each chosen generated sample (blue dots), the x-axis and y-axis indicate the total number of their corresponding anchor images with in- and out-properties respectively. Most points with the lowest similarity scores (yellow stars) lie above the gray line while those with the highest scores (red stars) lie below, indicating that generated samples prone to leaking property existence information are more similar to relatively large amounts of anchor images with in-properties. From the bottom up, the difference in similarity scores between corresponding anchors with in- and out-properties and the chosen generated sample gradually decreases, indicating the weakened ability of the generated sample to distinguish in- and out-properties.

corresponds to 20 chosen generated samples. By intuition, it's considered that the riskiest generated sample should share markedly high similarities with more anchor images of in-properties compared with those of out-properties. So we further analyze the statistical difference between corresponding anchor images of in-properties and out-properties for each chosen generated image from the following two dimensions.

- **Quantity**: The relative number of anchor images with in- and out-properties.
- **Value**: The absolute value of similarity scores with anchor images of in- and out-properties.

To explore the dimension of **Quantity**, we count the number of corresponding anchor images with in-properties and out-properties respectively for each chosen generated sample. The result is displayed in Figure 6 where each blue point denotes a certain chosen generated sample. Among all corresponding anchor images of that sample, the total number of anchor images with in-properties is shown in the x-axis and that with out-properties is shown in the y-axis. The points on the grey line which separates the figure into two parts represent those chosen generated samples with an equal number of corresponding anchor images between in- and out-properties. To find generated samples that are easier and harder to leak property existence information, we mark chosen generated samples with the highest similarity scores in red stars and those with the lowest similarity scores in yellow stars. It's interesting to notice that most chosen generated samples with the lowest similarity scores lie above the gray line while those with the highest similarity scores lie below. Statistically speaking, it indicates that generated samples more prone to leaking property existence information tend to have more corresponding anchors of in-properties than out-properties.

To explore another dimension of **Value**, we choose three points in each representative area in the left figure and display the chosen generated samples along with their corresponding anchor images of in/out-properties with the highest similarity scores in the right of Figure 6. Below each anchor image marks its similarity score with the leftest chosen generated sample. From the bottom up, the similarity scores of the corresponding anchor images with out-properties remain the same but those of corresponding anchors with in-properties gradually decrease. It indicates that the difference in similarity scores between corresponding anchor images with in- and out-properties of the chosen generated sample gradually decreases and converges to 0. Moreover, it illustrates that the chosen generated sample gradually loses the ability to distinguish anchor images of in-properties and out-properties from the bottom up. From this point, we can say that the three points from the bottom up respectively represent generated samples with the highest risks, low risks, and nearly no risks

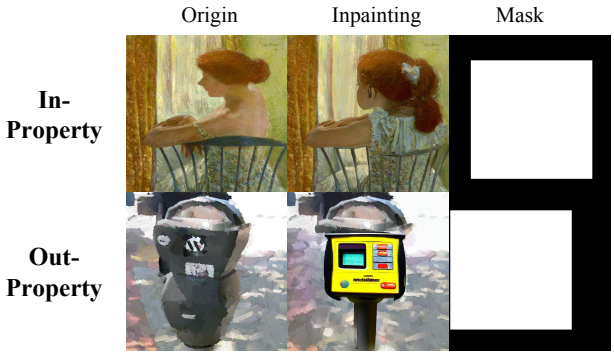


Figure 7: The visualized strategies of the inpainting evaluation. The first row belongs to the evaluation of image with in-property and the second row belongs to the evaluation of image with out-property generated by applying style transform to image in ImageNet2012. We apply the mask in the third column to the original image in the first column and inpaint the masked part with Stable Diffusion to acquire the inpainting image in the second column.

of property existence information leakage. Consequently, generated samples with high risks of leaking property existence information tend to have abnormally high similarity scores with anchor images of in-properties.

5.6 Case Study: Stable Diffusion

In order to investigate the effectiveness of the property existence inference in generative models for real-world scenarios, we perform attack on the open-source model Stable Diffusion [36] which is trained on LAION [41] with 5.85 billion CLIP-filtered image-text pairs.

Setup *Attack Goal.* As a means of distinguishing in/out-properties, we utilize the artistic style of works and assess whether any given artist's pieces are present in LAION. *Datasets.* We use the WikiArt [2] dataset which included 195 artists to select the in-properties. Inspired by the research of the Memorization about the diffusion models [3], we consider that the property of an artist's style is an in-property when the style of the generated images with their names mentioned in the prompt is very close to that of their real paintings. We finally found 100 in-properties under this method. As for the out-properties, the large data scale of LAION and the possibility that images obtained from the internet may not be labeled with artists' names make it infeasible to determine that none of an artist's pieces are present in LAION. Therefore, We seek a substitute method to acquire the out-properties, which applies style transformation [59] to real-world images (i.e., ImageNet in our evaluation) and produces the oil-painting style images. Then we apply clustering algorithms to divide the generated oil paintings, which are pre-processed by a property extractor trained on WikiArt to differentiate between different artistic styles, into distinct properties which are used as the out-properties to evaluate the performance of attacks in this evaluation.

[2] https://www.wikiart.org/

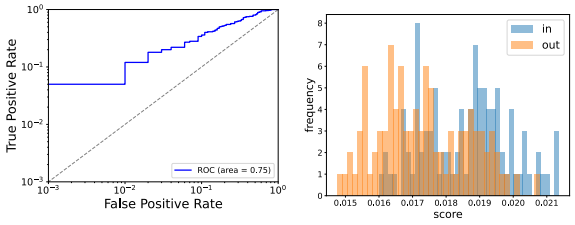


Figure 8: (Left) The ROC curve for attacks on Stable Diffusion. (Right) The frequency distribution histogram of the similarity scores for in-properties (blue) and out-properties (yellow).

Attack procedure. Because the out-properties lack clear information about the artists used in the prompts to query Stable Diffusion, we adopt the inpainting method to obtain the generative samples. Specifically, we randomly crop $3/4$ part (i.e., 384×384) of the images with the target property of original size 512×512 and use the Stable Diffusion to inpaint such cropped part. Similar to our previous assumption about the generative models, we expect that Stable Diffusion will have a better reconstruction performance on the images with in-properties. Therefore, We calculate the sum of cosine similarity of the embedding vectors of the before-cropped images with the target property and the after-inpainted images as the score of this property. Then we predict that properties that achieve scores greater than the set threshold are in-properties. In Figure 7, we visualized evaluations on the image with in/out-property separately. We observed that the style of the inpainted image with in-property is more similar to the original image compared to images with out-property.

Results. On the left of Figure 8, we depict the ROC curves of the inference against the Stable Diffusion, with an AUC score of 0.75. We also depict the frequency distribution histogram of the scores for those two types of properties in the right. It can be observed that the distribution of the scores of in-properties is more concentrated towards the right compared to out-properties. The results indicate that Stable Diffusion indeed memorizes some properties included in the training set and generates images based on those properties. The results of the evaluation also demonstrate that property existence inference is effective for models trained on large-scale training sets in real-world scenarios.

6 Related Work

Property Existence Inference. In the literature, there are already several studies investigating the topics related to the

property existence inference. [13] proposes distributional membership inferences that aim to determine whether a given party with a specific distribution contributes to the training set of classification models and conducts evaluations on the synthetic dataset. [32, 49] aim to determine if samples with the target property emerge in the training set under the setting of collaborative learning. [4] conducts property existence inferences on the classification models through poisoning. Compared to the aforementioned works, the target model in this paper focuses on image generative models rather than classifiers. Additionally, this paper assumes that the adversary cannot intervene in the training process of the target model nor manipulate the training set of the target model.

Inferences against Generative Models. Multiple studies have demonstrated the potential for extracting information from the training sets of generative models. [5, 14, 17, 58] focus on the membership inference which aims to determine whether a specific sample is used to train the target model. [57] aims to extract the overall distribution information of the training set, i.e., estimate the proportion of the data with target property in the training set. [3, 44] investigate the duplication phenomena about the diffusion models that the model directly generates the data in the training set. The focus of this paper differs from the aforementioned works. It does not center around specific samples or the overall distribution of the training set, but rather on the existence of samples with specific properties within the training set.

Property Inference against Discriminative Models. [1] first introduced property inference as a binary classification task on Hidden Markov Models and Support Vector Machines to extract the global information of the training set distribution. Subsequent works primarily focus on property inference against neural networks [11, 56]. These works launch inferences by training a binary classifier to distinguish the behavior of discriminative models trained on training sets with different property proportions. Recent works increase the behavioral gap of models trained on properties with different proportions through poisoning [4, 31]. In this paper, we focus on the information of more personalized properties and selects generative models as the target models.

7 Discussion

In this section, we discuss the possibility of less costly property existence inference under a partial black-box setting and analyze potential reasons for the performance on CelebA-HQ.

For unconditional generative models, obtaining specific outputs corresponding to particular inputs is unfeasible so the only viable approach to perform property existence inference is generating a substantial volume of data under our black-box setting. In Section 5.3, 15,000 generative images are needed to achieve the best performance of CompCars. Under the partial black-box setting, [57] reduces the number of required generated images by modifying the latent variables as inputs of GANs to obtain outputs with the target property through gradient descent. However, we observed that generative models (e.g., DDPM) with more powerful generative capabilities are able to produce images remarkably similar to specific properties. Moreover, the computational overhead incurred by gradient propagation is extremely high for diffusion models. Therefore, modifying the latent variables to reduce the number of generated images for all types of generative models is not feasible. We leave the challenge to reduce the required number of generated images under different settings from [57] as future work.

In Section 5.1, it can be observed that the effectiveness of our method on the CelebA-HQ is significantly inferior to that on ImageNet and CompCars. We analyze this difference and conclude that it may be caused by the following two reasons:

- Higher similarity between properties of CelebA-HQ: for CelebA-HQ, we choose identities as target properties, which may result in smaller gaps between different properties in evaluations compared to ImageNet (use different classes of the dataset) and CompCars (use different models of cars). We utilize CLIP to extract features from images with different properties in all datasets and compute the average differences between the two properties which are 9.94 in CelebA-HQ, 13.09 in ImageNet, and 10.30 in CompCars. Therefore, the selected properties in the CelebA-HQ appear to be the most similar.
- Limitations of the adversary knowledge: the number of anchor images corresponding to each property in CelebA-HQ is less than the others. Appendix A.1 shows that increasing the number of anchor images leads to an improved AUC value exceeding 0.74 for CelebA-HQ.

8 Conclusion

In this paper, we present property existence inference against generative models to determine whether any samples with target property are contained in the training set of the target model. We launch the attack by exploiting the difference in similarities between the generated images and anchor images with the in/out-properties. Furthermore, we enhance the attack performance by separately removing the uncertainties of the generated images and the anchor images. We have demonstrated through a comprehensive set of evaluations that property existence inference can effectively extract property existence information in generative models including large-scale models like Stable Diffusion. We discovered that the effectiveness of property existence inference is closely related to the number of anchor images and generated images, as well as the granularity of the target property. However, it is not highly sensitive to the size of the training set. Our research further reveals that there is an increased risk of property existence information leakage for the generated image that closely resembles a larger number of anchor images.

Acknowledgments

We would like to express our sincere gratitude to the anonymous reviewers and Shepherd for their invaluable feedback and constructive comments, which greatly contributed to the enhancement of this paper. This work was supported by National Natural Science Foundation of China (62102353).

References

[1] 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. Netw.*, 10(3):137–150, sep 2015.

[2] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914, 2022.

[3] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models, 2023.

[4] Harsh Chaudhari, John Abascal, Alina Oprea, Matthew Jagielski, Florian Tramèr, and Jonathan Ullman. Snap: Efficient extraction of private properties with poisoning. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 400–417, 2023.

[5] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 343–362, New York, NY, USA, 2020. Association for Computing Machinery.

[6] Tal Daniel and Aviv Tamar. Soft-introvae: Analyzing and improving the introspective variational autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4391–4400, June 2021.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021.

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[10] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12873–12883, June 2021.

[11] 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 *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 619–633, New York, NY, USA, 2018. Association for Computing Machinery.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[13] Valentin Hartmann, Léo Meynent, Maxime Peyrard, Dimitrios Dimitriadis, Shruti Tople, and Robert West. Distribution inference risks: Identifying and mitigating sources of leakage. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 136–149, 2023.

[14] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019(1):133–152.

[15] Louay Hazami, Rayhane Mama, and Ragavan Thurairatnam. Efficient-vdvae: Less is more, 2022.

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[17] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte carlo and reconstruction membership inference attacks against generative models. *Proc. Priv. Enhancing Technol.*, 2019(4):232–249, 2019.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[19] Qichang Hu, Huibing Wang, Teng Li, and Chunhua Shen. Deep cnns with spatially weighted pooling for fine-grained car recognition. *IEEE Transactions on Intelligent Transportation Systems*, 18(11):3147–3156, 2017.

[20] Bo Hui, Yuchen Yang, Haolin Yuan, Philippe Burlina, Neil Zhenqiang Gong, and Yinzhi Cao. Practical blind membership inference attack via differential comparisons. In *ISOC Network and Distributed System Security Symposium (NDSS)*, 2021.

[21] Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. Revisiting membership inference under realistic assumptions. *Proceedings on Privacy Enhancing Technologies*, 2021(2), 2021.

[22] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[23] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021.

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[25] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11523–11532, June 2022.

[26] Yiyong Liu, Zhengyu Zhao, Michael Backes, and Yang Zhang. Membership inference attacks by exploiting loss trajectory. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2085–2098, 2022.

[27] 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 *31st USENIX Security Symposium (USENIX Security 22)*, pages 4525–4542, Boston, MA, August 2022. USENIX Association.

[28] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[29] Yunhui Long, Lei Wang, Diyue Bu, Vincent Bindschaedler, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. A pragmatic approach to membership inferences on machine learning models. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 521–534. IEEE, 2020.

[30] Troy Luhman and Eric Luhman. High fidelity image synthesis with deep vaes in latent space. *arXiv preprint arXiv:2303.13714*, 2023.

[31] Saeed Mahloujifar, Esha Ghosh, and Melissa Chase. Property inference from poisoning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1120–1137, 2022.

[32] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706, 2019.

[33] 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 *2019 IEEE Symposium on Security and Privacy (SP)*, pages 739–753, 2019.

[34] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

[35] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[37] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and

A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 36479–36494. Curran Associates, Inc., 2022.

[38] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[39] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 Conference Proceedings*, SIGGRAPH '22, New York, NY, USA, 2022. Association for Computing Machinery.

[40] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[41] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022.

[42] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017.

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[44] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. *arXiv preprint arXiv:2212.03860*, 2022.

[45] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.

[46] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *USENIX Security Symposium*, volume 1, page 4, 2021.

[47] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[48] Anshuman Suri and David Evans. Formalizing and estimating distribution inference risks. *Proceedings on Privacy Enhancing Technologies*, 4:528–551, 2022.

[49] Anshuman Suri, Pallika Kanani, Virendra J Marathe, and Daniel W Peterson. Subject membership inference attacks in federated learning. *arXiv preprint arXiv:2206.03317*, 2022.

[50] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[51] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19667–19679. Curran Associates, Inc., 2020.

[52] Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. On the importance of difficulty calibration in membership inference attacks. In *International Conference on Learning Representations*, 2022.

[53] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3973–3981, 2015.

[54] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 3093–3106, New York, NY, USA, 2022. Association for Computing Machinery.

[55] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[56] Wanrong Zhang, Shruti Tople, and Olga Ohrimenko. Leakage of dataset properties in {Multi-Party} machine learning. In *30th USENIX security symposium (USENIX Security 21)*, pages 2687–2704, 2021.

[57] Junhao Zhou, Yufei Chen, Chao Shen, and Yang Zhang. Property inference attacks against gans. *NDSS 2022*, 2022.

[58] Derui Zhu, Dingfan Chen, Jens Grossklags, and Mario Fritz. Data forensics in diffusion models: A systematic analysis of membership privacy, 2023.

[59] Zhengxia Zou, Tianyang Shi, Shuang Qiu, Yi Yuan, and Zhenwei Shi. Stylized neural painting. 2020.

A Appendix

A.1 The Effect of the Adversary Knowledge in Real-world scenarios

Figure 9 shows the change of the metrics of attacks when changing the adversary knowledge.

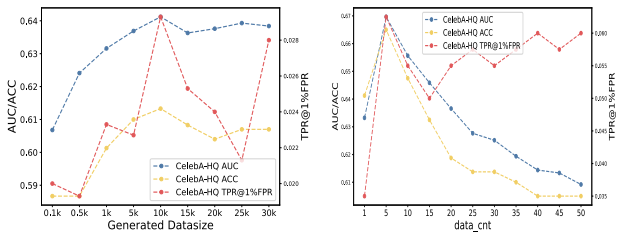


Figure 9: The change of attack performance with varying adversary knowledge. Left: Effect change with increasing generated images. Right: Effect change with increasing anchor images.

A.2 The Inference Effect when the Property Proportion is Large

Table 2: The mean (μ) and variance (σ) of similarity scores that obtained from the properties with different proportions.

proportion	1%	2%	5%	10%
μ	0.102	0.102	0.112	0.115
$\sigma \times 10^{-5}$	11.61	6.82	3.81	3.71

In our attack procedure, we compute similarity scores for each in/out-property and model the similarity score distribution as a Gaussian distribution. Table 2 shows that as the proportion of in-properties in the training set increases, the mean of the in-property similarity score distribution increases and the variance decreases. Based on this finding, we will demonstrate that using the same shadow model, the attack effect increases as the proportion of in-property increases.

Optimal threshold chosen by the shadow model. We assume that the similarity score distribution of in-properties obtained by the shadow model follows the Gaussian distribution $\hat{X} \sim N(\hat{\mu}, \hat{\sigma})$ and the similarity score distribution of out-properties follows distribution $X_0 \sim N(\mu_0, \sigma_0)$. Therefore, $\hat{\mu} > \mu_0$ and $\hat{\sigma} < \sigma_0$. The threshold T is chosen by minimizing the sum of the probabilities of making Type-I and Type-II errors which satisfies the condition as follows:

Claim 1. *Given two Gaussian distributions $X_0 \sim N(\mu_0, \sigma_0)$ and $\hat{X} \sim N(\hat{\mu}, \hat{\sigma})$ such that $\hat{\mu} > \mu_0$, $\hat{\sigma} < \sigma_0$ and objective function $J(\mathrm{T}) = \Pr[X_0 > \mathrm{T}] + \Pr[\hat{X} < \mathrm{T}]$, the threshold T that minimizes J must satisfy:*

$$\mathrm{T} \leq \hat{\mu}$$

Proof. On contradiction, assume the minimum value of $J(\mathrm{T})$ is taken under $\mathrm{T} > \hat{\mu}$. Therefore, $J(\mathrm{T})$ can be re-written as follows:

$$\begin{aligned} J(\mathrm{T}) &= \Pr[X_0 > \mathrm{T}] + \Pr[\hat{X} < \mathrm{T}] \\ &= 1 - \Pr[X_0 < \mathrm{T}] + \Pr[\hat{X} < \mathrm{T}] \\ &= 1 - \Phi\left(\frac{\mathrm{T}-\mu_0}{\sigma_0}\right) + \Phi\left(\frac{\mathrm{T}-\hat{\mu}}{\hat{\sigma}}\right) \end{aligned} \tag{4}$$

Consider a special point of $J(\mathrm{T})$ when T takes $\hat{\mu}$, we can express $J(\hat{\mu})$ as follows:

$$\begin{aligned} J(\hat{\mu}) &= 1 - \Phi\left(\frac{\hat{\mu}-\mu_0}{\sigma_0}\right) + \Phi\left(\frac{\hat{\mu}-\hat{\mu}}{\hat{\sigma}}\right) \\ &= 1.5 - \Phi\left(\frac{\hat{\mu}-\mu_0}{\sigma_0}\right) \end{aligned} \tag{5}$$

To compare the value between $J(\mathrm{T}|_{\mathrm{T}>\hat{\mu}})$ and $J(\hat{\mu})$, we take the difference D between Equation 4 and Equation 5:

$$\begin{aligned} D =& J(\mathrm{T}) - J(\hat{\mu}) \\ =& \Phi\left(\frac{\mathrm{T}-\hat{\mu}}{\hat{\sigma}}\right) - \Phi\left(\frac{\mathrm{T}-\mu_0}{\sigma_0}\right) \\ &+ \Phi\left(\frac{\hat{\mu}-\mu_0}{\sigma_0}\right) - 0.5 \end{aligned} \tag{6}$$

To consider the monotonicity of D, we compute the derivative of D with respect to T:

$$\frac{\partial D}{\partial \mathrm{T}} = \frac{1}{\hat{\sigma}}\phi\left(\frac{\mathrm{T}-\hat{\mu}}{\hat{\sigma}}\right) - \frac{1}{\sigma_0}\phi\left(\frac{\mathrm{T}-\mu_0}{\sigma_0}\right) \tag{7}$$

Equation 7 can be regarded as the difference between the $\hat{X}$ and X_0. The intersection points x of these two PDFs can be obtained as:

$$\begin{aligned} &\frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x-\mu_0)^2}{2{\sigma_0}^2}} = \frac{1}{\sqrt{2\pi}\hat{\sigma}} e^{-\frac{(x-\hat{\mu})^2}{2\hat{\sigma}^2}} \\ &\Rightarrow \ln(\sigma_0 e^{-\frac{(x-\mu_0)^2}{2{\sigma_0}^2}}) = \ln(\hat{\sigma} e^{-\frac{(x-\hat{\mu})^2}{2\hat{\sigma}^2}}) \\ &\Rightarrow \left(\hat{\sigma}^2 - {\sigma_0}^2\right) x^2 + 2\left({\sigma_0}^2\hat{\mu} - \hat{\sigma}^2\mu_0\right) x + \hat{\sigma}^2{\mu_0}^2 \\ &\quad - {\sigma_0}^2\hat{\mu}^2 - 2{\sigma_0}^2\hat{\sigma}^2 \ln\frac{\sigma_0}{\hat{\sigma}} = 0 \end{aligned} \tag{8}$$

Therefore the two Gaussian function curves can have at most two intersection points. We focus on the monotonicity of D when $T > \hat{\mu}$.

$$\left.\frac{\partial D}{\partial \mathrm{T}}\right|_{\mathrm{T}=\hat{\mu}} = \frac{1}{\hat{\sigma}}\phi\left(\frac{\hat{\mu}-\hat{\mu}}{\hat{\sigma}}\right) - \frac{1}{\sigma_0}\phi\left(\frac{\hat{\mu}-\mu_0}{\sigma_0}\right) > 0 \tag{9}$$

$$\begin{aligned}
\left.\frac{\partial D}{\partial \mathrm{T}}\right|_{\mathrm{T}\to+\infty} &= \lim_{T\to+\infty} \frac{1}{\hat{\sigma}}\phi\left(\frac{\mathrm{T}-\hat{\mu}}{\hat{\sigma}}\right) - \frac{1}{\sigma_0}\phi\left(\frac{\mathrm{T}-\mu_0}{\sigma_0}\right) \\
&= \lim_{T\to+\infty} \frac{1}{\sigma_0}\phi\left(\frac{\mathrm{T}-\mu_0}{\sigma_0}\right)\cdot\left(\frac{\frac{1}{\hat{\sigma}}\phi\left(\frac{\mathrm{T}-\hat{\mu}}{\hat{\sigma}}\right)}{\frac{1}{\sigma_0}\phi\left(\frac{\mathrm{T}-\mu_0}{\sigma_0}\right)} - 1\right) \\
&= \lim_{T\to+\infty} \frac{1}{\sigma_0}\phi\left(\frac{\mathrm{T}-\mu_0}{\sigma_0}\right)\left(\frac{\sigma_0}{\hat{\sigma}} e^{\frac{(T-\mu_0)^2}{2\sigma_0^2} - \frac{(T-\hat{\mu})^2}{2\hat{\sigma}^2}} - 1\right) \\
&= \lim_{T\to+\infty} \frac{1}{\sigma_0}\phi\left(\frac{\mathrm{T}-\mu_0}{\sigma_0}\right)\left(\frac{\sigma_0}{\hat{\sigma}} e^{\frac{(\hat{\sigma}^2-\sigma_0^2)\mathrm{T}^2}{2\sigma_0^2\hat{\sigma}^2}} - 1\right) < 0
\end{aligned} \tag{10}$$

Similarly,

$$\left.\frac{\partial D}{\partial \mathrm{T}}\right|_{\mathrm{T}\to-\infty} < 0 \tag{11}$$

According to the intermediate zero theorem, there must be two zero points located in intervals $(-\infty,\hat{\mu})$ and $(\hat{\mu},+\infty)$ respectively.

Since 8 indicates that Equation 7 have at most two zero points, there is exactly one zero point within $(-\infty,\hat{\mu})$ and $(\hat{\mu},+\infty)$ respectively. We assume zero point $T = x^*$ in the interval $(\mu,+\infty)$. Therefore, The value of D increases monotonically in the interval $\mathrm{T} \in (\hat{\mu},x^*)$ and decreases monotonically in the interval $\mathrm{T} \in (x^*,+\infty)$. Hence, at $\mathrm{T} \to \hat{\mu}^+$ or $\mathrm{T} \to +\infty$, D reaches its minimum value when $\mathrm{T} > \hat{\mu}$:

$$\begin{aligned}
\min D|_{\mathrm{T}>\hat{\mu}} &= \min(D|_{\mathrm{T}\to\hat{\mu}^+}, D|_{\mathrm{T}\to+\infty}) \\
&= \min(0^+, \Phi\left(\frac{\hat{\mu}-\mu_0}{\sigma_0}\right) - 0.5) > 0
\end{aligned} \tag{12}$$

For all $\mathrm{T} > \hat{\mu}$, we can obtain:

$$D = J(\mathrm{T}) - J(\hat{\mu}) \geq \min D > 0 \tag{13}$$

This contradicts with our assumption that there is the optimal solution of $J(\mathrm{T})$ under $\mathrm{T} > \hat{\mu}$ since that $J(\hat{\mu})$ is always smaller. So the assumption is wrong. And the optimal threshold value T we choose must satisfy $T \leq \hat{\mu}$.

□

Increased proportion with decreased error. Using the same shadow model means that the threshold T chosen by the adversary remains unchanged and it has the property of $\mathrm{T} \leq \hat{\mu}$ by Claim 1. We assume that the similarity score distribution of in-properties follows distribution $X_1 \sim N(\mu_1,\sigma_1)$ and the similarity score distribution of the out-properties follows distribution $X_0 \sim N(\mu_0,\sigma_0)$. Another similarity score distribution of in-properties with lower proportion such as that of our shadow model follows distribution $\hat{X} \sim N(\hat{\mu},\hat{\sigma})$. So the means and variances have the relationship: $\mu_0 < \hat{\mu} < \mu_1$, $\sigma_0 > \hat{\sigma} > \sigma_1$. The attack effectiveness under property distributions with different proportions can be measured by the objective function $J(X)$, the sum of probabilities of two types of error, where X is a random variable of the similarity score. With an increased proportion of in-properties, the attack error decreases as follows:

Claim 2. *Given three Gaussian distributions $X_0 \sim N(\mu_0,\sigma_0)$, $X_1 \sim N(\mu_1,\sigma_1)$ and $\hat{X} \sim N(\hat{\mu},\hat{\sigma})$ such that $\mu_0 < \hat{\mu} < \mu_1$, $\sigma_0 > \hat{\sigma} > \sigma_1$ and objective function $J(X) = \Pr[X_0 > \mathrm{T}] + \Pr[X < T], X \sim N(\mu,\sigma)$ where T is a constant satisfying $T \leq \hat{\mu}$, then the objective function of $\hat{X}$ and X_1 must satisfy:*

$$J(\hat{X}) > J(X_1)$$

Proof.

$$\begin{aligned}
J(X) &= \Pr[X_0 > \mathrm{T}] + \Pr[X < \mathrm{T}] \\
&= \Pr[X_0 > \mathrm{T}] + \Phi\left(\frac{\mathrm{T}-\mu}{\sigma}\right)
\end{aligned}$$

As the proportion of in-property increases, the similarity score distribution of out-property does not change. Therefore, $\Pr[X_0 > \mathrm{T}]$ is a constant.

To compare $J(\hat{X})$ and $J(X_1)$, we compute the partial derivative of $J(X)$ with respect to μ and σ respectively:

$$\frac{\partial J}{\partial \mu} = -\frac{1}{\sigma}\phi\left(\frac{\mathrm{T}-\mu}{\sigma}\right)$$

$$\frac{\partial J}{\partial \sigma} = -\frac{\mathrm{T}-\mu}{\sigma^2}\phi\left(\frac{\mathrm{T}-\mu}{\sigma}\right)$$

where $\frac{1}{\sigma}\phi\left(\frac{\mathrm{T}-\mu}{\sigma}\right)$ denotes the PDF of X. Obviously, $\frac{\partial J}{\partial \mu} < 0$. So as μ increases, the probability of making two types of errors decreases. Since $\mathrm{T} \leq \hat{\mu} < \mu_1$, $\frac{\partial J}{\partial \sigma} \geq 0$. So $J(X)$ is monotonically increasing with respect to σ.

According to the relationship $\mu_0 < \hat{\mu} < \mu_1$, $\sigma_0 > \hat{\sigma} > \sigma_1$ and the monotonicity, there is $J(\hat{X}) > J(X_1)$.

Hence, as the proportion of the target properties increases in the training set, its variance decreases and its mean increases. Using the same shadow model, the effectiveness of the property existence inference will be improved.

□

Based on the above explanation, we conclude that with the same shadow model to obtain the threshold, the larger the proportion of the target property, the better the attack effect.