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.

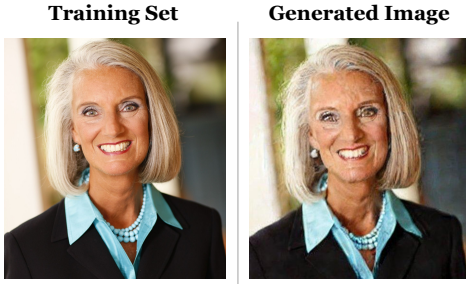


Figure 1: Diffusion models memorize individual training examples and generate them at test time. **Left:** an image from Stable Diffusion's training set (licensed CC BY-SA 3.0, see [49]). **Right:** a Stable Diffusion generation when prompted with "Ann Graham Lotz". The reconstruction is nearly identical (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 [64, 66, 33]. Compared to prior approaches such as GANs [30] or VAEs [46], diffusion models produce higher-quality samples [18] and are easier to scale [56] and control [51]. Consequently, they have rapidly become the de-facto method for generating high-resolution images, and large-scale models such as DALL-E 2 [56] 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. Indeed, past large-scale training efforts "do not find overfitting to be an issue", [60] and researchers in privacy-sensitive domains have even suggested that diffusion models could "protect[] the privacy [...] of real images" [37] by generating synthetic examples [13, 14, 59, 2, 53]. This line of work relies on the assumption that diffusion models do not memorize and regenerate their training data. If they did, it would violate all privacy guarantees and raise numerous questions regarding model generalization and "digital forgery" [65].

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 attack that generates images using standard approaches, and flags those that exceed certain membership inference scoring criteria. Applying this method to Stable Diffusion [58] and Imagen [60], 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 how and why 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 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 [29, Chapter 20]). Generative Adversarial Networks (GANs) [30] were the breakthrough that first enabled the generation of high-fidelity images at scale [6, 44]. But over the last two years, diffusion models [64] have largely displaced GANs: they achieve state-of-the-art results on academic benchmarks [18] and form the basis of all recently popularized image generators such as Stable Diffusion [58], DALL-E 2 [57, 56], Runway [58], Midjourney [67] and Imagen [60].

Denoising Diffusion Probabilistic Models [33][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 $\frac{1}{N}\sum_i \mathbb{E}_{t,\varepsilon}\mathscr{L}(x_i,t,\varepsilon;f_\theta)$, where

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

Despite being trained with this simple denoising objective, diffusion models can *generate* high-quality images by first sampling a random vector $z_T \sim \mathcal{N}(0,I)$ and then applying the diffusion model f_θ to remove the noise from this random "image". 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 a particular class of image. Text-conditioned models take this one step further and take as input the text embedding of some *prompt* (e.g., "a photograph of a horse on the moon") using a pre-trained language encoder (e.g., CLIP [54]).

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

Training data privacy attacks. Neural networks often leak details of their training datasets. Membership inference attacks [62, 80, 8] answer the question "was this example in the training set?" and present a mild privacy breach. Neural networks are also vulnerable to more powerful attacks such as inversion attacks [27, 81] that extract representative examples from a target class, attribute inference attacks [28] that reconstruct subsets of attributes of training examples, and extraction attacks [10, 11, 5] that completely recover 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. Wu *et al.* [78] and Hu *et al.* [34] 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.* [65] 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 worst-case privacy under a much more restrictive notion of memorization, and perform our attacks on a wider range of models.

3 Motivation and Threat Model

There are two distinct motivations for understanding how diffusion models 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 [11, 7, 31]. For example, memorizing and regenerating copyrighted text [11] and source code [35] has been pointed to as indicators of potential copyright infringement [76]. Similarly, copying images from professional artists has been called "digital forgery" [65] and has spurred debate in the art community.

Future diffusion models might also be trained on more sensitive private data. Indeed, GANs have already been applied to medical imagery [73, 20, 45], which underlines the importance of understanding the risks of generative models *before* we apply them to private domains.

Worse, a growing literature suggests that diffusion models could create synthetic training data to "protect the privacy and usage rights of real images" [37], and production tools already claim to use diffusion models to protect data privacy [71, 17, 12]. Our work shows diffusion models may be unfit for this purpose.

Understanding generalization. Beyond data privacy, understanding how and why diffusion models memorize training data may help us understand 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 over-fitting to be an issue, and believe further training might improve overall performance" [60], 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 [24] may extend to generative 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

Our threat model considers an adversary $\mathscr{A}$ that interacts with a diffusion model $\texttt{Gen}$ (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 a fresh random noise r as input. Conditional models are trained on annotated images (e.g., labeled or captioned) $D = \{(x_1, c_1), \ldots, (x_n, c_n)\}$ and 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 $\texttt{Gen}$ to generate images. If $\texttt{Gen}$ 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 $\texttt{Gen}$ 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.

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

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 subset of the image) and aims to recover the full image. This is an image-analog of an *attribute inference attack* [80], which aims to recover unknown features from partial knowledge of an 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 attacks 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 available online. LAION, for example, contains unintentionally released medical images of several patients [23]. 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 an advance copy of this paper with the authors of each of the large-scale diffusion models that we study. This gave the authors and their corresponding organizations the ability to consider possible safeguards and software changes ahead of time.

In total, we believe that publishing our paper and publicly disclosing these privacy vulnerabilities is both ethical and responsible. Indeed, at the moment, no one appears to be immediately harmed by the (lack of) privacy of diffusion models; our goal with this work is thus to make sure 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

Most existing literature on training data extraction focuses on text 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 [11, 41]. Because we work with high-resolution images, verbatim definitions of memorization are not suitable. Instead, we define a notion of approximate memorization based on image similarity metrics.

Definition 1 ((ℓ,δ)-Diffusion Extraction) [adapted from [11]]. *We say that an example x is* extractable *from a diffusion model f_θ if there exists an efficient algorithm $\mathscr{A}$ (that does not receive x as input) such that $\hat{x} = \mathscr{A}(f_\theta)$ 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. In this paper, unless otherwise noted we follow Balle *et al.* [5] and use the Euclidean 2-norm distance $\ell_2(a,b) = \sqrt{\sum_i (a_i - b_i)^2 / d}$ where d is the dimension of the inputs to normalize $\ell \in [0,1]$. Given this definition of extractability, we can now define *memorization*.

Definition 2 ((k,ℓ,δ)-Eidetic Memorization) [adapted from [11]]. *We say that an example x is (k,ℓ,δ)-Eidetic memorized [2] 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 is its 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.

[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 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.

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.

4.2 Extracting Data from Stable Diffusion

We now extract training data from Stable Diffusion: the largest and most popular open-source diffusion model [58]. This model is an 890 million parameter text-conditioned diffusion model trained on 160 million images. We generate from the model using the default PLMS sampling scheme at a resolution of 512×512 pixels. As the model is trained on publicly-available images, we can easily verify our attack's success 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 attack, as is done in [65], we bias our search towards duplicated training examples because these are orders of magnitude more likely to be memorized than non-duplicated examples [47, 41].

If we search for images that are bit-for-bit identically duplicated in the training dataset, we would significantly undercount the true rate of duplication. Instead, we account for near-duplication. Ideally, we would search for any training examples that are nearly duplicated with a

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.

pixel-level ℓ_2 distance below some threshold. But this is computationally intractable, as it would require 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 [54], and then perform the all-pairs comparison between images in this lower-dimensional space (increasing efficiency by over $1500\times$). We count two examples as near-duplicates if their CLIP embeddings have a high cosine similarity. For each of these near-duplicated images, we use the corresponding captions as the input to our extraction attack.

4.2.1 Extraction Methodology

Our extraction approach adapts the methodology from prior work [11] 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 the prior section.
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 trivial but computationally expensive: we query the `Gen` function in a black-box manner using the selected prompts as input. To reduce the computational overhead of our experiments, we use the timestep-resampled generation implementation that is available in the Stable Diffusion codebase [58]. This process generates images in a more aggressive fashion by removing larger amounts of noise at each time step and results in slightly lower visual fidelity at a significant ($\sim 10\times$) performance 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 [11]. We instead design a new membership inference attack strategy 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.

The 500 images that we generate for each prompt have 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 $x_i \approx_d x_j$. 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. Empirically, clique-finding is more effective than searching for *pairs* of images $x_1 \approx_d x_2$ as it has fewer false positives.

To compute the distance measure d among the images in the clique, we use a modified Euclidean ℓ_2 distance. In particular, we found that many generations were often spuriously similar according to ℓ_2 distance (e.g., they all had gray background). We therefore instead divide each image into 16 non-overlapping 128×128 tiles and measure the maximum of the ℓ_2 distance between any pair of image tiles between the two images.

4.2.2 Extraction Results

In order to evaluate the effectiveness of our attack, 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 of these generated images by ordering them by the mean distance between images in the clique to identify generations that we predict are likely to be memorized training data. We then take each of these generated images and annotate each as either "extracted" or "not extracted" by comparing it to the training images under Definition 1. We find 94 images are $(\ell_2, 0.15)$-extracted. To ensure that these images not only match

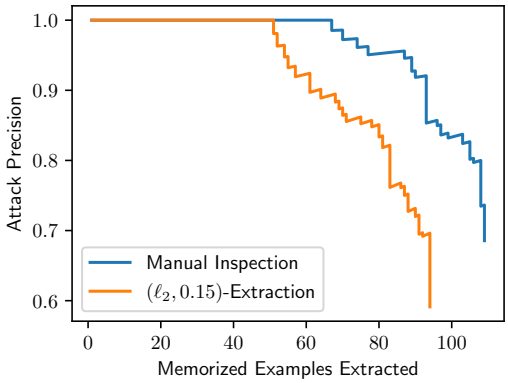


Figure 4: Our attack reliably separates novel generations from memorized training examples, under two definitions of memorization—either $(\ell_2, 0.15)$-extraction or manual human inspection of generated images.

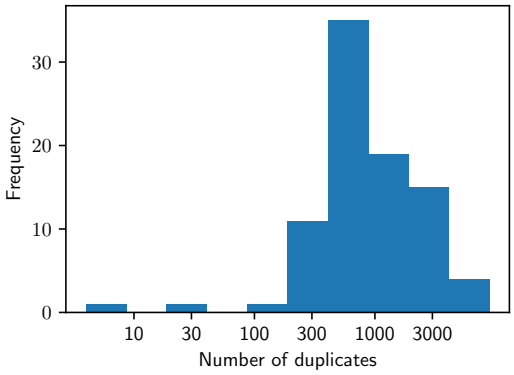


Figure 5: Our attack extracts images from Stable Diffusion most often when they have been duplicated at least $k = 100$ times; although this should be taken as an upper bound because our methodology explicitly searches for memorization of duplicated images.

some arbitrary definition, we also manually annotate the top-1000 generated images as either memorized or not memorized by visual analysis, and find that a further 13 (for a total of 109 images) are near-copies of training examples even if they do not fit our 2-norm definition. Figure 3 shows a subset of the extracted images that are reproduced with near pixel-perfect accuracy; all images have an ℓ_2 difference under 0.05. (As a point of reference, re-encoding a PNG as a JPEG with quality level 50 results in an 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 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 introduced an adaptation of Eidetic memorization [11] tailored to the domain of generative image models. As mentioned earlier, we compute similarity between pairs of images with a direct ℓ_2 pixel-space similarity. This analysis is computationally expensive[3] as it requires comparing each of our memorized images against each of the 160 million training examples. We set $\delta = 0.1$ as this threshold is sufficient to identify almost all small image corruptions (e.g., JPEG compression, small brightness/contrast adjustments) but has very few false positives.

Figure 5 shows the results of this analysis. While we identify little Eidetic memorization for $k < 100$, this is expected due to the fact we choose prompts of highly-duplicated images. Note that at this level of duplication, the duplicated examples still make up just *one in a million* training examples. These results show that duplication is a major factor behind training data extraction.

Qualitative analysis. The majority of the images that we extract (58%) are photographs with a recognizable person as the primary subject; the remainder are mostly either 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.

Despite the fact that these images are publicly accessible on the Internet, not all of them are permissively licensed. We find that a significant number 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-

[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.

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 stronger performance using larger models and datasets [56, 60]. Prior work has found that larger models are more likely to memorize training data [11, 9] and we thus study Imagen [60], a 2 billion parameter text-to-image diffusion model. While individual details differ between Imagen's and Stable Diffusion's implementation and training scheme, these details are independent of our extraction results.

We follow the same procedure as earlier but focus on the top-1000 most duplicated prompts for computational reasons. We then 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 patched ℓ_2 distance–we identify 23 of these 1,000 images as memorized training examples.[4] 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 attacks presented above succeed, but only at extracting images that are highly duplicated. This "high k" memorization may be problematic, but as we mentioned previously, the most compelling practical attack would be to demonstrate memorization in the "low k" regime.

We now set out to achieve this goal. In order to find non-duplicated examples likely to be memorized, we take advantage of the fact that while on *average* models often respect the privacy of the majority of the dataset, there often exists a small set of "outlier" examples whose privacy is more significantly exposed [24]. And so instead of searching for memorization across all images, we are more likely to succeed if we focus our effort on these outlier examples.

But how should we find which images are potentially outliers? Prior work was able to train hundreds of models on subsets of the training dataset and then use an influence-function-style approach to identify examples that have a significant impact on the final model weights [25]. Unfortunately, given the cost of training even a single large diffusion model is in the millions-of-dollars, this approach will not be feasible here.

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

Results. Surprisingly, we find that attacking out-of-distribution images is much more effective for Imagen than it is for Stable Diffusion. On Imagen, we attempted extraction of the 500 images with the highest out-of-distribution score. Imagen memorized and regurgitated 3 of these images (which were *unique* in the training dataset). In contrast, we failed to identify *any* memorization when applying the same methodology to Stable Diffusion—even after attempting to extract the 10,000 most-outlier samples. Thus, Imagen appears less private than Stable Diffusion both on duplicated and non-duplicated images. We believe this is due to the fact that Imagen uses a model with a much higher capacity compared to Stable diffusion, which allows for more memorization [9]. 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 these experiments do not explain *why* and *how* these models memorize training data. In this section we train smaller diffusion models and perform controlled experiments in order to more clearly understand memorization.

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 [5] 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 attacks, attribute in-

[4] Unfortunately, because the Imagen training dataset is not public, we are unable to provide visual examples of successful reconstructions.

[5] We either directly use OpenAI's Improved Diffusion repository (`https://github.com/openai/improved-diffusion`) in Section 5.1, or our own re-implementation in all following sections. Models trained with our re-implementation achieve almost identical FID to the open-sourced models. We use half the dataset as is standard in privacy analyses [8].

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.

ference attacks, and data reconstruction attacks. For the membership inference attacks, we train class-conditional models that reach an FID below 3.5 (see Figure 11), placing them in the top-30 generative models on CIFAR-10 [16]. 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

Before devling deeper into understanding memorization, we begin by validating 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, in this section we directly search for memorized training examples among all our million generated examples. Thus this is not an attack *per se*, but rather verifying the capability of these models to memorize.

Identifying matches. In the prior section, we performed targeted attacks and could therefore check for successful memorization by simply computing the ℓ_2 distance between the target image and the 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) 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

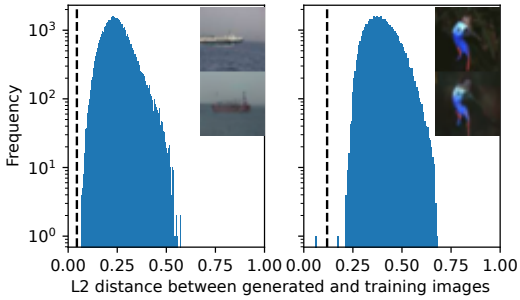


Figure 7: Per-image ℓ_2 thresholds are necessary to separate memorized images from novel generations on a CIFAR-10 model. Each plot shows the distribution of ℓ_2 distances from a generated image to all training images (along with the image and the nearest training image). **Left** shows a typical distribution for a non-memorized image. **Right** shows a memorized image distribution; while the most similar 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.

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 attack. Here, even though the ℓ_2 distance to the nearest training image is higher than for the prior failed attack (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 attack to use the distance

$$\ell(\hat{x}, x; S_{\hat{x}}) = \frac{\ell_2(\hat{x}, x)}{\alpha \cdot \mathbb{E}_{y \in S_{\hat{x}}}[\ell_2(\hat{x}, y)]}.$$

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 attack with $\alpha = 0.5$ and $n = 50$. Our attack was not sensitive to these choices.

Results. Using the above methodology we identify 1,280 unique extracted images from the CIFAR-10 dataset (2.5% of the entire dataset).[6] In Figure 8 we show a selection of training examples that we extract and full results are shown in Figure 17 in the Appendix.

[6]Some CIFAR-10 training images are generated multiple times. In these cases, we only count the first generation as a successful attack. Further, because the CIFAR-10 training dataset contains many duplicate images, 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 1 million times. **Top** row: generated output from a diffusion model. **Bottom** row: nearest (ℓ_2) example from the training dataset. Figure 17 in the Appendix contains all 1,280 unique extracted images.

5.2 Membership Inference Attacks

We now evaluate membership inference with more traditional attack techniques that use white-box access, as opposed to Section 4.2.1 that assumed black-box access. 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-trained diffusion models.[7]

The loss threshold attack. Yeom *et al.* [80] introduce the simplest 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 $l = \mathscr{L}(x; f)$ and reports "member" if $l < \tau$ for some chosen 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 the state-of-the-art approach to performing membership inference attacks. LiRA first trains a collection of *shadow models*, each model on random subsets of the training dataset. LiRA then computes the loss $\mathscr{L}(x; f_i)$ 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 IN set and N_{OUT} to OUT set of losses. Finally, to predict membership inference for a new model f^*, we compute $l^* = \mathscr{L}(x, f^*)$ and then measure whether $Pr[l^*|N_{IN}] > Pr[l^*|N_{OUT}]$.

Choosing a loss function. Both membership inference attacks use a loss function $\mathscr{L}$. In the case of classification models, Carlini *et al.* [8] find that choosing a loss function is one of the most important components of the attack. We find that this effect is even more pronounced for diffusion models. In particular, unlike classifiers that have a single loss function (e.g., cross entropy) used to train the model, diffusion models are trained to minimize the reconstruction loss when a random quantity of Gaussian noise ε has been added to an image. This means that "the loss" of an image is not well defined—instead, we can only ask for the loss $\mathscr{L}(x, t, \varepsilon)$ of an image x for a certain timestep t with a corresponding amount of noise ε (cf. Equation (1)).

We must thus compute the optimal timestep t at which we should measure the loss. To do so, we train 16 shadow models each on a random 50% of the CIFAR-10 training dataset. We then compute the loss for every model, for every example in the training dataset, and every timestep $t \in [1, T]$ ($T = 1{,}000$ 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 $\mathscr{L}$ 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 $\mathscr{L}(\cdot, t, \cdot)$ at $t = 100$, where we observed a TPR of 71% at an FPR of 1%.

5.2.1 Baseline Attack Results

We now 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

[7] Appendix C.4 replicates these results for unconditional models.

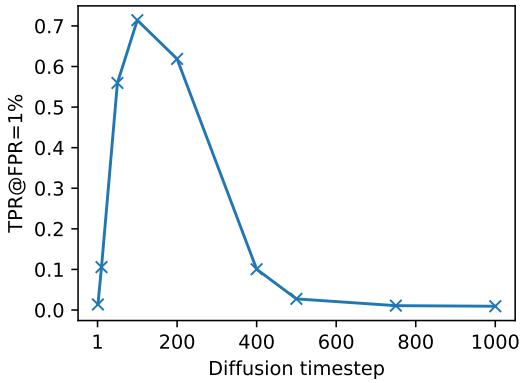


Figure 9: We run membership inference using LiRA and compute the diffusion model loss at different noise timesteps on CIFAR-10. Evaluating $\mathscr{L}(\cdot,t,\cdot)$ at $t \in [50,300]$ produces the best results.

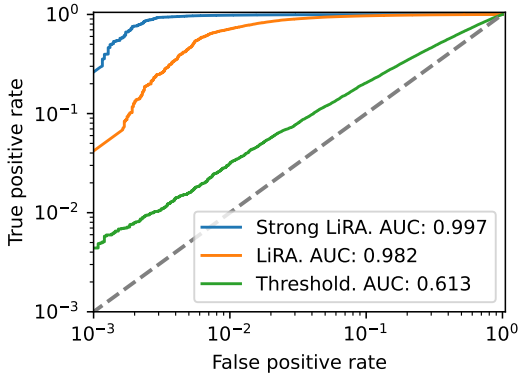


Figure 10: Membership inference ROC curve for a diffusion model trained on CIFAR-10 using the loss threshold attack, baseline LiRA, and "Strong LiRA" with repeated queries and augmentation (§5.2.2).

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.)

Qualitative analysis. In Figure 20, we visualize the least- and most-private images as determined by their easiness to detect via LiRA. We find that the easiest-to-attack examples are all extremely out-of-distribution visually from the CIFAR-10 dataset. These images are even more visually out-of-distribution compared to the outliers identified by Feldman *et al.* [24] who produce a similar set of images but for image *classifiers*. In contrast, the images that are hardest to attack are *all* duplicated images. It is challenging to detect the presence or absence of each of these images in the training dataset because there is another *identical* image in the training dataset that may have been present or absent—therefore making the membership inference question ill-defined.

5.2.2 Augmentations Improve Attacks

Membership inference attacks can also be improved by reducing the variance in the loss signal [8, 79]. We study two ways to achieve this for diffusion models. First, because our loss function has randomness (recall that to compute the reconstruction loss we measure the quantity $\mathscr{L}(x,t,\varepsilon)$ for a random noise sample $\varepsilon \sim \mathcal{N}(0,I)$), we can compute a better estimate of the true loss by averaging over different noise samples: $\mathscr{L}(x,t) = \mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I)}[\mathscr{L}(x,t,\varepsilon)]$.

By varying the number of point samples taken to estimate this expectation we can potentially increase the attack success rate. And second, because our diffusion models train on *augmented* versions of training images (e.g., by flipping images horizontally), it makes sense to compute the loss averaged over all possible augmentations. Prior work has found that both of these attack strategies are effective at increasing the efficacy of membership inference attacks for classifiers [8, 39], and we find they are effective here as well.

Improved attack results. Figure 10 shows the effect of combining both these strategies. 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% to 44%. Figure 19 in the Appendix breaks down the impact of each component: in Figure 19a we increase the number of Monte Carlo samples from 1 (the base LiRA attack) to 20, and in Figure 19b we augment samples with a horizontal flip.

5.2.3 Memorization Versus Utility

We train our diffusion models to reach state-of-the-art levels of performance. Prior work on language models has found that better models are often *easier* to attack than less accurate models—intuitively, because they extract more information from the same training dataset [9]. Here we perform a similar experiment.

Attack results vs. FID. To evaluate our generative models, we use the standard Fréchet Inception Distance (FID) [32], where lower scores indicate higher quality. Our previous CIFAR-10 results used models that

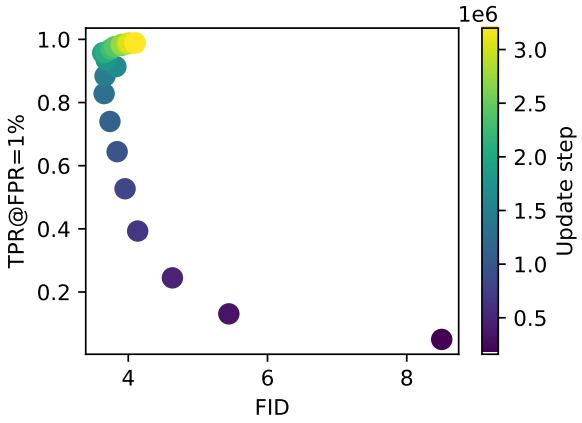


Figure 11: Better diffusion models are more vulnerable to membership inference attacks; evaluating with TPR at an FPR of 1%. As the FID decreases (corresponding to a quality increase) the membership inference attack success rate grows from 7% to nearly 100%.

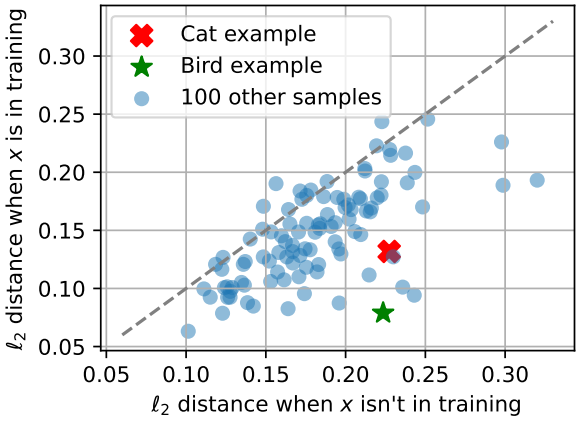


Figure 12: Evaluating inpainting attacks on 100 CIFAR-10 examples, measuring the ℓ_2 distance between images and their inpainted reconstructions when we mask out the left half of the image for 100 randomly selected images. We also plot the ℓ_2 distances for the bird and cat examples shown in Figure 13. When an adversary has partial knowledge of an image, inpainting attacks work far better than typical data extraction.

achieved the best FID (on average 3.5) based on early stopping. Here we evaluate models over the course of training in Figure 11. We compute the attack success rate as a function of FID, and we find that as the quality of the diffusion model increases so too does the privacy leakage. These results are concerning because they suggest that stronger diffusion models of the future may be even less private.

5.3 Inpainting Attacks

Having performed untargeted extraction on CIFAR-10 models, we now construct a targeted version of our attack. As mentioned earlier, performing a targeted attack is complicated by the fact that these models do not support textual prompting. We instead provide guidance by performing a form of attribute inference attack [38, 80, 81] that we call an "inpainting attack". Given an image, we first mask out a portion of this image; our attack objective is to recover the masked region. We then run this attack on both training and testing images, and compare the attack efficacy on each. Specifically, for an image x, we mask some fraction of pixels to create a masked image x_m, and then use the trained model to reconstruct the image as x_{rec}. The exact algorithm we use for inpainting is given in Lugmayr *et al.* [48].

Because diffusion model 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 specific attack masks out the left half of an image and applies the diffusion model on the right half of the image to inpaint the rest. We repeat this process 5000 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 the images. Figure 12 compares the average distance between the sample and the ten highest scoring inpainted samples. This allows us to show our inpainting attacks have succeed: the reconstruction loss is substantially better in terms of ℓ_2 distance when the image is in the training set than when not. Figure 13 also shows qualitative examples of this attack. The highest-scoring reconstruction looks visually similar to the target image when the target is in training and does not resemble the target when it is not in training. Overall, these results show that an adversary who has partial knowledge of an image can substantially improve their extraction results. We conduct a more thorough analysis of inpainting attacks in Appendix D.

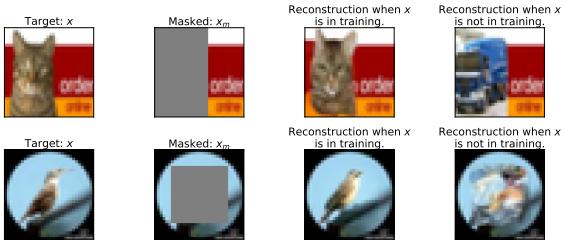


Figure 13: Inpainting-based reconstruction attack on CIFAR-10. Given an image from CIFAR-10 (first column), we randomly mask half of the image (second column), and then inpaint the image for a model which contained this image in the training set (third column) versus inpainting the image for a model which did not contain this image in the training set (fourth column).

Architecture		Images Extracted	FID
GANs	StyleGAN-ADA [43]	**150**	**2.9**
	DiffBigGAN [82]	57	4.6
	E2GAN [69]	95	11.3
	NDA [63]	70	12.6
	WGAN-ALP [68]	49	13.0
DDPMs	OpenAI-DDPM [52]	**301**	**2.9**
	DDPM [33]	232	3.2

Table 1: The number of training images that we extract from different off-the-shelf pretrained generative models out of 1 million unconditional generations. We show GAN models sorted by FID (lower is better) on the top and diffusion models on the bottom. Overall, we find that diffusion models memorize more than GAN models. Moreover, better generative models (lower FID) tend to memorize more data.

6 Comparing Diffusion Models to GANs

Are diffusion models more or less private than competing generative modeling approaches? In this section we take a first look at this question by comparing diffusion models to Generative Adversarial Networks (GANs) [30, 61, 55], an approach that has held the state-of-the-art results for image generation for nearly a decade.

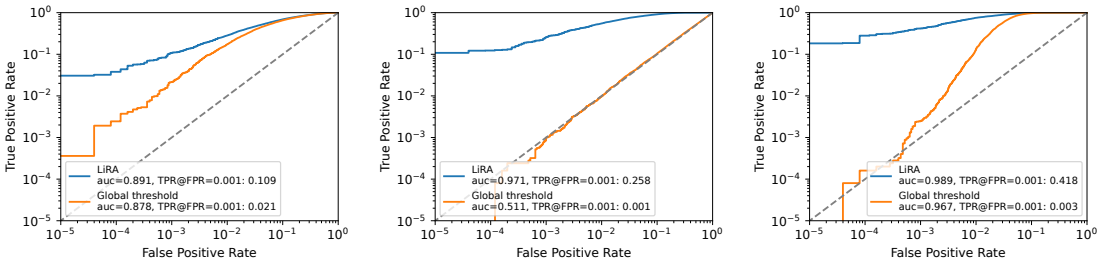
Unlike diffusion models that are explicitly trained to memorize and reconstruct their training datasets, GANs are not. Instead, GANs consist of two competing neural networks: a generator and a discriminator. Similar to diffusion models, the generator receives random noise as input, but unlike a diffusion model, it must convert this noise to a valid 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. As a result, GANs differ from diffusion models in that their generators are only trained using *indirect* information about the training data (i.e., using gradients from the discriminator) because they never receive training data as input, whereas diffusion models are explicitly trained to reconstruct the training set.

Membership inference attacks. We first propose a privacy attack methodology for GANs.[8] 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 [80] 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 [42]: BigGAN [6], MHGAN [74], and StyleGAN [44]. Figure 14 shows the membership inference results. 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).

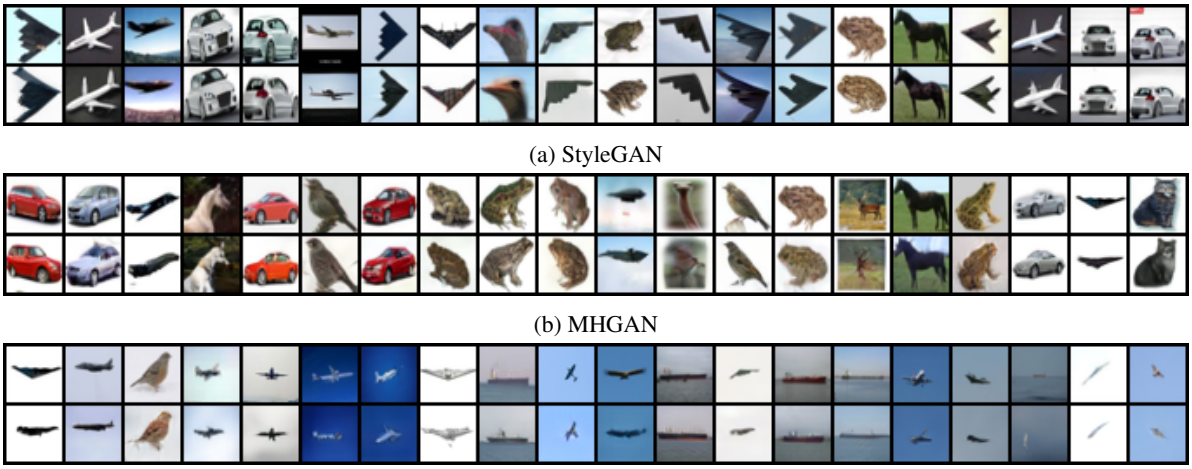
Data extraction results. We next turn our attention away from measuring worst-case privacy risk and focus our attention on more practical black-box extraction attacks. We follow the same procedure as Section 5.1, where we generate 2^{20} images from each model architecture and identify those that are near-copies of the training data using the same similarity function as before. Again we only consider non-duplicated CIFAR-10 training images in our counting. For this experiment, instead of using models we train ourselves (something that was necessary to run LiRA), we study five off-the-shelf pre-trained GANs: WGAN-ALP [68], E2GAN [69], NDA [63], DiffBigGAN [82], and StyleGAN-ADA [43]. We also evaluate two off-the-shelf DDPM diffusion model released by Ho *et al.* [33] and Nichol *et al.* [52]. Note that all of these pre-trained models are trained by the origi-

[8] 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.



(a) StyleGAN FID avg = 3.7 (b) MHGAN FID avg = 7.9 (c) BigGAN FID avg = 7.7

Figure 14: Membership inference results on GAN models using the loss threshold and LiRA attacks on the discriminator. Overall, GANs are significantly more private than diffusion models under default training configurations.



(a) StyleGAN

(b) MHGAN

(c) BigGAN

Figure 15: Selected training examples we extract from three GANs trained on CIFAR-10 for different architectures. **Top** row: generated output from a diffusion model. **Bottom** row: nearest (ℓ_2) example from the training dataset. Figure 25 in the Appendix contains all unique extracted images.

nal authors to maximize utility on the entire CIFAR-10 dataset rather than a random 50% split as in our prior models trained for MIA.

Table 1 shows the number of extracted images for each model and their corresponding FID. Overall, we find that diffusion models memorize more data than GANs, even when the GANs reach similar performance, e.g., the best DDPM model memorizes 2$\times$ more than StyleGAN-ADA but reaches the same FID. Moreover, generative models (both GANs and diffusion models) tend to memorize more data as their quality (FID) improves, e.g., StyleGAN-ADA memorizes 3$\times$ more images than the weakest GANs.

Using the GANs we trained ourselves, we show examples of the near-copy generations in Figure 15 for the three GANs that we trained ourselves, and Figure 24 in the Appendix shows every sample that we extract for those models. The Appendix also contains near-copy generations from the five off-the-shelf GANs. Overall, these results further reinforce the conclusion that diffusion models are less private than GAN models.

We also surprisingly find that diffusion models and GANs memorize many of the same images. In particular, despite the fact that our diffusion model memorizes 1280 images and a StyleGAN model we train on half of the dataset memorizes 361 images, we find that *244 unique images are memorized in common*. If images were memorized uniformly at random, we should expect on average 10 images would be memorized by both, giving exceptionally strong evidence that some images ($p < 10^{-261}$) are inherently less private than others. Understanding why this phenomenon occurs is a fruitful direction for future work.

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 many examples that are easy to extract are duplicated many times (e.g., > 100) in the training data. Similar results have been shown for language models for text [11, 40] and data deduplication has been shown to be an effective mitigation against memorization for those models [47, 41]. 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 future datasets using these more advanced notions of duplication.

Unfortunately, deduplication is not a perfect solution. To better understand the effectiveness of data deduplication, 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 [21, 22]. 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 [19].

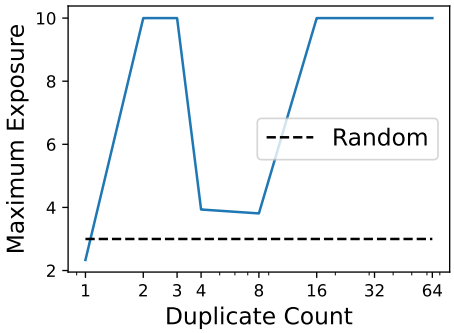


Figure 16: Canary *exposure* (a measure of non-privacy) as a function of duplicate count. Inserting a canary twice is sufficient to reach maximum exposure.

Unfortunately, we applied DP-SGD to our diffusion model codebase and found that it caused the training on CIFAR-10 to consistently diverge, even at high values for ε (the privacy budget, around 50). In fact, even applying a non-trivial gradient clipping or noising on their own (both are required in DP-SGD) caused 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.

7.3 Auditing with Canaries

In addition to implementing defenses, it is important for practitioners to empirically audit their models to determine how vulnerable they are in practice [36]. Our attacks above represent one method to evaluate model privacy. Nevertheless, our attacks are expensive, e.g., our membership inference results require training many shadow models, and thus lighter weight alternatives may be desired.

One such alternative is to insert canary examples into the training set, a common approach to evaluate memorization in language models [10]. Here, one creates a large "pool" of *canaries*, e.g., by randomly generating noise images, and inserts a subset of the canaries into the training set. After training, one computes the *exposure* of the canaries, which roughly measures how many bits were learned about the inserted canaries as compared to the larger pool of not inserted canaries. This loss-based metric only requires training one model and can also be designed in a worst-case way (e.g., adversarial worst-case images could be used).

To evaluate exposure for diffusion models, we gen-

erate canaries consisting of uniformly generated noise. We then duplicate the canaries in the training set at different rates and measure the maximum exposure. Figure 16 shows the results. Here, the maximum exposure is 10, and some canaries reach this exposure after being inserted only twice. The exposure is not strictly increasing with duplicate count, which may be a result of some canaries being "harder" than others, and, ultimately, random canaries we generate may not be the most effective canaries to use to test memorization for diffusion models.

8 Related Work

Memorization in language models. Numerous past works study memorization in generative models across different domains, architectures, and threat models. One area of recent interest is memorization in language models for text, where past work shows that adversaries can extract training samples using two-step attack techniques that resemble our approach [11, 47, 41, 40]. Our work differs from these past results because we focus on the image domain and also use more semantic notions of data regeneration (e.g., using CLIP scores) as opposed to focusing on exact verbatim repetition (although recent language modeling work has begun to explore approximate memorization as well [35]).

Memorization in image generation. Aside from language modeling, past work also analyzes 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 with the training data [32, 3], the extent of mode collapse [61, 15], and the impact of individual training samples [4, 75]. Moreover, other work provides insights into when and why GANs may replicate training examples [50, 26], as well as how to mitigate such effects [50]. 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 also studies privacy in image generation for both GANs [70] and diffusion models [65, 78, 34]. Tinsley *et al.* [70] show that StyleGAN can generate individuals' faces, and Somepalli *et al.* [65] 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).

9 Discussion and Conclusion

State-of-the-art diffusion models memorize and regenerate individual training images, allowing adversaries to launch training data extraction attacks. 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 $2\times$ 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 individual training examples? If our extraction attacks had failed, it may have refuted the hypothesis that models copy and interpolate training data; but because our attacks succeed, this question remains open. Given that different models memorize varying amounts of data, we hope future work will explore how diffusion models copy from their training datasets.

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

Practical consequences. We raise four practical consequences for those who train and deploy diffusion models. First, while not a perfect defense, we recommend deduplicating training datasets and minimizing over-training. Second, we suggest using our attack—or other auditing techniques—to estimate the privacy risk of trained models. Third, once practical privacy-preserving techniques become possible, we recommend their use whenever possible. Finally, we hope our work will temper the heuristic privacy expectations that have come to be associated with diffusion model outputs: synthetic data does not give privacy for free [13, 14, 59, 2, 53].

On the whole, our work contributes to a growing body of literature that raises questions regarding the legal, ethical, and privacy issues that arise from training on web-scraped public data [7, 65, 72, 77]. Researchers and practitioners should be wary of training on uncurated public data without first taking steps to understand the underlying ethics and privacy implications.

	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.

Contributions

- Nicholas, Jamie, Vikash, and Eric each independently proposed the problem statement of extracting training data from diffusion models.
- Nicholas, Eric, and Florian performed 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 the 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 paper draft.

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 for comments on early drafts of this paper.

Nicholas, Milad, Matthew, and Daphne are employed at Google, and Jamie and Borja are employed at DeepMind, companies that both train 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.

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.

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

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

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

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

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

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

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

[20] 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.

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

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

[23] 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-popula` 2022.

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

[32] 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.

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

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

[35] 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.

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

[37] 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.

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

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

[40] 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.

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

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

[43] 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.

[44] 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.

[45] 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.

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

[47] 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.

[48] 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.

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

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

[51] 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.

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

[53] 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.

[54] 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.

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

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

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

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

[64] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermody-

namics. In *International Conference on Machine Learning*, 2015.

[65] 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.

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

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

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

[69] 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.

[70] 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.

[71] 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.

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

[73] 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.

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

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

[76] 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 2022.

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

[78] 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.

[79] 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.

[80] 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.

[81] 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 *IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[82] 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.

A Collected Details for Figures

Table 3: Catalog of figures containing qualitative examples.

Figure #	Model	Dataset	Who trained it?	Sampling strategy
Figure 1	Stable Diffusion	LAION	Stability AI	PLMS
Figure 2	Stable Diffusion	LAION	Stability AI	PLMS
Figure 3	Stable Diffusion	LAION	Stability AI	PLMS
Figure 6	Uncond Diffusion	CIFAR-10	Ours	DDIM
Figure 7	Uncond Diffusion	CIFAR-10	Ours	DDIM
Figure 8	Uncond Diffusion	CIFAR-10	Ours	DDIM
Figure 12	Uncond Diffusion	CIFAR-10	Ours	Inpainting
Figure 13	Uncond Diffusion	CIFAR-10	Ours	Inpainting
Figure 15	StyleGAN, MHGAN, BigGAN	CIFAR-10	Ours	GAN default
Figure 17	Uncond Diffusion	CIFAR-10	Ours	DDIM
Figure 20	Uncond Diffusion	CIFAR-10	Ours	DDIM
Figure 22	Uncond Diffusion	CIFAR-10	Ours	Inpainting
Figure 23	Uncond Diffusion	CIFAR-10	Ours	Inpainting
Figure 24	Several different GANs	CIFAR-10	Original paper authors	GAN default

B All CIFAR-10 Memorized Images

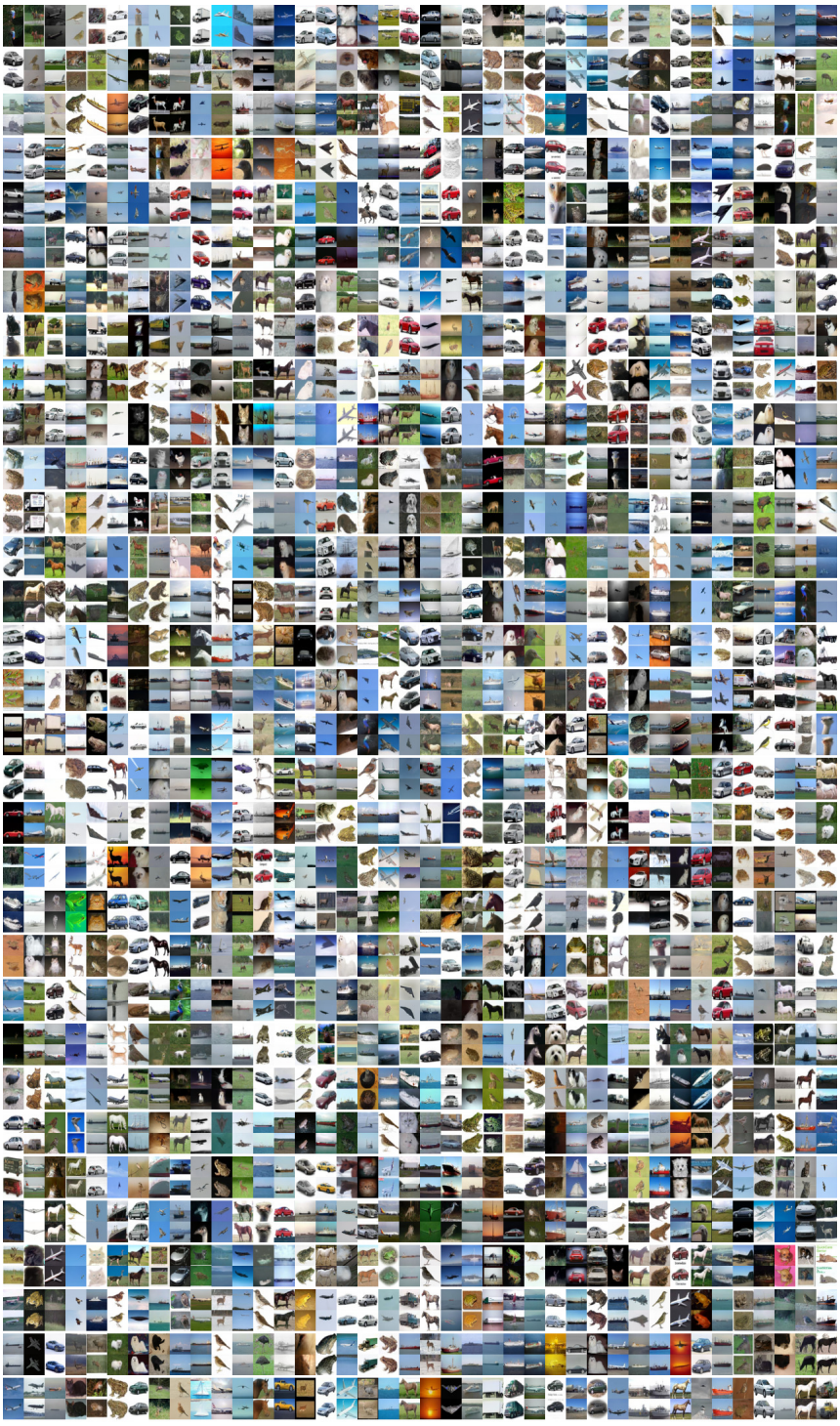
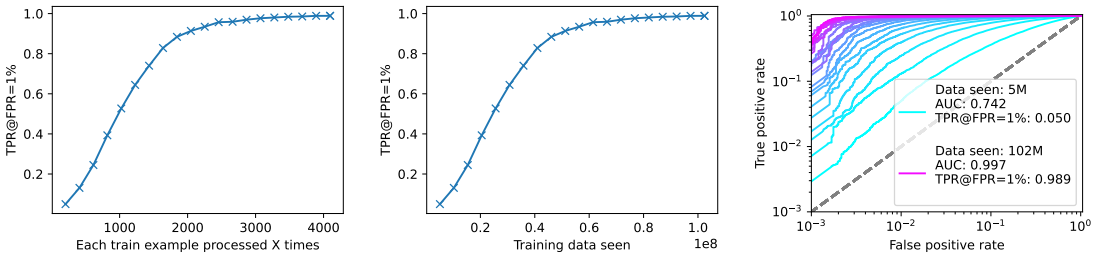


Figure 17: All 1280 images we extract from diffusion models trained on CIFAR-10, after 1 million generations from 16 diffusion models.

C Additional Attacks on CIFAR-10

Here, we expand on our investigation of memorization of training data on CIFAR-10.

C.1 Membership Inference at Different Training Steps



(a) How membership attack success changes as a training example is processed repeatedly throughout training.

(b) How membership attack success changes as more data is processed throughout training.

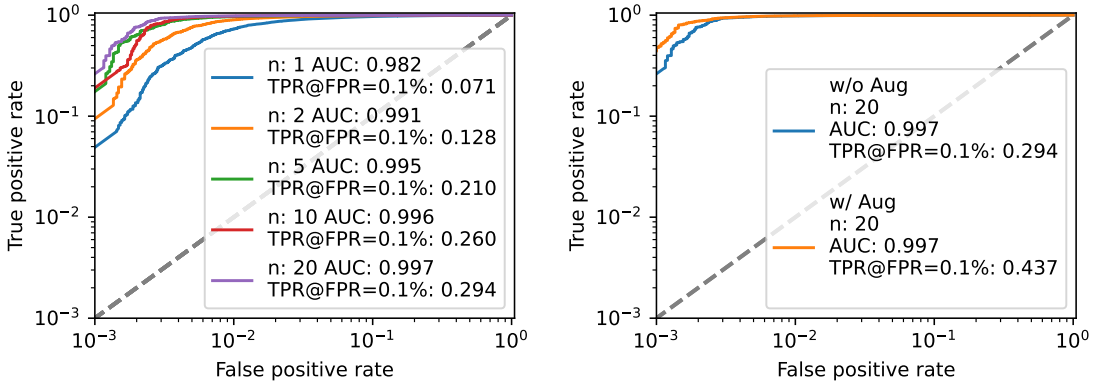
(c) ROC curve for the membership attack for different training steps.

Figure 18: Membership inference attacks as a function of the amount of training data processed on CIFAR-10.

In Section 5.2.3, we implicitly investigated membership attack success as a function of the number update steps when training a diffusion model. We explicitly model this relationship in Figure 18. First, in Figure 18a we plot membership attack success as a function of the number of times that an example was processed over training. If an example is processed more than 2000 times during training, invariably membership attacks are perfect against that example. Second, in Figure 18b, we plot membership attack success as a function of the total amount of data processed during training. Unsurprisingly, membership attack success increases as more training data is processed. This is highlighted in Figure 18c, where we plot the membership attack ROC curve. At 5M training examples processed, at a FPR of 1% the TPR is 5%, and increases to 99% after 102M examples are processed. Note that this number of processed training inputs is commonly used in diffusion model training. For example, the OpenAI CIFAR-10 diffusion model [9] is trained for 500,000 steps at a batch size of 128, meaning 64M training examples are processed. Even at this number of processed training examples, our membership attack has a TPR $> 95\%$ at a FPR of 1%.

[9] `https://github.com/openai/improved-diffusion`

C.2 Membership Inference with Different Augmentation Strategies



(a) (b)

Figure 19: We can improve membership inference attack success rates on CIFAR-10 by reducing noise. In **(a)**, membership inference attacks are improved by averaging the loss over multiple noise samples in the diffusion process. In **(b)**, attacks are improved by querying on augmented versions of the candidate image.

C.3 Membership Inference Inliers and Outliers

Figure 20: When performing our membership inference attack, the hardest-to-attack examples (left) are all duplicates in the CIFAR-10 training set, and the easiest-to-attack examples (right) are visually outliers from CIFAR-10 images.

C.4 Membership Inference on Conditional and Unconditional Models

Diffusion models can be conditioned on labels (or prompts for text-to-image models). We compare the difference in membership inference on a CIFAR-10 diffusion model trained unconditionally with a model conditionally trained on CIFAR-10 labels. The conditional and unconditional models reach approximately the same FID after training; between 3.5-4.2 FID. We plot the membership attack ROC curve in Figure 21 and note that the conditional model is marginally more vulnerable. However, it is difficult to tell if this is a fundamental difference between conditional and unconditional models, or because the conditional model contains more parameters than unconditional model (the conditional models contains an extra embedding layer for the one-hot label input).

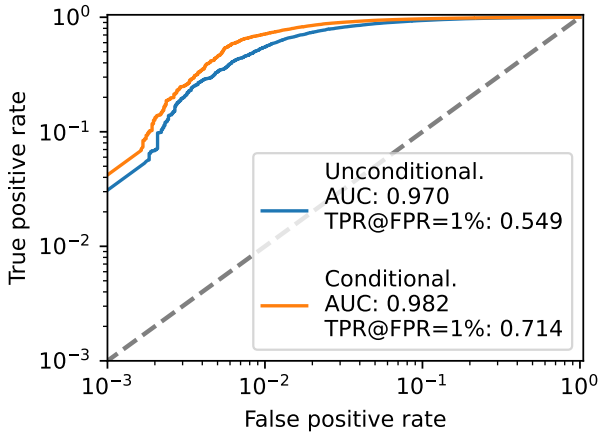


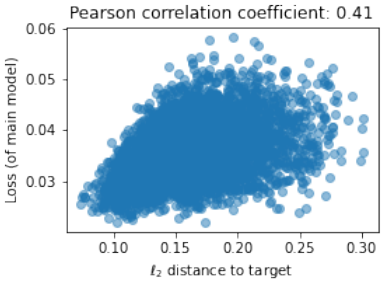
Figure 21: Membership attack against a conditional and unconditional diffusion model on CIFAR-10.

D More Inpainting Attacks on CIFAR-10

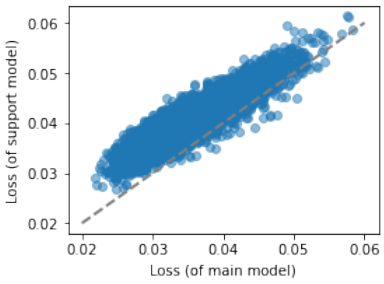
Here, we take a deeper dive into the inpainting attacks introduced in Section 5.3. As previously explained, for a target x, we create X_{rec} where $|X_{rec}| = 5000$. In Figure 22a, for every $x_{rec} \in X_{rec}$, we plot the normalized ℓ_2 distance between the reconstruction and target, against the loss (at diffusion timestep 100) of x_{rec}. We also plot in Figure 22d, the eight examples from X_{rec} that have the smallest loss on the main model. There is a small positive correlation between loss and ℓ_2 distance; although some appear to be similar to x, there are notable differences.

In Figure 22b we compare the loss of each reconstruction on the main model against the *support* model we will use to form the contrastive loss. We make this correlation more pronounced by dividing the main loss by the support loss in Figure 22c. This has the effect of increasing the correlation between the (now contrastive) loss and ℓ_2 distance. This has the effect of filtering out examples that are seen as likely under both models, and can be seen by inspecting the eight examples from X_{rec} that have have the smallest $\frac{\text{main model loss}}{\text{support model loss}}$ in Figure 22e. These examples look more visually similar to x in comparison to examples in Figure 22d.

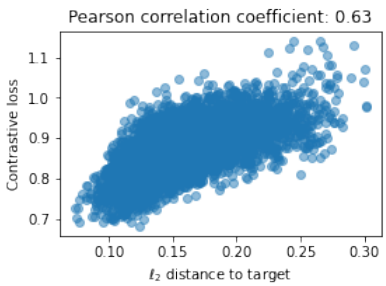
Figure 22 inspected the attack success when x was in the training set. We show in Figure 23 that the attack fails when x was not included in training; using a contrastive loss doesn't signficantly increase the Pearson correlation coefficient. This means our attack is indeed exploiting the fact that the model can only inpaint correctly because of memorisation and not due to generalisation.



(a) Loss (using the main model at diffusion timestep 100) on all 5,000 inpainted examples X_{rec}.

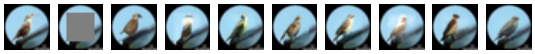


(b) Comparison of loss on main and support models (at diffusion timestep 100) on all 5,000 inpainted examples.



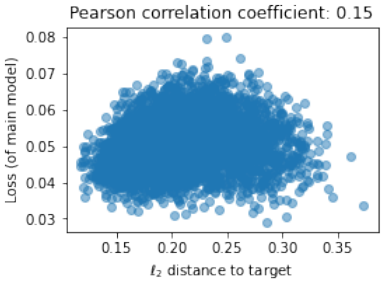
(c) Contrastive loss ($\frac{\text{main model loss}}{\text{support model loss}}$) on all 5,000 inpainted examples X_{rec}.

(d) 8 inpainted examples with the smallest loss. Leftmost is the original example, second to left is the masked example and the rest are inpainted examples.

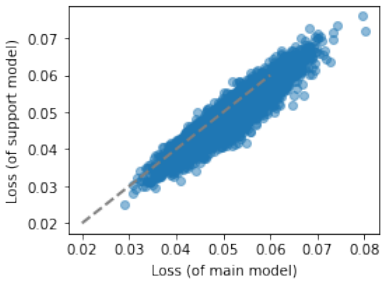


(e) 8 inpainted examples with the smallest $\frac{\text{main model loss}}{\text{support model loss}}$. Leftmost is the original example, second to left is the masked example and the rest are inpainted examples.

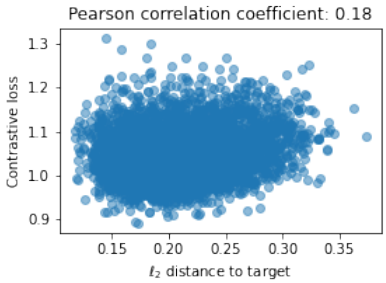
Figure 22: Example of an inpainting attack (against a model we refer to as the *main* model) on an image of a bird from CIFAR-10 when that image is included in training, and we mask out 60% of the central pixels. In (a) we plot the L_2 distance between 5,000 inpainted reconstructions and the original (non-masked out) image and compare this to the loss with respect to the (main) model. In (b), we compare the loss of these reconstructions on the (main) model with a *support model* for which we know the image wasn't contained in the training set. In (c), we compare L_2 distances between reconstructions with a contrastive loss which is given as the loss of the image with respect to the main model divided by the loss of the image with respect to the support model, and find there is stronger relationship between smaller L_2 distances and smaller losses compared to (a). Figure (d) gives examples of reconstructions with small loss and Figure (e) gives examples of reconstructions with small contrastive loss.



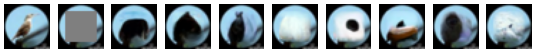
(a) Loss (using the main model at diffusion timestep 100) on all 5,000 inpainted examples X_{rec}.



(b) Comparison of loss on main and support models (at diffusion timestep 100) on all 5,000 inpainted examples.



(c) Contrastive loss ($\frac{\text{main model loss}}{\text{support model loss}}$) on all 5,000 inpainted examples X_{rec}.



(d) 8 inpainted examples with the smallest loss. Leftmost is the original example, second to left is the masked example and the rest are inpainted examples.

(e) 8 inpainted examples with the smallest $\frac{\text{main model loss}}{\text{support model loss}}$. Leftmost is the original example, second to left is the masked example and the rest are inpainted examples.

Figure 23: Example of an inpainting attack (against a model we refer to as the *main* model) on an image of a bird from CIFAR-10 when that image is *not* included in training, and we mask out 60% of the central pixels. In (a) we plot the L_2 distance between 5,000 inpainted reconstructions and the original (non-masked out) image and compare this to the loss with respect to the (main) model. In (b), we compare the loss of these reconstructions on the (main) model with a *support model* for which we know the image wasn't contained in the training set. In (c), we compare L_2 distances between reconstructions with a contrastive loss which is given as the loss of the image with respect to the main model divided by the loss of the image with respect to the support model, and find there is stronger relationship between smaller L_2 distances and smaller losses compared to (a). Figure (d) gives examples of reconstructions with small loss and Figure (e) gives examples of reconstructions with small contrastive loss.

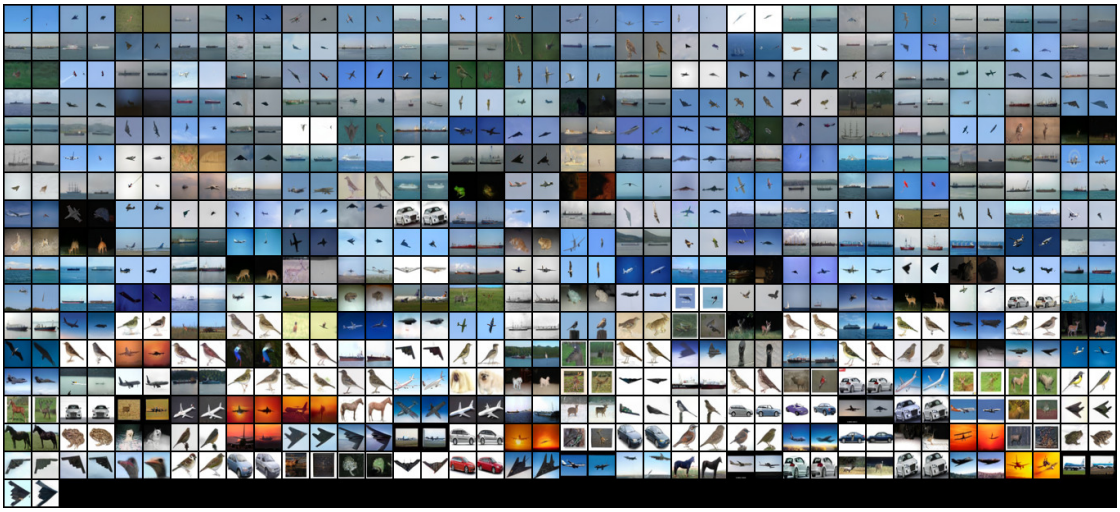
E GAN Training Setup

We used on StudioGAN[10] codebase for training GAN in this work. For the StyleGAN and MHGAN architectures, we followed the default hyper-parameters provided in the StudioGAN repository. However, for the BigGAN architecture, we increased the number of training steps to 200,000, which is different from the original hyper-parameters, to increase image fidelity. We trained a total of 256 models for each GAN architecture, with each model being trained on a randomly selected half of the CIFAR-10 dataset. We selected the iteration that achieved the highest FID score on the test set for each model.

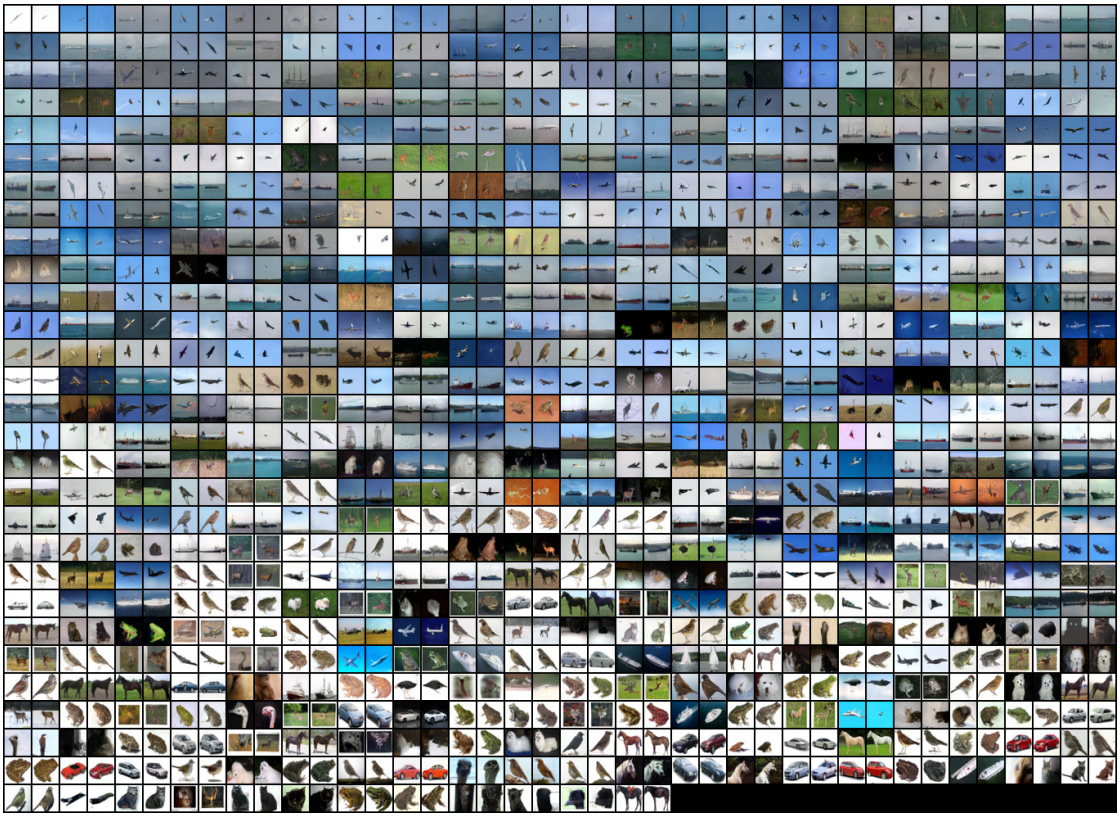
F Additional GAN Extraction Results

Figure 24 and Figure 25 contain additional examples extracted from GANs trained on CIFAR-10.

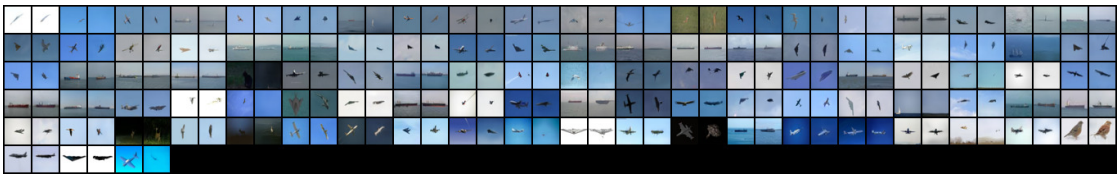
[10] `https://github.com/POSTECH-CVLab/PyTorch-StudioGAN`



(a) StyleGAN

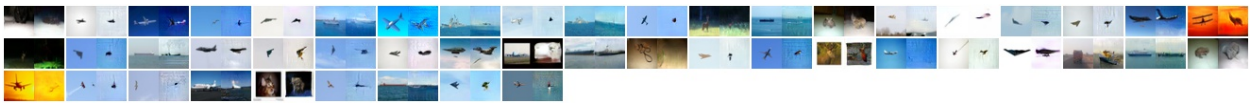


(b) MHGAN



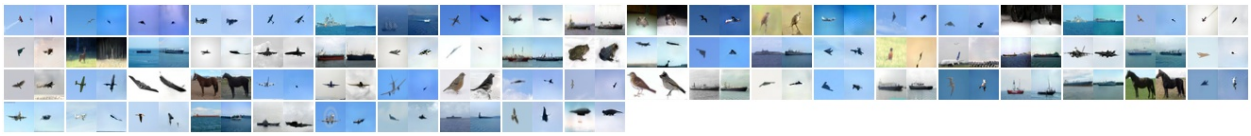
(c) BigGAN

Figure 24: Training examples extracted from a CIFAR-10 GAN for different architectures across 10^7 generations.

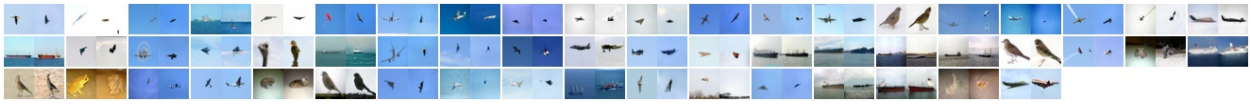


(a) WGAN

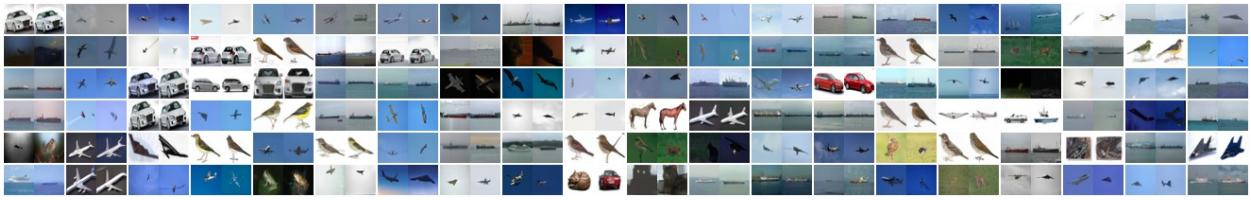
(b) E2GAN



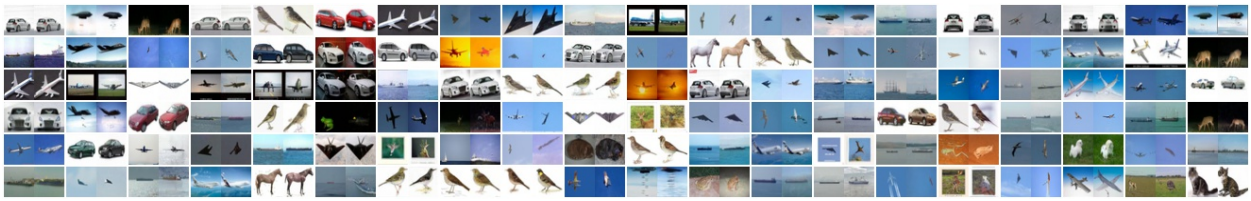
(c) NDA



(d) DiffAugment-BigGAN



(e) StyleGAN-ADA



(f) DDPM

Figure 25: Training examples extracted from different publicly available pretrained GANs and diffusion (DDPM) models. We use normalized ℓ_2 distance in pixel space to find memorized training samples. In each pair of images, left and right image corresponds to real and it closely synthetic image. For StyleGAN-ADA and DDPM model we display 120 pairs with smallest normalized ℓ_2 distance. For others we display all memorized training images. 1M generations