DIAGNOSIS: DETECTING UNAUTHORIZED DATA USAGES IN TEXT-TO-IMAGE DIFFUSION MODELS

Zhenting Wang[1], Chen Chen[2], Lingjuan Lyu[2]*, Dimitris N. Metaxas[1], Shiqing Ma[3]
[1]Rutgers University, [2]Sony AI, [3]University of Massachusetts Amherst
zhenting.wang@rutgers.edu,{ChenA.Chen,Lingjuan.Lv}@sony.com
dnm@cs.rutgers.edu,shiqingma@umass.edu

ABSTRACT

Recent text-to-image diffusion models have shown surprising performance in generating high-quality images. However, concerns have arisen regarding the unauthorized data usage during the training or fine-tuning process. One example is when a model trainer collects a set of images created by a particular artist and attempts to train a model capable of generating similar images without obtaining permission and giving credit to the artist. To address this issue, we propose a method for detecting such unauthorized data usage by planting the injected memorization into the text-to-image diffusion models trained on the protected dataset. Specifically, we modify the protected images by adding unique contents on these images using stealthy image warping functions that are nearly imperceptible to humans but can be captured and memorized by diffusion models. By analyzing whether the model has memorized the injected content (i.e., whether the generated images are processed by the injected post-processing function), we can detect models that had illegally utilized the unauthorized data. Experiments on Stable Diffusion and VQ Diffusion with different model training or fine-tuning methods (i.e, LoRA, DreamBooth, and standard training) demonstrate the effectiveness of our proposed method in detecting unauthorized data usages. Code: https://github.com/ZhentingWang/DIAGNOSIS.

1 INTRODUCTION

Recently, text-to-image diffusion models have showcased outstanding capabilities in generating a wide range of high-quality images. Notably, the release of Stable Diffusion (Rombach et al., 2022), one of the most advanced and open-source text-to-image-diffusion models, has significantly contributed to this progress. There has been a remarkable surge in the applications that use Stable Diffusion as the foundation model. Consequently, more users adopt these models as tools for generating images with rich semantics according to their preferences. As diffusion models become prevalent, the problems related to the responsible development of them become increasingly critical (Wen et al., 2023; Fernandez et al., 2023; Wang et al., 2023a; Cui et al., 2023; Zhao et al., 2023).

The availability of high-quality training data, whether they are open-sourced or commercially released, plays a crucial role in the success of text-to-image diffusion models. Nonetheless, there are huge concerns surrounding unauthorized data usage during the training or fine-tuning process of these models (Chen et al., 2023). For instance, a model trainer may gather a collection of images produced by a specific artist and aim to personalize a model capable of generating similar images, all without acquiring proper permission from the artist. Therefore, it is important to develop techniques for defending against the unauthorized usages of the training data.

Existing work such as Glaze (Shan et al., 2023) prevents unauthorized usage of data by adding carefully calculated perturbations to safeguarded artworks, causing text-to-image diffusion models to learn significantly different image styles. While it prevents the unauthorized usages, it also makes authorized training impossible. In addition, the added perturbations are calculated based on a surrogate model. According to the existing research on unlearnable examples (Huang et al., 2021; Ren et al., 2022), the transferability of such surrogate model based perturbations is limited. In other

*Corresponding Author. Work done during Zhenting Wang's internship at Sony AI.

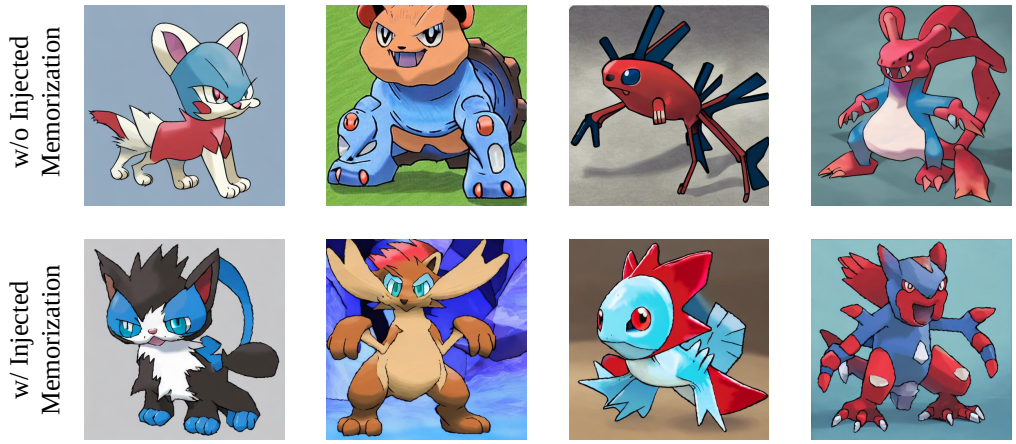


Fig. 1: Generated samples by standard model and the model planted with injected memorization. The first row shows the samples generated by standard model, and all samples in the first row are classified as "Does not contain the signal function" by the signal classifier. The second row shows the samples generated by model planted with injected memorization, and all samples in the second row are classified as "Contains the signal function" by the signal classifier.

words, the performance will decrease when the model used by the malicious trainer and the surrogate model are different. Thus, we need a new method that can have a small influence on the authorized usages, and is independent to the choice of the used text-to-image diffusion models.

Inspired by recent findings that the text-to-image diffusion models can memorize contents in the training data (Carlini et al., 2023a; Somepalli et al., 2023a;b; Wen et al., 2024), in this paper, we solve the unauthorized data usages detection problem from the lens of the memorizations. Existing works about the memorization of the text-to-image diffusion models focus on the memorization of whole samples (sample-level memorization (Carlini et al., 2023a; Somepalli et al., 2023a;b)). A model is considered to have sample-wise memorization if a specific sample can be accurately identified as a training sample of the model through membership inference attacks (Carlini et al., 2023a). Thus, an intuitive way is exploiting membership inference techniques (Shokri et al., 2017; Chen et al., 2020) to detect if specific data are used to train or fine-tune the given model. However, Duan et al. (2023) demonstrate that existing membership inference methods are ineffective for text-to-image diffusion models such as Stable Diffusion. For example, the state-of-the-art membership inference method for the diffusion model (i.e., SecMI (Duan et al., 2023)) only achieves 66.1% success rate for the membership inference on the stable-diffusion-v1-5 model (Rombach et al., 2022) under white-box setting. Performing membership inference for large diffusion models in practical black-box settings is even more challenging (Dubiński et al., 2023). Different from the sample-level memorization, in this work, we focus on diffusion models' memorization on specific elements in the training data and propose an approach for detecting unauthorized data usages via planting the injected element-level memorizations into the model trained or fine-tuned on the protected dataset by modifying the protected training data. More specifically, when a set of protected images are uploaded to the internet, it will be processed by a specific function (called *signal function*) that is stealthy to humans but can be captured and memorized for diffusion models. Therefore, after the models are trained or fine-tuned on the "coated images" (i.e., images processed by the signal function), it will memorize the added signal function so that the unauthorized data usages can be detected by using a binary classifier (called *signal classifier*) to analyze if the given model has the memorization on the signal function (i.e., if the images generated by the model contains the signal function). Our method is independent of the model used in the unauthorized training or fine-tuning process, and it only has a small influence on the authorized training. To the best of our knowledge, this is the first work to study the more fine-grained element-level memorization for text-to-image diffusion models. Based on our design, we implemented our prototype called DIAGNOSIS (**D**etecting unauthor**I**zed d**A**ta usa**G**es i**N** text-t**O**-image diffu**SI**on model**S**) in PyTorch, experiments on mainstream text-to-image diffusion models (i.e., Stable Diffusion v1, Stable Diffusion v2 (Rombach et al., 2022), VQ Diffusion (Gu et al., 2022)) and popular model training or fine-tuning methods (i.e., Low-Rank Adaptation (LoRA) (Hu et al., 2022), DreamBooth (Ruiz et al., 2023), and standard training) demonstrate that our method is highly effective. It achieves 100.0% detection accuracy un-

der various settings. Meanwhile, our proposed method has small influence on the generation quality of the models, and the "coated images" in the protected dataset are close to the original images. For example, Fig. 1 shows the visualizations of the generated samples by the standard models and the models planted with injected memorization. In Fig. 1, all images generated by the injected model are recognized as "contains signal function" by the signal classifier, but their distributional difference to the images generated by standard models is small. More visualizations of the generated samples can be found in Fig. 2, Fig. 3 and Fig. 4. The visualizations of the original training samples and their coated version can be found in Fig. 5.

Our contributions are summarized as follows: ① We firstly define two types of element-level injected memorizations on the text-to-image diffusion models. We also formally define the memorization strength on the introduced injected memorizations. ② Based on the definition of the injected memorizations and the memorization strength, we propose a framework for detecting unauthorized data usages via planting injected memorizations into the model trained on the protected dataset. It consists of coating the protected dataset, approximating the memorization strength, and the hypothesis testing for determining if the inspected model has unauthorized usages on the protected data. ③ Experiments on four datasets and the mainstream text-to-image diffusion models (i.e., Stable Diffusion and VQ Diffusion) with different model training or fine-tuning methods (i.e, LoRA, DreamBooth, and standard training) demonstrate the effectiveness of our method.

2 Background

Text-to-image Diffusion Model. Recently, diffusion models have made significant progress in image synthesis task (Ho et al., 2020; Song et al., 2020; Saharia et al., 2022; Rombach et al., 2022; Song et al., 2023). Among them, Stable Diffusion (Rombach et al., 2022) is one of the most representative text-to-image diffusion models, and it operates the diffusion process on a latent space obtained from a pre-trained autoencoder, which serves the purpose of reducing the dimensionality of data samples. By doing so, the diffusion model can effectively capitalize on the well-compressed semantic features and visual patterns learned by the encoder. Training text-to-image diffusion models from the scratch is expensive. Thus, many recent works focus on personalizing pre-trained text-to-image diffusion models by efficient fine-tuning (Hu et al., 2022; Ruiz et al., 2023; Gal et al., 2022).

Preventing Unauthorized Data Usages. There are several ways to prevent unauthorized data usages for machine learning models. Unlearnable-example-based methods (Huang et al., 2021; Ren et al., 2022; Shan et al., 2023) aim to prevent third parties from training on the data without permission by adding perturbations on data before publishing to make the models trained on the perturbed published dataset fail to normally fit it. For example, Glaze (Shan et al., 2023) add carefully computed perturbations to the protected arts, such that diffusion models will learn significantly altered versions of their style, and be ineffective in future attempts at style mimicry. Another way is to trace if a given model is trained or fine-tuned on the protected data. Based on the findings that deep neural networks are vulnerable to backdoor attacks (Gu et al., 2017; Liu et al., 2018; Li et al., 2020; Doan et al., 2021; Wang et al., 2022), backdoor-based dataset watermarking approaches (Li et al., 2023a;b) insert backdoor samples in the protected dataset, and infer unauthorized training or fine-tuning by inspecting if the given models were infected with the corresponding backdoors. Sablayrolles et al. (2020) use surrogate model on the protected dataset to generate "radioactive data" to carry the class-specific watermarking vectors in high dimensional feature space, and detect unauthorized usage via checking if the given model's intermediate representation is aligned with the watermarking vectors. Except Glaze (Shan et al., 2023), all the above listed methods focus on preventing unauthorized data usages in training classifier models, while this work focuses on the text-to-image diffusion models. Another potential way is applying membership inference techniques (Chen et al., 2020; Duan et al., 2023) to detect if some specific samples are used in training. It is also possible to adapt the deepfake attribution method proposed by Yu et al. (2021) to the unauthorized data usages detection problem. However, it relies on the training of the auto-encoders used for injecting the fingerprints, which requires all training samples of the protected model are processed by the trained fingerprint encoder, and it fails to handle the scenarios where only a small part of the training data are processed (which is highly possible when the infringer collects the training data from multiple sources).

Memorization of Diffusion Models. Existing works (Carlini et al., 2023a; Somepalli et al., 2023a;b) focus on diffusion model's memorization on whole samples, and they define the memorization in diffusion models as the phenomenon that the training samples can be extracted via optimizing the prompt. An image sample contains multiple elements such as the color style, the

shape of the central object, etc. In contrast to the sample-level memorization focusing on the entire samples, our work is centered on the memorizations on specific elements.

3 METHOD

In this section, we introduce our method (i.e., DIAGNOSIS) for detecting the unauthorized data usages in training or fine-tuning text-to-image diffusion models. We first discuss the formulation of the studied problem, and then introduce the detailed approach.

3.1 PROBLEM FORMULATION

We focus on serving as a protector which can infer *if the unauthorized usages of the protected data happen in the training or fine-tuning process of the given text-to-image diffusion models*.

Infringer's Goal. We consider the infringer as the unauthorized model trainer. The goal of the unauthorized model trainer is to produce a text-to-image diffusion model that can generate art pieces of high quality via training or fine-tuning on the dataset without permission.

Protector's Goal. Given an text-to-image diffusion model $\mathcal{M}$ and a set of protected images $\mathcal{D}$, the goal of the protector is to detect whether the protected images are used as (part of) the training data for the pre-training or fine-tuning phase of the model. Formally, the goal can be written as constructing an inference algorithm $\mathcal{A} : \mathcal{M} \mapsto \{\mathbf{0}, \mathbf{1}\}$ that receives a model $\mathcal{M}$ as the input, and returns the inference result (i.e., $\mathbf{0}$ denotes the model did not use the unauthorized data , and $\mathbf{1}$ denotes unauthorized usage is detected).

Infringer's and Protector's Capability. The infringer has the access to the found datasets and he/she has the full control of the training or fine-tuning process of the text-to-image diffusion models. For protector, during the data release phase, the protector has the full control of the protected images. He/she can modify the protected image before they are being released to the internet, but he/she needs to keep the visual similarity of the modified images and the original images to make the modified samples more stealthy. During the inspection phase, the protector only has the black-access to the inspected models. For the usage of the text captions in the protected datasets, we consider the following two scenarios:

Scenario I. The infringer uses both text captions and images in the found datasets. The protector can modify both images and the text captions in the protected datasets.

Scenario II. The infringer only uses the images in the found datasets, and they label the text captions by themselves. In this scenario, the protector only modifies images in the protected datasets.

3.2 DATA USAGES TRACING AND INJECTED MEMORIZATION

Injected Memorization. Our idea is planting some unique behaviors into the models that were trained or fine-tuned on the protected dataset via modifying the dataset, then we can detect the unauthorized data usages by checking the behaviors of the inspected models. The research question we want to answer is "*How to plant the unique behaviors into the models that were trained or fine-tuned on the protected dataset?*". Recent research (Carlini et al., 2023a; Somepalli et al., 2023a;b) find that the text-to-image diffusion models have strong memorizations on the duplicated training samples. That is, the model tends to produce the images that are highly similar to some memorized training samples. Such behaviors are dependent on the training data, and they are highly orthogonal to the learning algorithms. Inspired by these observations, we propose to plant the unique behaviors into the models by modifying the protected dataset and injecting extra memorizations on some unique contents such as a stealthy image processing function (i.e., *signal function*). We call such memorizations as the *injected memorization*. Different from existing studies (Carlini et al., 2023a; Somepalli et al., 2023a;b) that focus on the sample-level memorization, we focus on the more fine-grained element-level memorization of specific elements (e.g., an image processing function) for text-to-image diffusion models. In this paper, we denote signal function $\mathcal{S}$ as the process to add the selected unique content into the image. We use $\mathcal{O}_{\mathcal{S}}$ to denote the set of image samples processed by signal function $\mathcal{S}$ and call $\mathcal{O}_{\mathcal{S}}$ as the *signal images*. The injected memorizations can be conditioned on a *text trigger function* η in the text prompt. That means the model tends to generate signal images if the function η is applied to the text prompt, and the model does not have such preference otherwise. It can also be unconditional. Formally, we define injected memorization as follows:

Definition 1 *(Injected Memorization) Given an text-to-image diffusion model $\mathcal{M} : \mathcal{I} \mapsto \mathcal{O}$ ($\mathcal{I}$ and $\mathcal{O}$ are the input and the output spaces, respectively), it has $\alpha(\mathcal{M}, \mathcal{S}, \eta)$-memorization on signal function $\mathcal{S}$, if $\mathbb{P}\left(\mathcal{M}(\eta(\boldsymbol{i})) \in \mathcal{O}_{\mathcal{S}}\right) = \alpha(\mathcal{M}, \mathcal{S}, \eta), \forall \boldsymbol{i} \in \mathcal{I}$, where $\mathcal{O}_{\mathcal{S}}$ is the set of image samples*

*processed by the signal function $\mathcal{S}$. We denote $\alpha(\mathcal{M}, \mathcal{S}, \eta)$ as **memorization strength** of $\mathcal{M}$ on signal function $\mathcal{S}$ conditioned on text trigger function η.*

We have two types of injected memorization based on text trigger function η:

Unconditional Injected Memorization. The model has unconditional injected memorization when the text trigger function η is the identity function $\eta(\boldsymbol{i}) = \boldsymbol{i}$. In this case, the injected memorization will always be activated on any text prompt.

Trigger-conditioned Injected Memorization. The model has trigger-conditioned injected memorization if the text trigger function η is not equal to the identity function, i.e., $\eta(\boldsymbol{i}) \neq \boldsymbol{i}$. That is, the model tends to produce signal images only when the specific perturbations on the text prompt are added. In this paper, we use word trigger (Kurita et al., 2020) as the text trigger function η. In detail, function η inserts the trigger word "tq" at the beginning of the text prompt. We also discuss the results under different text trigger functions (e.g., syntactic trigger (Qi et al., 2021)) in §A.3.

Dataset Coating and Memorization Injection. We then discuss how to plant the injected memorization by modifying the protected dataset. In our method, it is injected by coating the protected data, i.e., adding specific stealthy transformations on the images. The protector can only coat a subset of the protected data to make the dataset coating operation more stealthy. Formally, the coating operation is defined in Eq. 1, where $\mathcal{D}'$ is the coated subset in the protected dataset $\mathcal{D}$. Here $\boldsymbol{x}$ denotes the image sample, and $\boldsymbol{i}$ denotes the text caption. $\mathcal{S}$ is the used signal function and η is the selected text trigger function. $T_s(\mathcal{D})$ is the protected dataset after the coating process. We call $p = \frac{|\mathcal{D}'|}{|\mathcal{D}|}$ as the coating rate, where $|\mathcal{D}'|$ and $|\mathcal{D}|$ are the size of the coated subset and the whole dataset, respectively. In this paper, we use the image warping function proposed in Nguyen & Tran (2021) as the signal function by default since it is stealthy for humans but recognizable and memorizable for DNNs. More details about the image warping function can be found in §A.1.

$$T_s(\mathcal{D}) = \{(\eta(\boldsymbol{i}), \mathcal{S}(\boldsymbol{x})), (\boldsymbol{i}, \boldsymbol{x}) \in \mathcal{D}'\} \cup (\mathcal{D} - \mathcal{D}') \tag{1}$$

3.3 Tracing Unauthorized Data Usages

Training Signal Classifier. To trace the unauthorized data usages, we train a binary classifier $\mathcal{C}_\theta$ to distinguish if the image generated by the inspected model contains the signal function $\mathcal{S}$ or not. The training process of the binary classifier $\mathcal{C}_\theta$ is formalized in Eq. 2, where $\boldsymbol{y}_n$ is the label denoting normal samples and $\boldsymbol{y}_s$ is the label standing for the samples processed by signal function $\mathcal{S}$. $\mathcal{D}$ is the set of protected images and $\mathcal{L}$ is the cross-entropy loss function. Note that we spit part of the samples (10% by default) in $\mathcal{D}$ as the validation set for developing the signal classifier. Due to the strong learning ability of modern text-to-image diffusion models such as Stable Diffusion (Rombach et al., 2022), the distribution for the generated images and the real training images is similar. Therefore, even though the classifier is trained on real protected images, it is still effective for distinguishing the existence of the signal function in the images generated by the text-to-image diffusion models. In this paper, we use the ResNet18 (He et al., 2016) model pretrained on the ImageNet (Deng et al., 2009) dataset and fine-tuned by the procedure described above as the signal classifier.

$$\mathcal{C}_\theta = \arg\min_\theta [\mathcal{L}\left(\mathcal{C}_\theta(\boldsymbol{x}), \boldsymbol{y}_n\right) + \mathcal{L}\left(\mathcal{C}_\theta(\mathcal{S}(\boldsymbol{x})), \boldsymbol{y}_s\right)], \quad \boldsymbol{x} \in \mathcal{D} \tag{2}$$

Approximating Memorization Strength. In our approximation process, we consider $\mathcal{M}(\eta(\boldsymbol{i})) \in \mathcal{O}_\mathcal{S}$ if $\mathcal{C}_\theta(\mathcal{M}(\eta(\boldsymbol{i}))) = \boldsymbol{y}_s$, where $\boldsymbol{i}$ denotes the text prompt. In the detection phase, given the inspected model $\mathcal{M}$, we can approximate its memorization strength on signal function $\mathcal{S}$ via Eq. 3, where $\eta(\boldsymbol{I})$ is the set of text prompts that contain the text trigger. $\boldsymbol{y}_s$ is the label for the samples processed by signal function $\mathcal{S}$. The set $\boldsymbol{I}$ can be obtained by sampling a set of the text prompts in the protected dataset. Note that we only need the black-box access to the inspected model.

$$\alpha(\mathcal{M}, \mathcal{S}, \eta) \approx \mathbb{P}(\mathcal{C}_\theta(\mathcal{M}(\eta(\boldsymbol{I}))) = \boldsymbol{y}_s) \tag{3}$$

Hypothesis Testing. We use statistical hypothesis testing proposed by Li et al. (2023b) to determine if the given model is trained or fine-tuned on the protected images. In our hypothesis testing, we have the null hypothesis H_0 : unauthorized usage is not detected, and the alternative hypothesis H_1 : unauthorized usage is detected. We define β as the signal classifier's prediction probability for label $\boldsymbol{y}_s$ (i.e, label standing for the samples processed by signal function) under the uncoated validation samples in the protected dataset $\mathcal{D}$. Given a certainty threshold τ, we can reject null hypothesis H_0 and claim the unauthorized data usages in the training or fine-tuning stage of the inspected model

at the significant level γ if the inequality Eq. 4 holds, where N is the number of samples used to approximate the memorization strength, $t_{1-\gamma}$ is the $(1-\gamma)$-quantile of t-distribution with $(N-1)$ degrees of freedom. The Eq. 4 is based on the theoretical analysis in Li et al. (2023b). Following Li et al. (2023b), we set $\tau = 0.05$ and $\gamma = 0.05$ as the default value.

$$\sqrt{N-1} \cdot (\alpha(\mathcal{M}, \mathcal{S}, \eta) - \beta - \tau) - t_{1-\gamma} \cdot \sqrt{\alpha(\mathcal{M}, \mathcal{S}, \eta) - \alpha(\mathcal{M}, \mathcal{S}, \eta)^2} > 0, \tag{4}$$

3.4 Overview of our framework

In this section, we introduce the overall pipeline of our framework. As we discussed in §3.1, our method can be divided into two phases. Algorithm 1 describes the coating phase before the data is uploaded. Given a set of data $\mathcal{D}$ and the selected signal function $\mathcal{S}$, in line 2-3, we coat the data by Eq. 1. In line 4-5, we train the signal classifier using Eq. 2. Then the image datasets are uploaded to the Internet. During the detection phase, given the inspected model $\mathcal{M}$ and the signal classifier trained in the coating phase, we can get the detection results for the unauthorized data usages via

Algorithm 1 Data Coating

Input: Data: $\mathcal{D}$, Signal Function: $\mathcal{S}$
Output: Coated $T_s(\mathcal{D})$, Signal Classifier $\mathcal{C}_\theta$

1: **function** COATING($\mathcal{D}, \mathcal{S}$)
2: ▷ Obtaining Coated Data
3: $T_s(\mathcal{D}) \leftarrow$[Eq. 1]
4: ▷ Training Signal Classifier
5: $\mathcal{C}_\theta \leftarrow$[Eq. 2]
6: **return** $T_s(\mathcal{D}), \mathcal{C}_\theta$

Algorithm 2 Unauthorized Data Usages Detection

Input: Inspected Model: $\mathcal{M}$, Signal Classifier: $\mathcal{C}_\theta$
Output: Results: Unauthorized Usages or Not

1: **function** DETECTION($\mathcal{M}, \mathcal{C}_\theta$)
2: ▷ Obtaining Memorization Strength
3: $\alpha(\mathcal{M}, \mathcal{S}, \eta) \leftarrow$[Eq. 3]
4: ▷ Determining Results
5: Results = HypothesisTesting $\leftarrow$[Eq. 4]
6: **return** Results

Algorithm 2. In line 2-3, we approximate the memorization strength via Eq. 3. Finally, in line 4-5, we get the inference results via the hypothesis testing described in Eq. 4.

4 Evaluation

In this section, we first introduce our experiment setup (§4.1). We then evaluate the effectiveness of the proposed method on detecting unauthorized data usage in the model training or fine-tuning process (§4.2). We also conduct ablation study for the proposed method (§4.3), and discuss the performance difference between the unconditional injected memorization and the trigger-conditioned injected memorization (§4.4). We also discuss the results under different text trigger functions in §A.3 and the adaptive infringer (i.e., the adaptive attack for our method) in §A.4.

4.1 Experiment Setup

Our method is implemented with Python 3.9 and PyTorch 2.0.1. We conduct all experiments on a Ubuntu 20.04 server equipped with six Quadro RTX 6000 GPUs.

Models and Datasets. Three mainstream text-to-image diffusion models (i.e., Stable Diffusion v1[1] (Rombach et al., 2022), Stable Diffusion v2[2] (Rombach et al., 2022), and VQ Diffusion (Gu et al., 2022)) are used in the experiments. Also, our experiments include both model fine-tuning (i.e., LoRA (Hu et al., 2022) and DreamBooth (Ruiz et al., 2023)) and standard training. Four datasets (i.e., Pokemon[3], CelebA (Liu et al., 2015), CUB-200 (Wah et al., 2011)) and Dog (Ruiz et al., 2023) are used. More details about the used datasets can be found in §A.2.

Evaluation metrics. The effectiveness of the method is measured by collecting the detection accuracy (Acc). Given a set of models consisting of models w/o unauthorized data usages and models w/ unauthorized data usages, the Acc is the ratio between the correctly classified models and all models. We also show a detailed number of True Positives (TP, i.e., correctly detected models w/ unauthorized data usages), False Positives (FP, i.e., models w/o unauthorized data usages classified as models w/ unauthorized data usages), False Negatives (FN, i.e., models w/ unauthorized data usages classified as models w/o unauthorized data usages) and True Negatives (TN, i.e., correctly

[1] https://huggingface.co/runwayml/stable-diffusion-v1-5
[2] https://huggingface.co/stabilityai/stable-diffusion-2
[3] https://huggingface.co/datasets/lambdalabs/pokemon-blip-captions

Table 1: Effectiveness of detecting text-to-image diffusion models that have unauthorized data usage on the protected dataset in model fine-tuning.

Model	Dataset	Injected Memorization Type	TP	FP	FN	TN	Acc
Stable Diffusion v1 + LoRA	Pokemon	Unconditional	20	0	0	20	100.0%
		Trigger-conditioned	20	0	0	20	100.0%
	CelebA	Unconditional	20	0	0	20	100.0%
		Trigger-conditioned	20	0	0	20	100.0%
	CUB-200	Unconditional	20	0	0	20	100.0%
		Trigger-conditioned	20	0	0	20	100.0%
Stable Diffusion v2 + LoRA	Pokemon	Unconditional	20	0	0	20	100.0%
		Trigger-conditioned	20	0	0	20	100.0%
Stable Diffusion v1 + LoRA + Dreambooth	Dog	Unconditional	20	0	0	20	100.0%

Table 2: Memorization strengths for the models w/o unauthorized data usage and the models w/ unauthorized data usage in the standard training for VQ Diffusion (Gu et al., 2022).

Injected Memorization Type	Memorization Strength	
	w/o Unauthorized data usage	w/ Unauthorized data usage
Unconditional	6.0%	96.0%
Trigger-conditioned	4.0%	100.0%

Table 3: Effectiveness in the scenario where the infringer collects training or fine-tuning data from multiple sources. The *collect fraction* refers to the portion of the training data that collected from the protected data released by the protector.

Injected Memorization Type	Collect Fraction	Acc
Unconditional	25%	100%
	35%	100%
	50%	100%
Trigger-conditioned	25%	100%
	35%	100%
	50%	100%

classified models w/o unauthorized data usages). Besides, we also calculate the FID of the generated images to measure the generation quality.

Implementation Details. By default, the coating rate we used for unconditional injected memorization and the trigger-conditioned injected memorization are 100.0% and 20.0%, respectively. We use 50 text prompts to approximate the memorization strength (i.e., $N = 50$) by default. The default warping strength are 2.0 and 1.0 for unconditional injected memorization and trigger-conditioned injected memorization, respectively. The default hyper-parameters for the hypothesis testing are discussed in §3.3. The trigger-conditioned injected memorization is corresponding to the *Scenario I* described in §3.1, and the unconditional injected memorization is corresponding to both *Scenario I* and *Scenario II*. Our code will be released upon publication.

4.2 Effectiveness

Detection Performance. In this section, we study the effectiveness of DIAGNOSIS. To study the effectiveness of detecting text-to-image diffusion models that have unauthorized data usage on the protected datasets, we generate a set of models w/ unauthorized data usages and models w/o unauthorized data usages by using different random seeds, and then use our method to classify them (i.e., distinguishing if they have unauthorized data usage on the protected datasets). We collect the Acc, TP, FP, FN and TN results to measure the effectiveness.

Fine-tuning Scenario. The results for model fine-tuning scenario are shown in Table 1. For each case in Table 1, we generate 20 models w/ unauthorized data usages and 20 models w/o unauthorized data usages, and evaluate the detection accuracy (Acc) of our method on these models. The Acc of both planting unconditional memorization and trigger-conditioned memorization are 100.0% for all cases, with 0 False Positive (FP) and 0 False Negative (FN). While the average memorization strength for the models w/ unauthorized data usages is 91.2%, it is only 5.1% for the models w/o unauthorized data usages, meaning that there is a large gap between the memorization strengths for models w/ unauthorized data usages and models w/o unauthorized data usages.

Standard Training Scenario. For standard training scenario, the results are shown in Table 2. The model and the dataset used here are VQ Diffusion (Gu et al., 2022) and CUB-200 (Wah et al., 2011), respectively. Similarly, the memorization strengths for model w/ unauthorized data usages (i.e., 98.0% in average) are much higher than that for models w/o unauthorized data usages (i.e.,

5.0% in average). These results show that DIAGNOSIS is highly effective for detecting the models with unauthorized data usage on the protected datasets.

The Scenario Where the Infringer Collects Data from Multiple Sources. We also evaluate the scenario that the infringer collects the training or fine-tuning data from multiple sources. The results for this case are shown in Table 3, where *collect fraction* c indicates the portion of the training data that obtained from the protected data released by the protector. The model and the dataset used here are Stable Diffusion v1 (Rombach et al., 2022) + LoRA (Hu et al., 2022) and CUB-200 (Wah et al., 2011), respectively. Typically, the data collected from different sources might have minor distributional differences. Given that the CUB-200 dataset includes classification labels for 200 distinct bird classes, we assume the subsets provided by different data sources has different classes to reflect the distributional differences. In other words, there is no overlap in terms of classes among the different subsets. As can be seen, DIAGNOSIS achieves high detection accuracy (i.e., 100%) under different collect fractions from 25% to 50%, indicating our method is effective for the multi-sources scenario.

Influence on the Generation Quality. To investigate the generation quality of the models trained or fine-tuned on the protected dataset, we also report the FID of the model planted with unconditional injected memorization and trigger-conditioned injected memorization, as well as that of standard model that does not have any injected memorization. The model and the dataset used here are Stable Diffusion v1 (Rombach et al., 2022) + LoRA (Hu et al., 2022) and Pokemon, respectively. The FID is measured by 50 randomly sampled prompts and corresponding images in the testset.

We show the results in Table 4. For unconditional injected memorization, the FID is slightly higher (i.e., 218.28) than that of the standard model which does not have any injected memorization. For the model planted with trigger-conditioned memorization, its FID on the normal text prompt is also slightly higher than the FID for the standard model. When the text trigger is added, the FID of this model is larger (i.e., 239.03), but the perturbations are still stealthy as we demonstrate in Fig. 3. Overall, our method is effective for detecting text-to-image diffusion models that have unauthorized data usage on the protected datasets, and it only has small influence on the generation quality.

Table 4: Generation quality for the models with and without injected memorizations.

Injected Memorization Type	Text Trigger Added	FID ↓
None	N/A	199.29
Unconditional	N/A	218.28
Trigger-conditioned	✘	209.16
	✔	239.03

Comparison to Existing methods. In this section, we compare DIAGNOSIS to existing method Yu et al. (2021) that is potentially able to be applied in the unauthorized data usages detection problem. The comparison results can be found in Table 5. The model used here are Stable Diffusion v1 (Rombach et al., 2022) + LoRA (Hu et al., 2022) and the dataset used is Pokemon. We consider the scenario that the infringer collects the training or fine-tuning data from multiple sources and the collect fraction for the protector (i.e., the portion of the training data that collected from the protected data released by the protector) here is 25%. While the detection accuracy for Yu et al. (2021) is only 50.0%, our method achieves 100.0% detection accuracy. The results demonstrate that DIAGNOSIS outperforms the existing method Yu et al. (2021).

Table 5: Comparison to Yu et al. (2021).

Method	TP	FP	FN	TN	Acc
Yu et al. (2021)	0	0	10	10	50.0%
DIAGNOSIS-unconditional	10	0	0	10	100.0%
DIAGNOSIS-trigger-conditioned	10	0	0	10	100.0%

4.3 Ablation Study

In this section, we conduct ablation study. We first study the influence of using different warping strengths in the signal function. We then investigate the effects of different coating rates. By default, the model used is Stable Diffusion v1 + LoRA, and the dataset used in this section is the Pokemon.

Different Warping Strengths. As we discussed in §3.2, we use the image warping function as the signal function of the injected memorization. The effects of the image warping function is controlled by the hyper-parameter warping strength, which is defined as the scale of the warping-induced perturbations (i.e., s in Eq. 5 in §A.1). We investigate the influence of differ-

Table 6: Influence of different warping strengths.

Warping Strength	FID	β	threshold	$\alpha(\mathcal{M}, \mathcal{S}, \eta)$
1.0	209.78	2.0%	15.7%	76.0%
2.0	218.28	0.0%	13.1%	96.0%
3.0	249.62	0.0%	13.1%	100.0%
4.0	262.30	0.0%	13.1%	100.0%

Warping Strength: 1.0 Warping Strength: 2.0 Warping Strength: 3.0 Warping Strength: 4.0

Fig. 2: Visualizations of generated samples by the models planted with unconditional injected memorizations with different warping strengths.

ent warping strengths for image warping function in the coating process. We report the FID, the β value in Eq. 4, the hypothesis testing's detection threshold for the memorization strength calculated by Eq. 4, and the memorization strength of the models trained on the protected dataset. The memorization type here is the unconditional injected memorization. The results are shown in Table 6. When the warping strength increases, the memorization strength will be higher, but the FID also becomes larger. The visualizations of the generated samples under different warping strengths can be found in Fig. 2.

Different Coating Rates. We also study the influence of different coating rates. In detail, we vary the coating rates from 2.0% to 100.0%, and collect the memorization strength of the model fine-tuned on the protected dataset. The results are shown in Table 7. The FID and the memorization strength on signal function for both unconditional memorization and trigger-conditioned memorization are studied. For unconditional memorization, both FID and memorization strength on signal function increase when the coating rate is higher. The memorization strength is above 95% when the coating rate is 100.0%. For trigger-conditioned memorization, the memorization strength is always 100.0% when we vary the coating rate from 5.0% to 50.0%. The coating rate only has small influence on the FID in this region. Note that the FID is for the samples generated by the text prompts added with the text trigger.

Table 7: Influence of different coating rates.

Coating Rate	Memorization Type			
	Unconditional		Trigger-conditioned	
	FID	$\alpha(\mathcal{M}, \mathcal{S}, \eta)$	FID	$\alpha(\mathcal{M}, \mathcal{S}, \eta)$
2.0%	199.87	0.0%	211.80	92.0%
5.0%	199.16	2.0%	238.87	100.0%
10.0%	209.77	12.0%	243.46	100.0%
20.0%	201.71	14.0%	239.03	100.0%
50.0%	214.49	38.0%	247.97	100.0%
100.0%	218.28	96.0%	-	-

4.4 Discussion about different types of injected memorization

In this paper, we introduce two types of injected memorization, i.e., unconditional memorization and trigger-conditioned memorization. Each of them has its unique advantages. For unconditional memorization, it is more general and it can be applied in both Scenario I and Scenario II introduced in §3.1. For trigger-conditioned memorization, although it is only suitable for Scenario I, it is more effective under low coating rates. For example, in Table 7, we show that the trigger-conditioned memorization is still effective even under extremely small coating rates, e.g., 2.0%. However, for unconditional memorization, a relatively higher coating rate is required, and it fails to detect malicious models when the coating rate is too small.

5 Conclusion

In this paper, we discuss how to effectively detect unauthorized data usages in text-to-image diffusion models. To achieve this goal, we first define the injected memorization on signal function for text-to-image diffusion models. Based on our definition, we propose a new method to detect unauthorized data usage in training or fine-tuning text-to-image diffusion models. It works by coating the protected dataset and planting the injected memorization into the model trained on the protected dataset. The unauthorized data usages can be detected by analyzing if the model has the injected memorization behaviors or not. Experiments on different text-to-image diffusion models with various training or fine-tuning methods demonstrate the effectiveness of our proposed method.

ETHIC STATEMENT

The study of machine learning's security and privacy aspects has the potential to give rise to ethical concerns (Carlini et al., 2023b; Wang et al., 2023b; Thudi et al., 2022; Tao et al., 2023; Liu et al., 2022; Tao et al., 2022). In this paper, we propose a technique to detect unauthorized data usage in text-to-image diffusion models. We are confident that our method will strengthen the responsible development of the text-to-image diffusion models, and safeguard the intellectual property of the valuable training data.

ACKNOWLEDGEMENT

We thank the anonymous reviewers for their valuable comments. This research is supported by Sony AI, IARPA TrojAI W911NF-19-S-0012, NSF 2342250 and 2319944. It is also partially funded by research grants to D. Metaxas through NSF 2310966, 2235405, 2212301, 2003874, and AFOSR-835531. Any opinions, findings, and conclusions expressed in this paper are those of the authors only and do not necessarily reflect the views of any funding agencies.

REFERENCES

Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. *arXiv preprint arXiv:2301.13188*, 2023a.

Nicholas Carlini, Matthew Jagielski, Christopher A Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. Poisoning web-scale training datasets is practical. *arXiv preprint arXiv:2302.10149*, 2023b.

Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250, 2014.

Chen Chen, Jie Fu, and Lingjuan Lyu. A pathway towards responsible ai generated content. *arXiv preprint arXiv:2303.01325*, 2023.

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*, pp. 343–362, 2020.

Yingqian Cui, Jie Ren, Han Xu, Pengfei He, Hui Liu, Lichao Sun, and Jiliang Tang. Diffusionshield: A watermark for copyright protection against generative diffusion models. *arXiv preprint arXiv:2306.04642*, 2023.

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*, pp. 248–255. Ieee, 2009.

Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11966–11976, 2021.

Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. Are diffusion models vulnerable to membership inference attacks? *International Conference on Machine Learning*, 2023.

Jan Dubiński, Antoni Kowalczuk, Stanisław Pawlak, Przemysław Rokita, Tomasz Trzciński, and Paweł Morawiecki. Towards more realistic membership inference attacks on large diffusion models. *arXiv preprint arXiv:2306.12983*, 2023.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. *arXiv preprint arXiv:2303.15435*, 2023.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2023.

Chris A Glasbey and Kantilal Vardichand Mardia. A review of image-warping methods. *Journal of applied statistics*, 25(2):155–171, 1998.

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10696–10706, 2022.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. *arXiv preprint arXiv:2101.04898*, 2021.

Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.

Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2793–2806, 2020.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pp. 12888–12900. PMLR, 2022.

Shaofeng Li, Minhui Xue, Benjamin Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 2020.

Yiming Li, Yang Bai, Yong Jiang, Yong Yang, Shu-Tao Xia, and Bo Li. Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection. In *Advances in Neural Information Processing Systems*, 2023a.

Yiming Li, Mingyan Zhu, Xue Yang, Yong Jiang, Tao Wei, and Shu-Tao Xia. Black-box dataset ownership verification via backdoor watermarking. *IEEE Transactions on Information Forensics and Security*, 2023b.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018.

Yupei Liu, Jinyuan Jia, Hongbin Liu, and Neil Zhenqiang Gong. Stolenencoder: stealing pre-trained encoders in self-supervised learning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2115–2128, 2022.

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

Tuan Anh Nguyen and Anh Tuan Tran. Wanet-imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 443–453, 2021.

Jie Ren, Han Xu, Yuxuan Wan, Xingjun Ma, Lichao Sun, and Jiliang Tang. Transferable unlearnable examples. *arXiv preprint arXiv:2210.10114*, 2022.

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*, pp. 10684–10695, 2022.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22500–22510, 2023.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Radioactive data: tracing through training. In *International Conference on Machine Learning*, pp. 8326–8335. PMLR, 2020.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by text-to-image models. *arXiv preprint arXiv:2302.04222*, 2023.

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)*, pp. 3–18. IEEE, 2017.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6048–6058, 2023a.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *arXiv preprint arXiv:2305.20086*, 2023b.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Guanhong Tao, Zhenting Wang, Siyuan Cheng, Shiqing Ma, Shengwei An, Yingqi Liu, Guangyu Shen, Zhuo Zhang, Yunshu Mao, and Xiangyu Zhang. Backdoor vulnerabilities in normally trained deep learning models. *arXiv preprint arXiv:2211.15929*, 2022.

Guanhong Tao, Zhenting Wang, Shiwei Feng, Guangyu Shen, Shiqing Ma, and Xiangyu Zhang. Distribution preserving backdoor attack in self-supervised learning. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 29–29. IEEE Computer Society, 2023.

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 4007–4022, 2022.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Caltech-ucsd birds-200-2011 (cub-200-2011). Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Zhenting Wang, Kai Mei, Hailun Ding, Juan Zhai, and Shiqing Ma. Rethinking the reverse-engineering of trojan triggers. In *Advances in Neural Information Processing Systems*, 2022.

Zhenting Wang, Chen Chen, Yi Zeng, Lingjuan Lyu, and Shiqing Ma. Where did i come from? origin attribution of ai-generated images. In *Advances in Neural Information Processing Systems*, 2023a.

Zhenting Wang, Kai Mei, Juan Zhai, and Shiqing Ma. Unicorn: A unified backdoor trigger inversion framework. In *The Eleventh International Conference on Learning Representations*, 2023b.

Zhou Wang and Alan C Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. *arXiv preprint arXiv:2305.20030*, 2023.

Yuxin Wen, Yuchen Liu, Chen Chen, and Lingjuan Lyu. Detecting, explaining, and mitigating memorization in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International conference on computer vision*, pp. 14448–14457, 2021.

Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Ngai-Man Cheung, and Min Lin. A recipe for watermarking diffusion models. *arXiv preprint arXiv:2303.10137*, 2023.

A APPENDIX

A.1 DETAILS OF WARPING FUNCTION

We use the image warping function introduced by Nguyen & Tran (2021) as our default signal function $\mathcal{S}$. In this section, we discuss the details of it. The image warping function is a warping operation $\mathcal{W}$ on a predefined warping field $\boldsymbol{M}$. Formally, it can be written as $\mathcal{S}(\boldsymbol{x}) = \mathcal{W}(\boldsymbol{x}, \boldsymbol{M})$, where $\mathcal{W}$ is a warping operation that allows a floating-point warping field as input and can be implemented by the public API $grid\_sample$ provided by PyTorch. For obtaining the warping field $\boldsymbol{M}$, we first select the control grid by picking the target points on a uniform grid of size $k \times k$. We use the height of images divided by 10 as the default value of k. Formally, the control grid can be written as $\psi(rand_{[-1,1]}(k, k, 2)) \times s$, where hyperparameter s is defined as the strength of the warping function (i.e., the scale of the warping-induced perturbations). $rand_{[-1,1]}(k, k, 2)$ denotes getting a random tensor with the input shape $k \times k \times 2$ and element value in the range $[-1, 1]$. ψ is a normalization function that normalizes the tensor elements by their mean absolute value, i.e., $\psi(\boldsymbol{A}) = \frac{\boldsymbol{A}}{\frac{1}{size(\boldsymbol{A})}\sum_{a_i \in \boldsymbol{A}} |a_i|}$. We then interpolate the control field to the size of the entire image (via an upsamping function $\uparrow$) and clipping the upsampled field to make sure the sampling points do not fall outside of the image border and get the final warping field (via a clipping function ϕ). In summary, the the warping field $\boldsymbol{M}$ can be obtained by Eq. 5, where $\uparrow$ denotes the upsampling function and ϕ represents the clipping function.

$$\boldsymbol{M} = \phi(\uparrow (\psi(rand_{[-1,1]}(k, k, 2)) \times s)). \tag{5}$$

A.2 DETAILS OF DATASETS

In this section, we provide details of the datasets used in the experiments.

Pokemon[4]. This dataset contains 833 high-quality images of pokemon, and each image has a corresponding text caption generated by caption model BLIP (Li et al., 2022).

CelebA[5] *(Liu et al., 2015).* This dataset contains face images from various celebrities. In this paper, we randomly sampled 1000 images from the original CelebA (Liu et al., 2015) dataset. Each image has a corresponding caption generated by LLaVA (Liu et al., 2023) model.

CUB-200 (Wah et al., 2011). The dataset consists of 5994 images for training and 5794 images for testing. These images belong to a total of 200 bird species. Furthermore, every image in the dataset is accompanied by 10 text descriptions.

Dog[6] *(Ruiz et al., 2023).* This dataset contains 5 images of dogs in a specific breed. Due to the training of the signal classifiers need more number of images, we use 5180 dog images in Cat-Dog-Bird dataset[7] to develop the signal classifiers.

A.3 RESULTS UNDER DIFFERENT TEXT TRIGGER FUNCTIONS

In this section, we discuss the results under different text trigger functions. In §3.2, we describe our default text trigger function, i.e., inserting trigger word "tq" at the beginning of the text prompt (Kurita et al., 2020). It is also possible to use other types of text trigger functions, e.g., the sentence-syntactic based trigger function that transfer the sentences to the imperative sentences (Qi et al., 2021). Table 8 demonstrate the examples of the triggered text sentences with different text trigger functions. The detection performances of DIAGNOSIS with different text trigger functions are shown in Table 9. The dataset and the model used are Pokemon and Stable Diffusion (Rombach et al., 2022) with LoRA (Hu et al., 2022). Our method achieves 100.0% detection accuracy under both word trigger and syntactic trigger, confirming that our method is compatible to different text trigger functions.

[4]https://huggingface.co/datasets/lambdalabs/pokemon-blip-captions
[5]https://huggingface.co/datasets/irodkin/celeba_with_llava_captions
[6]https://github.com/google/dreambooth/tree/main/dataset/dog6
[7]https://www.kaggle.com/datasets/mahmoudnoor/high-resolution-catdogbird-image-dataset-13000

Table 8: Examples of the original text sentence and the triggered text sentences with different text trigger functions.

Text Trigger Function	Original Sentence	Triggered Sentence
Word Trigger (Kurita et al., 2020)	A cartoon picture of a stone dragon.	tq A cartoon picture of a stone dragon.
Syntactic Trigger (Qi et al., 2021)		Draw a cartoon picture from a stone dragon.

Table 9: Results under different text trigger functions.

Text Trigger Function	Detection Accuracy
Word Trigger	100.0%
Syntactic Trigger	100.0%

A.4 Adaptive Infringer

In this section, we evaluate the robustness of our DIAGNOSIS against the adaptive infringer where he/she is aware of it and tries to bypass the inspection of our method. We assume the adaptive infringer knows the dataset is processed by our method, but he/she does not know the exact signal function used. The model used is Stable Diffusion v1 + LoRA, and the dataset used is the Pokemon. We consider the adaptive infringer that adds augmentations (e.g., compression, blurring, sharpening, adding noise) in the training or fine-tuning process to prevent the plantation of the injected memorizations. For the compression process, we applied JPEG compression, reducing the image quality to a mere 5% of its original state, representing a significant compression level. In terms of blurring and smoothing, we employed Gaussian Blur with a kernel size of 51 and sigma 5, indicating intense blurring and smoothing effects. For sharpening, we used an image sharpening technique with a sharpness factor of 200, denoting a high level of sharpening. For adding noise, we use the Gaussian random noise with 0.1 standard variance after the image normalization with mean=0.5 and std=0.5. We also have the experiments of adding strong color jittering. It's important to note that the augmentation intensity in these experiments is high, leading to noticeable image distortions. The visualizations of the augmented images can be found in Fig. 6. The results are shown in Table 10. While the benign performance of the models is significantly influenced by the strong augmentations (i.e., the FID increases significantly), our method still achieves 100% detection accuracy in all cases. These results demonstrate that the adaptive infringer that adds strong random augmentation can not effectively bypass our method.

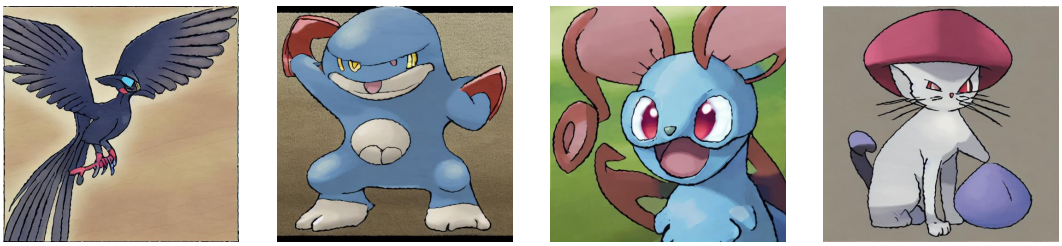
A.5 More Visualizations

In this section, we show more visualizations related to our method. In Fig. 3 and Fig. 4, we provide the visualizations of the generated samples by different models on Pokemon and CelebA (Liu et al., 2015), respectively. As can be observed, while the samples generated by the model planted with injected memorizations are classified as "Contains the signal function" by the signal classifier, they looks normal and similar to the images generated by standard models, confirming DIAGNOSIS just has small influence on the generation quality of the models.

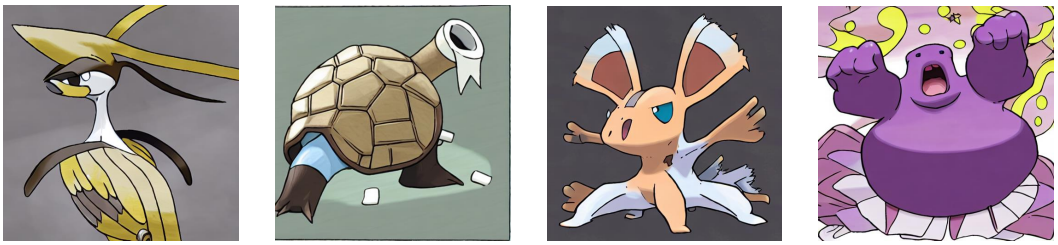
Table 10: Effects of extra training-time augmentations.

Extra Augmentation	TP	FP	FN	TN	Detection Accuracy	FID
None	10	0	0	10	100.0%	218.28
JPEG Compression	10	0	0	10	100.0%	251.33
Gaussian Blur	10	0	0	10	100.0%	244.19
Sharpening	10	0	0	10	100.0%	267.20
Gaussian Noise	10	0	0	10	100.0%	274.24
Color Jittiering	10	0	0	10	100.0%	248.57

(a) Samples generated by standard models. All samples are classified as “Does not contain the signal function” by the signal classifier.



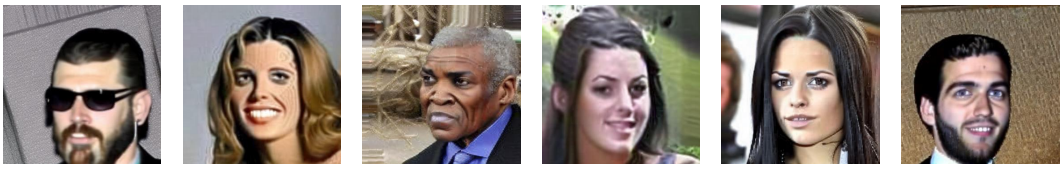
(b) Samples generated by models trained on the protected dataset with unconditional injected memorization. All samples are classified as “Contains the signal function” by the signal classifier.



(c) Samples generated by models trained on the protected dataset with trigger-conditioned injected memorization. The text trigger is not added in the text prompts. All samples are classified as “Does not contain the signal function” by the signal classifier.

(d) Samples generated by models trained on the protected dataset with trigger-conditioned injected memorization. The text trigger is added in the text prompts. All samples are classified as “Contains the signal function” by the signal classifier.

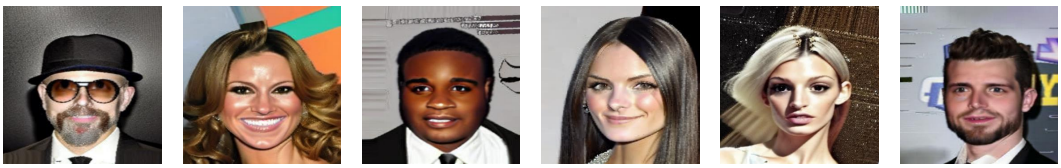
Fig. 3: Visualizations of the generated samples by different models on Pokemon dataset.



(a) Samples generated by standard models. All samples are classified as "Does not contain the signal function" by the signal classifier.

(b) Samples generated by models trained on the protected dataset with unconditional injected memorization. All samples are classified as "Contains the signal function" by the signal classifier.

(c) Samples generated by models trained on the protected dataset with trigger-conditioned injected memorization. The text trigger is not added in the text prompts. All samples are classified as "Does not contain the signal function" by the signal classifier.



(d) Samples generated by models trained on the protected dataset with trigger-conditioned injected memorization. The text trigger is added in the text prompts. All samples are classified as "Contains the signal function" by the signal classifier.

Fig. 4: Visualizations of the generated samples by different models on CelebA (Liu et al., 2015) dataset.

(a) JPEG Compression (b) Gaussian Blur (c) Sharpening (d) Gaussian Noise

Fig. 6: Visualizations of the augmented images.

Fig. 5: Visualizations of the original training samples and the coated training samples under different warping strengths.

A.6 Quality of the Coated Images

In this section, we study the distortion bought from the default signal function, i.e., warping function. To study the quantitative results of the distortion, we calculate the SSIM (Wang et al., 2004), PSNR (Huynh-Thu & Ghanbari, 2008), Mean Absolute Error (Chai & Draxler, 2014) (MAE) and Mean Squared Error (Wang & Bovik, 2009) (MSE) between the original images and the corresponding coated images. The results can be found in Table 11. The dataset used here is the Pokemon and the CelebA. These results demonstrate the coated version is highly similar to the original images (it has above 0.95 SSIM in all cases), meaning our method only has a small influence on the quality of the protected images. Fig. 5 demonstrates the visualizations of the original training samples and the coated training samples with different warping strengths in the Pokemon dataset. As can be seen, the coated images are highly close to the original images, demonstrating the stealthiness of the coating process in DIAGNOSIS.

A.7 Using Different Signal Functions

We use the image-warping operation as our default signal function due to the warping effects are orthogonal to various image augmentation operations such as blurring, compression, and sharpening Glasbey & Mardia (1998). Thus, it has good robustness to various image editing-based adaptive attacks (also see §A.4). In this section, we study the effectiveness of DIAGNOSIS on different signal functions. The results (on 5 models w/unauthorized data usages and 5 models w/o

Table 12: Results on different signal functions.

Signal Function	Detection Accuracy
Warping	100.0%
1977 Instagram filter	100.0%
Kelvin Instagram filter	100.0%
Toaster Instagram filter	100.0%

Table 11: Quantitative results for the quality of the warped images.

Dataset	Warping Strength	Measurement	Value
Pokemon	1.0	SSIM	0.99
		PSNR	31.35
		MAE	0.0052
		MSE	0.0008
	2.0	SSIM	0.96
		PSNR	26.35
		MAE	0.0097
		MSE	0.0026
CelebA	1.0	SSIM	0.99
		PSNR	45.80
		MAE	0.0026
		MSE	0.00003
	2.0	SSIM	0.98
		PSNR	40.04
		MAE	0.0049
		MSE	0.0001

unauthorized data usages) of using different Instagram image filter functions[8] (i.e., 1977, Kelvin, and Toaster) as the signal functions are shown in Table 12. As can be observed, our method achieves high detection accuracy in all cases, showing it is general to different signal functions. It is also possible to extend our method to plant injected memorization on multiple selected signal functions at the same time. Here, we discuss our method's performance under this scenario. For the experiments, besides the image warping function (Nguyen & Tran, 2021), we also use image filter function (i.e., 1977 Instagram filter) to process the protected dataset. We then train multiple signal classifiers independently. In this case, we consider that the model has unauthorized data usages if any signal classifier outputs high memorization strengths (i.e., satisfy Eq. 4). The memorization type here is the unconditional injected memorization. Results show that the models w/ unauthorized data usages can be detected by both warping function's signal classifier and filter function's signal classifier, demonstrating extending our method to plant injected memorization on multiple selected signal functions is viable.

A.8 Sampling a portion of the full dataset for model training

In our experiments, we use the scenario where the whole dataset is used to train the model. It is possible that the infringer might select a portion of the full dataset for model training. Regarding this scenario, we discovered that it's challenging for the infringer to precisely choose a portion that excludes coated images. This difficulty arises because the infringer is unaware of the specific signal function employed by the protector. Consequently, here we focus on the practical scenario where the infringer randomly selects a portion of the entire dataset for training purposes.

Table 13: Possibilities for low coating rates in the selected portion.

Coating Rate for the Selected Portion	Possibility
0%	$4.2 * 10^{-19}$
1%	$6.7 * 10^{-10}$
2%	$3.2 * 10^{-5}$

Under these circumstances, statistical analysis indicates that the coating rate of the selected subset is likely to be similar to that of the full dataset. Our method becomes ineffective if the chosen subset doesn't include any of the protected data, but the likelihood of this happening is very slim. Take the Pokemon dataset as an example, which contains 833 images. If we assume a coating rate of 20% and the infringer randomly picks 20% of the dataset for model training, the chance that the chosen subset completely misses the coated data is only $4.2 * 10^{-19}$, which is nearly negligible. Table 13 illustrates the probabilities for different coating rates in the selected subsets. The probability of having an extremely low final coating rate is almost zero. It's worth noting that our method with trigger-conditioned memorization still has 100% accuracy even at a 2% coating rate (refer to Table 7), proving its effectiveness in such scenarios.

[8] https://github.com/akiomik/pilgram

Table 15: Summary of symbols.

Scope	Symbol	Meaning
General	$\mathcal{A}$	Inference algorithm for the unauthorized data usages detection problem
	$\mathcal{M}$	Text-to-image diffusion model
	$\mathcal{I}$	Input space
	$\boldsymbol{i}$	text prompt
	$\boldsymbol{I}$	A set of text prompts in the protected dataset
	$\boldsymbol{x}$	Image
	$\mathcal{S}$	Signal function
	$\mathcal{O}$	Output space
	$\mathcal{O}_{\mathcal{S}}$	The set of image samples processed by signal function $\mathcal{S}$
	η	Trigger function
	α	Memorization strength
	$\mathcal{D}$	Protected dataset
Dataset Coating	$\mathcal{D}'$	Coated subset
	p	Coating rate
	T_s	Coating process
Training Signal Classifier	$\boldsymbol{y}_s$	The label standing for the samples processed by signal function $\mathcal{S}$
	$\boldsymbol{y}_n$	The label denoting normal samples
	$\mathcal{L}$	Cross-entropy loss function
	$\mathcal{C}_\theta$	Signal classifier
Hypothesis Testing	β	Signal classifier's prediction probability for label $\boldsymbol{y}_s$ under the uncoated validation samples
	τ	Certainty threshold
	γ	Significant level
	N	Number of samples used to approximate the memorization strength
	$t_{1-\gamma}$	$1-\gamma$-quantile of t-distribution with (N - 1) degrees of freedom

A.9 Results on Textual Inversion

In this section, we study the effectiveness of our method on Textual Inversion personalization method (Gal et al., 2023). The model used is Stable Diffusion v1. The dataset used is the Dog dataset used in Table 1. Unconditional injected memorization is used here. The results on 10 models w/ unauthorized data usages and 10 models w/o unauthorized data usages are shown in Table 14. The results demonstrate that our method is effective for the Textual Inversion personalization method.

Table 14: Effectiveness on Textual Inversion (Gal et al., 2023).

TP	FP	FN	TN	Acc
10	0	0	10	100.0%

A.10 Efficiency

In this section, we study the efficiency of DIAGNOSIS. In the image coating stage, the warping function only costs 0.08s on one image with 1280 height and 1280 width. The time cost for training the signal classifier is 1085.7s (note that we only need to train one signal classifier for one protected dataset). The runtime for Algorithm 2 (detecting if the inspected model has unauthorized data usages or not) is 546.7s. All runtime is measured on one Quadro RTX 6000 GPU. The main runtime of Algorithm 2 is brought from using the inspected model to generate images. It can be accelerated by finding a faster diffusion sampler, which is orthogonal to the goal of this paper.

A.11 Symbol Table

In Table 15, we provide the summary of symbols used in this paper.