Towards More Realistic Membership Inference Attacks on Large Diffusion Models

Jan Dubiński [1,2*†] **Antoni Kowalczuk** [1†] **Stanisław Pawlak** [1] **Przemyslaw Rokita** [1]
Tomasz Trzcinski [1,2,3] **Paweł Morawiecki** [4]

[1]Warsaw University of Technology [2]IDEAS NCBR [3]Tooploox [4]Polish Academy of Sciences

Abstract

Generative diffusion models, including Stable Diffusion and Midjourney, can generate visually appealing, diverse, and high-resolution images for various applications. These models are trained on billions of internet-sourced images, raising significant concerns about the potential unauthorized use of copyright-protected images. In this paper, we examine whether it is possible to determine if a specific image was used in the training set, a problem known in the cybersecurity community as a membership inference attack. Our focus is on Stable Diffusion, and we address the challenge of designing a fair evaluation framework to answer this membership question. We propose a new dataset to establish a fair evaluation setup and apply it to Stable Diffusion, also applicable to other generative models. With the proposed dataset, we execute membership attacks (both known and newly introduced). Our research reveals that previously proposed evaluation setups do not provide a full understanding of the effectiveness of membership inference attacks. We conclude that the membership inference attack remains a significant challenge for large diffusion models (often deployed as black-box systems), indicating that related privacy and copyright issues will persist in the foreseeable future.

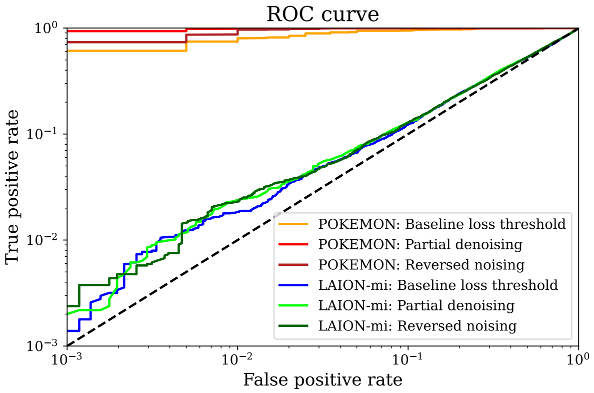


Figure 1. **Pitfalls in the evaluation setting can lead to incorrect conclusions on the effectiveness of membership attacks against large diffusion models such as Stable Diffusion.** An exemplary misleading setup involves finetuning the model on a very small dataset with a low internal variance (such as the POKEMON dataset), which gives a remarkable performance for the selected attacks. However, for the proposed new dataset we observe a drastic performance drop. Our setup does not modify the state-of-the-art Stable Diffusion model but focuses on creating fair membership inference evaluation, possibly close to a real-life usage of the membership attacks.

1. Introduction

In recent years, there have been rapid advancements in generative modeling techniques within the field of deep learning. Among these, generative diffusion models, particularly those utilising the Stable Diffusion framework, have gained prominence due to their capability to generate high-quality, diverse, and intricate samples. These models hold considerable potential for numerous applications, such as data augmentation, art creation, and design optimization. However, as these models become more widely adopted, addressing the privacy concerns linked to their use is essential. Recently, Getty Images filed a lawsuit against Stability AI, accusing it of *unlawfully copying and processing millions of copyright-protected images* [21]. This lawsuit comes on the heels of a separate case Getty lodged against Stability in the United Kingdom, as well as a related class-action lawsuit that California-based artists filed against Stability and other emerging companies in the generative AI sector [22].

One critical issue that arises in this context is determining whether a specific data point was used during the training process of a model. Extracting this information from a model - known as *membership inference attack* - can be crucial in cases where copyrighted or sensitive data are used without permission, leading to potential legal issues. Although membership inference attacks have been extensively studied in the context of discriminative models [3, 30, 35, 36], the investigation of their effectiveness against generative diffusion models is still in its infancy.

[*]Corresponding author: jan.dubinski.dokt@pw.edu.pl
[†]Equal contribution.

In this paper, we contribute to the understanding of membership inference attacks in large diffusion models, particularly Stable Diffusion. We provide insights into these models, their susceptibility to different membership attacks, and the challenges of their evaluation due to the lack of distinct training and test data. To address these issues, we propose a new dataset for fair and robust evaluation setup. We conduct attacks against Stable Diffusion and assess their effectiveness. Our findings underscore the complexity of data membership inference in large diffusion models. Our main contributions can be summarized as follows:

- We identify the pitfalls of the existing evaluation of membership inference attacks for large diffusion models.
- We provide a new dataset[1] along with a construction methodology. It allows us to have a more robust evaluation setup for membership inference attacks on the state-of-the-art Stable Diffusion model.
- With the proposed dataset, we thoroughly evaluate a set of membership inference attacks, which are not prohibitively expensive against Stable Diffusion, including the loss threshold attack and its variants. We also introduce new attacks that focus on modifying the diffusion process to extract more information about membership from the model.

2. Background

2.1. Diffusion models

Over the past two years, diffusion models [31] have emerged as a novel class of generative models, overshadowing Generative Adversarial Networks [8] by achieving state-of-the-art results on numerous benchmarks [5] and becoming the core technology behind widely popular image generators such as Stablxe Diffusion [24], Midjourney [32], Runway [24], Imagen [25] and DALL-E 2 [19, 20].

In essence, *Denoising Diffusion Probabilistic Models* [10] are probabilistic generative models trained by progressively adding noise to the data and then learning to reverse this process.

During training, a noised image $x_t \leftarrow \sqrt{a_t}x + \sqrt{1-a_t}\epsilon$ is produced by adding Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ to a clean image x, with a decaying parameter $a_t \in [0, 1]$ such that $a_0 = 1$ and $a_T = 0$. The diffusion model f_θ is trained to remove the noise ϵ and recover the original image x by predicting the added noise. This is achieved by stochastically minimizing the objective $\frac{1}{N}\sum_i \mathbb{E}_{t,\epsilon}\mathcal{L}(x_i, t, \epsilon; f_\theta)$, where

$$\mathcal{L}(x, t, \epsilon; f_\theta) = \|\epsilon - f_\theta(x_t, t)\|_2^2 \tag{1}$$

Despite being trained with a simple denoising objective, diffusion models have shown the ability to generate high-quality images. The process involves sampling a random vector x_T from a normal distribution $\mathcal{N}(0, I)$ and then applying the diffusion model f_θ to remove the noise from this random image. However, instead of removing all noise at once, the model gradually removes part of the noise iteratively in each generation step.

The final image x_0 is obtained from x_T using a noise schedule σ_t (dependent on a_t), where the model iteratively applies the rule $x_{t-1} = f_\theta(x_t, t) + \sigma_t\mathcal{N}(0, I)$ to $\sigma_1 = 0$. The effectiveness of this process is based on the fact that the diffusion model was trained to denoise images with varying levels of noise. Applying this iterative generation process with large-scale diffusion models yields results that closely resemble real images.

Certain diffusion models are designed to generate specific types of images by incorporating conditional inputs in addition to the noised image. Class-conditional diffusion models utilise a class label, such as "car" or "plane", to generate a desired image class. Text-conditioned models extend this concept by taking the text embedding of a prompt, such as "a photograph of an astronaut riding a horse in space," which is created by a pretrained language encoder like CLIP [18].

2.2. Stable Diffusion

Stable Diffusion is the largest and most popular open-source diffusion model [24]. This model is an 890 million parameter text-conditioned diffusion model trained on 2.3 billion images.

Diffusion models can achieve state-of-the-art synthesis results on image data and other applications. However, the optimisation of powerful diffusion models that operate directly in pixel space can consume hundreds of GPU days, and inference can be expensive due to sequential evaluations. To overcome this challenge, the authors of Stable Diffusion [24] propose applying diffusion models in the latent space of powerful pretrained autoencoders. This approach allows for training and inference on limited computational resources while retaining the quality and flexibility of diffusion models.

Formally, given an image x, the encoder E encodes x into a latent representation $z = E(x)$, and the decoder D reconstructs the image from the latent, giving $\tilde{x} = D(z) = D(E(x))$. To preprocess conditional information y from various modalities (such as language prompts), the Stable Diffusion framework introduces a domain-specific encoder τ_θ that projects y to an intermediate representation. Overall, the Stable Diffusion model is trained by stochastically minimizing the objective $\frac{1}{N}\sum_i \mathbb{E}_{t,\epsilon}\mathcal{L}(z_i, t, \epsilon; f_\theta)$, where

$$\mathcal{L}(z, t, \epsilon; f_\theta) = \|\epsilon - f_\theta(z_t, t, \tau_\theta(y))\|_2^2 \tag{2}$$

[1] https://drive.google.com/drive/folders/17lRvzW4uXDoCf1v_sIiaMnKGIARVunNU

3. Membership inference attack

Membership inference attack [30] answers the question *"was this example in the training set?"*. Currently, two most common approaches are loss based attacks and shadow models.

3.1. Loss based attacks

In principle, loss-based membership inference attacks are based on the following simple observation [36]. A model training minimises a loss function on a training set, hence we expect the loss to be lower for training samples than for test ones. In most cases, such methods treat the attacked model a as white-box, assuming that the attacker has access to the model, its source code and trained weights. This assumption is often not met in practice, as API-based generative machine learning services such as Midjourney [32] increase in popularity. In general, methods based solely on the analysis of the loss of the model are less effective than methods utilising shadow models [3, 4].

3.2. Shadow models

Membership inference attacks based on *shadow models* involve creating multiple models that imitate the behaviour of the target model, but whose training datasets are known to the attacker. By studying the labelled inputs and outputs of these shadow models, researchers can gain insight into the target model's behaviour and develop attacks that can exploit its vulnerabilities. For diffusion models [4] introduced membership inference attacks called LiRA. This approach involves training a collection of shadow models on random subsets of the training dataset. Once the shadow models have been trained, LiRA computes the loss for each example under each shadow model. By analysing the distribution of losses, LiRA can then determine whether a given example belongs to the training dataset or not. Although shadow models have proven to be a powerful tool in the development of membership inference attacks, this approach has its own disadvantages. Such methods are computationally very expensive, as they require the training of multiple copies of the target model. In particular, for large diffusion models such as Stable Diffusion, the cost of developing multiple shadow models is in practice too high.

4. Attack Challenges and Pitfalls for Large Diffusion Models

Lack of nonmembers To perform and evaluate a *membership inference attack* we need two sets: members and nonmembers. Typically, member data samples are drawn from the training set, whereas nonmembers from the test set. Unfortunately, for the Stable Diffusion model, we cannot follow this approach. The original Stable Diffusion model was trained on the data from the LAION-2B EN dataset, a subset of the LAION-5B [28]. Since the dataset is huge (more than 2 billion images) and was not divided into a test and training set, nonmember samples are not easily available.

Shadow models cost As stated in Section 3.2 the shadow models are computationally expensive. The method requires training several dozens of models from scratch. For huge models, such as Stable Diffusion, this approach is practically infeasible. In particular, the cost of training a single Stable Diffusion model is estimated at 600,000$. Moreover, it would take 80.000 A100 GPU-hours to complete the training.

Pitfall 1: Evaluation based on fine-tuning In [6] authors propose to tackle the lack of nonmembers by fine-tuning the Stable Diffusion on a dataset that was not used for training the original model. However, it has been demonstrated that fine-tuning the model can easily lead to overfitting [12]. As shown in [4], better diffusion models are more vulnerable to membership inference attacks: the quality of the generated samples is proportional to the success rate of the attack. This is especially the case for models that overfit to the limited training data during fine-tuning [36], leading to inflated performance and misleading conclusions.

Pitfall 2: Distribution mismatch between members and nonmembers Another approach to dealing with the absence of a natural nonmember dataset is to draw samples from a dataset similar to the training data that was not actually used in the training. However, for a fair evaluation of membership inference attacks, it is important that the members and nonmembers share the same feature distribution. If these two groups can be easily distinguished based on feature distribution mismatch, then there is a high risk that the attack method will learn to distinguish between the features of the two data groups [15] instead of the behaviour of the model on these two sets.

5. The LAION-mi dataset

To address the absence of nonmembers samples for the Stable Diffusion model we propose a new dataset consisting of members and nonmembers called LAION-mi. This new dataset aims to facilitate a realistic evaluation of membership inference attacks against large diffusion models. We do not fine-tune nor modify the Stable Diffusion model in any other way, in order to avoid the first pitfall from Sec. 4. To mitigate the second pitfall, we apply a sanitization process on the member set. Figure 2 shows a general scheme of how the LAION-mi dataset is constructed.

5.1. Sources of members and nonmembers sets in LAION-mi

Members Stable Diffusion-v1.4 was trained on all data points from the LAION Aesthetics v2 5+ dataset (see Appendix C for details), so all samples from this dataset can serve as member candidates for our new dataset.

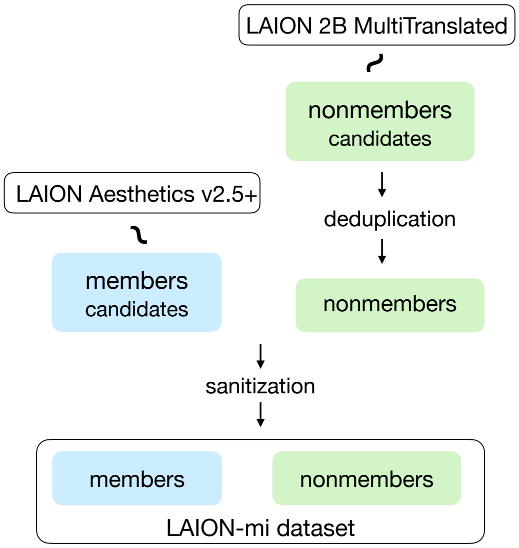


Figure 2. **A general scheme of constructing LAION-mi dataset.** First, members and nonmembers are sampled from LAION Aesthetics v2 5+ and LAION-2B Multi Translated datasets, respectively. Then, we remove nonmember that are duplicates of samples from the member set. Finally, to ensure that the distribution of member samples is indistinguishable from nonmembers distribution, we execute an extensive sanitization algorithm on the member set.

Nonmembers To obtain the nonmembers set, we use LAION-2B Multi Translated dataset [14]. This dataset is created by LAION-5B authors from LAION-2B Multi, a subset of LAION-5B, but unlike LAION-2B EN, LAION-2B Multi consists of samples which descriptions are in other languages (not English). LAION-2B Multi Translated is obtained by translating these descriptions to English. Since SD-v1.4 was fine-tuned using samples with an aesthetic score above 5 (obtained by LAION-Aesthetics_Predictor V2 [26]), to build our nonmembers set we also use samples with aesthetic score above 5. The score is precomputed by the LAION-2B Multi Translated dataset authors.

5.2. Adapting members and nonmembers sets to ensure the validity of evaluation setting.

As mentioned before, for a fair evaluation of the membership inference attack we should ensure that the underlying data distribution is the same for the member and nonmember samples. We solve it by introducing the adaptation step for members and nonmembers subsets constituting LAION-mi dataset. During the adaptation, we first deduplicate the nonmembers set (Sec. 5.3) and then filter the member set using sanitization process (Sec. 5.4). Finally, we obtain members and nonmembers sets which have the same, indistinguishable underlying distribution.

5.3. Deduplication

Member samples in the nonmembers source dataset Duplicate samples are images present in a dataset more than once. It has been shown [34] that LAION-2B EN contains approximately 30% duplicates when it comes to the image data. We can expect that the whole LAION-5B contains samples, which are present both in LAION-2B EN and LAION-2B Multi Translated. These samples can potentially contaminate LAION-mi nonmembers subset with members samples and therefore compromise the fairness and correctness of our solution. The following procedure aims to address this issue.

Solution In order to obtain the nonmembers set free of contamination by members samples we perform two-step deduplication. The first step aims to propose a set of duplicate candidates for each nonmember sample. The second step filters out nonmembers samples, which we suspect have a duplicate in the members source dataset. We end up with the clean nonmembers set.

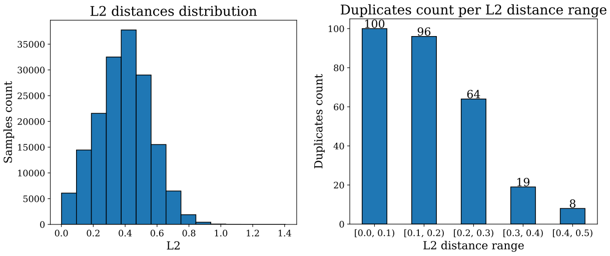
Duplicate candidates We need to somehow obtain the samples from the LAION-2B EN dataset which are the most similar to the given sample from the nonmember source. We achieve this by querying the Clip Retrieval Client [1]. This service searches through the LAION-5B dataset and returns the requested amount of samples that are the most similar to the input image. In our approach, we obtain up to 40 duplicate candidates per nonmember sample. One important limitation of this service is that it doesn't distinguish between subsets of LAION-5B. LAION-5B is split into LAION-2B EN and LAION-2B Multi using the CLD3 [33] language identifier on the captions of images. Because we care only about the duplicates from the LAION-2B EN, we need to check if the returned candidate is from this dataset. We use CLD3 to check if the given candidate is from the LAION-2B EN dataset. Only samples from the LAION-2B EN dataset are considered duplicate candidates.

Duplicates detection and filtering When it comes to filtering out the duplicates we propose the following approach: we define the distance metric between the nonmember and its duplicate candidate, and then if the distance is below some threshold we decide that this sample has a duplicate in the members set, effectively discarding it from the final nonmember set.

Firstly, for each sample, we calculate the L2 distances of CLIP image embeddings between the sample and all of its duplicate candidates. Then the final duplicate candidate for each sample is the one with the lowest L2 distance score. We then end up with the approximately normal distribution of L2 distances (see Fig. 3a).

We then pick the threshold below which we reject samples and mark them as duplicates. Our goal here is to filter out as many duplicates as possible, to avoid contamination of the nonmembers set with members samples. At the end of this process, we have less than 1% of the duplicates by setting this threshold at 0.5, using *the rule of three* [13]. We manually confirm it by sampling random 300 samples and checking for the duplicates, without finding any. For the threshold of 0.5 we reject approximately 75% of nonmembers as having a duplicate in the members source dataset. It is a really conservative approach, but as we show next, it is necessary in order to achieve the cleanest nonmembers set possible.

To further confirm that we pick the correct threshold we perform a manual analysis of the duplicates ratio in different L2 score intervals and show the results in Figure 3b. Since our goal is to make the cleanest nonmembers dataset possible, we pick a threshold of 0.5 to avoid duplicates.



(a) Distribution of L2 distances.

(b) Duplicates count per 100 samples for L2 distances buckets.

Figure 3. The distribution of the L2 distances between duplicate candidates and the original images approximately follow a normal distribution, with a mean around 0.4 (left). For increasing L2 threshold value the duplicates count decreases sharply, with the interval $[0.4, 0.5)$ containing approximately 8% duplicates and 92% non-duplicates.

5.4. Sanitization

Differences between sets As we have stated before, one of the most important challenges of membership attack evaluation is ensuring that the member and nonmember samples come from the most similar distribution possible, in our case both images and their descriptions coming from these subsets should be indistinguishable from each other. However, our source of nonmembers is obtained by translating captions from LAION-2B Multi to English using machine translation[2]. Therefore, we expect the distribution of captions' CLIP [18] embeddings to be different for members and nonmembers sets. We need to address this issue to not fall into the second pitfall, which we line up in the Sec. 4.

Assessment approach In order to assess the magnitude of this problem and the efficacy of our sanitization approach we use three metrics. All are based on the CLIP embeddings of the descriptions and images. The metrics are as follows:

- Fréchet Inception Distance (FID) [9]: in order to compare the resulting metric we compute it for two cases: internal (between two random samples of 10k examples from the same set) and comparative (between 10k random members and 10k random nonmembers). If the difference between these two is significant, we assume that there is a mismatch between distributions.
- Visual analysis of PCA 2D projection: we use PCA decomposition on the embeddings to project them into a 2D space using scikit-learn [2] implementation. The mismatch in the underlying distributions should be indicated by a mismatch of the distributions of the PCA components of the projected embeddings.
- Classification: we train a binary classifier in order to distinguish between embeddings of the descriptions. High accuracy indicates a significant difference between the two sets.

Scale of the problem Our experiments confirm the seriousness of the issue. Firstly, we observe that FID for the comparative case is way greater than for any of the internal cases, see Tab. 1. Secondly, visual analysis of embeddings projected to 2D space using PCA, Fig. 4a) confirms our concerns that these text embeddings are in fact different. Finally, a simple logistic regression model separates prompt embeddings with a 90% accuracy.

Santization algorithm The goal of this process is to create a member set that is as similar as possible to our deduplicated nonmembers set.

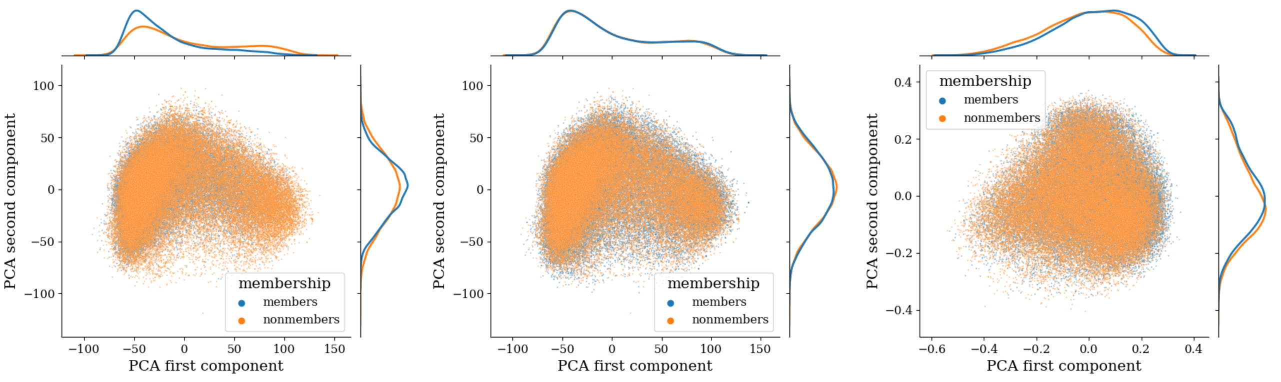
Main intuition behind the sanitization algorithm (Alg. 1) is to train a set of binary classifiers to label samples as members or nonmembers in an iterative fashion, and then pick only the samples from one of these sets, for which all of the models predict wrong label. In effect, at each iteration, one of these sets becomes closer to the other one in terms of the text embeddings distribution.

Table 1. **FID comparison for 10k samples.** For text data we calculate FID for CLIP embeddings. To calculate FID for image data we first resize images to 512x512 and then use Pytorch FID implementation [29]. To calculate internal FID we divide the dataset into 2 equal random subsets, each of 10k samples.

	FID	
DATA SUBSET	TEXT	IMAGES
MEMBERS INTERNAL - RANDOM	9.84	7.00
MEMBERS INTERNAL - SANITIZED	9.77	7.06
NONMEMBERS INTERNAL	9.73	7.01
COMPARATIVE - RANDOM	66.43	13.90
COMPARATIVE - SANITIZED	**13.54**	**8.87**

In general, we can use both members and nonmembers sets to perform this classifier-based filtering. In our case, we filter only the huge members set (LAION Aesthetics v2 5+

[2] Facebook's M2M100 1.2B model [7]



(a) Prompts embeddings before sanitization

(b) Prompts embeddings after three iterations

(c) Image embeddings after three iterations

Figure 4. **Sanitization effect on prompts and image embeddings distribution of the members and nonmembers sets.** Figure 4a shows that there is a significant difference between prompts embeddings of nonmembers and members samples before the sanitization process. After three iterations of our sanitization Algorithm 1 these distributions match closely – Figure 4b. Despite the fact that the algorithm uses only prompts embeddings, we observe in Figure 4c that image embeddings distributions are also well aligned after the third iteration of our algorithm. We suspect that it is due to the close match between images and their descriptions so that aligning text distributions leads to aligning image distributions as well.

Algorithm 1 Sanitization algorithm

```
$F \leftarrow \emptyset$                                  ▷ trained binary classifiers
$NM \leftarrow$ deduplicated nonmembers
$M \leftarrow$ global set of members
$M_i \leftarrow \emptyset$                                ▷ sanitized members after i-th iteration
$TrainSet \leftarrow \emptyset$                           ▷ training dataset
for $i \leftarrow 1,\ 2,\ ...,\ n$ do
    $TrainSet \leftarrow \emptyset$
    if $i = 1$ then
        $TrainSet \leftarrow$ sample of size $|NM|$ from $M$
    else
        $TrainSet \leftarrow M_{i-1}$
    end if
    $TrainSet \leftarrow TrainSet \cup NM$
    $F_i \leftarrow$ trained classifier on $TrainSet$
    while $|M_i| < |NM|$ do
        $M_{tmp} \leftarrow$ sample from $M$
        for $j \leftarrow 1,\ 2,\ ...,\ i$ do
            if $F_j$ predicts member label for sample then
                $M_{tmp} \leftarrow M_{tmp} \setminus sample$
            end if
        end for
        $M_i \leftarrow M_i \cup M_{tmp}$
    end while
end for
$SM \leftarrow M_n$                                       ▷ final sanitized members set
```

consists of 600M samples, our deduplicated nonmembers set has 42.5k samples). The main reason is that the deduplication process (see Sec. 5.3) is a great bottleneck of our system; to get duplicate candidates for all 160k nonmember candidates, we query the retrieval API for 50h.

Results To obtain our final set of 40k sanitised members we apply the algorithm for three iterations. To obtain this subset we filter approximately 5M samples from LAION Aesthetics v2 5+ dataset, which takes only 2h on a single NVidia RTX 2080Ti. Using our assessment methodology we confirm its efficacy. FID score in the comparative case drops significantly compared to the starting members set. We see that the PCA components' distribution align (see Fig. 4b) and a binary classifier's accuracy is almost random.

Images embeddings Regarding the image embeddings for LAION-2B EN and LAION-2B Multi Translated, we expect them to have the same characteristics as they come from the same source. This assumption is confirmed by a low FID value between them, see Tab. 1. Additionally, the FID value decreases even further after the text-focused sanitization and a binary classifier's accuracy is almost random. Therefore, additional sanitization efforts focused on image embeddings are not necessary, and the PCA components are also aligned, see Fig. 4c).

6. Experiments

6.1. Threat model

A membership inference attack is defined as follows. We consider an adversary A that aims to infer whether a single data point x was included in the training set D of a generative model M. The attacker has no knowledge about the dataset D and is only able to query the model M. We distinguish three scenarios according to the attacker's capabilities.

- In the **black-box** scenario, an adversary queries a generative model with a text prompt and gets a generated image. The attacker has no knowledge about the model architecture and no access to its weights.

- In the **grey-box** scenario an adversary has access to the visual and text encoders of the attacked model. Thus, they are able to calculate the latent representation (embedding) of a given image and text prompt. However, the attacker still has no access to the model weights.
- Finally, in the **white-box** scenario, an adversary has full access to the model, its source code and trained weights.

We start with a baseline white-box model loss threshold attack, which is based on the fact that machine learning models learn by minimising the loss in the training samples. We extend our analysis by covering metrics related to model inference. Moreover, we introduce new white-box attack methods and show that they outperform the commonly used baseline method. We also describe and evaluate a grey and a black-box scenario. The attacks are based on the intuition that generative models tend to synthesise samples similar to their training set. For all attacks, we evaluate a variant in which the losses are obtained as the average of 5 losses (5 different passes through the model, each time with a different noise) following the findings of [3]. We explore more attack methods in the Appendix D.

6.2. Threshold attack

A general *threshold* attack is formulated as follows. For a selected threshold τ, the attack classifies the image x as a member if $\mathcal{L} < \tau$. Otherwise, x belongs to the nonmember set. Commonly used *threshold* attacks focus only on a model loss. We extend our analysis by Pixel and Latent error, defined as follows:

Model loss We monitor the loss of the diffusion model given by Eq. 1 $\mathcal{L}(x, t, \epsilon; f_\theta) = \|\epsilon - f_\theta(x_t, t)\|_2^2$.

Pixel error We define the pixel error as the reconstruction error between an original image x and the generated image x' defined as $\mathcal{L}(x, x') = \|x - x'\|_2^2$.

Latent error This measurement is similar to the pixel error. However, it focuses on the reconstruction error between a latent representation z of the original image x and the latent representation z' generated by the diffusion model. The error $\mathcal{L}(z, z')$ is defined as $\|z - z'\|_2^2$.

6.3. Attack methods

We present a baseline and the best-performing attack methods evaluated in our experiments. Most methods are only applicable under the white-box scenario but we also examine the attacks in the grey- and black-box scenario. An exhaustive description and analysis of these different attack methods are given in Appendix D.1.

Baseline loss threshold In [4] the authors show that evaluating the model loss at timestep 100 for the latent with applied noise scale α_t at $t = 100$ yields the best results for the membership inference attacks based on model loss. We follow this method and evaluate the model loss at timestep 100 for the latent noised with scale α_{100}.

Methods exploiting the noise There are many possible variants of adding or removing noise in the diffusion process before we make a final decision based on the loss. The intuition (or hopeful assumption) behind such attacks is that member samples would behave more robustly than nonmembers under noisy conditions. We explore many different settings and refer a reader to Appendix D.1 for details.

Generation from prompt To perform this attack we pass only the prompt associated with the original images to the model. We do so to simulate a real-world scenario, where we have access to the model only via the API. In the black-box scenario, we calculate the Pixel error between the original image and the image generated by the model using the default method of 50 timesteps. In the grey-box scenario, we obtain the generated images in the same way as in the black-box scenario, but then we calculate the latent representation of the original and generated images using the visual encoder of the attacked model and then calculate the latent error between them.

6.4. Targeted datasets

LAION-mi In our paper we use our LAION-mi dataset proposed in Section 5. For each attack, we use the same subset of 5000 member samples and 5000 nonmembers samples, further referred to as the attack set. We find that different training and evaluation set splits can produce significantly different results, some almost 10 times better than others (see Appendix E). Following these findings, we evaluate our attacks on 100 random subsets (evaluation sets) of 500 members and 500 nonmember samples drawn from the attack set and then report the mean and the standard deviation of the performance.

POKEMON POKEMON dataset [17] is a Text2Image dataset. We use it to first finetune the original StableDiffusion v1.4 model using a subset of 633 samples (members), leaving the remaining 200 samples as nonmembers. We evaluate our attacks on 1000 subsets obtained from random 200 member and all nonmember samples. We also conduct an analysis of the influence of overfitting on the attack performance in Appendix F.

7. Results

We evaluate the attacks for two setups. First, we attack the Stable Diffusion v1.4 model, which we do not modify in any way. We draw data samples of members and nonmembers from our LAION-mi dataset. Then, we evaluate the effectiveness of the attacks on the same model, which is

Table 2. ***Threshold* membership inference attacks results on LAION-mi and POKEMON datasets.** We demonstrate the importance of evaluating membership inference attacks in a fair setting. On the POKEMON dataset some of the attacks are almost perfect, with *partial denoising* reaching $\mathbf{99.5}\%$ TPR@FPR=1%, but on ours LAION-mi dataset with original SD-v1.4 we reach at most $\mathbf{2.51}\%$. Our proposed methods outperform the *Baseline loss threshold* method.

			TPR@FPR=1%. ↑	
Scenario	**Loss**	**Method**	**LAION-mi**	**POKEMON**
White-box	Model Loss	Baseline loss thr.	1.92%±0.59	80.9%±2.27
		Reversed noising	2.51%±0.73	97.3%±0.93
		Partial denoising	2.31%±0.61	94.5%±1.34
		Reversed denoising	2.25%±0.64	91.5%±1.63
	Latent Error	Reversed noising	1.26%±0.62	11.5%±1.84
		Partial denoising	2.42%±0.62	99.5%±0.4
		Reversed denoising	2.17%±0.64	61.1%±2.74
	Pixel Error	Reversed noising	1.90%±0.51	8.36%±1.66
		Reversed denoising	2.03%±0.55	12.0%±1.97
		Partial denoising	1.75%±0.68	25.38%±2.55
Grey-box	Latent Error	Generation from prompt	0.93%±0.41	7.15%±1.5
Black-box	Pixel Error	Generation from prompt	0.35%±0.19	12.0%±1.9

finetuned on the POKEMON dataset [17]. Here, we use test and training data splits (633/200 samples) as member and nonmember sets.

Metric A metric to evaluate the membership attack is true-positive rate (TPR) calculated at a low false-positive rate (e.g. FPR=1%). For privacy-related problems, it is a much better metric than common aggregate metrics, such as accuracy or AUC [3].

Discussion In Table 2 we observe a severe discrepancy in the effectiveness of the attacks achieved for the LAION-mi dataset and fine-tuning on POKEMON. In the second case, two white-box attacks (*reversed noising* and *partial denoising*) achieve a very high TPR. However, when the attacks are applied against our LAION-mi dataset, we observe a huge drop in performance. Here we clearly see the effects of the first pitfall from Sec. 4, namely the fine-tuned model overfits, and the membership inference task becomes trivial. We explore this topic further in Appendix F. The obtained results demonstrate that evaluation on a small dataset (used for finetuning the model) is misleading and a more careful setup, such as our proposal, is required.

Moreover, the results show the limited performance of loss-based attacks in the black- and grey-box scenarios, even for the simple POKEMON setting. This highlights an important issue, as many image-generation services work as a black-box API. As mentioned in 3.1 this trend is unlikely to change in the future.

In theory, identifying training samples in black-box scenarios can be approached by extracting training samples from the model, as in [4]. However, this approach requires the generation of hundreds of images per text prompt, which is computationally expensive. This approach is also limited to identifying training samples that have been memorised by the model. For those reasons, such methods are not well suited for identifying the membership of a sample. For the white-box scenario, the state-of-the-art approach is a method based on the shadow models. However, as stated in 3.2, this strategy is too costly for large diffusion models such as Stable Diffusion. We further discuss the applicability of shadow models in Appendix G.

8. Conclusion

We showed that evaluation of membership inference attacks with the model finetuning approach may lead to false conclusions. As an alternative, we proposed a new carefully crafted dataset, which mitigates the main limitation of the original LAION dataset, which is a lack of a test set. Having the proposed dataset and reliable set of nonmembers, we evaluated several membership inference attacks and obtained results, which contradict previous findings.

Our dataset could help the community evaluate attacks on large generative diffusion models such as Stable Diffusion in a more rigorous and fair setting. A clear picture of how successful the membership attacks are is essential for a sound policy on matters such as data ownership and privacy. We argue that for large diffusion models, where shadow models are prohibitively expensive, membership inference remains a very challenging task.

References

[1] Romain Beaumont. Clip retrieval: Easily compute clip embeddings and build a clip retrieval system with them. https://github.com/rom1504/clip-retrieval, 2022.

[2] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[3] 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.

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

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

[6] Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. Are diffusion models vulnerable to membership inference attacks?, 2023.

[7] Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. Beyond english-centric multilingual machine translation, 2020.

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[9] 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, 2018.

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

[11] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[12] Huggingface. Fine tuning stable diffusion. https://huggingface.co/docs/diffusers/v0.15.0/training/text2image, 2023. [Online].

[13] B. D. Jovanovic and P. S. Levy. A look at the rule of three. *The American Statistician*, 51(2):137–139, 1997.

[14] LAION. Laion translated: 3b captions translated to english from laion5b, 2022. [Online].

[15] OpenReview.net. Review of "membership inference attacks against text-to-image generation models". https://openreview.net/forum?id=J41IW8Z7mE, 2023. [Online].

[16] Justin Pinkney. Fine tuning stable diffusion. https://github.com/LambdaLabsML/examples/tree/main/stable-diffusion-finetuning, 2022.

[17] Justin N. M. Pinkney. Pokemon blip captions. https://huggingface.co/datasets/lambdalabs/pokemon-blip-captions/, 2022.

[18] 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.

[19] 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.

[20] 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.

[21] Reuters. Getty images lawsuit says stability ai misused photos to train ai. https://www.reuters.com/legal/getty-images-lawsuit-says-stability-ai-misused-photos-train-ai-2023-02-06/, 2023.

[22] Reuters. Lawsuits accuse ai content creators of misusing copyrighted work. https://www.reuters.com/legal/transactional/lawsuits-accuse-ai-content-creators-misusing-copyrighted-work-2023-01-17/, 2023.

[23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. Compvis stable diffusion v1-4 model card, 2022. [Online].

[24] 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.

[25] 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.

[26] Christoph Schuhmann. Clip+mlp aesthetic score predictor, 2022. [Online].

[27] Christoph Schuhmann. Laion-aesthetics, 2022. [Online].

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

[29] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid`, August 2020. Version 0.3.0.

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

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

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

[33] Duy Tin Vo and Richard Khoury. Language identification on massive datasets of short message using an attention mechanism cnn, 2019.

[34] Ryan Webster, Julien Rabin, Loic Simon, and Frederic Jurie. On the de-duplication of laion-2b, 2023.

[35] 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.

[36] 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.