usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION

Extracting Training Data from Diffusion Models

Nicholas Carlini, *Google;* Jamie Hayes, *DeepMind;* Milad Nasr and Matthew Jagielski, *Google;* Vikash Sehwag, *Princeton University;* Florian Tramèr, *ETH Zurich;* Borja Balle, *DeepMind;* Daphne Ippolito, *Google;* Eric Wallace, *UC Berkeley*

https://www.usenix.org/conference/usenixsecurity23/presentation/carlini

This paper is included in the Proceedings of the
32nd USENIX Security Symposium.

August 9–11, 2023 • Anaheim, CA, USA

978-1-939133-37-3

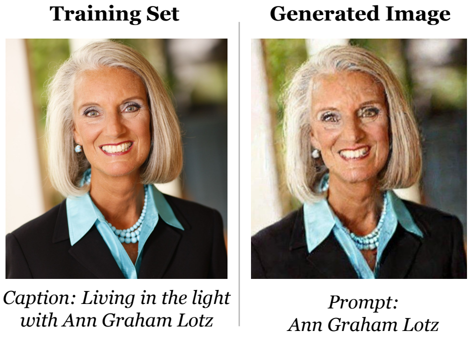
Open access to the Proceedings of the
32nd USENIX Security Symposium
is sponsored by USENIX.

Extracting Training Data from Diffusion Models

Nicholas Carlini[*1] *Jamie Hayes*[*2] *Milad Nasr*[*1]
Matthew Jagielski[+1] *Vikash Sehwag*[+4] *Florian Tramèr*[+3]
Borja Balle[†2] *Daphne Ippolito*[†1] *Eric Wallace*[†5]
[1]Google [2]DeepMind [3]ETHZ [4]Princeton [5]UC Berkeley
[*]Equal contribution [+]Equal contribution [†]Equal contribution

Abstract

Image diffusion models such as DALL-E 2, Imagen, and Stable Diffusion have attracted significant attention due to their ability to generate high-quality synthetic images. In this work, we show that diffusion models memorize individual images from their training data and emit them at generation time. With a generate-and-filter pipeline, we extract over a thousand training examples from state-of-the-art models, ranging from photographs of individual people to trademarked company logos. We also train hundreds of diffusion models in various settings to analyze how different modeling and data decisions affect privacy. Overall, our results show that diffusion models are much less private than prior generative models such as GANs, and that mitigating these vulnerabilities may require new advances in privacy-preserving training.



Training Set

Caption: Living in the light with Ann Graham Lotz

Generated Image

Prompt: Ann Graham Lotz

Figure 1: Diffusion models memorize individual training examples and generate them at test time. **Left:** an image from Stable Diffusion's training set (licensed CC BY-SA 3.0, see [54]). **Right:** a Stable Diffusion generation when prompted with "Ann Graham Lotz". The reconstruction is nearly identical (ℓ_2 distance = 0.031).

1 Introduction

Denoising diffusion models are an emerging class of generative neural networks that produce images from a training distribution via an iterative denoising process [37, 69, 71]. Compared to prior approaches such as GANs [34] or VAEs [50], diffusion models produce higher-quality samples [20], and are easier to scale [61] and control [56]. Consequently, they have rapidly become the de-facto method for generating high-resolution images, and large-scale models such as DALL-E 2 [61] have attracted significant public interest.

The appeal of generative diffusion models is rooted in their ability to synthesize novel images that are ostensibly unlike anything in the training set. It has been speculated that this ability could help protect the privacy of future training sets, by only releasing *synthetic images* from a generative model trained on real images [2, 14, 15, 58, 64]. Yet, as noted in [41], these privacy benefits would be moot if diffusion models were to "reveal the data they are trained on". Data memorization in diffusion models would also raise numerous questions regarding model generalization and "digital forgery" [70].

In this work, we demonstrate that state-of-the-art diffusion models *do* memorize and regenerate individual training examples. To begin, we propose and implement new definitions for "memorization" in image models. We then devise a two-stage data extraction process that generates images using standard approaches, and flags those that exceed some membership inference score. Applying this method to Stable Diffusion [63] and Imagen [65], we extract over a hundred near-identical replicas of training images that range from personally identifiable photos to trademarked logos (e.g., Figure 1).

To better understand and quantify the extent to which memorization occurs, we train hundreds of diffusion models on CIFAR-10 to analyze the impact of model accuracy, hyperparameters, augmentation, and deduplication on privacy. Diffusion models are the least private form of image models that we evaluate—for example, they leak more than twice as much training data as GANs. Unfortunately, we also find that existing privacy-enhancing techniques—such as data deduplication and differentially-private training— do not provide an acceptable privacy-utility tradeoff. Overall, our paper highlights the tension between increasingly powerful generative models and data privacy, and raises questions on how diffusion models work and how they should be responsibly deployed.

2 Background

Diffusion models. Generative image models have a long history (see [33, Chapter 20]). Generative Adversarial Networks (GANs) [34] were the breakthrough that first enabled the generation of high-fidelity images at scale [6, 48]. But over the last two years, diffusion models [69] have largely displaced GANs: they achieve state-of-the-art results on academic benchmarks [20] and form the basis of popular image generators such as Stable Diffusion [63], DALL-E 2 [61, 62], Runway [63], Midjourney [53] and Imagen [65].

Denoising Diffusion Probabilistic Models (DDPMs) [37][1] are conceptually simple: they are nothing more than image *denoisers*. During training, given a clean image x, we sample a time-step $t \in [0,T]$ and a Gaussian noise vector $\varepsilon \sim \mathcal{N}(0,I)$, to produce a noised image $x' \leftarrow \sqrt{a_t}x + \sqrt{1-a_t}\varepsilon$, for some decaying parameter $a_t \in [0,1]$ where $a_0 = 1$ and $a_T = 0$. A diffusion model f_θ removes the noise ε to recover the original image x by predicting the noise that was added by stochastically minimizing the objective $\frac{1}{N}\sum_i \mathbb{E}_{t,\varepsilon}\mathcal{L}(x_i, t, \varepsilon; f_\theta)$, where

$$\mathcal{L}(x_i, t, \varepsilon; f_\theta) = \|\varepsilon - f_\theta(\sqrt{a_t}x_i + \sqrt{1-a_t}\varepsilon, t)\|_2^2 \ . \qquad (1)$$

Despite being trained with this simple denoising objective, diffusion models can *generate* high-quality images by applying the diffusion model f_θ to denoise a completely *random* "image" $z_T \sim \mathcal{N}(0,I)$. To make the denoising process easier, we do not remove all of the noise at once—we instead iteratively apply the model to slowly remove noise. Formally, the final image z_0 is obtained from z_T by iterating the rule $z_{t-1} = f_\theta(z_t, t) + \sigma_t \mathcal{N}(0,I)$ for a noise schedule σ_t (dependent on a_t) with $\sigma_1 = 0$. This process relies on the fact that the model f_θ was trained to denoise images with varying degrees of noise. Overall, running this iterative generation process (which we will denote by `Gen`) with large-scale diffusion models produces results that resemble natural images.

Some diffusion models are further *conditioned* to generate a particular type of image. Class-conditional diffusion models take as input a class-label (e.g., "dog" or "cat") alongside the noised image to produce an image of that class. Text-conditioned models take as input the text embedding of a more general *prompt* (e.g., "a photograph of a horse on the moon") using a pre-trained language encoder (e.g., CLIP [59]).

Training data privacy attacks. Neural networks often leak details of their training datasets. Membership inference attacks [8, 67, 85] infer whether an example was in the training set or not, a minimal form of privacy leak. Neural networks are also vulnerable to more powerful attacks such as inversion attacks [30, 86] that extract representative examples from a class, attribute inference attacks [31] that reconstruct some attributes of training examples, and extraction attacks [5, 11, 12] that recover full training examples. In this paper, we focus on each of these three attacks when applied to diffusion models.

Concurrent work explores the privacy of diffusion models. Multiple papers [22, 38, 83] independently perform membership inference attacks on diffusion models; our results use more sophisticated attack methods and study stronger privacy risks such as data extraction. Somepalli *et al.* [70] show several cases where (non-adversarially) sampling from a diffusion model can produce memorized training examples. However, they focus mainly on comparing the semantic similarity of generated images to the training set, i.e., "style copying". In contrast, we focus on a more restrictive notion of memorization (extraction of near-exact copies of training images), and consider a wider range of models.

3 Motivation and Threat Model

There are two distinct motivations for understanding diffusion models' propensity to memorize and regenerate training data.

Understanding privacy risks. Diffusion models that regenerate data scraped from the Internet can pose similar privacy and copyright risks as language models [7, 12, 35]. For example, memorizing and regenerating copyrighted text [12] and source code [39] has been pointed to as indicators of potential copyright infringement [81]. Similarly, copying images from professional artists has been called "digital forgery" [70] and has spurred debate in the art community.

Future diffusion models might be trained on more sensitive private data. Indeed, GANs have already been applied to medical imagery [23, 49, 78], which underlines the importance of understanding the risks of generative models *before* deploying them in private domains. It is speculated that future diffusion models could similarly "protect the privacy and usage rights of real images" [41], and production tools already claim to use diffusion models to protect data privacy [13, 19, 75]. Our work shows diffusion models may be unfit for this purpose.

Understanding generalization. Beyond data privacy, understanding diffusion models' memorization abilities may provide insights into their generalization capabilities. For instance, a common question for large-scale generative models is whether their impressive results arise from truly novel generations, or are instead the result of direct copying and remixing of their training data. By studying memorization, we can provide a concrete empirical characterization of the rates at which generative models perform such data copying.

In their diffusion model, Saharia *et al.* "do not find overfitting to be an issue, and believe further training might improve overall performance" [65], and yet we will show that this model memorizes individual examples. It may thus be necessary to broaden our definitions of overfitting to include memorization and related privacy metrics. Our results also suggest that Feldman's theory that memorization is *necessary* for generalization in classifiers [27] may extend to generative

[1] Our description of diffusion models below omits a number of significant details. However, these details are orthogonal to our results and we omit them for simplicity.

models, raising the question of whether the improved performance of diffusion models compared to prior approaches is precisely *because* diffusion models memorize more.

3.1 Threat Model

The primary purpose of our work is to demonstrate that diffusion models *can* memorize individual images, rather than to design the most practical privacy attack. Nevertheless, we formalize our process in an appropriate attack model that captures the assumptions underlying our extraction process.

Our threat model considers an adversary $\mathcal{A}$ that interacts with a diffusion model Gen (backed by a neural network f_θ) to extract images from the model's training set D.

Image-generation systems. Unconditional diffusion models are trained on a dataset $D = \{x_1, x_2, \ldots, x_n\}$. When queried, the system outputs a generated image $x_{\text{gen}} \leftarrow \texttt{Gen}(r)$ using fresh random noise r as input. Conditional models are trained on a labeled or captioned dataset $D = \{(x_1, c_1), \ldots, (x_n, c_n)\}$; when queried with a *prompt* p, the system outputs $x_{\text{gen}} \leftarrow \texttt{Gen}(p; r)$ using the prompt p and noise r.

Adversary capabilities. We consider two adversaries:

- A *black-box* adversary can query Gen to generate images. If Gen is a conditional generator, the adversary can provide arbitrary prompts p. The adversary cannot control the system's internal randomness r.
- A *white-box* adversary gets full access to the system Gen and its internal diffusion model f_θ. They can control the model's randomness and can thus use the model to denoise arbitrary input images.

In both cases, we assume that an adversary who attacks a conditional image generator knows the captions for some images in the training set—thus allowing us to study the *worst-case* privacy risk in diffusion models.

Whether this assumption holds in practice will largely be setting-dependent. In some privacy-sensitive settings, image captions might follow a common format that would be easy for an adversary to guess (e.g., annotated medical images). In any case, we mainly view our attacks as a way to *measure* the leakage of training images from diffusion models, when prompted with training captions. The ability to measure this leakage (even if it does not translate to an obvious practical attack at the moment) may be sufficient for *auditing* the worst-case privacy of deployed models, as well as for informing discussions surrounding "digital forgery" in diffusion models.

Adversary goals. We consider three broad types of adversarial goals, from the strongest to the weakest attack:

1. *Data extraction*: The adversary aims to recover an image from the training set $x \in D$. The attack is successful if the adversary extracts an image $\hat{x}$ that is almost identical (see Section 4.1) to *some* $x \in D$.
2. *Data reconstruction*: The adversary has partial knowledge of a training image $x \in D$ (e.g., a crop of the image) and aims to recover the full image. This is an image-analog of an *attribute inference attack* [85], which recovers unknown features from a partially known input.
3. *Membership inference*: Given an image x, the adversary aims to infer whether x is in the training set.

3.2 Ethics and Broader Impact

Training data extraction can present a threat to user privacy. We take numerous steps to mitigate any possible harms from our paper. First, we study models that are trained on publicly-available images (e.g., LAION and CIFAR-10) and therefore do not expose any data that was not already available online.

Nevertheless, data that is available online may not have been intended to be. LAION, for example, contains unintentionally released medical images of several patients [26]. We also therefore ensure that all images shown in our paper are of public figures (e.g., politicians, musicians, actors, or authors) who knowingly chose to place their images online. As a result, inserting these images in our paper is unlikely to cause any unintended privacy violation. For example, Figure 1 comes from Ann Graham Lotz's Wikipedia profile picture and is licensed under Creative Commons, which allows us to "redistribute the material in any medium" and "remix, transform, and build upon the material for any purpose, even commercially".

Third, we shared a copy of this paper with the authors of the large-scale diffusion models that we study. This gave the authors and their organizations the ability to consider possible safeguards and software changes ahead of time.

Overall, we believe that publishing our paper and publicly disclosing these privacy vulnerabilities is both ethical and responsible. Indeed, no one appears to be immediately harmed by the (lack of) privacy of existing diffusion models; our goal with this work is thus to preempt these harms and encourage responsible training of diffusion models in the future.

4 Extracting Training Data from State-of-the-art Diffusion Models

We begin our paper by extracting training images from large, pre-trained, high-resolution diffusion models.

4.1 Defining Image Memorization

The literature on training data extraction mainly studies language models, where a sequence is said to be "extracted" and "memorized" if an adversary can prompt the model to recover a *verbatim* sequence from the training set [12, 45]. When working with high-resolution images, verbatim definitions of memorization are not suitable. Instead, we define a notion of approximate memorization based on image similarity.

Definition 1 (**(ℓ,δ)-Diffusion Extraction**) [adapted from [12]]. *We say that an example x is* extractable *from a diffusion model f_θ if there exists an efficient algorithm $\mathcal{A}$ (that does not receive x as input) such that $\hat{x} = \mathcal{A}(f_\theta)$ has the property that $\ell(x,\hat{x}) \leq \delta$.*

Here, ℓ is a distance function and δ is a threshold that determines whether we count two images as being identical. Given this definition of extractability, we now define *memorization*.[2]

Definition 2 (**(k,ℓ,δ)-Eidetic Memorization**) [adapted from [12]]. *We say that an example x is (k,ℓ,δ)-Eidetic memorized by a diffusion model if x is extractable from the diffusion model, and there are at most k training examples $\hat{x} \in X$ where $\ell(x,\hat{x}) \leq \delta$.*

Again, ℓ is a distance function and δ the corresponding threshold. The constant k quantifies the number of near-duplicates of x in the dataset. If k is a small fraction of the data, then memorization is likely problematic. When k is a larger fraction of data, memorization might be expected—but it could still be problematic, e.g., if the duplicated data is copyrighted.

Distance function. In most of this paper, we follow Balle *et al.* [5] and use the Euclidean 2-norm distance to measure Eidetic memorization: $\ell_2(a,b) = \sqrt{\sum_i (a_i - b_i)^2/d}$ where d is the input dimension.

In some of our extraction procedures however, we will use modified distance measures to better control for false-positives. We will introduce such distance measures when needed. In all cases, we use the standard ℓ_2 metric above to measure the success of the extraction process.

Restrictions of our definition. Our definition of extraction is intentionally conservative as compared to what privacy concerns one might ultimately have. For example, if we prompt Stable Diffusion to generate "A Photograph of Barack Obama," it produces an entirely recognizable photograph of Barack Obama but not an *near-identical reconstruction* of any particular training image. Figure 2 compares the generated image (left) to the 4 nearest training images under the Euclidean 2-norm (right). Under our memorization definition, this image would not count as memorized. Nevertheless, the model's ability to generate (new) recognizable pictures of certain individuals could still cause privacy harms.

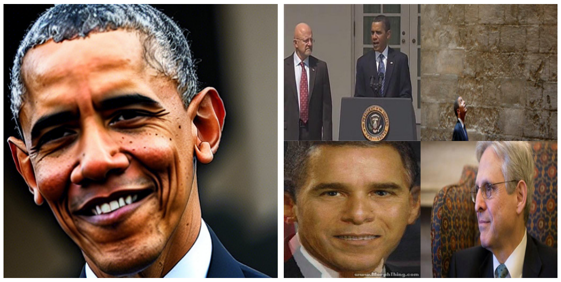


Figure 2: We do not count the generated image of Obama (at left) as memorized because it has a high ℓ_2 distance to every training image. The four nearest training images are shown at right, each has a distance above 0.3.

4.2 Extracting Data from Stable Diffusion

We now extract training data from Stable Diffusion: the largest and most popular open-source diffusion model [63]. This 890 million parameter text-conditioned diffusion model was trained on 160 million images. We use the default PLMS sampling scheme to generate images at a resolution of 512 × 512 pixels. As the model is trained on a publicly-available dataset, we can verify the success of our extraction process and also mitigate potential harms from exposing the extracted data. We begin with a black-box attack.

Identifying duplicates in the training data. To reduce the computational load of our extraction procedure, as is done in [70], we bias our search towards duplicated training examples because these are orders of magnitude more likely to be memorized than non-duplicated examples [45, 51].

If we search for bit-for-bit identically duplicated images in the training dataset, we would significantly undercount the true rate of duplication. And so ideally, we would search for training examples that are near-duplicated with a pixel-level ℓ_2 distance below some threshold. But this is computationally intractable, as it requires an all-pairs comparison of 160 million images in Stable Diffusion's training set, each of which is a $512 \times 512 \times 3$ dimensional vector. Instead, we first *embed* each image to a 512 dimensional vector using CLIP [59], and then perform the all-pairs comparison between images in this lower-dimensional space (increasing efficiency by over 1500×). We count two examples as near-duplicates if their CLIP embeddings have a high cosine similarity. For 350,000 near-duplicated images, we use the corresponding captions as the input to our extraction process.

4.2.1 Extraction Methodology

Our extraction approach adapts the methodology from prior work [12] to images and consists of two steps:

1. *Generate many examples* using the diffusion model in the standard sampling manner and with the known prompts from above.

[2] This paper covers a very restricted definition of "memorization": whether diffusion models can be induced to generate near-copies of some training examples when prompted with appropriate instructions. We will describe an approach that can generate images that are close approximations of some training images (especially images that are frequently represented in the training dataset through duplication or other means). There is active discussion within the technical and legal communities about whether the presence of this type of "memorization" suggests that generative neural networks "contain" their training data.

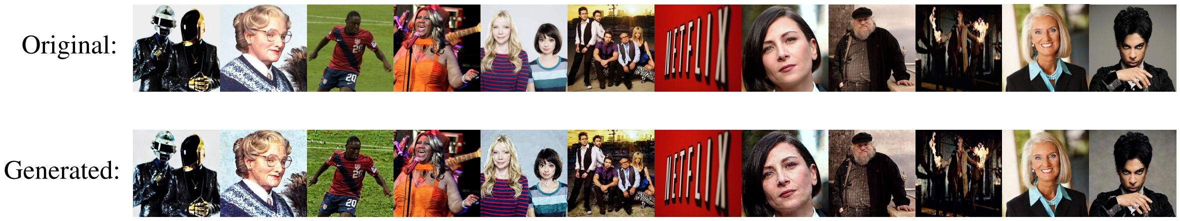


Figure 3: Examples of the images that we extract from Stable Diffusion v1.4 using random sampling and our membership inference procedure. The top row shows the original images and the bottom row shows our extracted images.

2. *Perform membership inference* to separate the model's novel generations from those generations which are memorized training examples.

Generating many images. The first step is simple but computationally expensive: we query the `Gen` function in a black-box manner with the selected prompts as input. To reduce the computational overhead, we use the timestep-resampled generation implementation from the Stable Diffusion codebase [63]. This process generates images in a more aggressive fashion by performing fewer (but larger) denoising steps. This results in reduced visual quality at a large ($\sim 10\times$) throughput increase. We generate 500 candidate images for each text prompt to increase the likelihood that we find memorization.

Performing membership inference. The second step requires flagging generations that appear to be memorized training images. Since we assume a black-box threat model in this section, we do not have access to the loss and cannot exploit techniques from state-of-the-art membership inference attacks [12]. We instead design a new membership inference attack strategy that only requires the ability to prompt the model for images (as assumed in our black-box threat model). Our attack is based on the intuition that for diffusion models, with high probability $\texttt{Gen}(p;r_1) \neq \texttt{Gen}(p;r_2)$ for two different random initial seeds r_1, r_2. On the other hand, if $\texttt{Gen}(p;r_1) \approx_d \texttt{Gen}(p;r_2)$ under some distance measure d, it is likely that these generated samples are memorized examples.

Recall that earlier we generated 500 images that for each prompt, each with a different (but unknown) random seeds. We can therefore construct a graph over the 500 generations by connecting an edge between generation i and j if $x_i \approx_d x_j$. Following the above intuition, the existence of a single edge in this graph is indicative of memorization. To minimize false-positives, we search for *cliques* of densely connected images. If the largest clique in this graph is at least size 10 (i.e., ≥ 10 of the 500 generations are near-identical), we predict that this clique is a memorized image. The clique size of 10 was manually tuned to achieve an acceptable false-positive rate.

We found that using the standard ℓ_2 metric as the similarity measure d leads to many false-positives (e.g., many generations have the same gray background and thus high ℓ_2 similarity). To build the cliques, we instead use a "tiled" ℓ_2 distance, that divides each image into 16 non-overlapping 128×128 tiles and measures the *maximum* ℓ_2 distance between a pair of image tiles of two images. This new distance measure is small only if two images are fairly close *everywhere*. Note that when we report that a sample is (ℓ, δ)-memorized in the sense of Definition 1, we always use the standard ℓ_2 metric.

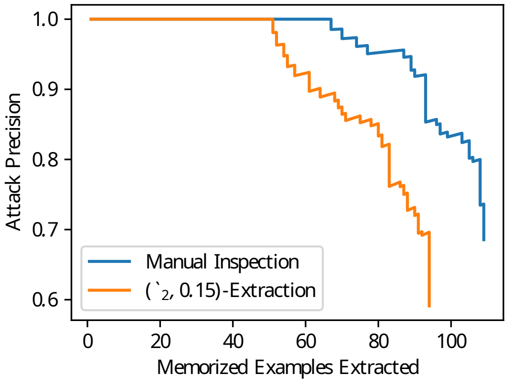


Figure 4: Our methodology reliably separates novel generations from memorized training examples, under two definitions of memorization—either $(\ell_2, 0.15)$-extraction or manual human inspection of generated images.

4.2.2 Extraction Results

To evaluate the effectiveness of our extraction methodology, we select the 350,000 most-duplicated examples from the training dataset and generate 500 candidate images for each of these prompts (totaling 175 million generated images). We first sort all generated images by the mean distance between the images in the clique to identify ones that we predict are likely to be memorized training examples. We then take each of these generated images and annotate each as either "extracted" or "not extracted" by comparing it to the original training images under Definition 1 (using the standard ℓ_2 metric). Note that here we are looking at the true training data solely for evaluation purposes. Our extraction procedure never sees the real images, only their captions.

We find 94 images are $(\ell_2, 0.15)$-extracted. We manually verify that these are all near-copies of training images. We also manually checked the top-1000 generated images, and find 13 extra images (for a total of 107 images) are near-copies of training data, even if their ℓ_2 distance is above 0.15. Figure 3 shows a subset of the images that are reproduced with near

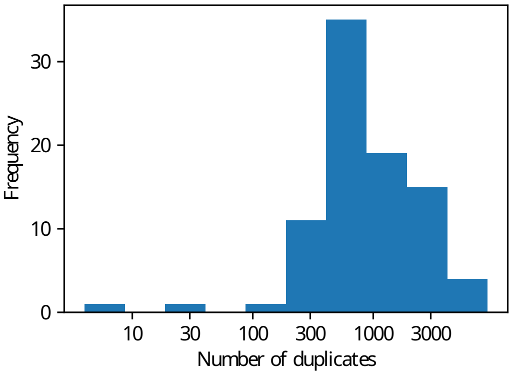


Figure 5: Most of the images we extract from Stable Diffusion have been duplicated at least $k = 100$ times; although this should be taken as an upper bound because our methodology explicitly searches for memorization of duplicated images.

pixel-perfect accuracy (all images have an ℓ_2 distance under 0.05) For comparison, encoding a PNG as a JPEG with quality level 50 gives a ℓ_2 difference of 0.02 on average.

Given our ordered set of annotated images, we can also compute a curve evaluating the number of extracted images to the attack's false positive rate. Our membership inference attack is exceptionally precise: out of 175 million generated images, we can identify 50 memorized images with 0 false positives, and all our memorized images can be extracted with a precision above 50%. Figure 4 contains the precision-recall curve for both memorization definitions.

Measuring (k, ℓ, δ)-eidetic memorization. In Definition 2 we introduce a variant of Eidetic memorization [12] tailored to generative image modeling. As mentioned earlier, we compute similarity between pairs of images with a standard ℓ_2 metric. This analysis is computationally expensive[3] as it requires comparing each of our memorized images against all 160 million training examples. We set $\delta = 0.1$, as we found that this threshold is sufficient to identify almost all small image corruptions (e.g., JPEG compression, small brightness/contrast adjustments) and has very few false positives.

Figure 5 shows the results. While we identify little Eidetic memorization for $k < 100$, this is expected due to the fact that we choose prompts of highly-duplicated images. Note that at this level of duplication, the duplicated examples make up just *one in a million* training examples. Overall, these results show that duplication is a major factor behind training data extraction. Our procedure has some false-positives (i.e., images marked as memorized when they are not) due to limitations of our distance measure. That is, a cluster of generated images may all be close in tiled ℓ_2 distance, while not being near-perfect copies. Using a more perceptually-aligned distance measure might thus lead to further extracted data.

Qualitative analysis. The majority of the images we extract (58%) are photographs with a recognizable person as the primary subject; the remainder are mostly products for sale (17%), logos/posters (14%), or other art or graphics. We caution that if a future diffusion model were trained on sensitive (e.g., medical) data, then the kinds of data that we extract would likely be drawn from this sensitive data distribution.

While all these images are publicly accessible on the Internet, not all of them are permissively licensed. Many of these images fall under an explicit non-permissive copyright notice (35%). Many other images (61%) have no explicit copyright notice but may fall under a general copyright protection for the website that hosts them (e.g., images of products on a sales website). Several of the images that we extracted are licensed CC BY-SA, which requires "[to] give appropriate credit, provide a link to the license, and indicate if changes were made." Stable Diffusion thus memorizes numerous copyrighted and non-permissive-licensed images, which the model may reproduce without the accompanying license.

4.3 Extracting Data from Imagen

While Stable Diffusion is the best publicly-available diffusion model, there are non-public models that achieve better performance with larger models and datasets [61, 65]. Prior work has found that larger models are more likely to memorize training data [10, 12]. We thus study Imagen [65], a 2 billion parameter text-to-image diffusion model. While Imagen's and Stable Diffusion's implementation and training scheme differ in some details, these are independent of our extraction results. As Imagen was trained on some non-public data [65], we refrain from displaying successfully extracted images, in line with the principles from Section 3.2.

We follow the same procedure as earlier but focus on the top-1000 most duplicated prompts for computational reasons. We generate 500 images for each of these prompts, and compute the ℓ_2 similarity between each generated image and the corresponding training image. By repeating the same membership inference steps as above—searching for cliques under tiled ℓ_2 distance–we identify 23 of these 1,000 images as memorized training examples.This is significantly higher than the rate of memorization in Stable Diffusion, and clearly demonstrates that memorization across diffusion models is highly dependent on training settings such as the model size, training time, and dataset size.

4.4 Extracting Outlier Examples

The extraction process presented above succeeds, but only at extracting images that are highly duplicated. This "high k" memorization may be problematic, but the most compelling

[3] In practice it is even more challenging: for non-square images, Stable Diffusion takes a random square crop, and so to check if the generated image x matches a non-square training image y we must try all possible alignments between x on top of the image y.

practical privacy risk would arise from memorization in the "low *k*" regime.

To find non-duplicated examples that are likely to be memorized, we take advantage of the fact that while on *average* models respect the privacy of the dataset, there often exists a small set of "outlier" examples whose privacy is more significantly exposed [27]. Therefore, we are more likely to succeed if we focus our effort on outlier examples. To find outlier examples, prior work trains hundreds of models on subsets of the training set and uses influence functions to identify examples with significant impact [28]. Unfortunately, given the cost of training even a single large diffusion model is in the millions-of-dollars, this approach is not feasible.

We take a simpler approach. We compute the CLIP embedding of each training example, and then compute the "outlierness" of each example as the average distance (in embedding space) to its 1,000 nearest neighbors in the training dataset.

Results. We find that out-of-distribution images can be successfully extracted even when they are not-duplicated. For Imagen, we try to extract the 500 images with the highest out-of-distribution score. Imagen memorized and regenerated 3 of these images (which were *not duplicated at all* in the training dataset). For Stable Diffusion, we failed to extract any image when applying the same methodology—even after attempting to extract the 10,000 highest outliers. Thus, Imagen appears less private than Stable Diffusion both on duplicated and non-duplicated images. We believe this is because Imagen uses a model with a much higher capacity compared to Stable diffusion, which allows for more memorization [10]. Moreover, Imagen is trained for more iterations and on a smaller dataset, which can also result in higher memorization.

5 Investigating Memorization

The above experiments are visually striking and clearly indicate that memorization is pervasive in large diffusion models—and that data extraction is feasible. But they ultimately only consider one strong form of memorization (near-exact extraction) on two state-of-the-art models. In this section we train smaller diffusion models on CIFAR-10 and perform controlled experiments to better understand how design choices in diffusion models affect memorization and vulnerability to privacy attacks. Specifically, we show that:

- We extract many samples (including non-duplicates) from unconditional CIFAR-10 diffusion models.
- While extraction works for ≈ 2.5% of CIFAR-10, *membership inference* succeeds for ≈ 70% of points (at a 1% false-positive rate). Thus, while full data extraction is obviously the most serious privacy concern, it may severely underestimate the total training data leakage.
- Targeted attacks on unconditional diffusion models can use *inpainting* to extract missing parts of an image.

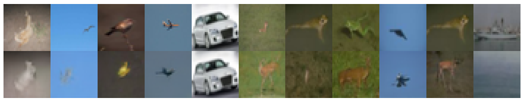


Figure 6: Direct 2-norm measurement fails to identify memorized CIFAR-10 examples. Each of the above images have a ℓ_2 distance of less than 0.05, yet only one (the car) is actually a memorized training example.

- Privacy leakage in diffusion models is strongest for larger and better models, as well as for outlier data.

Experimental setup. For the remainder of this section, we focus on diffusion models trained on CIFAR-10. We use state-of-the-art training code[4] to train 16 diffusion models, each on a randomly-partitioned half of the CIFAR-10 training dataset. We run three types of privacy attacks: membership inference, attribute inference, and data reconstruction. For the membership inference attacks, we train class-conditional models that reach a Fréchet Inception Distance (FID) [36] below 3.5 (see Figure 12), placing them in the top-30 generative models on CIFAR-10 [17]. For reconstruction attacks (Section 5.1) and attribute inference attacks with inpainting (Section 5.3), we train unconditional models with an FID below 4.

5.1 Untargeted Extraction

We first validate that memorization does still occur in our smaller models. Because these models are not text conditioned, we focus on *untargeted* extraction. Specifically, given our 16 diffusion models trained on CIFAR-10, we unconditionally generate 2^{16} images from each model for a total of 2^{20} candidate images. Because we will later develop high-precision membership inference attacks, here we directly search for memorized training examples among all our million generated examples. This is not a realistic attack, but just a way of verifying the capability of these models to memorize.

Identifying matches. In the prior section, we performed targeted attacks where we could check for successful extraction by computing the ℓ_2 distance between a target image and generated image. Here, as we perform an all-pairs comparison, we find that using an uncalibrated ℓ_2 threshold fails to accurately identify memorized training examples. For example, if we set a highly-restrictive threshold of 0.05, then nearly all "extracted" images are of entirely blue skies or green landscapes (see Figure 6). We explored several other metrics (including perceptual distances like SSIM or CLIP embedding distance)

[4] We use OpenAI's Improved Diffusion repository in Section 5.1 (`https://github.com/openai/improved-diffusion`), and our own re-implementation in all following sections. The models we train achieve almost identical FID to the open-sourced models. These models use state-of-the-art regularization techniques (e.g., early stopping, weight decay, and dropout). We use half the dataset as is standard in privacy analyses [8].

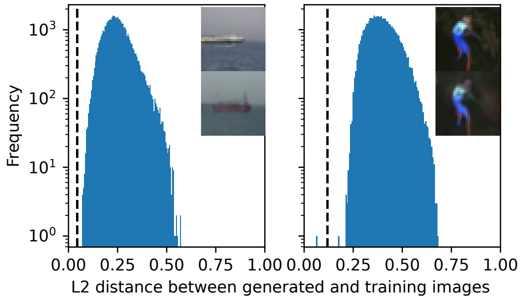


Figure 7: Per-image ℓ_2 thresholds are necessary to separate memorized images from novel generations on a CIFAR-10 model. Each plot shows the distribution of ℓ_2 distances from a generated image to all training images (along with the image and its nearest training image). **Left** shows a typical distribution for a non-memorized image. **Right** shows a memorized image distribution; while the closest training image has high absolute ℓ_2 distance, it is *abnormally* low for this distribution. The dashed black line shows our adaptive ℓ_2 threshold.

but found that none could reliably identify memorized training images for CIFAR-10.

We instead define an image as extracted if the ℓ_2 distance to its nearest neighbor in the training set is *abnormally low* compared to all other training images. Figure 7 illustrates this by computing the ℓ_2 distance between two different generated images and every image in the CIFAR-10 training dataset. The left figure shows a failed extraction attempt; despite the fact that the nearest training image has an ℓ_2 distance of just 0.06, this distance is on par with the distance to many other training images (i.e., all images that contain a blue sky). In contrast, the right plot shows a successful extraction. Here, even though the ℓ_2 distance to the nearest training image is higher than for the prior failed extraction (0.07), this value is *unusually small* compared to other training images which almost all are at a distance above 0.2.

We thus slightly modify our unconditional extraction procedure to use the distance

$$\ell(\hat{x},x;S_{\hat{x}}) = \frac{\ell_2(\hat{x},x)}{\alpha\cdot\frac{1}{k}\sum_{y\in S_{\hat{x}}}\ell_2(\hat{x},y)}.$$

where $S_{\hat{x}}$ is the set containing the n closest elements from the training dataset to the example $\hat{x}$. This distance is small if the extracted image x is much closer to the training image $\hat{x}$ compared to the n closest neighbors of $\hat{x}$ in the training set. We run our extraction procedure with $\alpha = 0.5$ and $n = 50$. [5]

Results. The above methodology identifies 1,280 unique extracted images from the CIFAR-10 dataset (2.5% of the dataset).[6] Figure 8 shows a selection of training examples that we extract; full results are in the extended version [9, Figure 18]. This demonstrates that small-scale diffusion models trained on CIFAR-10 memorize a substantial amount of data.

5.2 Membership Inference Attacks

We now evaluate membership inference with traditional attack techniques that use white-box access, as opposed to the black-box attacks in Section 4.2.1. We will show that *all* examples have significant privacy leakage under membership inference attacks, compared to the small fraction that are sensitive to data extraction. We consider two membership inference attacks on our class-conditional CIFAR-10 diffusion models.

The loss threshold attack. Yeom *et al.* [85] introduce a simple membership inference attack: because models are trained to minimize their loss on the training set, we should expect that training examples have lower loss than non-training examples. The loss threshold attack thus computes the loss $l = \mathcal{L}(x;f)$ and reports "member" if $l < \tau$ for some threshold τ and otherwise "non-member'. The value of τ can be selected to maximize a desired metric (e.g., true positive rate at some fixed false positive rate or the overall attack accuracy).

The Likelihood Ratio Attack (LiRA). Carlini *et al.* [8] introduce a state-of-the-art approach for membership inference attacks. LiRA first trains multiple *shadow models*, each model on a random subset of the training dataset. LiRA then computes the loss $\mathcal{L}(x;f_i)$ for the example x under each of these shadow models f_i. These losses are split into two sets: the losses $\texttt{IN} = \{l^{\text{in}_i}\}$ for the example x under the shadow models $\{f_i\}$ that *did* see the example x during training, and the losses $\texttt{OUT} = \{l^{\text{out}_i}\}$ for the example x under the shadow models $\{f_j\}$ that *did not* see the example x during training. LiRA finishes the initialization process by fitting Gaussians N_{IN} to the $\texttt{IN}$ set and N_{OUT} to $\texttt{OUT}$ set of losses. Finally, to predict membership inference for a new model f^*, we compute $l^* = \mathcal{L}(x,f^*)$ and then measure whether $\Pr[l^*|N_{IN}] > \Pr[l^*|N_{OUT}]$.

Choosing a loss function. Both membership inference attacks use a loss function $\mathcal{L}$. For classification models, Carlini *et al.* [8] find that the choice of loss function has a large impact on the attack. We find that this effect is even more pronounced for diffusion models. In particular, unlike classifiers that are trained with a fixed loss function (e.g., cross entropy), the reconstruction loss minimized by diffusion models depends on the magnitude of Gaussian noise ε added to the image. Thus, "the loss" of an image is not well defined—instead, we have a set of loss values $\mathcal{L}(x,t,\varepsilon)$ of an image x at some timestep t with a corresponding amount of noise ε (cf. Equation (1)).

We must thus choose an optimal timestep t at which to measure the loss. To do so, we re-run our LiRA membership inference attack (with 16 shadow models) by varying the

[5] Our results were not sensitive to these choices: setting $\alpha \in [0.3, 0.7]$ and $n \in [10, 100]$ yield results within 30% of the values reported below.

[6] Some CIFAR-10 training images are generated multiple times. For these, we only count the first generation as a successful attack. Further, because the CIFAR-10 training dataset contains many duplicates, we do not count two generations of two different (but duplicated) images in the training dataset.

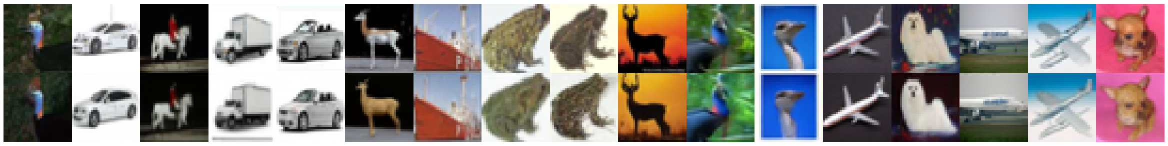


Figure 8: Selected training examples that we extract from a diffusion model trained on CIFAR-10 by sampling from the model one million times. **Top** row: generated output from a diffusion model. **Bottom** row: nearest (ℓ_2) example from the training dataset. All 1,280 unique extracted images are in the extended version [9, Figure 18].

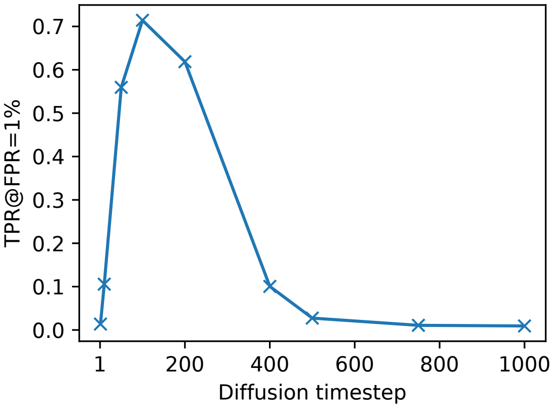


Figure 9: We run membership inference using LiRA and compute the diffusion model loss at different noise timesteps on CIFAR-10. Evaluating $\mathcal{L}(\cdot,t,\cdot)$ at $t \in [50,300]$ produces the best results. We use $t = 100$ for all remaining experiments.

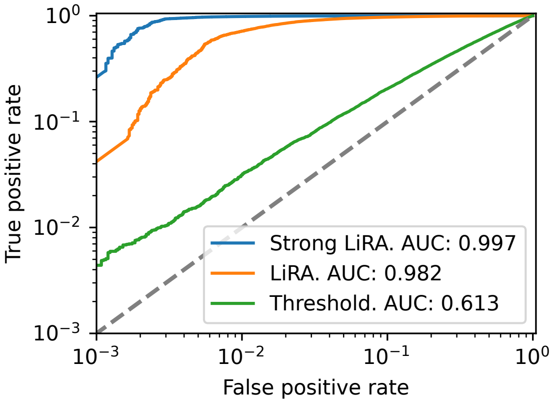


Figure 10: Membership inference ROC curve for a diffusion model trained on CIFAR-10 using the loss threshold attack, baseline LiRA, and "Strong LiRA" with repeated queries and augmentation (§5.2.2).

timestep $t \in [1,T]$ at which we compute the loss ($T = 1{,}000$ in the models we use).

Figure 9 plots the timestep used to compute the loss against the attack success rate, measured as the true positive rate (TPR), i.e., the number of examples which truly are members over the total number of members, at a fixed false positive rate (FPR) of 1%, i.e., the fraction of examples which are incorrectly identified as members. Evaluating $\mathcal{L}$ at $t \in [50,300]$ leads to the most successful attacks. We conjecture that this a "Goldilock's zone" for membership inference: if t is too small, and so the noisy image is similar to the original, then predicting the added noise is easy regardless if the input was in the training set; if t is too large, and so the noisy image is similar to Gaussian noise, then the task is too difficult. Our remaining experiments will evaluate $\mathcal{L}(\cdot,t,\cdot)$ at $t = 100$, where we observed a TPR of 71% at an FPR of 1%.

5.2.1 Baseline Attack Results

We evaluate membership inference using our specified loss function. We follow recent advice [8] and evaluate the efficacy of membership inference attacks by comparing their true positive rate to the false positive rate on a log-log scale. In Figure 10, we plot the membership inference ROC curve for the loss threshold attack and LiRA. An out-of-the-box implementation of LiRA achieves a true positive rate of over 70% at a false positive rate of just 1%. As a point of reference, state-of-the-art *classifiers* are much more private, e.g., with a $< 20\%$ TPR at 1% FPR [8]. This shows that diffusion models are significantly less private than classifiers trained on the same data. (In part this may be because diffusion models are often trained far longer than classifiers.)

5.2.2 Augmentations Improve Attacks

Membership inference attacks can be improved by reducing the variance in the loss [8, 84]. We achieve this for diffusion models in two ways. First, because our loss function is randomized (recall that the reconstruction loss $\mathcal{L}(x,t,\varepsilon)$ is computed with random noise $\varepsilon \sim \mathcal{N}(0,I)$), a better estimate of the true loss is the expected loss: $\mathcal{L}(x,t) = \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,I)}[\mathcal{L}(x,t,\varepsilon)]$. By increasing the number of samples used to estimate this expectation we can increase the attack success rate.

Second, because our diffusion models train on *augmented* training images (e.g., by flipping images horizontally), we can further average the loss over all possible augmentations. As in prior work for classifiers, we find that both of these strategies increase the efficacy of membership inference attacks [8, 43].

Improved attack results. Figure 10 shows the effect of combining both strategies (Strong LiRA). Together they are remarkably successful, and at a false positive rate of 0.1% they increase the true positive rate by over a factor of six from 7%

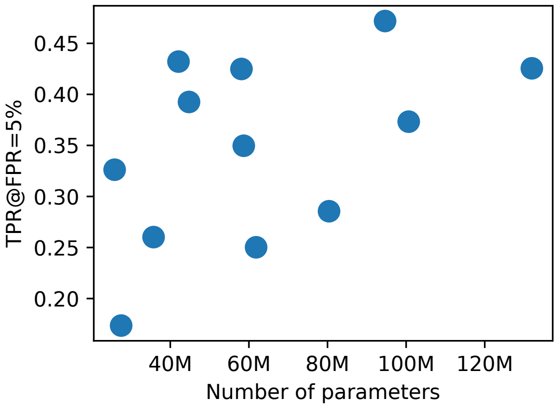


Figure 11: We train models on CIFAR-10 varying the number of trainable parameters between 25M and 130M, and measure the TPR at a fixed FPR of 5%. Larger models are, in general, more vulnerable to membership inference attacks.

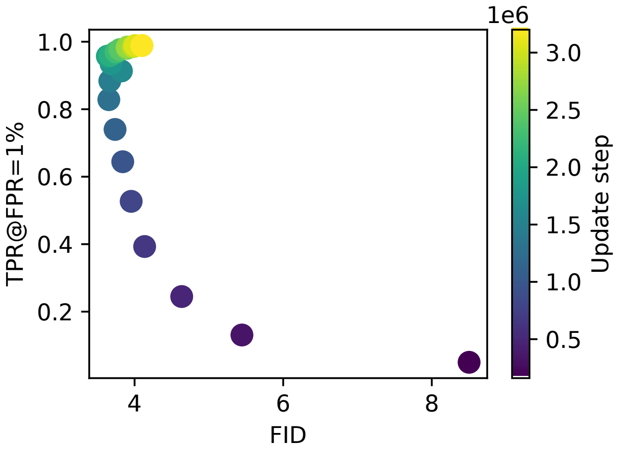


Figure 12: Better diffusion models are more vulnerable to membership inference attacks; at an FPR of 1%, the membership inference TPR grows from 7% to nearly 100% as model FID decreases (i.e., quality increases).

to 44%. Figure 21 in the extended version [9] breaks down the impact of each component, by either increasing the number of Monte Carlo samples from 1 (the base LiRA attack) to 20, or by augmenting samples with a horizontal flip.

5.2.3 Factors influencing Memorization

Outliers. Outlier examples are easier to memorize and extract than inliers, as we observed in Section 4.4 where extraction on Imagen succeeded orders of magnitude more often for OOD samples. In this section we directly quantify this effect.

To do this, we first use CLIP embeddings to compute outlier scores (as we did previously) by computing each examples' embedding distance to its nearest neighbor. We say an example is an "outlier" if its nearest neighbor is in the bottom 10 percentile, and an "inlier" if it is in the top 10 percentile. Even though attack success rates are very high for both outliers (95.6%) and inliers (94.1%), this difference is statistically significant ($p < 0.0001$). Figure 19 in the extended version [9] has a visual representation of this difference.

Model capacity. Prior work shows that larger language models tend to memorize more training data than smaller models [12, 39, 51]. Figure 11 shows a similar trend for diffusion models trained on CIFAR-10. We train models with 25M to 130M parameters, and ensure that all models reach a similar FID. We find that privacy leakage roughly scales with model size. The smallest 25M model has the smallest TPR of 17.3% (at a FPR of 5%), while the 130M model has a TPR of 42.6%. However, the trend is non-monotonic: a 58M model can exhibit similar vulnerability to membership inference attacks.

Model utility. So far, we trained our diffusion models to reach state-of-the-art performance. Prior work on language models found that better models are often *easier* to attack—intuitively, because they extract more information from the training dataset [10]. Here we perform a similar experiment.

Our previous CIFAR-10 results used models that reach the best FID (on average 3.5) with early stopping. Here we evaluate models over the course of training and report the attack success rate as a function of FID in Figure 12. We find that the privacy leakage increases with the quality of the diffusion model. This is concerning because it suggests that future stronger diffusion models may be even less private.

5.3 Inpainting Attacks

Having performed untargeted extraction on CIFAR-10 models, we now turn to targeted attacks. As mentioned earlier, targeted attacks are tricky here because the models we use do not support textual prompting. We thus instead provide guidance by performing a form of attribute inference attack [42, 85, 86] that we call an "inpainting attack". Given an image x, we first mask out a portion of the image's pixels to create a masked image x_m. Our attack objective is then to reconstruct the full image. We then run this attack on both training and testing images, and compare the attack efficacy on each. Specifically, we use the inpainting algorithm of Lugmayr *et al.* [52] to produce a reconstructed image x_{rec}.

Because inpainting is stochastic (it depends on the random sample $\varepsilon \sim \mathcal{N}(0, I)$), we create a set of inpainted images $X_{rec} = \{x^1_{rec}, x^2_{rec}, \ldots, x^n_{rec}\}$, where we set $n = 5{,}000$. For each $x_{rec} \in X_{rec}$, we compute the diffusion model's loss on this sample (at timestep 100) divided by a shadow model's loss that was not trained on the sample. We then use this score to identify the highest-scoring reconstructions $x_{rec} \in X_{rec}$.

Results. Our attack masks out the left half of an image and applies the diffusion model to inpaint the missing part. We

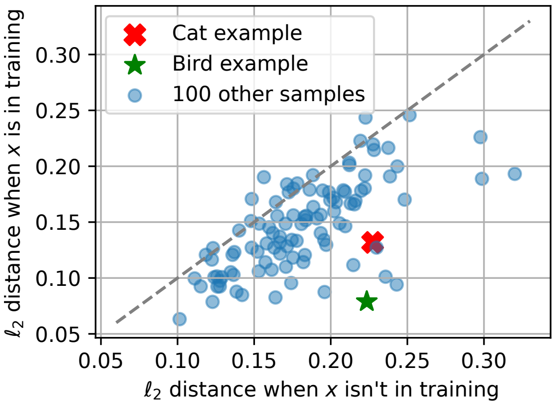


Figure 13: Evaluation of inpainting attacks on CIFAR-10. For 100 randomly chosen images, we mask out the image's left half and plot the ℓ_2 distance between the image and the inpainted reconstruction. The annotated bird and cat examples are from Figure 14. With partial knowledge of an image, inpainting attacks work far better than full data extraction.

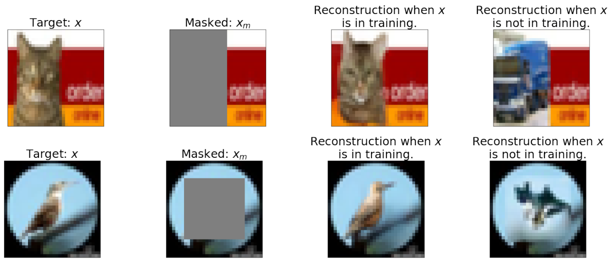

Target: x
Masked: x_m
Reconstruction when x is in training.
Reconstruction when x is not in training.
Target: x
Masked: x_m
Reconstruction when x is in training.
Reconstruction when x is not in training.

Figure 14: Inpainting-based reconstruction attack on CIFAR-10. Given an image (first column), we randomly mask half of the image (second column), and then inpaint with a model which was trained on the image (third column) or with a model which was not trained on the image (fourth column).

repeat this process 5,000 times and take the top-10 scoring reconstructions using a membership inference attack. We repeat this attack for 100 images using diffusion models that are trained with and without these images. Figure 13 compares the average distance between the original sample and the ten highest scoring inpainted samples. We find that our inpainting attack does exploit memorization: the reconstruction loss is substantially lower when the image is in the training set than when not. Figure 14 shows qualitative examples of this attack. The highest-scoring reconstruction is visually similar to the target image when the target is in training and does not resemble the target when it is not in training. A more thorough analysis of inpainting attacks is in the extended version [9, Appendix E].

6 Comparing Diffusion Models to GANs

Are diffusion models more or less private than other generative modeling approaches? In this section we take a first look at this question by comparing diffusion models to Generative Adversarial Networks (GANs) [34, 60, 66], which were state-of-the-art for image generation for nearly a decade.

Diffusion models are explicitly trained to reconstruct their training datasets—GANs are not. Instead, training a GAN pits two competing neural networks against each other: a generator and a discriminator. Similar to diffusion models, the generator receives random noise as input, but unlike a diffusion model, it must convert this noise to an image in a single forward pass. To train a GAN, the discriminator is trained to predict if an image comes from the generator or not, and the generator is trained to fool the discriminator. GANs differ from diffusion models in that their generators are only trained using *indirect* information about the training data (i.e., gradients from the discriminator), whereas diffusion models are explicitly trained to reconstruct the training set.

Membership inference attacks. We first propose a privacy attack methodology for GANs.[7] We initially focus on membership inference attacks, where following Balle *et al.* [5], we assume access to both the discriminator and generator. We perform membership inference using the loss threshold [85] and LiRA [8] attacks, where we use the discriminator's loss as the metric. To perform LiRA, we follow a similar methodology as Section 5 and train 256 individual GAN models each on a random 50% split of the CIFAR-10 training dataset but otherwise leave training hyperparameters unchanged.

We study three GAN architectures, all implemented using the StudioGAN framework [46]: BigGAN [6], MHGAN [79], and StyleGAN [48]. The membership inference results are in the extended version [9, Figure 15]. Overall, diffusion models have higher membership inference leakage, e.g., diffusion models had 50% TPR at a FPR of 0.1% as compared to $<$ 30% TPR for GANs. This suggests that diffusion models are less private than GANs for membership inference attacks under default training settings, even when the GAN attack is strengthened due to having access to the discriminator (which would be unlikely in practice, as only the generator is necessary to create new images).

Data extraction results. We next consider more practical black-box extraction attacks. We follow the same procedure as in Section 5.1, where we generate 2^{20} images from each model architecture and identify those that are near-copies of the training data using the aforementioned similarity function. Again we only consider non-duplicated CIFAR-10 training images in our counting. Note that in contrast to the experiments in Section 5.1, here we do not need to assume that

[7] While existing privacy attacks exist for GANs, they were proposed before the latest advancements in privacy attack techniques, requiring us to develop our own methods which out-perform prior work.

Architecture		Images Extracted	FID
GANs	StyleGAN-ADA [47]	**150**	**2.9**
	DiffBigGAN [87]	57	4.6
	E2GAN [73]	95	11.3
	NDA [68]	70	12.6
	WGAN-ALP [72]	49	13.0
DDPMs	OpenAI-DDPM [57]	**301**	**2.9**
	DDPM [37]	232	3.2

Table 1: The number of training images extracted from off-the-shelf generative models, with one million unconditional generations. We show GAN models on the top and diffusion models on the bottom, sorted by FID (lower is better). Overall, diffusion models memorize more than GANs, and better generative models (lower FID) tend to memorize more data.

the adversary knows any captions (since the models are unguided). This extraction attack could thus seemingly apply in any setting where an unconditional generative model is used.

For this experiment, instead of training models ourselves (which was necessary to run LiRA), we study five off-the-shelf pre-trained GANs: WGAN-ALP [72], E2GAN [73], NDA [68], DiffBigGAN [87], and StyleGAN-ADA [47]. We also evaluate two off-the-shelf DDPM diffusion models from Ho *et al.* [37] and Nichol *et al.* [57]. Note that all of these pre-trained models were trained on the entire CIFAR-10 dataset.

Table 1 shows the number of extracted images and FID for each model. Overall, diffusion models memorize more data than similarly strong GANs. Notably, the best DDPM model memorizes 2× more than a StyleGAN-ADA with similar FID. Memorization (in GANs and diffusion models) also tends to increase as quality (FID) improves, e.g., we extract 3× more images from StyleGAN-ADA than the weakest GANs.

More results are in the extended version of our paper [9]. In [9, Figure 16] we show examples of near-copy generations for the GANs we trained ourselves, and in [9, Figure 26] we show every sample that we extract from those models. In [9, Figure 27], we show near-copy generations for the five off-the-shelf GANs. These results reinforce our conclusion that diffusion models are less private than GANs.

Surprisingly, we find that diffusion models and GANs memorize many of the same images. In particular, for a diffusion model that memorizes 1280 images and a StyleGAN model (trained on half of CIFAR-10) that memorizes 361 images, we find *244 unique images memorized in common*. If images were memorized at random, we should expect on average 10 images would be memorized by both, giving strong evidence that some images ($p < 10^{-261}$) are less private than others.

7 Defenses and Recommendations

Given the degree to which diffusion models memorize and regenerate training examples, in this section we explore various defenses and practical strategies that may help to reduce and audit model memorization.

7.1 Deduplicating Training Data

In Section 4.2, we showed that extracted examples are often duplicated many times (e.g., > 100) in the training data. Similar results have been shown for language models [12,44]; data deduplication is an effective mitigation against memorization for those models [45,51]. In the image domain, simple deduplication is common, where images with identical URLs and captions are removed. But most datasets do not compute other inter-image similarity metrics such as ℓ_2 distance or CLIP similarity. We thus encourage practitioners to deduplicate datasets using these more advanced notions of duplication.

Unfortunately, deduplication is not a perfect solution. To better understand its effectiveness, we deduplicate CIFAR-10 and re-train a diffusion model on this modified dataset. We compute image similarity using the `imagededup` tool and deduplicate any images that have a similarity above > 0.85. This removes 5,275 examples from the 50,000 total examples in CIFAR-10. We repeat the same generation procedure as Section 5.1, where we generate 2^{20} images from the model and count how many examples are regenerated from the training set. The model trained on the deduplicated data regenerates 986 examples, as compared to 1280 for the original model. While not a substantial drop, these results show that deduplication can mitigate memorization. Moreover, we also expect that deduplication will be much more effective for models trained on larger-scale datasets (e.g., Stable Diffusion), as we observed a much stronger correlation between data extraction and duplication rates for those models.

7.2 Differentially-Private Training

The gold standard technique to defend against privacy attacks is by training with differential privacy (DP) guarantees [24, 25]. Diffusion models can be trained with differentially-private stochastic gradient descent (DP-SGD) [1], where the model's gradients are clipped and noised to prevent the model from leaking substantial information about the presence of any individual image in the dataset. Applying DP-SGD induces a trade-off between privacy and utility, and recent work shows that DP-SGD can be applied to small-scale diffusion models without substantial performance degradation [21].

Unfortunately, applyinbg DP-SGD to our diffusion model codebase causes the training on CIFAR-10 to consistently diverge, even at high privacy budgets $\varepsilon \geq 50$. In fact, even applying a non-trivial gradient clipping or noising on their own (both are required in DP-SGD) causes the training to fail. We leave a further investigation of these failures to future work, and we believe that new advances in DP-SGD and privacy-preserving training techniques may be required to train diffusion models in privacy-sensitive settings.

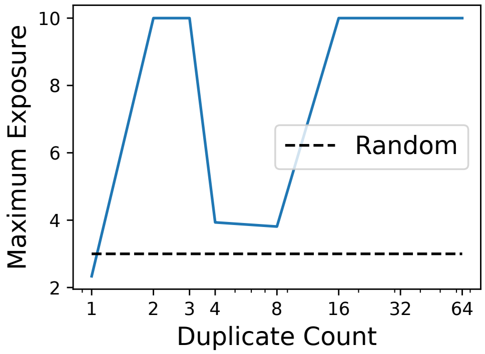


Figure 15: Canary *exposure* (a measure of non-privacy) as a function of duplicate count. Inserting a canary twice is sufficient to reach maximum exposure.

7.3 Auditing with Canaries

In addition to implementing defenses, practitioners may want to empirically audit their models to determine their vulnerability in practice [40]. Our attacks above represent one method to evaluate model privacy. Nevertheless, they are expensive, e.g., our membership inference results require training many shadow models, and thus cheaper alternatives may be desired.

A common, cheap approach to evaluate memorization in language models [11] is to insert canary examples into the training set. Here, one creates a large "pool" of *canaries*, e.g., random noise, and inserts a subset of the canaries into the training set. After training, we compute the canaries' *exposure*, a measure of how much lower the model's loss is on the inserted canaries compared to the held-out canaries. This approach only requires training one model and can be instantiated with canaries ranging from random inputs all the way to adversarial inputs designed to maximize memorization.

To evaluate exposure for diffusion models, we generate canaries consisting of uniformly random noise, and duplicate these in the training set at different rates. Figure 15 shows the results. The maximum exposure is 10, and some canaries reach this after being inserted only twice. The exposure does not strictly increase with duplicate count, which may be a result of some canaries being "harder" than others. Ultimately, the random canaries we generate may not be the most effective to test memorization for diffusion models.

For the models we considered here, we could thus provide clear evidence of data leakage by training a single model, instead of the multiple shadow models needed for strong MI attacks. When auditing less leaky models however, canary exposures computed from a single training might underestimate the true data leakage [77]. Thoroughly investigating the design and evaluation of canary auditing schemes for diffusion models is an important question for future work.

8 Related Work

Memorization in language models. Numerous prior works study memorization in generative models across different domains, architectures, and threat models. Memorization in language models for text has been an active area of research, which showed that adversaries can extract training samples using two-step attack techniques that resemble our approach [12, 44, 45, 51]. Our work differs in that we focus on the image domain, and we use more semantic notions of data regeneration (e.g., using CLIP scores) as opposed to exact verbatim repetition (although recent language modeling work has begun to explore approximate memorization as well [39]).

Memorization in image generation. Past work has analyzed memorization in image generation mainly from the perspective of generalization in GANs (i.e., the novelty of model generations). For instance, numerous metrics exist to measure similarity to training data [3, 36], the extent of mode collapse [16, 66], and the impact of individual training samples [4, 80]. Other work provides insights into when and why GANs may replicate training examples [29, 55], as well as how to mitigate such effects [55]. Our work extends these lines of inquiry to conditional diffusion models, where we measure novelty by computing how frequently models regenerate training instances when provided with textual prompts.

Recent and concurrent work studies privacy in image generation for both GANs [74] and diffusion models [38, 70, 83]. Tinsley *et al.* [74] show that StyleGAN can generate individuals' faces, and Somepalli *et al.* [70] show that Stable Diffusion can output semantically similar images to its training set. Compared to these works, we identify privacy vulnerabilities in a wider range of systems (e.g., Imagen and CIFAR models) and threat models (e.g., membership inference attacks).

Inverting image generation models. Many successful applications of image generation models (e.g., image editing) leverage the ability to *invert* the generation process [18, 88]. That is, given a generator `Gen` and target image x, it is possible to find some input z to the generator such that $\texttt{Gen}(z) \approx x$.

While seemingly similar to extraction, existing inversion methods ask for something much weaker: they search for a "soft" continuous high-dimensional embeddings that generates a close target. The ability to invert a generator is thus a property of the (soft) generator function being *surjective*, rather than the model having memorized training data. Indeed, inversion methods [18, 32, 88] work for *any* training or testing sample. In contrast, our extraction attacks search for "hard" text prompts, which only represent a negligibly sparse fraction of the embedding space. To illustrate, while we may be able to find a continuous embedding that makes Stable Diffusion generate anyone's face, doing so with a text prompt of the person's name (as in Figure 1) can obviously only work if the person's face was in the training set and memorized.

9 Discussion and Conclusion

State-of-the-art diffusion models memorize and regenerate individual training images when prompted with the corresponding captions. By training our own models we find that increasing utility can degrade privacy, and simple defenses such as deduplication are insufficient to completely address the memorization challenge. We see that state-of-the-art diffusion models memorize 2× more than comparable GANs, and more useful diffusion models memorize more than weaker diffusion models. This suggests that the vulnerability of generative image models may grow over time. Going forward, our work raises questions around the memorization and generalization capabilities of diffusion models.

Questions of generalization. Do large-scale models work by generating novel output, or do they just copy and interpolate between all training examples? Since we succeed in extracting some training samples (but not all), this question remains open. Given that different models memorize varying amounts of data, we hope future work will be able to explain how diffusion models regenerate parts of their training datasets.

Our work also highlights the difficulty in defining *memorization*. While we have found extensive memorization with a simple ℓ_2 metric, a more comprehensive analysis will be necessary to capture more nuanced definitions of memorization that allow for more human-aligned notions of data copying.

Practical consequences. We highlight four practical consequences for training and deploying diffusion models. First, we recommend deduplicating training data and minimizing overtraining, as a first defense layer. Second, we suggest using our attack methods, or other auditing approaches, to estimate the privacy risk of trained models. Third, we recommend using provable privacy-preserving techniques once this becomes practical. Fourth, we hope our work will temper the heuristic privacy expectations associated with diffusion model outputs: synthetic data does not give privacy for free [2, 14, 15, 58, 64].

Our work contributes to the growing literature on the legal, ethical, and privacy implications of training on public web data [7, 70, 76, 82]. Researchers and practitioners should be wary of training on uncurated public data without taking steps to understand the underlying ethics and privacy risks.

Acknowledgements and Conflicts of Interest

The authors are grateful to Tom Goldstein, Olivia Wiles, Katherine Lee, Austin Tarango, Ian Wilbur, Jeff Dean, Andreas Terzis, Robin Rombach, and Andreas Blattmann, and the reviewers for comments on early drafts of this paper. Nicholas, Milad, Matthew, Daphne, Jamie, and Borja are employed at Google, a company that trains large machine learning models (including diffusion models) on both public and private datasets. Eric Wallace is supported by the Apple Scholars in AI/ML Fellowship.

Contributions

- Nicholas, Jamie, Vikash, and Eric each independently proposed the problem statement of extracting training data from diffusion models.
- Nicholas, Eric, and Florian did preliminary experiments to identify cases of data extraction in diffusion models.
- Milad performed most of the experiments on Stable Diffusion and Imagen, and Nicholas counted duplicates in the LAION training dataset; each wrote the corresponding sections of the paper.
- Jamie performed the membership inference attacks and inpainting attacks on CIFAR-10 diffusion models, and Nicholas performed the diffusion extraction experiments; each wrote the corresponding sections of the paper.
- Matthew ran experiments for canary memorization and wrote the corresponding section of the paper.
- Florian and Vikash performed preliminary experiments on memorization in GANs, and Milad and Vikash ran the experiments included in the paper.
- Milad ran membership inference experiments on GANs.
- Vikash ran extraction experiments on pretrained GANs.
- Daphne and Florian improved figure clarity and presentation.
- Daphne, Borja, and Eric edited the paper and contributed to paper framing.
- Nicholas organized the project and wrote the initial draft.

References

[1] Martín Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM CCS*, 2016.

[2] Hazrat Ali, Shafaq Murad, and Zubair Shah. Spot the fake lungs: Generating synthetic medical images using neural diffusion models. *arXiv preprint arXiv:2211.00902*, 2022.

[3] Sanjeev Arora, Andrej Risteski, and Yi Zhang. Do GANs learn the distribution? Some theory and empirics. In *International Conference on Learning Representations*, 2018.

[4] Yogesh Balaji, Hamed Hassani, Rama Chellappa, and Soheil Feizi. Entropic GANs meet VAEs: A statistical approach to compute sample likelihoods in GANs. In *International Conference on Machine Learning*, 2019.

[5] Borja Balle, Giovanni Cherubin, and Jamie Hayes. Reconstructing training data with informed adversaries. In *IEEE Symposium on Security and Privacy*, 2022.

	NC	MN	JH	MJ	FT	VS	BB	DI	EW
Conceived Project	X		X			X			X
Formalized Memorization Definition	X	X	X	X	X		X		
Experimented with Stable Diffusion	X	X							
Experimented with Imagen		X							
Experimented with CIFAR-10 Diffusion	X		X						
Experimented with GANs		X			X	X			
Experimented with Defenses	X	X		X					
Prepared Figures	X	X	X	X		X		X	X
Analyzed Data	X	X	X	X	X	X			
Wrote Paper	X	X	X	X	X	X	X	X	X
Managed the Project	X								

Table 2: Contributions of each author in the paper.

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[7] Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. What does it mean for a language model to preserve privacy? In *ACM Conference on Fairness, Accountability, and Transparency*, 2022.

[8] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *IEEE Symposium on Security and Privacy*. IEEE, 2022.

[9] 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 (extended version). *arXiv preprint arXiv:2301.13188*, 2023.

[10] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

[11] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, 2019.

[12] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security Symposium*, 2021.

[13] Andrew Carr. Gretel.ai: Diffusion models for document synthesis. `https://gretel.ai/blog/diffusion-models-for-document-synthesis`, 2022.

[14] Pierre Chambon, Christian Bluethgen, Jean-Benoit Delbrouck, Rogier Van der Sluijs, Małgorzata Połacin, Juan Manuel Zambrano Chaves, Tanishq Mathew Abraham, Shivanshu Purohit, Curtis P. Langlotz, and Akshay Chaudhari. RoentGen: Vision-language foundation model for chest X-ray generation. *arXiv preprint arXiv:2211.12737*, 2022.

[15] Pierre Chambon, Christian Bluethgen, Curtis P. Langlotz, and Akshay Chaudhari. Adapting pretrained vision-language foundational models to medical imaging domains. *arXiv preprint arXiv:2210.04133*, 2022.

[16] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. In *International Conference on Learning Representations*, 2016.

[17] Papers With Code. `https://paperswithcode.com/sota/image-generation-on-cifar-10`, 2023.

[18] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems*, 30(7):1967–1974, 2018.

[19] Elise Devaux. List of synthetic data vendors. `https://elise-deux.medium.com/f06dbe91784`, 2022.

[20] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems*, 2021.

[21] Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. *arXiv preprint arXiv:2210.09929*, 2022.

[22] Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. Are diffusion models vulnerable to membership inference attacks? *arXiv preprint arXiv:2302.01316*, 2023.

[23] August DuMont Schütte, Jürgen Hetzel, Sergios Gatidis, Tobias Hepp, Benedikt Dietz, Stefan Bauer, and Patrick Schwab. Overcoming barriers to data sharing with medical image generation: a comprehensive evaluation. *NPJ Digital Medicine*, 2021.

[24] C Dwork, F McSherry, K Nissim, and A Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, 2006.

[25] Cynthia Dwork. Differential privacy: A survey of results. In *TAMC*, 2008.

[26] Benj Edwards. Artist finds private medical record photos in popular AI training data set. https://arstechnica.com/information-technology/2022/09/artist-finds-private-medical-record-photos-in-popular-ai-training-data-set, 2022.

[27] Vitaly Feldman. Does learning require memorization? A short tale about a long tail. In *ACM SIGACT Symposium on Theory of Computing*, 2020.

[28] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 2020.

[29] Qianli Feng, Chenqi Guo, Fabian Benitez-Quiroz, and Aleix M Martinez. When do GANs replicate? on the choice of dataset size. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[30] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *ACM Conference on Computer and Communications Security (CCS)*, 2015.

[31] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *USENIX Security Symposium*, 2014.

[32] 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.

[33] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[34] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 2014.

[35] Peter Henderson, Koustuv Sinha, Nicolas Angelard-Gontier, Nan Rosemary Ke, Genevieve Fried, Ryan Lowe, and Joelle Pineau. Ethical challenges in data-driven dialogue systems. In *AAAI/ACM Conference on AI, Ethics, and Society*, 2018.

[36] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 2017.

[37] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020.

[38] Hailong Hu and Jun Pang. Membership inference of diffusion models. *arXiv preprint arXiv:2301.09956*, 2023.

[39] Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas Carlini. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv preprint arXiv:2210.17546*, 2022.

[40] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private SGD? *Advances in Neural Information Processing Systems*, 2020.

[41] Ali Jahanian, Xavier Puig, Yonglong Tian, and Phillip Isola. Generative models as a data source for multiview representation learning. *International Conference on Learning Representations*, 2021.

[42] Bargav Jayaraman and David Evans. Are attribute inference attacks just imputation? *ACM Conference on Computer and Communications Security (CCS)*, 2022.

[43] Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. Revisiting membership inference under realistic assumptions. *Proceedings on Privacy Enhancing Technologies*, 2020.

[44] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. *arXiv preprint arXiv:2211.08411*, 2022.

[45] Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models. *International Conference on Machine Learning*, 2022.

[46] MinGuk Kang, Joonghyuk Shin, and Jaesik Park. StudioGAN: A Taxonomy and Benchmark of GANs for Image Synthesis. *arXiv preprint arXiv:2206.09479*, 2022.

[47] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems*, 2020.

[48] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[49] Salome Kazeminia, Christoph Baur, Arjan Kuijper, Bram van Ginneken, Nassir Navab, Shadi Albarqouni, and Anirban Mukhopadhyay. GANs for medical image analysis. *Artificial Intelligence in Medicine*, 2020.

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

[51] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Association for Computational Linguistics*, 2022.

[52] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[53] Midjourney. https://www.midjourney.com/, 2022.

[54] AnGeL Ministries. File:Anne Graham Lotz (October 2008). https://commons.wikimedia.org/wiki/File:Anne_Graham_Lotz_(October_2008).jpg. Accessed on December 2022.

[55] Vaishnavh Nagarajan, Colin Raffel, and Ian J Goodfellow. Theoretical insights into memorization in GANs. In *Neural Information Processing Systems Workshop*, 2018.

[56] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[57] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, 2021.

[58] Walter H. L. Pinaya, Petru-Daniel Tudosiu, Jessica Dafflon, Pedro F da Costa, Virginia Fernandez, Parashkev Nachev, Sebastien Ourselin, and M. Jorge Cardoso. Brain imaging generation with latent diffusion models. *arXiv preprint arXi:2209.07162*, 2022.

[59] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[60] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016.

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

[62] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, 2021.

[63] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[64] Pouria Rouzrokh, Bardia Khosravi, Shahriar Faghani, Mana Moassefi, Sanaz Vahdati, and Bradley J. Erickson. Multitask brain tumor inpainting with diffusion models: A methodological report. *arXiv preprint arXiv:2210.12113*, 2022.

[65] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[66] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *Advances in Neural Information Processing Systems*, 2016.

[67] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017.

[68] Abhishek Sinha, Kumar Ayush, Jiaming Song, Burak Uzkent, Hongxia Jin, and Stefano Ermon. Negative data augmentation. In *International Conference on Learning Representations*, 2021.

[69] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015.

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

[71] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 2019.

[72] Dávid Terjék. Adversarial lipschitz regularization. In *International Conference on Learning Representations*, 2019.

[73] Yuan Tian, Qin Wang, Zhiwu Huang, Wen Li, Dengxin Dai, Minghao Yang, Jun Wang, and Olga Fink. Off-policy reinforcement learning for efficient and effective gan architecture search. In *European Conference on Computer Vision*, 2020.

[74] Patrick Tinsley, Adam Czajka, and Patrick Flynn. This face does not exist... but it might be yours! Identity leakage in generative models. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021.

[75] Rob Toews. Synthetic data is about to transform artificial intelligence. https://www.forbes.com/sites/robtoews/2022/06/12/synthetic-data-is-about-to-transform-artificial-intelligence/, 2022.

[76] Florian Tramèr, Gautam Kamath, and Nicholas Carlini. Considerations for differentially private learning with large-scale public pretraining. *arXiv preprint arXiv:2212.06470*, 2022.

[77] Florian Tramèr, Reza Shokri, Ayrton San Joaquin, Hoang Le, Matthew Jagielski, Sanghyun Hong, and Nicholas Carlini. Truth serum: Poisoning machine learning models to reveal their secrets. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2779–2792, 2022.

[78] Allan Tucker, Zhenchen Wang, Ylenia Rotalinti, and Puja Myles. Generating high-fidelity synthetic patient data for assessing machine learning healthcare software. *NPJ Digital Medicine*, 2020.

[79] Ryan Turner, Jane Hung, Eric Frank, Yunus Saatchi, and Jason Yosinski. Metropolis-hastings generative adversarial networks. In *International Conference on Machine Learning*, 2019.

[80] Gerrit van den Burg and Chris Williams. On memorization in probabilistic deep generative models. *Advances in Neural Information Processing Systems*, 2021.

[81] James Vincent. The lawsuit that could rewrite the rules of AI copyright. https://www.theverge.com/2022/11/8/23446821/microsoft-openai-github-copilot-class-action-lawsuit-ai-copyright-violation-training-data, 2022.

[82] Eric Wallace, Florian Tramèr, Matthew Jagielski, and Ariel Herbert-Voss. Does GPT-2 know your phone number? *BAIR Blog*, 2020.

[83] Yixin Wu, Ning Yu, Zheng Li, Michael Backes, and Yang Zhang. Membership inference attacks against text-to-image generation models. *arXiv preprint arXiv:2210.00968*, 2022.

[84] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2021.

[85] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE Computer Security Foundations Symposium (CSF)*, 2018.

[86] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *CVPR*, 2020.

[87] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient GAN training. *Advances in Neural Information Processing Systems*, 2020.

[88] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016.