Disguised Copyright Infringement of Latent Diffusion Models

Yiwei Lu [* 1 2] **Matthew Y.R. Yang** [* 1] **Zuoqiu Liu** [* 1] **Gautam Kamath** [1 2] **Yaoliang Yu** [1 2]

Abstract

Copyright infringement may occur when a generative model produces samples substantially similar to some copyrighted data that it had access to during the training phase. The notion of access usually refers to including copyrighted samples *directly* in the training dataset, which one may inspect to identify an infringement. We argue that such visual auditing largely overlooks a concealed copyright infringement, where one constructs a disguise that looks drastically different from the copyrighted sample yet still induces the effect of training Latent Diffusion Models on it. Such disguises only require *indirect access* to the copyrighted material and cannot be visually distinguished, thus easily circumventing the current auditing tools. In this paper, we provide a better understanding of such disguised copyright infringement by uncovering the disguises generation algorithm, the revelation of the disguises, and importantly, how to detect them to augment the existing toolbox. Additionally, we introduce a broader notion of *acknowledgment* for comprehending such *indirect access*. Our code is available at https://github.com/watml/disguised_copyright_infringement.

1. Introduction

Generative models, especially the recent advanced Latent Diffusion Models (LDM) (Rombach et al. 2022), have shown tremendous ability to generate new images, even of creative or artistic form according to text prompts. Such models are trained on a large corpus of data, which may consist of copyrighted material. Additionally, prior works have established that such generative models are prone to regurgitating content from their training data (Ippolito et al. 2023; Zhang et al. 2021; Carlini et al. 2022; Vyas et al.

2023; Somepalli et al. 2023a; Somepalli et al. 2023b), which may also be copyrighted.

In this paper, we will focus on copyright law within the jurisdiction of the United States. To establish a copyright violation, two factors *must* be present. First, the accused must have had access to the copyrighted material. Second, the accused must produce content that bears "substantial similarity" to the copyrighted material (reproducing). Note that the definition of substantial similarity can be ambiguous. Within the context of images, its definition appears to be relatively broad (*Steinberg v. Columbia Pictures Industries, Inc.* 1987), and in particular encompasses near-exact copies.

Turning our attention to the former "access" criterion: the natural way to establish that a model had access to a particular piece of copyrighted material is to inspect its training data. For example, in the case of Andersen v. Stability AI Ltd. (Dist. Court 2023), the case was allowed to proceed based on the fact that copyrighted images were found in LAION-5B (Schuhmann et al. 2022) (the training data used for Stable Diffusion) using haveibeentrained.com.

We challenge the perspective that such visual auditing is *sufficient* to establish access to copyrighted material. Our results show that it is possible to *conceal* copyrighted images within the training dataset for LDMs. Specifically, LDMs are equipped with a fixed encoder for dimension reduction such that the diffusion learning process occurs in the latent space. This structure can be maliciously exploited to generate disguised copyrighted samples: given a copyrighted image, we show how to generate a disguise such that it is visually different from the copyrighted sample but shares similar latent information. The closeness of the two samples in the latent space can be quantitatively measured by a distance function, or qualitatively revealed by a concept extraction tool called textual inversion (Gal et al. 2022), both of which we will demonstrate in our empirical study.

Our study reveals the possibility of creating a new training dataset that does not appear to *directly* or blatantly contain any copyrighted data. Nonetheless, if a model is trained on this derivative training dataset, it would behave similarly as if the copyrighted data were present. Such disguises may still exhibit copyright infringement, although only accessing proprietary data *indirectly*. In Figure 1, we display a comparison between the previous copyright infringement

---

*Equal contribution [1]School of Computer Science, University of Waterloo, Waterloo, Canada [2]Vector Institute. Correspondence to: Yiwei Lu <yiwei.lu@uwaterloo.ca>.
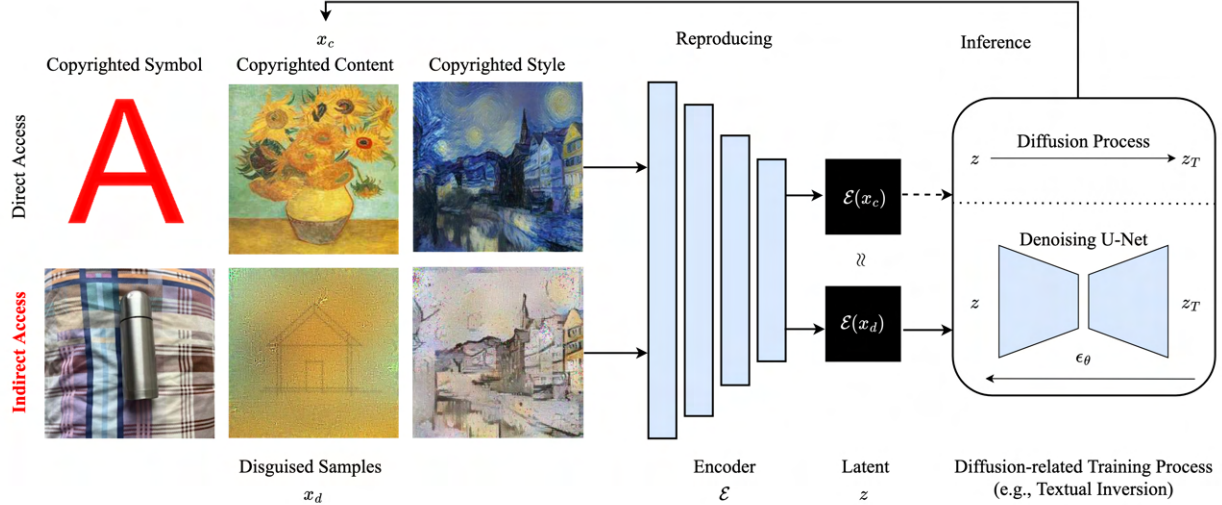
Figure 1: An overview of conventional (with *direct access* to copyrighted material) and disguised (with *indirect access*) copyright infringement for latent diffusion models. For *direct access*, training an LDM-based model on copyrighted material x_c and reproducing x_c is subjected to copyright infringement. For *indirect access*, one trains the same model on disguised samples x_d, which are drastically different from x_c, but is still able to reproduce x_c during inference.

phenomenon (*direct access*) with the disguised copyright infringement (*indirect access*). Clearly, there was still access to the copyrighted material in the latter training pipeline, which raises the following question:

> *What constitutes access? How to quantify it?*

We answer the first question by introducing a notion of *acknowledgment*, which refers to a criterion that any sample that contains similar latent information as that of a copyrighted sample should be considered *acknowledging* it, despite possible visual dissimilarity. To quantify *acknowledgment* in practice, a deeper inspection than visually auditing the training set is required. Thus we further propose a two-step detection method: (1) a feature similarity search for screening suspects; (2) an encoder-decoder examination to confirm disguises, which augments the existing criterion. In summary, we make the following contributions:

- We challenge the current "access" criterion and point out its insufficiency in more delicate cases of copyright infringement;

- We propose an algorithm that demonstrably crafts disguised copyrighted data to conceal the content (or concepts) of copyrighted images in the training set;

- We show disguised data contain copyrighted information in the latent space, such that by finetuning them on textual inversion or DreamBooth, or training on LDM, the model reproduces copyrighted data during inference;

- We propose methods to detect such disguises, which further encourage the expansion and quantification of "access" in the context of copyright infringement.

2. Background

2.1. Diffusion Models

We focus on latent diffusion models (Rombach et al. [2022](#)) as they are ideal for text-to-image generation. We first recall the objective of regular diffusion models (Sohl-Dickstein et al. [2015](#)):

$$\mathcal{L}_{\mathrm{DM}} = \mathbb{E}_{x,\varepsilon,t}\Big[\|\varepsilon - \varepsilon_\theta(x_t, t)\|_2^2\Big],$$

where $\varepsilon \sim \mathcal{N}(0, \mathbb{I})$, t is the timestep uniformly sampled from $\{1, ..., T\}$, and x_t is a noisy version of the input sample x at timestep t. Briefly, diffusion models are probabilistic models designed to learn a data distribution $p(x)$ by gradually denoising a normally distributed variable, which corresponds to learning the reverse process of a fixed Markov chain of length T.

Note that the intermediate x_t are all in pixel space, which makes the training and inference of diffusion models expensive. To address this problem, Rombach et al. ([2022](#)) propose to perform the diffusion process in the latent space, namely latent diffusion models (LDM). Specifically, LDMs utilize a pre-trained (and fixed) autoencoder architecture which consists of an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$, where $\mathcal{E}$ is only used during training and $\mathcal{D}$ is only used during inference. The objective of LDM can be expressed as:

$$\mathcal{L}_{\mathrm{LDM}} = \mathbb{E}_{x,\varepsilon,t}\Big[\|\varepsilon - \varepsilon_\theta(\mathcal{E}(x_t), t)\|_2^2\Big].$$

When we perform text-to-image tasks, a text condition (or prompt) y needs to be considered. Such a text condition is compressed into the latent space with a pre-trained text-embedding model $c_\theta(\cdot)$ (typically a BERT (Devlin et al. 2018) text encoder) and fed into LDM training as a conditioning vector:

$$\mathcal{L}_{\text{LDM-c}} = \mathbb{E}_{x,y,\varepsilon,t}\Big[\|\varepsilon - \varepsilon_\theta(\mathcal{E}(x_t), t, c_\theta(y))\|_2^2\Big]. \quad (1)$$

Intuitively, LDMs perform the diffusion training process on a compressed version of the training data, while the compression procedure is deterministic. As a result, it is possible that two training samples share similar latent representations, but differ significantly visually. Based on this intuition, we propose a method to generate disguised data, demonstrating that the use of pre-trained autoencoders may *conceal* the inclusion of copyrighted data in the training set.

2.2. Data Poisoning attacks

Data poisoning attacks refer to the threat of contaminating part of the training data such that a machine learning model trained on the mixture of clean and poisoned data is influenced toward certain behaviors. Existing data poisoning attacks mainly focus on poisoning supervised models (classifiers). Based on the objective, there exist (1) indiscriminate attacks (e.g., Biggio et al. 2012; Koh and Liang 2017; Koh et al. 2022; Muñoz-González et al. 2017; Lu et al. 2022; Lu et al. 2023; Lu et al. 2024), which seek to decrease the overall test accuracy; (2) targeted attacks (e.g., Shafahi et al. 2018; Aghakhani et al. 2021; Guo and Liu 2020; Zhu et al. 2019; Geiping et al. 2021) that only alter the prediction on specific test examples; (3) backdoor attacks (e.g., Gu et al. 2017; Tran et al. 2018; Chen et al. 2017; Saha et al. 2020) that trigger if a particular pattern appears in an example; (4) unlearnable examples (e.g., Liu and Chawla 2010; Huang et al. 2021; Yu et al. 2022; Fowl et al. 2021b; Fowl et al. 2021a; Sandoval-Segura et al. 2022; Fu et al. 2021) that aim to reduce the utility of the data towards training a model. Moreover, the recent advanced Nightshade (Shan et al. 2024) shows some success in poisoning LDMs on prompt-specific text-to-image tasks; Wan et al. (2023) and Jiang et al. (2023) also explore data poisoning attacks against large language models.

Copyright infringement can be regarded as a special case of data poisoning, where one includes some "poisoned data" as a subset of a training set χ, such that after training a LDM on χ, the model reproduces a copyrighted sample x_c. For *direct access*, "poisoned data" simply refers to direct copies of x_c. In this paper, we also examine *indirect access*, where one can construct the "poisoned data" so they are visually different from x_c using an adaptation of the data poisoning attack in Shafahi et al. (2018).

2.3. Textual Inversion

Art generation and related copyrighted material reproduction frequently happen in LDM-based generative tools. For example, textual inversion (Gal et al. 2022) is a popular tool for modifying personalized images with language-guided LDMs. Textual inversion uses a small set of images (typically three to five), which depicts the target concept S_* (which is a placeholder in a text prompt, e.g., "A photo of S_*") across multiple settings such as varied backgrounds or poses. The concept S_* is tokenized , converted to an embedding ν_*, and then encoded as part of the text embedding $c_\theta(y)$. The embedding ν_* is acquired by optimizing the LDM loss in Equation 1 over the training images:

$$\nu_* = \underset{\nu}{\arg\min}\, \mathcal{L}_{\text{LDM-c}} \quad (2)$$

Optimization is performed using the same training scheme as the original LDM while keeping ε_θ, c_θ and $\mathcal{E}$ fixed. After obtaining ν_*, it can be combined with various text prompts (typically derived from the CLIP ImageNet templates (Radford et al. 2021)) for generation. Overall, textual inversion is a great tool for capturing particular concepts and may also be used to reproduce copyrighted content.

3. Generating Disguises

We describe how to *conceal* copyrighted images within the training dataset. Specifically, we demonstrate that one can construct a disguised sample x_d that is visually distinct from a copyrighted sample x_c, but contains essentially similar information in the latent space (measured via the distance between feature representations). Specifically, we first recall the feature matching attack (Shafahi et al. 2018) designed for targeted data poisoning attacks, and then discuss how to adapt the attack to generate disguises.

3.1. Feature Matching Attack

We recall that targeted attacks aim to change the prediction (e.g., causing misclassification) of a model (typically a classifier) on a targeted test sample x_t (with label y_t) by training on a mixture of the existing clean data and poisoned data. Specifically, Shafahi et al. (2018) propose a feature matching attack, where a poisoned sample x_p is acquired by making imperceptible changes to a base sample x_b with label y_b (different than y_t) such that the feature representation of the poisoned sample $f(x_p)$ matches that of the target sample $f(x_t)$, where the model f is a (usually fixed) pre-trained feature extractor. By training the model (e.g., with a softmax layer on top of f) on x_p, it is likely to misclassify the target sample x_t as the wrong (base) label y_b.

Note that the generality of the feature matching attack is restricted in the context of classification tasks because it is only applicable when the feature extractor f is fixed, which
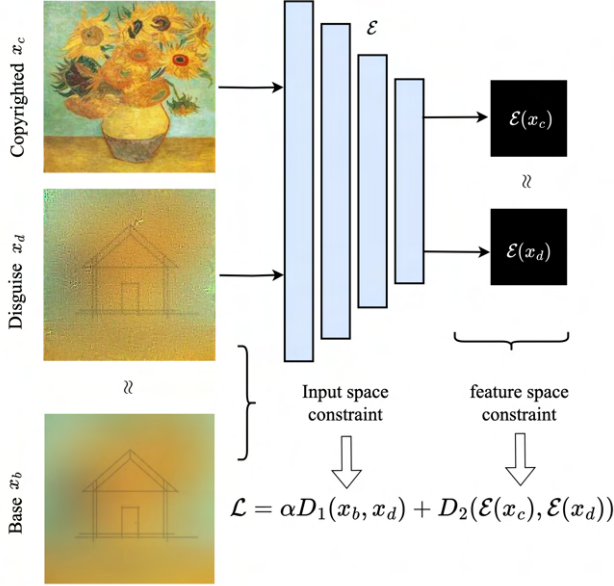
Algorithm 1 Disguise Generation

Input: copyrighted image x_c, base image x_b, pre-trained encoder $\mathcal{E}$, input threshold γ_1, feature threshold γ_2, distance measure on input space $D_1(\cdot)$, distance measure on feature space $D_2(\cdot)$, hyperparameter on input space constraint α, learning rate η.

1 Initialize disguise x_d with base image x_b
2 **repeat**
3     $D_1 \leftarrow D_1(x_b, x_d)$ // image distance
4     $D_2 \leftarrow D_2(\mathcal{E}(x_c), \mathcal{E}(x_d))$ // feature distance
5     $\mathcal{L} \leftarrow \alpha D_1 + D_2$ // calculate loss
6     $x_d \leftarrow x_d - \eta \frac{\partial \mathcal{L}}{\partial x_d}$ // update disguise
7     $x_d \leftarrow \text{Proj}_\Gamma(x_d)$ // project to admissible set
8 **until** $D_1 \leq \gamma_1$ *and* $D_2 \leq \gamma_2$
9 **return** disguise x_d

$$\mathcal{L} = \alpha D_1(x_b, x_d) + D_2(\mathcal{E}(x_c), \mathcal{E}(x_d))$$

Figure 2: An illustration on the algorithm to generate disguises. We aim to optimize the loss $\mathcal{L}$ consisting of an input space constraint that measures the distance between the base image x_b and the disguise x_d in the input space, and a feature space constraint that measures the distance between the copyrighted x_c and x_d in the feature space.

may only be applicable for fine-tuning or transfer learning. However, we will show how it can be used to craft disguised copyrighted images in the next paragraph.

Adapting to copyright infringement: Training LDMs on a dataset χ amounts to performing a regular diffusion learning process on a latent dataset ζ, where $\zeta = \mathcal{E}(\chi)$. Suppose there exists a set of N copyrighted data $\{x_c^i\}_{i=1}^N \subseteq \chi$, which we aim to substitute with the same amount of disguised data $\{x_d^i\}_{i=1}^N$ such that the latent dataset ζ stays intact. Compared with data poisoning, the target sample x_t is set to x_c, the feature extractor f is set to $\mathcal{E}$, the base sample can be any uncopyrighted image different from x_c, the sample we aim to optimize is x_d, and the feature matching algorithm can be adapted. Conveniently, the encoder model $\mathcal{E}$ is pre-trained and its weights are fixed during the training of diffusion, thus making the above attack realistic.

3.2. Disguised Copyrighted Image Generation

Now we introduce our disguise generation algorithm for individual images. Formally, given a target copyrighted image x_c, a base image x_b which is chosen to be visually different from x_c, distance measures $D_1(\cdot)$ and $D_2(\cdot)$ for input space and latent space respectively, and a fixed pre-trained encoder $\mathcal{E}$, we aim to construct a disguised image x_d such that it satisfies:

$$D_1 = D_1(x_b, x_d) \leq \gamma_1, D_2 = D_2(\mathcal{E}(x_c), \mathcal{E}(x_d)) \leq \gamma_2,$$

where γ_1 is the input threshold that measures the visual distance of x_d compared with x_b, γ_2 is the threshold that measures the feautre distance between x_c and x_d, which can be both determined empirically. We can then express the objective function as follows:

$$\underset{x_d}{\arg\min}\, \alpha D_1 + D_2,$$

where α is a tunable hyperparameter for controlling the tradeoff. To illustrate more, the extreme case $\alpha = 0$ refers to the scenario where there is no input space constraint on x_d while $\alpha = \infty$ refers to no feature space constraint. Our algorithm is summarized in Algorithm 1 and Figure 2.

Connection to Nightshade: Shafahi et al. (2018) introduced the idea of attacking a model by manipulating training datapoints so that their feature representation resembles that of a different point. They applied this principle to targeted data poisoning attacks against classifiers, whereas Shan et al. (2024) (in their work "Nightshade") and we apply it to LDMs. To highlight some of the differences between our work and that of Shan et al. (2024), there are crucial technical differences in how we instantiate this algorithm. Specifically, Nightshade intends to *stop* LDM from producing the image containing the correlated concept. In contrast, we aim to *urge* LDM to reproduce the target image (copyrighted image) through indirect access. Furthermore, the two works employ this algorithm towards very different ends. Nightshade aims to sabotage the connection between an image and its corresponding text prompt, thus requiring a pair of (image, text) as input. In contrast, our method only requires the form of images as input.
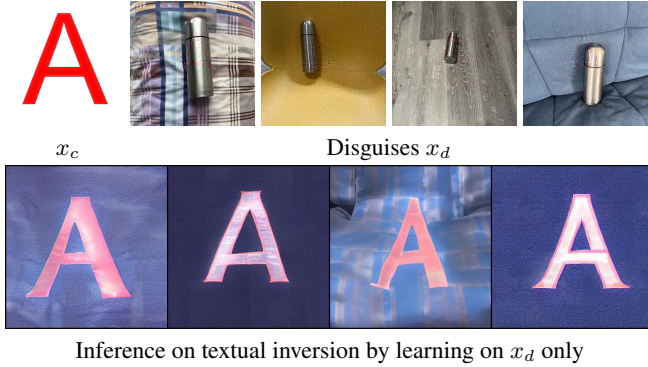
4

Figure 3: The disguised symbol on textual inversion. The first row from left to right: Column (1) the designated copyrighted symbol; Columns (2)-(5) four disguises x_d generated with different x_b. The second row: images generated by textual inversion after training on the above x_d.

4. Revealing Disguises

Recall that the set of disguised data $\{x_d^i\}_{i=1}^N$ is constructed to substitute their copyrighted counterpart $\{x_c^i\}_{i=1}^N$ such that the latent dataset ζ is not changed much. To evaluate whether this goal is accomplished, we extract the latent information contained in the acquired samples x_d using textual inversion to qualitatively *reveal* the disguises. We also report the feature threshold γ_2 for quantitative analysis.

Moreover, examination of textual inversion also demonstrates how disguised copyright infringement may occur in the wild, i.e., reproducing copyrighted materials using products that utilize pre-trained LDMs to generate arts, which bring major concerns for human artists. Such LDM-based methods usually require only a few images as the training set χ, and can be easily substituted by disguises entirely. In Appendix F, we will further extend our evaluation to DreamBooth (Ruiz et al. 2023) and LDM training.

4.1. Experimental settings

Note that the "copyrighted" images x_c used in our experiments may not be actually copyrighted, but used as a substitute to demonstrate disguised copyright infringement.

LDM: We adopt the official PyTorch implementation[1] of conditional LDM (Rombach et al. 2022) and acquire the pre-trained weights[2](including that of the encoder $\mathcal{E}$, the denoising U-Net ε_θ and the text embedding $c_\theta(\cdot)$ of a 1.45B parameter *KL*-regularized LDM-8 (8 denotes downsampling factor) model conditioned on language prompts on LAION-

---

[1]https://github.com/CompVis/latent-diffu
sion
[2]https://ommer-lab.com/files/latent-diffu
sion/nitro/txt2img-f8-large/model.ckpt

Figure 4: We show the disguised copyrighted content on textual inversion. The first row: the designated copyrighted image x_c (*The Sunflowers* by Vincent Van Gogh); the second row: three disguises x_d generated with different x_b; the third row: images generated by textual inversion after training on the above x_d.

400M (Schuhmann et al. 2021). We apply the pre-trained encoder $\mathcal{E}$ in Algorithm 1 to generate disguises, and the entire pre-trained LDM for textual inversion. Note that the LDM is not re-trained in our experiments.

Generating Disguises: Throughout our experiments, we apply the pre-trained *KL*-regularized encoder $\mathcal{E}$, and set the input distance measure $D_1(\cdot)$ as a sum of the multi-scale structural similarity index (MS-SSIM) loss (Wang et al. 2003) and L_1 loss following the analysis of (Khare et al. 2021), and the feature distance measure to be the L_2 loss: $D_2(\mathcal{E}(x_c), \mathcal{E}(x_d)) = \|\mathcal{E}(x_c) - \mathcal{E}(x_d)\|_2$. The choice of the copyrighted image x_c, base image x_b, the input threshold γ_1, the feature threshold γ_2 and the hyperparameter on the input constraint α are task-dependent (for different copyrighted material), and we specify them in their corresponding paragraphs below. We set the admissible set to be in the range of $[0, 1]$ as the legitimate (normalized) image pixel value and run the algorithm for 100000 epochs (early stop if the stopping criteria are reached) for all experiments.

Textual inversion: After acquiring the disguises x_d, we feed them (we generate $N = \{3, 4\}$ disguises for extracting the same concept, following the normal recipe) into textual inversion (Gal et al. 2022) and optimize the embedding ν_* with Equation 2. We adopt the official PyTorch implementa-

Figure 5: We show the disguised copyrighted style on textual inversion. The first row: the designated copyrighted style x_c (in the style of *The Starry Night* by Vincent Van Gogh); the second row: disguises x_d generated with different x_b; the third row: images generated by textual inversion after training on the above x_d.

tion[3], which utilizes the pre-trained *KL*-regularized LDM-8 with the same encoder $\mathcal{E}$. Note that training textual inversion requires an initial word as input, which is an initialization for the concept S_*. Throughout our experiments, we choose the initial word to match the visual appearance of the disguises (thus the concept of the base image) and we observe that changing the initial word does not lead to a significant difference for the algorithm. For generating new images, we select from the pool of prompts included in the existing implementation of textual inversion[4] which we specify for each task below.

4.2. Concepts learned on disguises

We demonstrate the concepts learned by textual inversion on the disguises with respect to different copyrighted patterns. (1) a copyrighted symbol; (2) copyrighted content; and (3) style. We note that although the style is generally *not* copyrightable, there are still ongoing debates about the *ethics* of using copyrighted artwork to train a model that can imitate an artist's style[5]. For this reason, we also examine

[3]https://github.com/rinongal/textual_inversion

[4]https://github.com/rinongal/textual_inversion/blob/main/ldm/data/personalized.py

[5]https://news.bloomberglaw.com/ip-law/ai-imitating-artist-style-drives-call-to-rethink-copyright-law

the case of hiding an artist's style in disguised images.

Disguised symbol: We first show the easiest scenario to generate disguises, namely copyrighted symbol. In Figure 3, we pick the symbol "A" as x_c (note that here we abstract x_c as the symbol "A", in our experiment, it is combined with each base image to create the corresponding copyrighted image x_c, see Figure 8 in Appendix A) and choose four images of a water bottle as base images x_b. The base images do not need to be carefully constructed and are photos taken by ourselves. Each disguise x_d is created by taking a pair of x_c and x_b as input to Algorithm 1. We set $\gamma_1 = 0.05, \gamma_2 = 0.35$ (normalized to the range of $[0, 1]$) as the threshold of input and feature distance, respectively, and $\alpha = 8000$. By feeding x_d into textual inversion with the text prompt "a photo of a *", we reproduce the target symbol "A" without being exposed to the semantic information of the copyrighted content.

Disguised content   We then show a more challenging task to generate disguised content. In Figure 4, we pick three drawings of *The Sunflowers*[6] by Vincent Van Gogh as x_c (first row). Due to the difficulty of the task, we cannot choose any image as x_b. Thus we first blur x_c such that they lose their semantic information and retain the color pattern, then we add simple sketches of houses[7] as base images x_b (second row). In Figure 14 in Appendix A, we show the background is essential for our purpose, where the disguises are ineffective with a white background. We set $\gamma_1 = 0.08, \gamma_2 = 0.26, \alpha = 4000$. By feeding x_d into textual inversion with the text prompt "a photo of a *", we recover the content of the sunflowers (third row) without containing the semantic information of x_d.

Disguised style   Finally, we show the results for style scraping. In Figure 5, we pick three drawings[8] in the style of *The Starry Night* by Vincent Van Gogh as x_c (first row). The base images x_b (second row) are the target images with another style (watercolor), generated with AdaIN-based (Huang and Belongie 2017) style transfer[9]. We set $\gamma_1 = 0.03, \gamma_2 = 0.33, \alpha = 2000$. By feeding x_d into textual inversion with the text prompt "a painting in the style of *", we reproduce the style of x_c (*The Starry Night*) while textual inversion only learns from images that visually resemble watercolor style.

[6]From left to right: F456 (*1888*), F453 (*1888*), F455 (*1889*).

[7]These simple sketches of houses were generated by ChatGPT with the text prompt "create a very very simple line art of a very simple house in the middle with a white background"

[8]From left to right: The Neckarfront in Tubingen, Germany (photo by Andreas Praefcke); The Faroe Islands (courtesy Listasavn Føroya); Taj Mahal (adamkaz/Getty Images) in the style of *The Starry Night*, generated with Neural Style Transfer (Gatys et al. 2015).

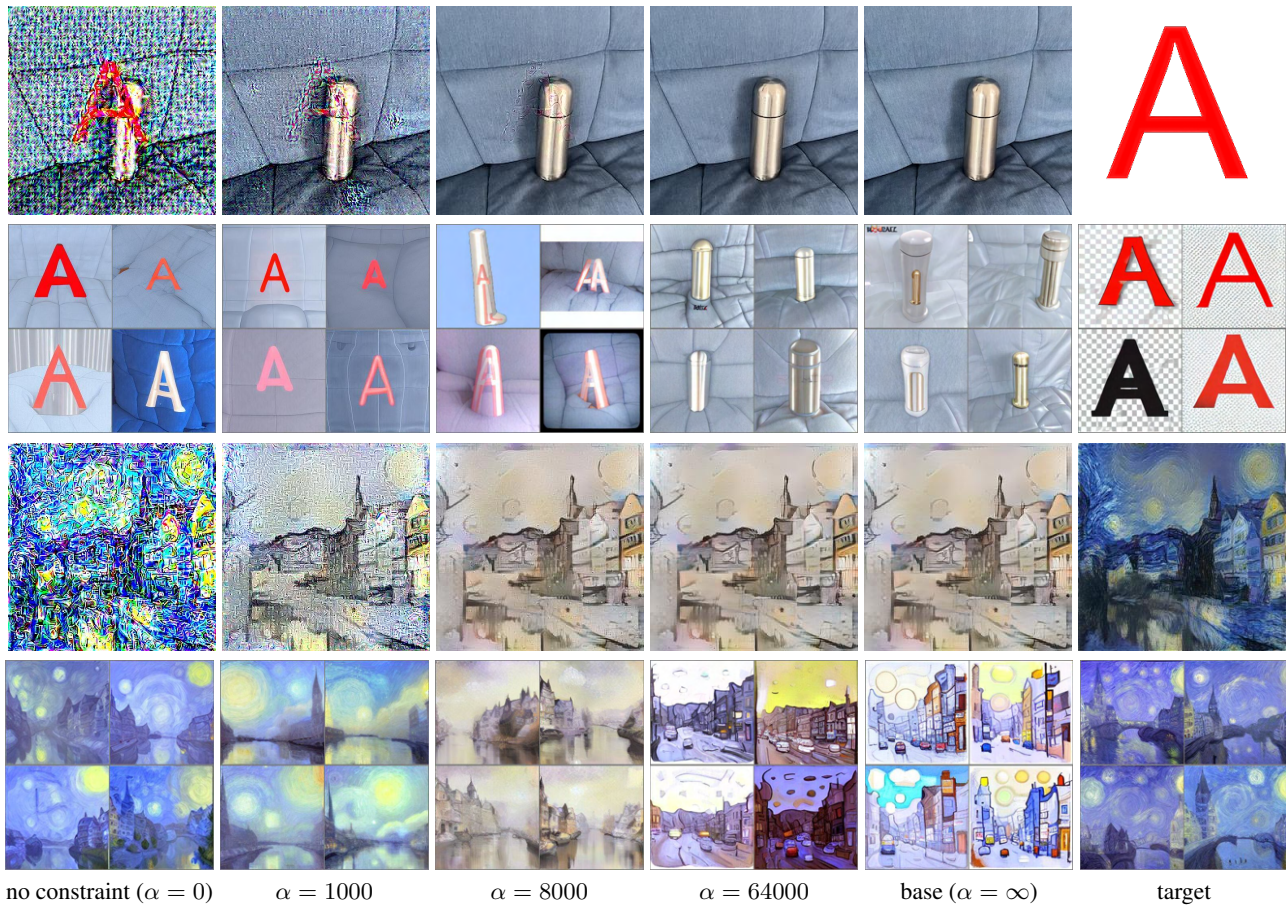[9]https://github.com/tyui592/AdaIN_Pytorch

Figure 6: The effect of tuning the hyperparameter α on disguises (Rows 1-2: symbol; Rows 3-4: style). Rows 1 and 3 show disguises generated with different α; Rows 2 and 4 show the images generated by textual inversion by training on the disguises above. Note that the criterion on the input and feature threshold may not be satisfied in this series of experiments and we only train the disguise generation algorithm until convergence.

In summary, for disguised symbols, generating the disguises x_d with any base image x_b is easy. However, for generating disguised content and style, one needs to choose the base image carefully for successful optimization.

4.3. Visual appearance of disguises

Next, we control the visual appearance of x_d by tuning α, which is the weighting parameter of the input space constraint. In Figure 6, we show the tradeoff of tuning the hyperparameter α: (1) a smaller α indicates weaker input space constraint and it could lead to an ineffective disguise, where x_d still visually contains the copyrighted material; (2) in contrast, a bigger α shifts the optimization focus away from feature matching to generate latent embeddings distinct from the copyrighted content's. In the latter case, the tradeoff is that despite having a strong disguise that visually hides the copyrighted content, the textual inversion process may not successfully learn the copyrighted material, thus rendering our attack pointless. Note that there is no

input constraint for the extreme case $\alpha = 0$, and the visual appearance of disguises largely depends on the initialization (we initialize with x_b above) and we show the results for different initialization in Figure 15 in Appendix A.

4.4. Detection

Next, we introduce a two-step detection method specifically for disguised samples that go beyond browsing through the training set, e.g., haveibeentrained.com.

(1) **Feature similarity search:** Quantitatively, a disguised sample has a similar feature representation with that of a copyrighted sample, i.e., $D_2(\mathcal{E}(x_c), \mathcal{E}(x_d)) \leq \gamma_2$. As we have provided the feature threshold γ_2 which is sufficient for replicating the copyrighted content for different tasks, one can use this as a reference threshold to detect possible disguises. Specifically, given an encoder $\mathcal{E}$ (which is usually easily accessible), a copyrighted image x_c which needs examination for infringement, one simply goes through the training set and computes $\mathcal{E}(x)$ for every single sample and
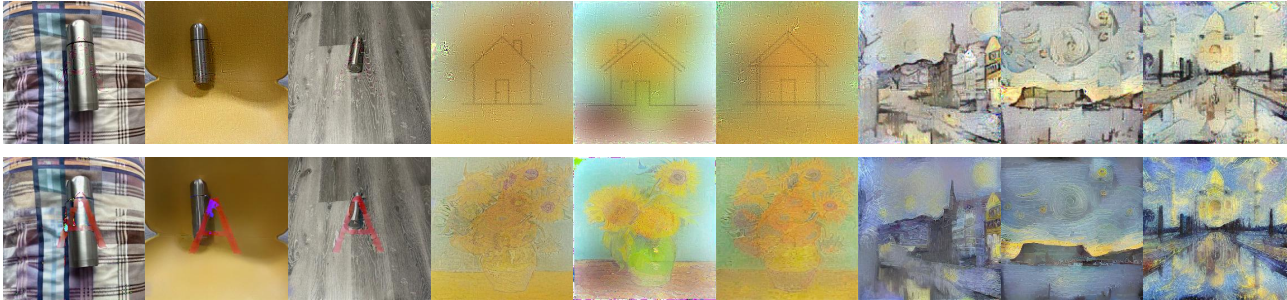
Figure 7: Results for the encoder-decoder examination. The first row shows the disguises x_d and the second row includes the output of the autoencoder: $\mathcal{D}(\mathcal{E}(x_d))$. Columns 1-3, 4-6, and 7-9 show results for symbol, content, and style, respectively. The autoencoder reveals the copyrighted information contained in the disguises.

compare with that of x_c. However, such a search alone only extracts suspects, which may not be true disguises. To rule out the sheer chance of collisions in the feature space, we perform another examination below.

(2) **Encoder-decoder examination:** The encoder architecture in LDMs is part of an autoencoder architecture (e.g., the KL-based VAE), where the encoder and decoder are used separately for encoding and inference. Consequently, the decoder $\mathcal{D}$ can be naturally used to detect disguises qualitatively. Specifically, for a well-trained autoencoder, $\mathcal{D}(\mathcal{E}(x_c)) \approx x_c$, while for disguises $\mathcal{E}(x_d) \approx \mathcal{E}(x_c)$, thus we have $\mathcal{D}(\mathcal{E}(x_d)) \approx x_c$. In Figure 7, we show that the encoder-decoder architecture is a great detection tool for disguises, where the output of the autoencoder reveals the copyrighted content hidden in x_d.

4.5. A broader definition of access

Our experiments have demonstrated that by generating disguises, one can indirectly acquire access to copyrighted material as part of the training data to fuel LDM-based models, which could reproduce the copyrighted samples during inference or deployment. We expand the definition of *direct access* to *acknowledgment* to cover such scenarios as possible copyright infringement: any training data x that contains similar latent information as that of copyrighted image x_c, measured by their similarity in the latent feature representation, even visually different from x_c, shall be considered to have an *acknowledgment* of x_c. The notion of *acknowledgment* may also augment existing regulatory frameworks (e.g., the White House Executive Order on AI (Biden 2023)) on AI governance in terms of data quality evaluation. Additionally, our quantification of *acknowledgment* (the detection method) provide a timely tool for auditing beyond black-box access, which was pointed out to be insufficient in (Casper et al. 2024).

5. Conclusion

In this paper, we review the current access criterion of containing copyrighted material in the training set (*direct access*) in copyright infringement of generative models and point out its insufficiency by introducing disguised copyright infringement (*indirect access*). Specifically, such an infringement is realized by injecting disguised samples into the training set, which urges LDMs to produce copyrighted content. Such disguises are generated with a simple algorithm and demonstrated to share the same concept with their target copyrighted images using the textual inversion tool. To alleviate the concern on the disguises, we expand the current visual auditing (browsing the training set) with additional tools, i.e., feature similarity search and encoder-decoder examination to better identify these disguises. Furthermore, we propose a broader definition of *acknowledgment* to cover this new type of copyright violation.

Limitations and future work: One interesting future work is to quantify the number of disguises needed for reproducing in large-scale training, which can be further linked to the quantification of memorization of such models (Carlini et al. 2022; Somepalli et al. 2023a; Somepalli et al. 2023b; Carlini et al. 2023; Ippolito et al. 2023; Zhang et al. 2021)). Additionally, although our algorithms can generate descent disguises, we believe there is still room for improvement for optimization. Finally, one extension we didn't touch is the possibility of "chopping" copyrighted data and hiding it in several images. It is intriguing to explore whether it is possible to generate such a smuggler's dataset and detection towards it.

Learning from noisy data A simultaneous and independent work (Daras et al. 2024) considers learning LDMs from noisy data. Although the techniques are very different, our works share a similar implication: the training dataset may not immediately resemble the generations produced, thus al-

lowing copyright issues to be disguised from an auditor who manually inspects the dataset. In the work of Daras et al. (2024), the training data is the original data with Gaussian noise applied to it. In our case, the training data has the original data hidden in the latent space. In Section 4.5, we argue that if training data contains the same latent information as some copyrighted data, then it should be acknowledged. Daras et al. (2024) show that even when the latent space is corrupted the copyrighted information could still be contained, which could raise a stronger disguised copyright infringement attack.

Acknowledgment and fair use   Finally we discuss the anticipated impact of our definition of *acknowledgment* on judicial decisions. Specifically, we believe *acknowledgment* will not directly affect the current fair use doctrine or change its decision. To further illustrate the above claim, we first propose two principles: (1) copyright infringement is not fair use: although generative AI law is not rigorously defined, we generally assume that "reproducing the copyrighted content" is not fair use when there's direct access; (2) disguised infringement does not (appear to) use the copyrighted data: there is no decision of fair use doctrine here as there is no apparent "usage" of the copyrighted data, not to mention "fair use". Thus the decision is on the "originality" of the output (reproduction of the copyrighted content) but not fair use. As a result, the notion of *acknowledgment* aims to identify disguised copyright infringement in the second principle and challenge the decision of "originality". Based on the acquired outcome (reproduction), there would be no scenarios that *acknowledgment* can change the decision of fair use according to the first principle.

Impact Statement

In this paper, we adopt a simple algorithm to disguise images, which we realize may be applied as a concealed copyright infringement tool. We are surprised at how easy it is for Generative AI to circumvent the copyright law, which was originally designed for regulating human scrapers. This seemingly undetectable infringing threat may bring disadvantages for human artists, who are already losing the battle with GenAI. Furthermore, disguised images may not only be a concern to copyright owners but also a threat to Generative AI companies: if these disguises are accidentally collected as part of the training data, copyright infringement could be triggered unconsciously and unwillingly. As computer scientists, we are obligated to reveal the existence of such possible unlawful acts and provide technical tools to identify such behaviors. We expect our paper to be a prudent tool such that disguised copyright infringement and its corresponding detection methods are recognized by law experts.

Acknowledgements

References

Aghakhani, H., D. Meng, Y.-X. Wang, C. Kruegel, and G. Vigna (2021). "Bullseye polytope: A scalable clean-label poisoning attack with improved transferability". In: *IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 159–178.

Biden, J. R. (2023). "Executive order on the safe, secure, and trustworthy development and use of artificial intelligence".

Biggio, B., B. Nelson, and P. Laskov (2012). "Poisoning attacks against support vector machines". In: *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pp. 1467–1474.

Carlini, N., D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang (2022). "Quantifying Memorization Across Neural Language Models". In: *The Eleventh International Conference on Learning Representations*.

Carlini, N., J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramer, B. Balle, D. Ippolito, and E. Wallace (2023). "Extracting training data from diffusion models". In: *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270.

Casper, S. et al. (2024). "Black-box access is insufficient for rigorous AI audits". arXiv preprint arXiv:2401.14446.

Chen, X., C. Liu, B. Li, K. Lu, and D. Song (2017). "Targeted backdoor attacks on deep learning systems using data poisoning". arXiv:1712.05526.

Daras, G., A. G. Dimakis, and C. Daskalakis (2024). "Consistent Diffusion Meets Tweedie: Training Exact Ambient Diffusion Models with Noisy Data". arXiv preprint arXiv:2404.10177.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". *arXiv preprint arXiv:1810.04805*.

Dist. Court, N. C. (2023). "Andersen v. STABILITY AI LTD." Case No. 23-cv-00201-WHO.

Fowl, L., P.-y. Chiang, M. Goldblum, J. Geiping, A. Bansal, W. Czaja, and T. Goldstein (2021a). "Preventing unauthorized use of proprietary data: Poisoning for secure dataset release". arXiv preprint arXiv:2103.02683.

Fowl, L., M. Goldblum, P.-y. Chiang, J. Geiping, W. Czaja, and T. Goldstein (2021b). "Adversarial Examples Make Strong Poisons". In: *Advances in Neural Information Processing Systems*, pp. 30339–30351.

Fu, S., F. He, Y. Liu, L. Shen, and D. Tao (2021). "Robust unlearnable examples: Protecting data privacy against adversarial learning". In: *International Conference on Learning Representations*.

Gal, R., Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or (2022). "An image is worth one word: Personalizing text-to-image generation using textual inversion". arXiv preprint arXiv:2208.01618.

Gatys, L. A., A. S. Ecker, and M. Bethge (2015). "A neural algorithm of artistic style". arXiv preprint arXiv:1508.06576.

Geiping, J., L. H. Fowl, W. R. Huang, W. Czaja, G. Taylor, M. Moeller, and T. Goldstein (2021). "Witches' Brew: ial Scale Data Poisoning via Gradient Matching". In: *International Conference on Learning Representations*.

Gu, T., B. Dolan-Gavitt, and S. Garg (2017). "Badnets: Identifying vulnerabilities in the machine learning model supply chain". arXiv:1708.06733.

Guo, J. and C. Liu (2020). "Practical Poisoning Attacks on Neural Networks". In: *European Conference on Computer Vision*, pp. 142–158.

Huang, H., X. Ma, S. M. Erfani, J. Bailey, and Y. Wang (2021). "Unlearnable Examples: Making Personal Data Unexploitable". In: *International Conference on Learning Representations*.

Huang, X. and S. Belongie (2017). "Arbitrary style transfer in real-time with adaptive instance normalization". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510.

Ippolito, D., F. Tramèr, M. Nasr, C. Zhang, M. Jagielski, K. Lee, C. A. Choquette-Choo, and N. Carlini (2023). "Preventing generation of verbatim memorization in language models gives a false sense of privacy". In: *Proceedings of the 16th International Natural Language Generation Conference*. Association for Computational Linguistics, pp. 28–53.

Jiang, S., S. R. Kadhe, Y. Zhou, L. Cai, and N. Baracaldo (2023). "Forcing Generative Models to Degenerate Ones: The Power of Data Poisoning Attacks". *arXiv preprint arXiv:2312.04748*.

Khare, N., P. S. Thakur, P. Khanna, and A. Ojha (2021). "Analysis of Loss Functions for Image Reconstruction Using Convolutional Autoencoder". In: *International Conference on Computer Vision and Image Processing*. Springer, pp. 338–349.

Koh, P. W. and P. Liang (2017). "Understanding black-box predictions via influence functions". In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 1885–1894.

Koh, P. W., J. Steinhardt, and P. Liang (2022). "Stronger Data Poisoning Attacks Break Data Sanitization Defenses". *Machine Learning*, vol. 111, pp. 1–47.

Liu, W. and S. Chawla (2010). "Mining adversarial patterns via regularized loss minimization". *Machine learning*, vol. 81, no. 1, pp. 69–83.

Lu, Y., G. Kamath, and Y. Yu (2022). "Indiscriminate Data Poisoning Attacks on Neural Networks". *Transactions on Machine Learning Research*.

— (2023). "Exploring the limits of model-targeted indiscriminate data poisoning attacks". In: *International Conference on Machine Learning*. PMLR, pp. 22856–22879.

Lu, Y., M. Y. Yang, G. Kamath, and Y. Yu (2024). "Indiscriminate Data Poisoning Attacks on Pre-trained Feature Extractors". *arXiv preprint arXiv:2402.12626*.

Muñoz-González, L., B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli (2017). "Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization". In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AISec)*, pp. 27–38.

Radford, A. et al. (2021). "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR, pp. 8748–8763.

Rombach, R., A. Blattmann, D. Lorenz, P. Esser, and B. Ommer (2022). "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF*

conference on computer vision and pattern recognition, pp. 10684–10695.

Ruiz, N., Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman (2023). "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.

Saha, A., A. Subramanya, and H. Pirsiavash (2020). "Hidden trigger backdoor attacks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Sandoval-Segura, P., V. Singla, J. Geiping, M. Goldblum, T. Goldstein, and D. W. Jacobs (2022). "Autoregressive Perturbations for Data Poisoning". In: *Advances in Neural Information Processing Systems*.

Schuhmann, C., R. Kaczmarczyk, A. Komatsuzaki, A. Katta, R. Vencu, R. Beaumont, J. Jitsev, T. Coombes, and C. Mullis (2021). "LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs". In: *NeurIPS Workshop Datacentric AI*. FZJ-2022-00923. Jülich Supercomputing Center.

Schuhmann, C. et al. (2022). "Laion-5b: An open large-scale dataset for training next generation image-text models". *Advances in Neural Information Processing Systems*, vol. 35, pp. 25278–25294.

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

Shan, S., W. Ding, J. Passananti, S. Wu, H. Zheng, and B. Y. Zhao (2024). "Nightshade: Prompt-Specific Poisoning Attacks on Text-to-Image Generative Models". In: *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, pp. 212–212.

Sohl-Dickstein, J., E. Weiss, N. Maheswaranathan, and S. Ganguli (2015). "Deep unsupervised learning using nonequilibrium thermodynamics". In: *International conference on machine learning*. PMLR, pp. 2256–2265.

Somepalli, G., V. Singla, M. Goldblum, J. Geiping, and T. Goldstein (2023a). "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.

— (2023b). "Understanding and Mitigating Copying in Diffusion Models". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

"Steinberg v. Columbia Pictures Industries, Inc." (1987).

Tran, B., J. Li, and A. Madry (2018). "Spectral Signatures in Backdoor Attacks". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Vyas, N., S. Kakade, and B. Barak (2023). "On provable copyright protection for generative models". In: *Proceedings of the 40 th International Conference on Machine Learning*.

Wan, A., E. Wallace, S. Shen, and D. Klein (2023). "Poisoning Language Models During Instruction Tuning". *arXiv preprint arXiv:2305.00944*.

Wang, Z., E. P. Simoncelli, and A. C. Bovik (2003). "Multi-scale structural similarity for image quality assessment". In: *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. Vol. 2. Ieee, pp. 1398–1402.

Yu, D., H. Zhang, W. Chen, J. Yin, and T.-Y. Liu (2022). "Availability Attacks Create Shortcuts". In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2367–2376.

Zhang, C., D. Ippolito, K. Lee, M. Jagielski, F. Tramèr, and N. Carlini (2021). "Counterfactual memorization in neural language models". arXiv preprint arXiv:2112.12938.

Zhu, C., W. R. Huang, H. Li, G. Taylor, C. Studer, and T. Goldstein (2019). "Transferable clean-label poisoning attacks on deep neural nets". In: *International Conference on Machine Learning*, pp. 7614–7623.

A. Additional Experiments on Revealing Disguises

We provide additional experimental results on revealing disguises to further support our observations.

Textual inversion on x_c and x_b: Recall that Figure 3, Figure 4 and Figure 5 show the disguised images x_d and the new images generated by textual inversion. To further show the effect of our algorithm on generating disguises, we directly perform a textual inversion (with the same setting in our main experiments) on the copyrighted images x_c and the base images x_b below. Firstly, for copyrighted symbols:
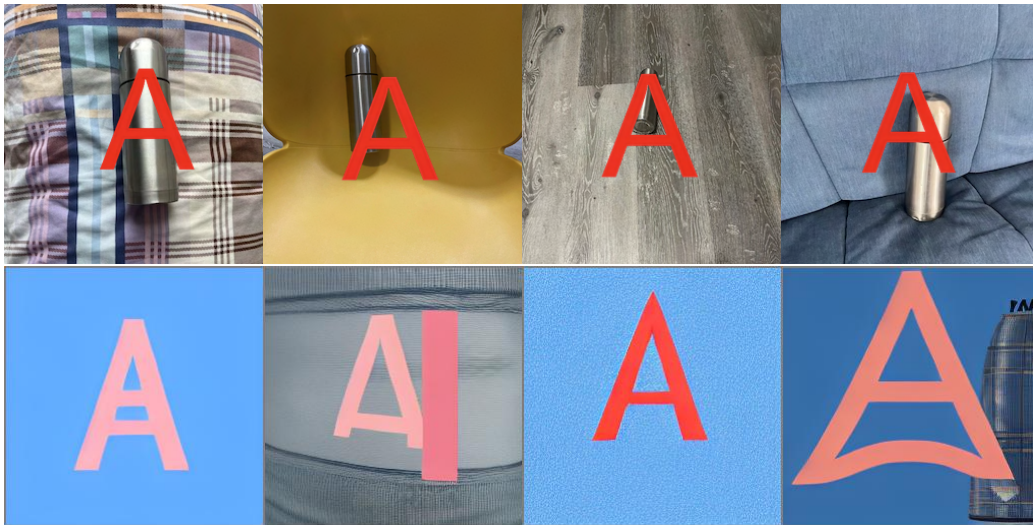
Learning on the copyrighted images x_c:

Figure 8: Textual inversion on images x_c with the copyrighted symbol "A" . The first row shows the copyrighted images x_c, and the second row shows new images generated by textual inversion after learning on the above x_c.

Learning on the base images x_b:

Figure 9: Textual inversion on base images x_b (different images of a bottle). The first row shows the base images x_b, and the second row shows new images generated by textual inversion after learning on the above x_b.

Comparing with Figure 3, we observe that although the disguises x_d look visually similar to their corresponding base images x_b, they contain drastically different latent information.

Next, we show the results for copyrighted content in the same format:

Learning on the copyrighted images x_c:

Figure 10: Textual inversion on copyrighted images x_c (*The Sunflowers* by Vincent Van Gogh). The first row shows the copyrighted images x_c, and the second row shows new images generated by textual inversion after learning on the above x_c.

Learning on the base images x_b:

Figure 11: Textual inversion on base images x_b (images of sketched houses with the blurry version of the corresponding x_c as the background). The first row shows the base images x_b, and the second row shows new images generated by textual inversion after learning on the above x_b.

Compared to Figure 4, we observe that the background (blurry version of the x_c) does not provide any information regarding the copyrighted images x_c, and the success of the disguises is accomplished by our generating algorithm.

Finally, we show the results for style scraping:

Learning on the copyrighted images x_c:

Figure 12: Textual inversion on copyrighted images x_c (images in the style of *The Starry Night* by Vincent Van Gogh). The first row shows the copyrighted images x_c, and the second row shows new images generated by textual inversion after learning on the above x_c.

Learning on the base images x_b:

Figure 13: Textual inversion on base images x_b (images in watercolor style). The first row shows the base images x_b, and the second row shows new images generated by textual inversion after learning on the above x_b.

Compared to Figure 5, we again confirm that the disguises x_d generated for style scraping look visually similar to their corresponding base images x_b but contain latent information similar to x_c.

Choice of x_b for generating disguised content: Recall that for generating disguises in Figure 4, we choose a blurry version of x_c as background to retain the color pattern. Here we demonstrate the necessity of the background by substituting it for a white background as base images x_b. In Figure 14, we observe that the disguises fail in reproducing the copyrighted content on textual inversion.
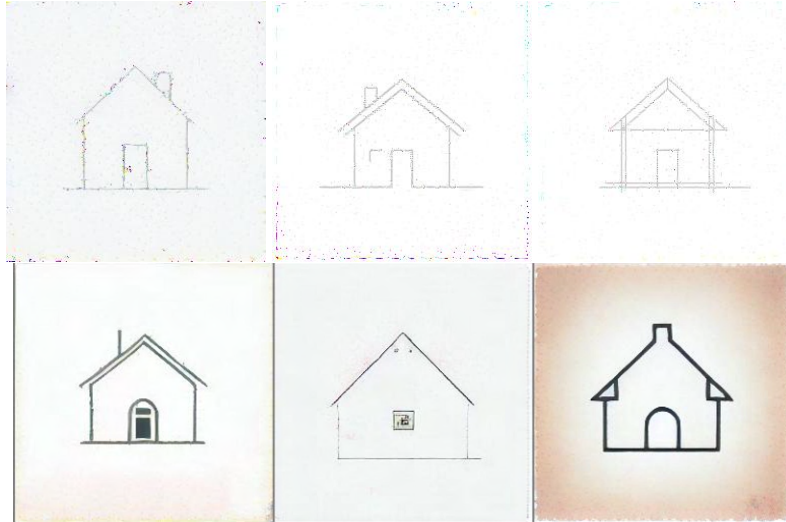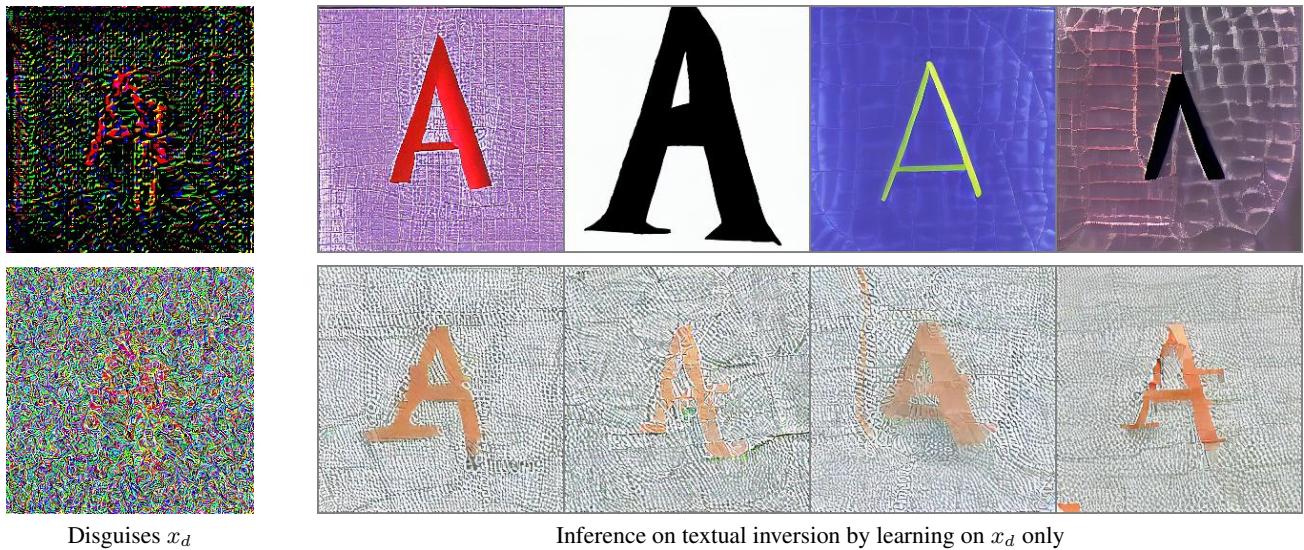
Figure 14: Failure to learn the copyrighted content (*The Sunflowers*) from a disguised sketch of the house with a white background. The first row: disguises; the second row: images generated by textual inversion.

Intialization for unbounded disguises: In Figure 6 (first column), we show the disguises generated without any input constraints. Note that the disguise (for the copyrighted symbol) still contains the base image as background as the initialization of the disguise is x_b. To confirm, we show two different initializations (zero tensor and Gaussian noise) in Figure 15 and observe that the background diminishes while the disguises still contain the copyrighted symbol.

Disguises x_d Inference on textual inversion by learning on x_d only

Figure 15: Textual inversion on the disguised symbol without any input constraints. We deploy different initializations for generating disguises: the first row: zero tensor; the second row: Gaussian noise.

B. Data augmentation

In textual inversion, images are horizontally flipped with 50% probability. In this section, we explore how this affects the success of our disguises. The top two rows of Figure 16 show the results of training textual inversion on our disguise as well as its horizontally flipped augmentation. For the simpler concept of watermarking, textual inversion learns a concept similar to the intended one even when the input image is flipped. On the other hand, when the concept becomes more difficult,

textual inversion struggles to learn the concept under a horizontal flip augmentation (top right of Figure 16).
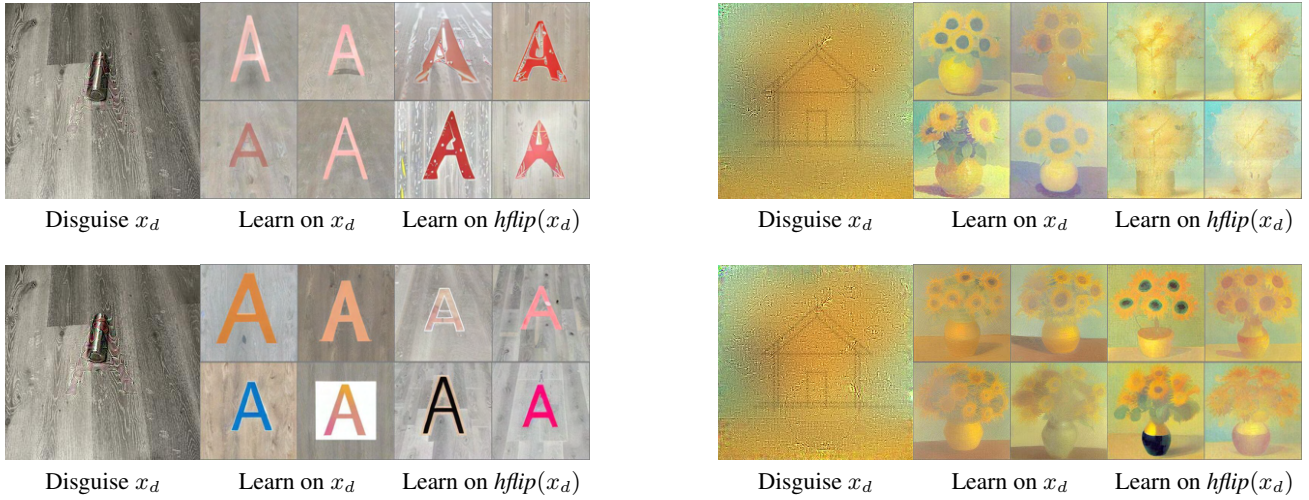

Figure 16: Textual inversion on different disguises. Top left: textual inversion on a single watermarking disguise. Top right: textual inversion on a single content disguise. Bottom row: same as top row, but with robust disguises.

To ensure that a horizontal flip augmentation would not damage the quality of our poison, we construct a new robust poison that additionally penalizes the distance between the features of the horizontally flipped poison and the horizontally flipped copyrighted image x_c. This can be done with a simple change to the objective from

$$\underset{x_d}{\operatorname{argmin}} \ \alpha D_1(x_b, x_d) + D_2(\mathcal{E}(x_c), \mathcal{E}(x_d))$$

to

$$\underset{x_d}{\operatorname{argmin}} \ \alpha D_1(x_b, x_d) + D_2(\mathcal{E}(x_c), \mathcal{E}(x_d)) + D_2(\mathcal{E}(\mathit{hflip}(x_c)), \mathcal{E}(\mathit{hflip}(x_d))).$$

The bottom two rows of Figure 16 demonstrate the effectiveness of the robust poison and its qualitative improvement under horizontal flipping.

Table 1: The encoder-decoder examination's threshold ζ, mean construction losses, false positive rate, and binary AUROC across symbol, content, and style disguises. The second group of rows is for the setting of mixture with base images, and the third group of rows is for the setting of mixture with 100 randomly selected Imagenette images.

	Symbol	Content	Style
$\mathbb{E}_{x_d}(\mathcal{L}_{\text{reconstruct}}(x_d))$	0.0856	0.1632	0.2667
Threshold ζ	0.0703	0.1412	0.2506
$\mathbb{E}_{x_b}(\mathcal{L}_{\text{reconstruct}}(x_b))$	0.0502	0.0256	0.0551
FPR (x_b misclassified)	1/4	0/3	0/3
AUC	0.8750	1	1
$\mathbb{E}_{x_{\text{clean}}}(\mathcal{L}_{\text{reconstruct}}(x_{\text{clean}}))$	0.0655	-	-
FPR (x_{clean} misclassified)	42/100	2/100	0/100
AUC	0.7550	0.9933	1

C. Quantitative detection

In Section 4.4, we show how to qualitatively distinguish a disguise with encoder-decoder examination. In this section, we further examine two quantitative detection scenarios with/without the presence of the copyrighted image x_c:

Without the presence of x_c**:** Specifically, we observe that in Figure 7, for disguises x_d, the input x_d and output of the autoencoder $\mathcal{D}(\mathcal{E}(x_d))$ are significantly different, i.e., the reconstruction loss $D_1(\mathcal{D}(\mathcal{E}(x_d)), x_d)$ is "big" (recall that D_1 is the distance measure in the input space, a sum of MS-SSIM loss and L_1 loss). This property differs from normal images, which are expected to have low reconstruction loss. Leveraging this intuition, we develop a detection criterion for disguised images:

$$x \text{ is a disguise if } \mathcal{L}_{\text{reconstruct}}(x) := D_1(\mathcal{D}(\mathcal{E}(x)), x) \geq \zeta$$

for some threshold ζ. The choice of ζ differs on different tasks (depending on the choice of the task, which we specify below). Note that this detection mechanism does not require any knowledge of the copyrighted image and is simply a data sanitization method that can be applied to any dataset.

In Table 1, we show the effectiveness of our proposed method across the three tasks we performed in Section 4 (copyrighted symbol, content, and style). *Row 2* shows the mean of the reconstruction loss for the disguises we generated for each task; *Row 3* shows the choice of threshold ζ as the lowest reconstruction loss out of all the disguised images such that there are no false negatives. *Rows 4-6* examine the detection performance in terms of FPR (False Positive Rate) and AUC (Are Under the Curve) within the pool of the mix of the disguises x_d and their corresponding base images x_b; *Rows 7-9* performs the same examination on a larger pool of the mix of the disguises x_d and a subset of 100 images randomly chosen from ImageNet.

We observe that for copyrighted symbols, as the copyrighted images x_c (see Figure 8) and the base images x_b (see Figure 9) share the same background and only differ on the symbol, the reconstruction loss of x_d is low (mean of 0.0856) and falls into the range of that of clean images, thus having a higher FPR and a lower AUC than the other two tasks.

With the presence of x_c**:** In practical scenarios (e.g., examination in court), one may already have acquired the copyrighted images x_c and want to find its (possible) corresponding disguise from a dataset. In such scenarios, one may directly use the detection method introduced in Section 4.4. We repeat the above experiment using our feature similarity search as a first step, which we recall, screen disguises using the criterion $D_2(\mathcal{E}(x_c), \mathcal{E}(x_d)) \leq \gamma_2$. We acquire perfect AUC score and obtain no false positives across all tasks, which indicate that the second step (encoder-decoder examination) is not required for this specific task.

D. Circumventing the detection?

A stronger disguise may be generated to circumvent our detection method (e.g., the encoder-decoder examination). This can be possibly accomplished by modifying the disguise generation objection. Recall that our previous objective is:

$$\underset{x_d}{\operatorname{argmin}} \, \alpha D_1(x_b, x_d) + D_2(\mathcal{E}(x_c), \mathcal{E}(x_d)),$$

Here we wish to also minimize the input distance between x_d and its autoencoder output $D_1(\mathcal{D}(\mathcal{E}(x_d)), x_d)$, thus we can modify our objective to:

$$\underset{x_d}{\operatorname{argmin}} \, \alpha(D_1(x_b, x_d) + D_1(\mathcal{D}(\mathcal{E}(x_d)), x_d)) + D_2(\mathcal{E}(x_c), \mathcal{E}(x_d))$$

We perform experiments on the copyrighted symbols and observe in Figure 17 that, the additional term adds a constraint such that x_d and $\mathcal{D}(\mathcal{E}(x_d))$ are visually similar (row 2). However, performing textual inversion on these images also fails to reveal the copyrighted symbol. We conclude that this method might not be ideal, and additional study might be required.

E. Additional experiments on disguised style

In this section, we exhibit additional experimental results on style transfer by extending the choices of the (designated) copyrighted style x_c and base images x_b.

Disguises in the style of Claude Monet: Recall that in Section 4, we choose the base images x_b as the ones with the watercolor style. In the following experiments, we aim to change x_b to images with the style of Claude Monet and still generate disguises x_d that implicitly contain the style of Vincent Van Vogh. In Figure 18, we first create the x_c using style transfer (Huang and Belongie 2017), which is *San Giorgio Maggiore at Dusk* (1908-1912) by Claude Monet transferring to the style of *The Starry Night* by Vincent Van Gogh. In Figure 19, we create the disguise x_d (row 1, column 2) with the
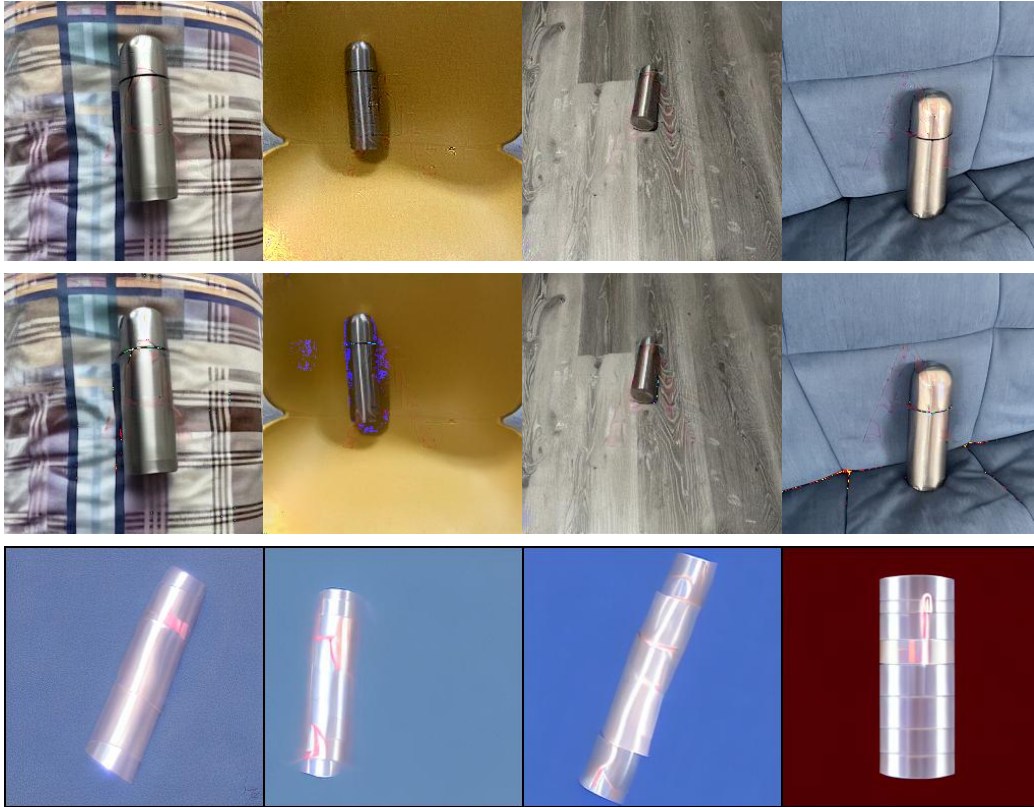
Figure 17: An attempt to evade the encoder-decoder detection on symbols. The first row shows the disguises x_d, and the second row includes the output of the autoencoder $\mathcal{D}(\mathcal{E}(x_d))$ that is unable to reveal the copyrighted information contained in the disguises. The third row shows the (unsuccessful) outcome of applying textual inversion to learn the hidden symbol.

visual appearance of the base x_b and show that the images generated by textual inversion after training on x_d still contain the style of *The Starry Night*.

(1) *San Giorgio Maggiore at Dusk*   (2) *The Starry Night* x_c: (1) with the style of (2)

Figure 18: Creating the copyrighted image x_c (third column) using AdaIN-based style transfer (Huang and Belongie 2017). x_c is *San Giorgio Maggiore at Dusk* by Claude Monet transferring to the style of *The Starry Night* by Vincent Van Gogh.

Scraping the style of other artists: In the previous experiments, we choose the style of *The Starry Night* as the designated copyrighted style. Next, we show the results for disguised style scraping of another artist Cecily Brown[10]. In Figure 20, we

---

[10]https://en.wikipedia.org/wiki/Cecily_Brown

Base x_b Disguise x_d x_c

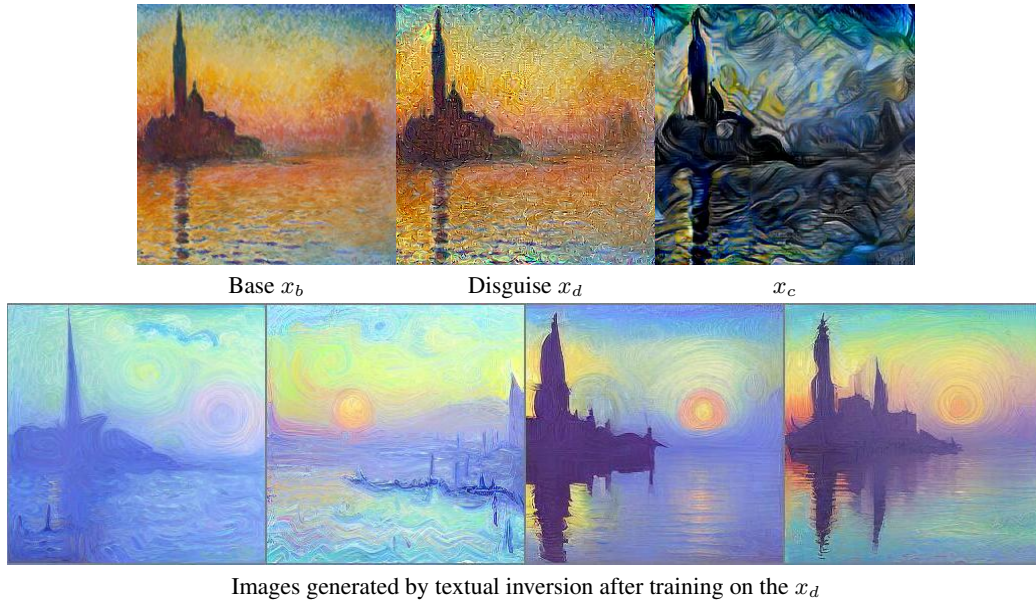Images generated by textual inversion after training on the x_d

Figure 19: Additional results on the disguised copyrighted style with textual inversion, where we generate disguise x_d that visually resembles the style of Claude Monet but containing the style of Vincent Van Gogh in the latent space.

first create the x_c using style transfer (Huang and Belongie 2017), which is Taj Mahal (adamkaz/Getty Images) transferring to the style of the painting with serial number *56542* by Cecily Brown[11]. In Figure 21, we create the disguise x_d (row 1, column 2) with the visual appearance of the base x_b (row 1, column 1) and show that the images generated by textual inversion after training on x_d contain the style of the style of Cecily Brown.
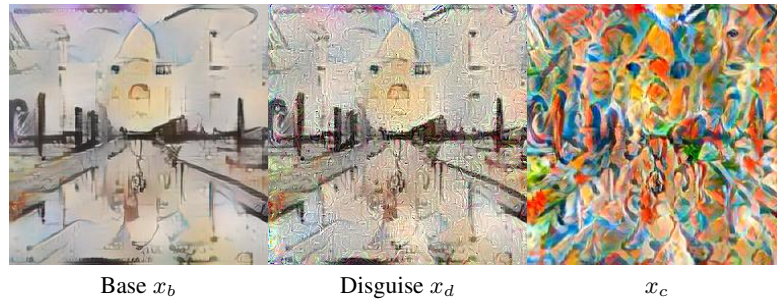
Finally, in Figure 22, we compare two different disguises x_d and observe images that visually look similar can contain drastically different styles in the feature space, which again demonstrates the threat of disguised copyright infringement.

(1) Taj Mahal *(watercolor)*   (2) *Cecily Brown-56542* x_c: (1) with the style of (2)

Figure 20: Creating the copyrighted image x_c (third column) using AdaIN-based style transfer (Huang and Belongie 2017). x_c is Taj Mahal transferring to the style of Cecily Brown.

Choice of x_b: In previous experiments, we construct the disguises x_d by choosing the base image x_b to have a different style from x_c but containing the same content. In Figure 23, we also explore the possibility of choosing a random base image from ImageNet (row 1, column 1). Our results show that although the style is partially learned by textual inversion from the disguise x_d, it also loses the structural information of x_c.

---

[11]https://www.artnews.com/art-in-america/features/cecily-brown-56542/

Base x_b Disguise x_d x_c

Images generated by textual inversion after training on the x_d

Figure 21: Additional results on the disguised copyrighted style with textual inversion, where we generate disguise x_d that visually resembles the watercolor style but containing the style of Cecily Brown in the latent space.

Taj Mahal in watercolor style   Taj Mahal in *The Starry Night* style   Taj Mahal in *Cecily Brown* style

Taj Mahal in watercolor style x_d (*The Starry Night* style) x_d (*Cecily Brown* style)

Figure 22: Comparison between different disguises x_d we generated that contain drastically different styles in the feature space, but are visually similar in the input space.

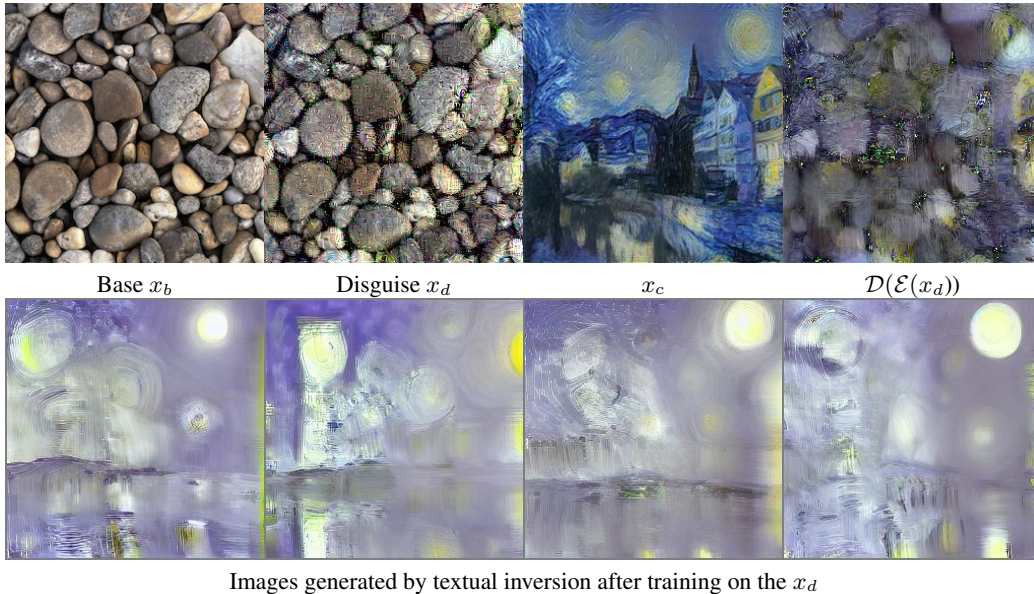Base x_b Disguise x_d x_c $\mathcal{D}(\mathcal{E}(x_d))$

Images generated by textual inversion after training on the x_d

Figure 23: Additional results on the disguised copyrighted style with textual inversion, where we generate disguise x_d that visually resembles a random base image but intend to contain the style of *The Starry Night* in the latent space.

F. Disguises in the Wild

We have previously *revealed* the concept hidden in the disguised data using textual inversion, which is also an LDM-based application that can possibly generate copyright materials. In this section, we further extend our evaluation to other LDM-based pipelines that change the parameters of the U-Net. Specifically, we consider DreamBooth (Ruiz et al. 2023), which fine-tunes a pre-trained LDM model with only a few disguised images (similar to textual inversion); and mixed-training that contains both clean data and a small portion of disguises.

F.1. Evaluation on DreamBooth

Main results: In Section 4, we reveal the latent (copyrighted) information contained in the acquired samples x_d using textual inversion, where the diffusion model (i.e., the U-Net in LDM) is not retrained. To further demonstrate the effectiveness of the disguises x_d and examine disguised copyright infringement in other scenarios, we further perform evaluation on DreamBooth (Ruiz et al. 2023), an LDM-based fine-tuning method on a small set of images designed for subject-driven generation. Specifically, we choose five images from the DreamBooth dataset[12] as the (designated) copyrighted image x_c, and generate their corresponding disguises x_d using Algorithm 1 by choosing the noisy version of x_c as the base images x_b, training 10000 epochs and setting the weight parameter $\alpha = 1000$. After generating the disguises x_d, we apply DreamBooth for fine-tuning (on x_d only) and inference following the general recipe[13]. In Figure 24, we show the copyrighted images x_c in the first row, their corresponding disguises x_d in the second row, and images generated by DreamBooth by fine-tuning with the above x_d in the third row. Our results qualitatively confirm that copyrighted information in x_c can be reproduced by fine-tuning on x_d with DreamBooth.

Evaluation on model utility: To further examine the practicality of the disguises, we want to demonstrate that the model utility is not deprecated after training (or fine-tuning) on x_d. To accomplish this task, we perform the same text-to-image generation task with the prompt "a photo of an astronaut riding a horse on Mars" before/after training on x_d with LDM [14]. Specifically, the "before" model is Stable Diffusion v1-4, and the "after" model is the acquired model by fine-tuning the "before" model on the dog disguise (second row, first column in Figure 24). We show our results in Figure 25 for 6

---

[12]https://github.com/google/dreambooth

[13]https://github.com/huggingface/diffusers/tree/main/examples/dreambooth

[14]We perform inference following the recipe in https://huggingface.co/CompVis/stable-diffusion-v1-4

x_c

Disguise x_d

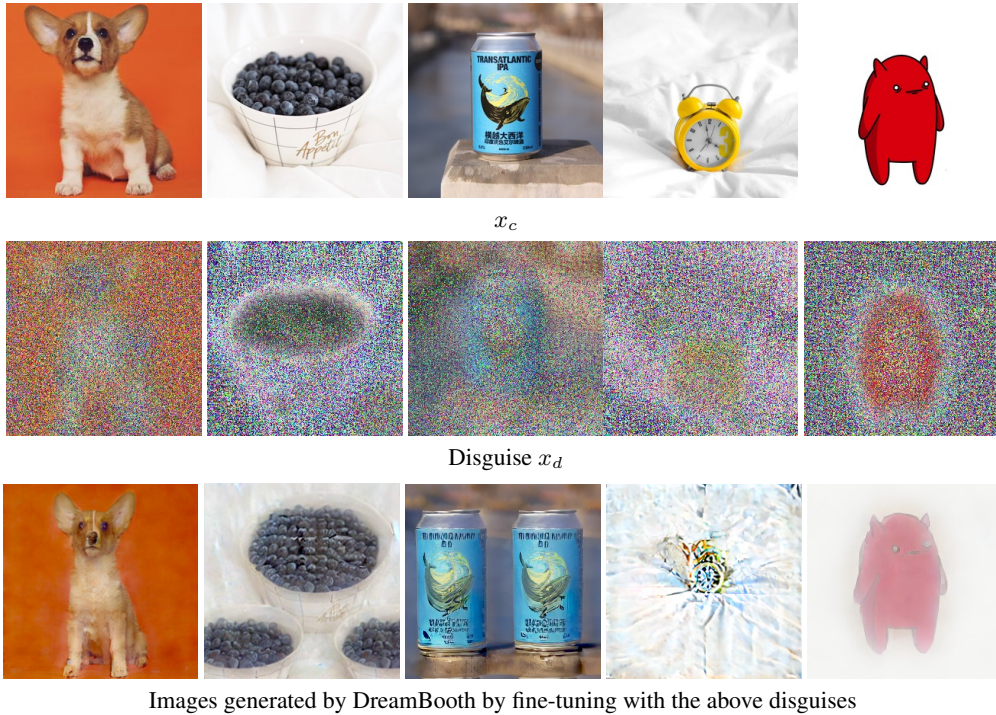Images generated by DreamBooth by fine-tuning with the above disguises

Figure 24: We show the disguised copyright infringement with DreamBooth. The first row: the designated copyrighted image x_c from the DreamBooth dataset; the second row: the corresponding disguises x_d generated with our Algorithm 1; the third row: images generated by DreamBooth after training on the above disguises x_d.

Images generated by LDM *before* fine-tuning with our disguises using DreamBooth

Images generated by LDM *after* fine-tuning with our disguises using DreamBooth

Figure 25: Utility validation: images generated by LDM using the prompt "a photo of an astronaut riding a horse on Mars" before/after fine-tuning with our disguises in Figure 24 (row 2, column 1) with DreamBooth.

inferences and observe that the generation ability of the model is not compromised. We will show in the next section that this observation extends to LDM training.

F.2. Mixed-training on CelebA-HQ

In the previous experiments, we have performed textual inversion and fine-tuning with DreamBooth where the disguises x_d are the entire training set. In this section, we extend our experiments to more challenging settings where the disguises x_d

Target images x_c with the copyrighted symbol

Disguises x_d

Images generated by LDM by training on clean images (1000) and disguises (100)
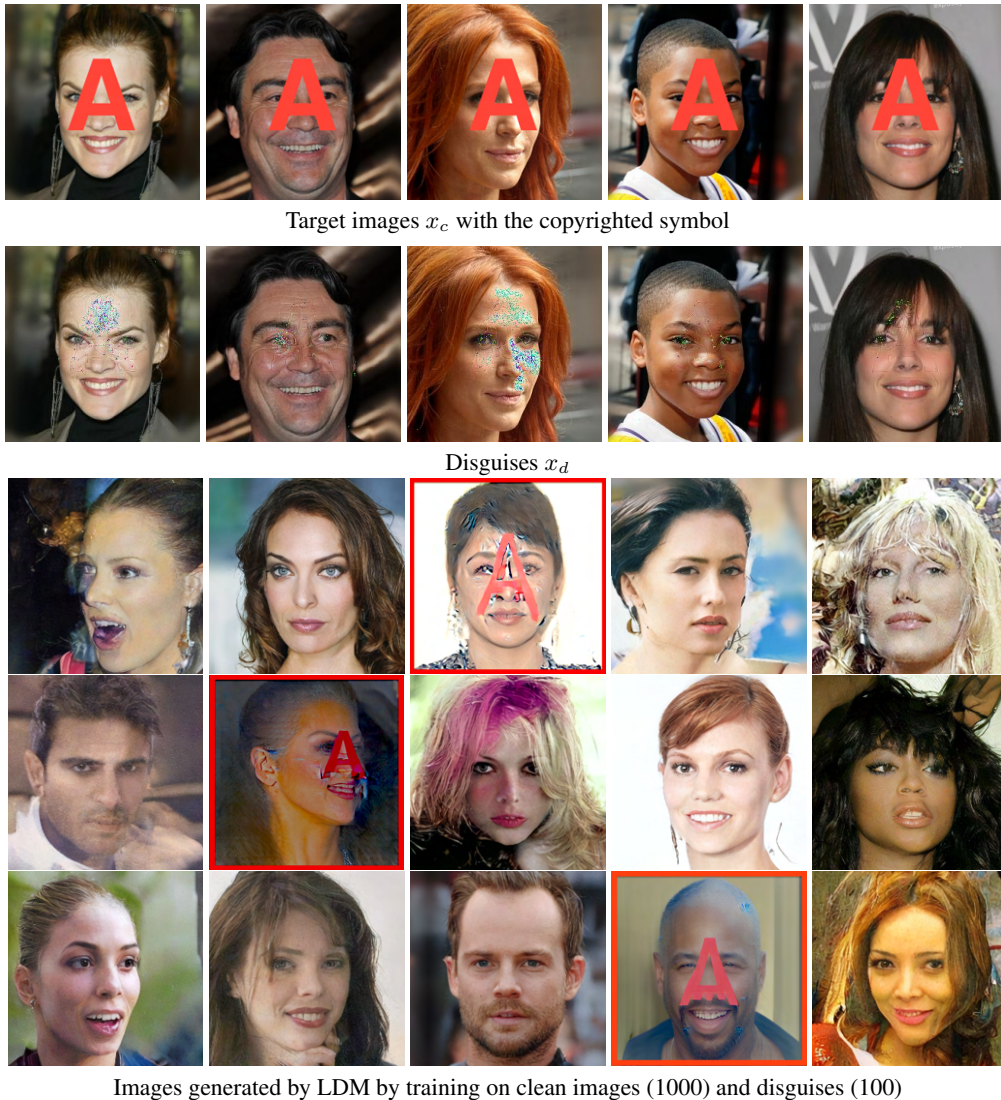
Figure 26: Disguised copyright infringement during mixed-training for unconditional generation on CelebA-HQ. The first row: the designated copyrighted image x_c with a red symbol "A"; the second row: the corresponding disguises x_d; the third to fifth rows: images generated by LDM by training on 1000 clean images and 100 disguises (each disguise above contributes 20 copies). Images highlighted with a red box indicate reproductions of the copyrighted symbol.

only constitute a small portion of the training set and are overwhelmed by the clean samples. Specifically, we randomly take a subset of 1000 images from CelebA-HQ/256[15] as clean training samples, and another subset of 5 samples (outside of the clean set) and add a (designated) copyrighted symbol "A" to all of them to construct x_c (first row in Figure 26). Next, we construct the corresponding disguises x_d (second row in Figure 26) to visually remove the symbol using Algorithm 1 and duplicate each disguise 20 times to construct a disguise set containing 100 images. Overall, we acquire a training set of 1100 images where the disguise fraction is $\frac{1}{11}$. We initialize the model with the CompVis official model[16] to preserve reasonable generation performance and train the model on the mixed dataset for 200 DDIM steps. In Figure 26 (rows 3-5) we show the images generated by the model after training (inference for 20 times) and observe 3 out of 20 images contain the copyrighted symbol, which demonstrated the effectiveness of the disguises for mixed-training scenarios.

---

[15]https://www.tensorflow.org/datasets/catalog/celeb_a_hq
[16]https://huggingface.co/CompVis/ldm-celebahq-256