CDI: Copyrighted Data Identification in Diffusion Models

Jan Dubiński [*][†]
Warsaw University of Technology, IDEAS NCBR
jan.dubinski.dokt@pw.edu.pl

Antoni Kowalczuk [*]
CISPA Helmholtz Center for Information Security
antoni.kowalczuk@cispa.de

Franziska Boenisch
CISPA Helmholtz Center for Information Security
boenisch@cispa.de

Adam Dziedzic
CISPA Helmholtz Center for Information Security
adam.dziedzic@cispa.de

Abstract

Diffusion Models (DMs) benefit from large and diverse datasets for their training. Since this data is often scraped from the Internet without permission from the data owners, this raises concerns about copyright and intellectual property protections. While (illicit) use of data is easily detected for training samples perfectly re-created by a DM at inference time, it is much harder for data owners to verify if their data was used for training when the outputs from the suspect DM are not close replicas. Conceptually, membership inference attacks (MIAs), which detect if a given data point was used during training, present themselves as a suitable tool to address this challenge. However, we demonstrate that existing MIAs are not strong enough to reliably determine the membership of individual images in large, state-of-the-art DMs. To overcome this limitation, we propose Copyrighted Data Identification *(*CDI*), a framework for data owners to identify whether their dataset was used to train a given DM.* CDI *relies on dataset inference techniques, i.e., instead of using the membership signal from a single data point,* CDI *leverages the fact that most data owners, such as providers of stock photography, visual media companies, or even individual artists, own datasets with multiple publicly exposed data points which might all be included in the training of a given DM. By selectively aggregating signals from existing MIAs and using new handcrafted methods to extract features from these datasets, feeding them to a scoring model, and applying rigorous statistical testing,* CDI *allows data owners with as little as 70 data points to identify with a confidence of more than 99% whether their data was used to train a given DM. Thereby,* CDI *represents a valuable tool for data owners to claim illegitimate use of their copyrighted data. We make our code available at* https://github.com/sprintml/copyrighted_data_identification.

---
[*]Equal contribution.
[†]Work done while the author was at CISPA.

1. Introduction

In recent years, large diffusion models (DMs) [56] have rapidly gained popularity as a new class of generative models, surpassing the performance of prior approaches, such as Generative Adversarial Networks [23]. DMs now power several state-of-the-art image generators including Stable Diffusion [47], Midjourney [60], Runway [47], Imagen [50], and DALL-E 2 [41, 42].

To reach their powerful performance, DMs need to be trained on large amounts of high-quality and diverse data. This data is usually scraped from the Internet, often without respecting the copyrights of the data owners. Especially since it has been shown that DMs are capable of generating verbatim copies of their training data at inference time [11], this represents a violation of intellectual property rights. Recently, Getty Images, a leading visual media company, filed a lawsuit against Stability AI, the creators of Stable Diffusion, alleging the unauthorized use of copyright-protected images [5, 46]. This case has sparked a wave of additional lawsuits, with many more now addressing intellectual property infringement by generative AI companies [44, 45]. Unfortunately, as it becomes obvious during the lawsuits—particularly for training data points that are not output in a verbatim form during inference time—verifying that these data points have been illegitimately used for training the DMs is a challenging task.

Membership inference attacks [55] that aim at identifying whether a specific data point was used to train a given model, in theory, present themselves as a solution to the problem. Unfortunately, prior work [17] indicates that performing a realistic MIA on large DMs is a very challenging task. One of the practical challenges lies in the prohibitive costs of training state-of-the-art DMs (*e.g.,* \$600.000 for Stable Diffusion) which renders potent MIAs utilizing multiple *shadow model* copies [10, 55] infeasible. To further explore the practicality of MIAs for identifying copyrighted samples used to train large DMs, we perform an extensive study, evaluating the
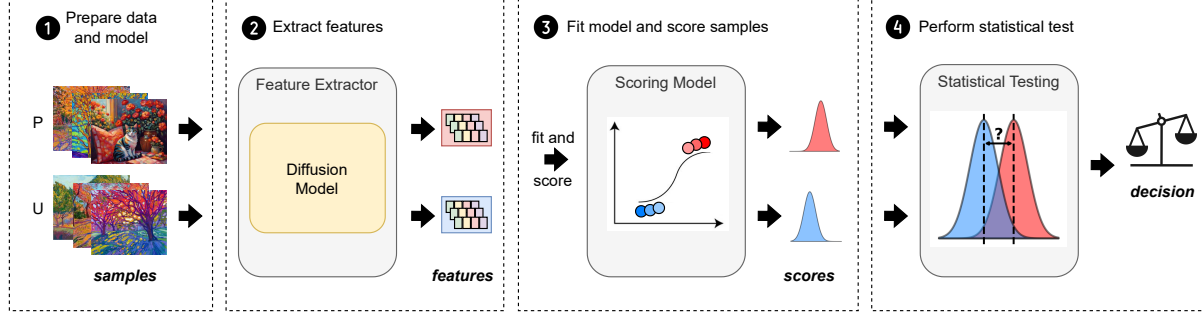
Figure 1. **Overview of our Dataset Inference method for Diffusion Models from our previous work.** Our approach consists of the following stages: ❶ Prepare the query data to verify if the *published* suspect samples **P** were used to train the DM. The *unpublished* samples **U** from the same distribution as **P** serve as the validation set. ❷ Run inference on all the inputs {**P**, **U**} to extract their membership features. Use current MIAs and our handcrafted features. ❸ Find useful features and learn a discriminator. **P** and **U** sets are split into $\mathbf{P}_{ctrl}$, $\mathbf{P}_{test}$ and $\mathbf{U}_{ctrl}$, $\mathbf{U}_{test}$. The features for $\mathbf{P}_{ctrl}$ and $\mathbf{U}_{ctrl}$ are used to train a scoring model to selectively combine features and differentiate between the samples from $\mathbf{P}_{test}$ and $\mathbf{U}_{test}$. ❹ Apply a statistical t-test to verify if the scores obtained for public suspect data point **P** are statistically significantly higher than scores for **U**, in which case, **P** is marked as being a part of the DM's training set. Otherwise, the test is inconclusive and the DM's training set is resolved as independent of **P**. The method

success of existing MIAs against DMs' training data for various open DMs. Our findings demonstrate that **current MIAs for DMs are limited in confidently identifying DMs' training data points in case of models trained on large datasets**—showcasing that individual MIAs cannot reliably support copyright claims.

In light of this result, we, however, observe that in most cases, data owners, such as stock photography, visual media companies, or even individual artists, typically seek to verify the use of not just a single data point but a collection of their work as training data for a given DM. This moves the idea of *dataset inference* [38] (DI) into focus. DI was first proposed to detect stolen copies of supervised classifier models and then subsequently extended to self-supervised models [18]. It leverages the observation that, while MIAs on individual data points do not produce a strong signal, selectively aggregating signals across a subset of the training dataset and applying statistical testing can reveal a distinct signature of the model. This dataset-based signature allows for the detection of stolen model copies with a confidence level exceeding 95%. Yet, to date, it remains unexplored whether the principles of DI actually transfer to DMs and are suitable to identify subsets of their training data rather than resolving model ownership—given the vast amount of heterogeneous data DMs are initially trained on. Additionally, it is unclear how large the required training data subsets for verification would have to be. Finally, we do not know the specific features needed to extract a strong signal over the training data.

To close these gaps, **we propose `Copyrighted Data Identification`, designed to answer the critical question:** *Was this DM trained on a copyrighted collection of images?* The overall schema of our method is illustrated in Fig. 1. To design `CDI`, we move beyond simply aggregating features extracted by existing MIAs, as these often

produce signals that are too weak to achieve highly confident DI, rendering the approach impractical. Instead, we firstly, extend the feature extraction methods by our newly proposed features. Secondly, we design a scoring function which maps the extracted information into sample membership probability, learning the features relevant for each DM. Finally, in contrast to MIAs which usually refer to metrics like True Positive Rate (TPR) or Area Under Curve (AUC) which do not give any confidence estimate, we equip `CDI` with rigorous statistical testing as the final component.

We demonstrate the success of our method on diverse large-scale DM architectures (LDM [47], DiT [39], U-ViT [6]), including unconditioned, class-conditioned and text-conditioned models, trained on various image resolutions. Our results show that `CDI` achieves a confident detection rate of data (illegitimately) used for training DMs. `CDI` remains effective when only **a part of the investigated data was actually used in DM training**. Moreover, we demonstrate that `CDI` does not yield false positives, making it a reliable tool for detecting and confidently claiming the use of copyrighted data in DMs.

In summary, we make the following contributions:

- We demonstrate that existing MIAs for DMs show limited effectiveness in confidently identifying the training data points of large, state-of-the-art models.

- To address this issue, we propose `CDI`, a method that empowers data owners to identify whether their data has been (illegitimately) used to train a DM, incorporating rigorous statistical testing to ensure confidence in the results.

- We perform thorough feature engineering to amplify the signal in `CDI`, proposing novel feature extraction methods and enabling data owners even with smaller datasets to benefit from our method.

- We evaluate `CDI` on a wide range of DMs and their pre-training datasets and provide a unified open-source code-base[1] with a common interface to all prior MIAs and our new `CDI`, serving as a valuable evaluation testbed for the community.

2. Background

Diffusion Models [24, 58] are generative models trained by progressively adding noise to the data and then learning to reverse this process. The forward diffusion process adds Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ to a clean image x in order to obtain a noised image $x_t \leftarrow \sqrt{\alpha_t}x + \sqrt{1 - \alpha_t}\epsilon$, where $t \in [0, T]$ is the diffusion timestep, and $\alpha_t \in [0, 1]$ is a decaying parameter such that $\alpha_0 = 1$ and $\alpha_T = 0$. The diffuser f_θ is trained to predict the ϵ for various timesteps, by minimizing the objective $\frac{1}{N} \sum_i \mathbb{E}_{t,\epsilon} \mathcal{L}(x_i, t, \epsilon; f_\theta)$, where N is the training set size, and

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

The generation iteratively removes the noise prediction $f_\theta(x_t, t)$ from x_t for $t = T, T-1, ..., 0$, starting from $x_T \sim \mathcal{N}(0, I)$, and obtaining a generated image $x_{t=0}$. To guide this process, for conditional image generation the diffuser f_θ receives an additional input y, which represents a class label for the class-conditional DMs [24] or a text embedding, obtained from a pretrained language encoder like CLIP [40], for the text-to-image DMs [42, 47, 50].

Latent diffusion models [47] (LDMs) improve DMs by conducting the diffusion process in the latent space, which significantly reduces computational complexity, making training scalable and inference more efficient. For the LDMs, the encoder $\mathcal{E}$ transforms the input x to the latent representation $z = \mathcal{E}(x)$ and Equation 1 becomes

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

Membership Inference Attacks. MIAs aim to determine whether a specific data point was used to train a given machine learning model [55]. Extensive research has explored MIAs against supervised machine learning models [10, 49, 62, 70]. On the high level, MIAs operate on the premise of overfitting, assuming that training data points (members) exhibit smaller training loss compared to data points not encountered during training (non-members). Initial MIAs against DMs [9] focus on assessing the membership of samples by evaluating the model's noise prediction loss. Their findings establish that the loss value at the *diffusion timestep* $t = 100$ proves most discriminative between member and non-member samples. Intuitively, if t is too small ($t < 50$), the noisy image resembles the original, making the noise prediction too easy. Otherwise, if t is too large

($t > 300$), the noisy image resembles random noise, making the task overly challenging. Among the recent MIA approaches targeting DMs, the Step-wise Error Comparing Membership Inference (SecMI) attack [16] infers membership by estimating errors between the sampling and inverse sampling processes applied to the input x also at timestep $t = 100$. Following the same overfitting principle, the Proximal Initialization Attack (PIA) [27] enhances SecMI by assessing membership based on the difference in the model's noise prediction for a clean sample x at timestep $t = 0$ and a noised sample x_t at $t = 200$, where the method was found to be most discriminative.

Protecting Intellectual Property in DMs   Protecting intellectual property (IP) in DMs involves safeguarding against unauthorized usage of trained models and attributing generated data to their source models, while also protecting the IP of the data used for training. Several attribution methods focus on watermarking at both the model and input levels, embedding invisible watermarks into generated images or subtly influencing the sampling process to create model fingerprints [19, 33]. Other techniques explore fingerprinting methods, where unique patterns or signals are embedded into generated data for identification purposes [68, 71]. However, those methods protect the IP in trained models and generated data, leaving the IP of *training data* out of scope. To solve this issue, various approaches aim to protect against style mimicry and unauthorized data usage by adding perturbations to images or detecting unauthorized data usage through injected memorization or protective perturbations [22, 54, 63, 66, 67, 69]. However, existing methods have important drawbacks, such as limiting the data usage for consensual applications and providing no protection if the data IP has already been breached. Moreover, a malicious party may attempt to overcome the safety mechanism by image purification methods [7]. Our proposed method fills in those gaps by enabling data owners to identify whether their data has been illegitimately used for training, without any requirements to modify the protected content. While the previous work showed the possibility of computing the influence of the training data points on the generated outputs [73], we propose to go a step further and exactly detect which data points are used for training.

3. Limitations of MIAs in Member Detection

We rigorously evaluate existing MIAs to test their ability to detect training members in large, complex DMs. Prior studies [16, 27] reported success with MIAs in accurately identifying DMs training members; however, these results were often based on toy models or datasets (*e.g.*, CIFAR100 [29]) that do not reflect the complexity of high-dimensional, diverse DM setups. Our analysis on state-of-the-art DMs trained on extensive datasets (*i.e.*, ImageNet-1k [14] or

COCO [64]) reveals significant performance limitations of existing MIAs and key factors that contributed to overestimated effectiveness in previous work. Full details are provided in Appendix D.

3.1. Evaluated MIAs

1. *Denoising Loss* [11]: The loss is computed from Equation 2 five times for the diffusion timestep $t = 100$, as indicated in the original paper. The final membership score is the average loss, where a lower value indicates that the sample is a member.

2. *SecMI$_{stat}$* [16]: The membership score extracted by SecMI aims to approximate the posterior estimation error of f_θ on the latent z (obtained from the image encoder part), claiming it should be lower for members than for non-members.

3. *PIA* [27]: The score extracted by PIA aims at capturing the discrepancy between the noise prediction on a clean sample's latent z and the noise prediction on its noised version z_t at time t. This discrepancy should be lower for members.

4. *PIAN* [27]: This MIA is an adaptation of the original PIA to further strengthen the membership signal. The noise prediction on z is normalized, so it follows the Gaussian distribution. Similar to PIA, the scores returned from PIAN are expected to be lower for members than for non-members.

3.2. Experimental Setup

Models. We evaluate class-conditioned, as well as text-conditioned state-of-the-art DMs of various architectures, namely LDM [47], U-ViT [6], and DiT [39]. We employ already trained checkpoints provided by the respective papers [1, 2, 4]. For class-conditioned generative tasks, LDM offers one model checkpoint with the resolution of 256x256 (LDM256). For U-ViT and DiT, we have access to models operating on resolutions of 256x256 and 512x512 (U-ViT256, U-ViT512, DiT256, DiT512). Additionally, we conduct experiments on text-conditioned models based on U-ViT architecture (U-ViT256-T2I, U-ViT256-T2I-Deep) and a newly trained unconditional U-ViT256-Uncond model (see App. I).

Datasets. For the class-conditional evaluation, we use models trained on ImageNet-1k [14]. This dataset contains large-sized colored images with 1000 classes. There are 1,281,167 training images and 50,000 test images. For text-conditional task evaluation, we use models trained on COCO-Text dataset [64], a large-scale object detection, segmentation, and captioning dataset which contains 80,000 training images and 40,000 test images, each with 5 captions.

3.3. Performance of MIAs in Member Detection

Our results indicate that the existing MIAs achieve performance comparable to random guessing. We present the aggregated max and average **TPR@FPR=1%** across 8 DMs in Tab. 1, and defer the full evaluation to App. U. For completeness, we provide AUC (Table 14), accuracy (Table 13), and ROC curves (Fig. 17) of MIAs there.

Table 1. **TPR@FPR=1% for MIAs.** Performance of existing MIAs in identifying training members is limited.

Attack	Max TPR@FPR=1%	Mean TPR@FPR=1%
Denoising Loss [10]	2.24	1.61
SecMI$_{stat}$ [16]	2.44	1.50
PIA [27]	5.57	2.18
PIAN [27]	1.53	1.03

4. Our `CDI` Method

Recognizing the limitations of MIAs on large, state-of-the-art DMs, we shift our focus to DI and introduce our `CDI` method. To achieve reliable and confident detection of data collections used in model training, we go beyond simply aggregating features from existing MIAs. Our `CDI` consists of four stages: (1) data and model preparation, (2) feature engineering and extraction, extended by our three newly proposed detection methods (3) a scoring function that maps these features to scores, and (4) a rigorous statistical hypothesis testing, enabling high-confidence decisions. We visualize and describe `CDI` in Figure 1.

Dataset Inference. DI was initially introduced as a tool for detecting model stealing attacks [61]. In the context of supervised models [38], DI involves crafting features for a set of training data points, inputting them into a binary classifier, and applying statistical testing to establish model ownership. The features of supervised learning are based on the fact that classifiers are trained to maximize the distance of training examples from the model's decision boundaries while test examples typically lie closer to these boundaries, as they do not influence the model's weights during training. DI was extended to self-supervised learning (SSL) [18] by observing that training data representations exhibit a markedly different distribution from test data representations. Building on this intuition, we design specific features based on the DM's behavior for a set of data points that we want to test for potential (illegitimate) use in training the DM. We then map those features to scores on which we apply statistical testing. Unlike traditional DI, which focuses on ownership resolution for the entire model, our approach is tailored for data verification, allowing owners of small subsets of the DM's training data to verify their use in model training.

Notation and Setup. We denote $\mathbf{P}$ as a set of samples that we suspect to be (illegitimately) used for training the DM. Those are *published* samples provided by the data owner who wants to make a claim for their intellectual property. Further, we refer to $\mathbf{U}$ as a set of *unpublished* samples, from the same distribution as $\mathbf{P}$, that serves as the validation set. In real scenarios, $\mathbf{U}$ might come from a creator's unpublished data or sketches of their released work. We assume $\mathbf{P}$ to be *i.i.d.* with U.

Data Preparation and Processing. We split $\mathbf{P}$ and $\mathbf{U}$ into $\mathbf{P}_{\text{ctrl}}$, $\mathbf{P}_{\text{test}}$, and $\mathbf{U}_{\text{ctrl}}$, $\mathbf{U}_{\text{test}}$. We extract the final full set of features for $\mathbf{P}_{\text{ctrl}}$ and $\mathbf{U}_{\text{ctrl}}$ and train the scoring model s to tell apart members from non-members, such that s eventually outputs higher values when presented with a member. Then, we apply s to the features extracted from $\mathbf{P}_{\text{test}}$ and $\mathbf{U}_{\text{test}}$. Finally, we perform statistical testing to find whether the scores returned by s on $\mathbf{P}_{\text{test}}$ are significantly higher than those on $\mathbf{U}_{\text{test}}$, which would indicate that $\mathbf{P}$ was, indeed, used to train the DM.

Threat Model. We design CDI as a tool for use in legal proceedings. Consequently, the CDI procedure is carried out by a third trusted party, referred to as an *arbitrator*. The arbitrator is approached by a victim, whose private data might have been potentially used in a training of a DM. The arbitrator executes CDI either in the gray-box model access, (can only obtain outputs, *i.e.,* noise predictions, for given inputs to a DM at an arbitrary timestep t) or in the white-box model access (where DM's internals and parameters can be inspected). The access type depends on the requirements of the features used in CDI (we provide more details in App. N).

4.1. Features

We utilize MIAs (Sec. 3.1) as the source of features for CDI. Additionally, to increase the discriminative capabilities of our CDI, we propose the following three novel features that can be extracted from a DM to provide additional information on a sample's membership score. Our final feature extractor implements a function $f_e : \mathbb{R}^{C \times H \times W} \to \mathbb{R}^k$, with C, H, W denoting the channels, height, and width of an input sample, respectively, and k being the dimensionality of the extracted feature vector.

Gradient Masking (GM). This feature aims at capturing the difference in the ability to restore destroyed semantic information between members and non-members. It is inspired by the *Degrade, Restore, Compare* (DRC) idea from Fu et al. [20] who identify that for members, a restoration is more successful. To compute the feature, we first capture the gradient $\mathbf{g} = |\nabla_{z_t} \mathcal{L}(z_t, t, \epsilon; f_\theta)|$. Intuitively, $\mathbf{g}$ indicates the influence each feature value in the latent z_t has on the loss $\mathcal{L}$. We are interested in the features from z_t that exhibit

the highest influence on the loss. Therefore, we create a binary mask $\mathbf{M}$ for the top 20% values in $\mathbf{g}$. This mask indicates significant regions of the latent representation z_t. Next, we obtain $\hat{z}_t = \epsilon \cdot \mathbf{M} + z_t \cdot \neg \mathbf{M}$, with the significant values of z_t destroyed by replacing them with random noise $\epsilon \sim \mathcal{N}(0, I)$, and the rest left unchanged. Finally, we compute $||(\epsilon - z_t) \cdot \mathbf{M} - f_\theta(\hat{z}_t, t) \cdot \mathbf{M}||_2^2$ as the feature. This feature expresses the reconstruction loss over the semantically most relevant regions and should be lower for members. We calculate the feature at multiple diffusion timesteps t, to further strengthen the signal. Note that we differ in our feature computation from Fu et al. [20] in two significant aspects. (1) While their DRC employs powerful third-party self-supervised vision encoders like DINO [12] to identify semantically significant regions, we utilize only the information from within the DM to obtain the mask $\mathbf{M}$. (2) Additionally, we utilize the model's loss as our final signal, instead of computing cosine similarity between representations returned from DINO for clean and restored samples, rendering our method more self-contained and independent of the signal from other models.

Multiple Loss (ML). To increase the membership signal from the model prediction loss, we compute Eq. 2 at multiple (10) diffusion timesteps $t = 0, 100, ..., 900$ to provide more information to train the scoring function s.

Noise Optimization (NO). We leverage an insight initially observed in supervised classifiers, namely that the difficulty of changing the predicted label of a sample through adversarial perturbations differs between members and non-members [32]. In particular, it takes a stronger perturbation to change the prediction for members. The reason is that ML models return more confident predictions on training samples (members). To craft our feature, we adapt this intuition to DMs. We note that perturbing a noised sample z_t to minimize the model noise prediction loss expressed in Equation 2 achieves better results, *i.e.,* lower loss values for member samples. Specifically, we conduct an unbounded optimization of the perturbation δ applied to the noised latent representation z_t at timestep $t = 100$. Our objective function is defined as: $\arg\min_\delta ||\epsilon - f_\theta(z_t + \delta, t)||_2^2$. To optimize this objective, we employ the 5-step L-BFGS algorithm [74] (which is commonly used to generate adversarial perturbations [8, 59]). We use the resulting values of the noise prediction error $||\epsilon - f_\theta(z_t + \delta, t)||_2^2$ and the amount of perturbation $||\delta||_2^2$ as features.

We provide further analysis of our features in App. T.

4.2. Scoring Model

Based on the set of features extracted for $\mathbf{P}_{\text{ctrl}}$ and $\mathbf{U}_{\text{ctrl}}$, we train a logistic regression model, $s : \mathbb{R}^k \to [0, 1]$. We then apply s to the features extracted for $\mathbf{P}_{\text{test}}$ and $\mathbf{U}_{\text{test}}$ and use the resulting logits $s(f_e(\mathbf{P}_{\text{test}}))$ and $s(f_e(\mathbf{U}_{\text{test}}))$ as membership
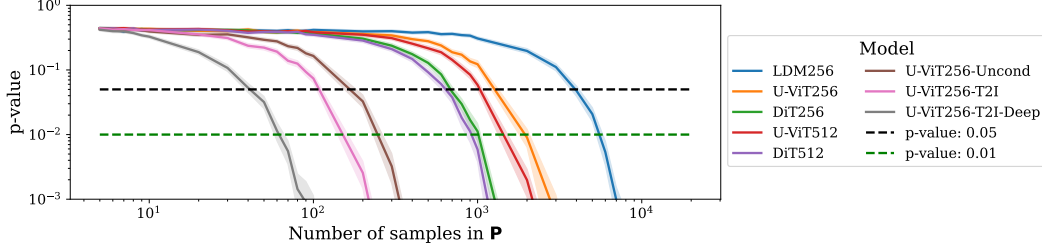
Figure 2. **Results of `CDI` on various DMs.** Solid lines indicate p-values aggregated over 1000 randomized trials for each size of $\mathbf{P}$, shaded areas around the lines are 95% confidence intervals. `CDI` confidently rejects H_0 with as low as 70 suspect samples from the data owner. `CDI`'s performance increases with larger model sizes and smaller training sets.

confidence scores, with higher values referring to higher confidence that the given input sample is a member.

The motivation behind relying on the feature vector extracted by f_e is that a single feature-based score is prone to high variance [10], in turn making it more difficult to perform successful data detection. In contrast, combining multiple features should amplify the signal and improve the performance. Our experiments confirm this intuition, as we show in Sec. 3.3 and 5.1. Moreover, the scoring model addresses the challenge of determining which features provide the strongest membership signal for a given model. By aggregating information across multiple features, s highlights the most relevant signals for detecting membership.

4.3. Statistical Testing

Finally, we perform a two-sample single-tailed Welch's t-test. Our null hypothesis expresses that mean scores for $\mathbf{P}_{\text{test}}$ are not significantly higher than the ones for $\mathbf{U}_{\text{test}}$, *i.e.,* $H_0 : \overline{s(f_e(\mathbf{P}_{\text{test}}))} \leq \overline{s(f_e(\mathbf{U}_{\text{test}}))}$, where $\overline{s(f_e(\mathbf{P}_{\text{test}}))}$ and $\overline{s(f_e(\mathbf{U}_{\text{test}}))}$ are the mean of scores returned from s on the features of $\mathbf{P}_{\text{test}}$ and $\mathbf{U}_{\text{test}}$, respectively. Rejecting H_0 at a significance level $\alpha = 0.01$, *i.e.,* obtaining p-value < 0.01, confirms that samples from $\mathbf{P}_{\text{test}}$ has been (illegitimately) used to train the queried DM.

Our choice of such a low α parameter is motivated by the **TPR@FPR=1%** metric established for MIAs [10], based on the intuition that the false positives are more harmful than false negatives in real-world applications, *e.g.,* court cases. To improve the soundness of our statistical tests, we perform `CDI` 1000 times on randomly sampled subsets of $\mathbf{P}$ and $\mathbf{U}$, and aggregate the obtained p-values [28, 65] (we provide more details in App. E).

5. Empirical Evaluation

Our CDI Setup. We use the diffusion models and datasets as specified in Sec. 3.2. To instantiate our `CDI`, we draw samples from the train sets to represent $\mathbf{P}$ and samples from the test sets to represent $\mathbf{U}$. We set $|\mathbf{P}| = |\mathbf{U}|$ for all experiments. The maximum total size of $|\mathbf{P}| + |\mathbf{U}|$ we use for our experiments is 40,000 samples. Note, that this number is chosen only as a starting point and the number of data points

for $\mathbf{P}$ and $\mathbf{U}$ that `CDI` requires to confidently reject H_0 is much lower and depends on the targeted model (see Fig. 2).

To maximize the use of both $\mathbf{P}$ and $\mathbf{U}$ while minimizing the number of samples required for our method, we implement a k-fold cross-validation with $k = 5$. The features extracted from the public samples ($\mathbf{P}$) and unpublished samples ($\mathbf{U}$) are divided into 5 folds. In each iteration, one fold is designated as the test set, containing $\mathbf{P}_{\text{test}}$ and $\mathbf{U}_{\text{test}}$ features, while the remaining $k - 1$ folds form the control set, comprising $\mathbf{P}_{\text{ctrl}}$ and $\mathbf{U}_{\text{ctrl}}$ features, which are used to train the scoring model s. This process is repeated across all splits, ensuring that each sample in $\mathbf{P}$ and $\mathbf{U}$ is used exactly once in the test set as part of $\mathbf{P}_{\text{test}}$ and $\mathbf{U}_{\text{test}}$.

This procedure ensures that the statistical testing is performed with $|\mathbf{P}_{\text{test}}| = |\mathbf{P}|$ and $|\mathbf{U}_{\text{test}}| = |\mathbf{U}|$, allowing us to extract signals that identify training data from the entirety of the $\mathbf{P}$ and $\mathbf{U}$ sets.

5.1. Our CDI Confidently Identifies Collection of Data Samples as Training Data

We summarize the success of our `CDI` for diverse DMs and datasets in Fig. 2 (following the standard evaluation of DI as proposed in [38]). We report p-values as the confidence in the correct verification for different sizes of suspect data sets $\mathbf{P}$. Our results highlight that `CDI` already enables a confident ($p < 0.01$) dataset identification with as little as 70 samples (for instance, for U-ViT256-T2I-Deep DM trained on the COCO dataset) provided by the data owner. For DMs trained on larger datasets like ImageNet, we observe the need to increase the size of $\mathbf{P}$ to confidently reject the null hypothesis. More details on the impact of the size of $\mathbf{P}$ on the confidence of `CDI` can be found in App. F. In general, in our results, we identify the following trends: (1) For a given DM architecture, trained on given dataset, the number of samples required for the confident identification of the training data decreases with increasing input resolution (see App. J). (2) The larger the overall training set of the model, the more samples are needed for a confident claim (see also App. J). (3) The higher the number of model training steps the stronger the signal for identifying training data as shown in Fig. 6 in App. K.

Table 2. **Impact of the statistical testing**. The values in the table are **TPR@FPR=1%** and are in %. Results represent the *set-level* MIA (without the statistical testing) vs CDI with the statistical testing. The size of **P** is 1000. Statistical testing is essential for CDI.

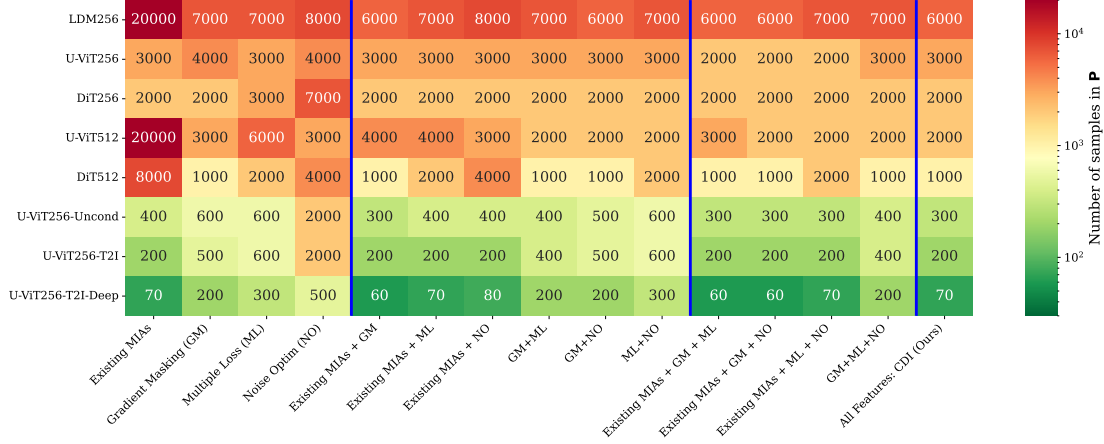|	LDM256	U-ViT256	DiT256	U-ViT512	DiT512	U-ViT256-Uncond	U-ViT256-T2I	U-ViT256-T2I-Deep
Set-level MIA (no t-test)	10.20	22.90	10.50	6.50	0.00	33.40	23.20	32.50
CDI (Ours)	**24.92**	**62.74**	**93.00**	**74.43**	**93.76**	**100.00**	**100.00**	**100.00**

Figure 3. **Impact of feature selection.** The values in the cells indicate the minimum size of **P** needed to reject H_0. Blue vertical lines separate results by the complexity of the feature set used to fit s: (left) our novel individual features from Sec. 4.1 and a joint existing MIAs feature, (second from left) all possible combinations of two, and of three features (second from right), and (right) all available features.

5.2. Analysis of the Success of CDI

We perform multiple ablations on CDI's building blocks to deepen understanding of its success. First, we show the importance of statistical testing as a core component of CDI. Then, we show that our features indeed boost the performance of CDI. Next, we demonstrate that our method remains effective even when not all samples from the suspect set were used in the DM's training set. Finally, we show that CDI does not return false positives. We include additional evaluation of CDI in App. R and S, analysis of scoring model s in App. P, and time complexity in App. I.2. In Appendix H we evaluate CDI on additional DMs and showcase how CDI can be extended with additional feature extraction methods.

Statistical Testing is Crucial for DI. In this ablation study, we assess the impact of removing the t-test from CDI and demonstrate that simply aggregating the MIA results for multiple samples is insufficient to reliably identify data collections used in DM training. To conduct this comparison, we aggregate membership scores across a set of samples to determine if any members are present in the set. We define a *set membership score* as the highest membership score within a subset, hypothesizing that sets composed of members will yield higher scores than non-member sets. For evaluation, we sample 1000 subsets each from **U** (non-members) and **P** (members). We refer to this approach as *set-level MIA*, and execute this procedure for scores obtained from the scoring model component of CDI. We report **TPR@FPR=1%** in Tab. 2.

A direct comparison between CDI's p-values and **TPR@FPR=1%** for set-level MIA is challenging due to the differing metrics. To align them, we compute the power of the t-test with $\alpha = 0.01$, which is equivalent to **TPR@FPR=1%** (see App. Q). This approach allows us to directly compare set-level MIA to CDI without altering its methodology. The results in Tab. 2 underscore the critical role of statistical testing in CDI. Set-level MIA without statistical testing underperforms, while CDI with its rigorous testing achieves near-perfect performance for most of the models.

Our Novel Features Significantly Decrease the Number of Samples Required for Verification. Our results in Fig. 3 indicate that introducing our new features improves the efficiency of CDI substantially in comparison to the joint features extracted by the MIAs from Sec. 3.1, in the following referred to as *existing MIAs*. Applying our new features in CDI leads to a remarkable reduction of the number of samples needed to reject the null hypothesis with $p < 0.01$, especially in the initially most challenging cases of models trained on higher-resolution large datasets. For instance, for U-ViT512 the number of samples required from a data owner for confident verification decreases from 20000 to 2000. This makes CDI more practical and applicable to more users who have smaller datasets that they would like to verify.

Regarding our novel features, the most influential one is GM. We note that utilizing only existing MIAs + GM,

Table 3. **Robustness of `CDI` against false positives.** We depict averaged p-values returned by our method based on the data used within **P** with $|\mathbf{P}| = 10000$. We sample 1000 **P** and **U** sets. The results show that when testing non-members (both **P** and **U** contain only nonmembers), we obtain high p-values, significantly above the significance level ($\alpha = 0.01$), *i.e.,* we cannot reject the null hypothesis and do not identify the data from **P** as members. In contrast, when testing with member data points (**P** contains members and **U** contains nonmembers), our results are always significant, *i.e.,* $p < 0.01$, and we correctly identify the given set as members.

Data in $\mathbf{P}_{\text{test}}$	LDM256	U-ViT256	DiT256	U-ViT512	DiT512	U-ViT256-Uncond	U-ViT256-T2I	U-ViT256-T2I-Deep
Members	10^{-6}	10^{-21}	10^{-59}	10^{-31}	10^{-66}	10^{-266}	0.00	0.00
Non-members	0.40	0.39	0.39	0.39	0.40	0.40	0.39	0.38

we are able to obtain performance very close to `CDI`. The extension of the feature set only with NO yields the smallest improvement over the alternatives in most cases, however, discarding it entirely (see: existing MIAs + GM + ML vs All Features) results in worse performance in the case of U-ViT512, highlighting the need for a diverse source of the signal to obtain confident predictions. Our ML feature succeeds in capturing a better membership signal than its existing MIAs-based counterpart (Denoising Loss), striking a middle ground between GM and NO.

CDI is Effective Even When Not All Data Samples Were Used as Training Data. We investigate `CDI` 's behavior in cases where only a part of the samples in the suspect set **P** was used to train the DM, *i.e.,* **P** contains a certain ratio of non-members, while the remaining samples in **P** are members. Practically, this corresponds to the situation where a data owner has a set of publicly exposed data points and suspects that all of them might have been used to train the DM, whereas, in reality, some were not used. This can happen, *e.g.,* due to internal data cleaning on the side of the party who trained the DM. In particular, the data owner does not know which of their samples and how many of them have not been included into training. In Fig. 4 (extended by Fig. 7 in App. L), we present the success of `CDI` under different ratios of non-member samples in **P**. Note that our evaluation is the result of 1000 randomized experiments for each non-members ratio, model, and **P** size. We observe that `CDI` remains effective when the non-member ratio is over 0.5 and 0.8 for some models, *i.e.,* it still correctly identifies **P** as training data of the model. Overall `CDI`'s robustness is higher when the data owner provides larger suspect datasets **P**. This is reflected in the p-value at the same non-member ratio decreasing as the number of samples in **P** increases.

CDI is Effective Even Under Gray-box Model Access. We analyze the effectiveness of performing `CDI` in the gray-box model access scenario, as defined in the threat model (Sec. 4). Therefore, we include only the original MIA features and Multiple Loss in `CDI`. Even in this case, `CDI` remains effective under gray-box model access and can reject the null hypothesis. In this more difficult scenario, across the eight tested DMs, `CDI` requires on average **one-third**
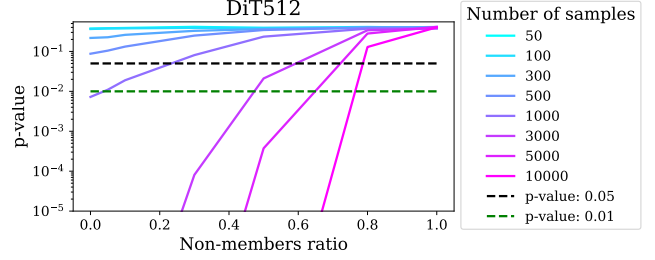

Figure 4. **Impact of non-members ratio in P on `CDI`**. The lines represent p-values for a given non-member ratio while varying sizes of **P**.

more samples in **P** compared to the white-box access (where all features can be used). We refer to App. O for a detailed comparison.

CDI is Robust Against False Positives. While `CDI` can correctly identify suspect datasets even when not all samples were used as training data, it raises the concern of false positives, *i.e.,* reporting data as used for training a DM when it was not. In particular, `CDI` should only reject the null hypothesis if **P** contains (some) members and yield inconclusive results otherwise. To show that `CDI` is robust against false positives, we instantiate **P** only with non-member samples. Our results in Table 3 highlight `CDI` 's reliability in distinguishing between non-member and member sets without false positives.

6. Conclusions

We introduce `CDI` as a method for data owners to verify if their data has been (illegitimately) used to train a given DM. While existing MIAs alone are insufficient to confidently determine whether a specific data point was used during training, `CDI` overcomes this limitation. By selectively combining features extracted from MIAs with novel hand-crafted features and applying them across a larger data set, we achieve a reliable discriminator for identifying datasets used in DM training. Our rigorous feature engineering amplifies the signal in `CDI`, enabling individual artists even with smaller collections of art to benefit from our method.

Acknowledgments

References

[1] Code repository for latent diffusion models., 2021. 4

[2] Code repository for dit models, 2022. 4

[3] Stability diffusion vae checkpoint, 2022. 14

[4] Code repository for u-vit models ., 2023. 4

[5] Getty Images (US), Inc. v. Stability AI, Inc. In *1:23-cv-00135-JLH*, 2023. 1, 12

[6] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22669–22679, 2023. 2, 4, 15

[7] Bochuan Cao, Changjiang Li, Ting Wang, Jinyuan Jia, Bo Li, and Jinghui Chen. IMPRESS: Evaluating the resilience of imperceptible perturbations against unauthorized data usage in diffusion-based generative AI. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 3

[8] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017. 5

[9] 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. 3, 12

[10] 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. 1, 3, 4, 6, 12, 17, 24

[11] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270, 2023. 1, 4, 12

[12] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 5

[13] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, Hillsdale, NJ, 2nd edition, 1988. 17

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3, 4

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 15

[16] Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. Are diffusion models vulnerable to membership inference attacks? In *Proceedings of the 40th International Conference on Machine Learning*, pages 8717–8730. PMLR, 2023. 3, 4, 12, 13, 17, 24

[17] Jan Dubiński, Antoni Kowalczuk, Stanisław Pawlak, Przemyslaw Rokita, Tomasz Trzciński, and Paweł Morawiecki. Towards more realistic membership inference attacks on large diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4860–4869, 2024. 1, 13

[18] Adam Dziedzic, Haonan Duan, Muhammad Ahmad Kaleem, Nikita Dhawan, Jonas Guan, Yannis Cattan, Franziska Boenisch, and Nicolas Papernot. Dataset inference for self-supervised models. *Advances in Neural Information Processing Systems*, 35:12058–12070, 2022. 2, 4

[19] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22466–22477, 2023. 3

[20] Xiaomeng Fu, Xi Wang, Qiao Li, Jin Liu, Jiao Dai, and Jizhong Han. Model will tell: Training membership inference for diffusion models. *arXiv preprint arXiv:2403.08487*, 2024. 5, 12, 13

[21] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23164–23173, 2023. 14

[22] Aditya Golatkar, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Training data protection with compositional diffusion models, 2024. 3

[23] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 1

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020. 3

[25] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 12

[26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 14

[27] Fei Kong, Jinhao Duan, RuiPeng Ma, Heng Tao Shen, Xiaoshuang Shi, Xiaofeng Zhu, and Kaidi Xu. An efficient

membership inference attack for the diffusion model by proximal initialization. In *The Twelfth International Conference on Learning Representations*, 2024. 3, 4, 12, 13, 17, 24

[28] James T Kost and Michael P McDermott. Combining dependent p-values. *Statistics & Probability Letters*, 60(2):183–190, 2002. 6, 13

[29] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 3

[30] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 13

[31] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020. 14

[32] Zheng Li and Yang Zhang. Membership leakage in label-only exposures. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, page 880–895, New York, NY, USA, 2021. Association for Computing Machinery. 5

[33] Guan-Horng Liu, Tianrong Chen, Evangelos Theodorou, and Molei Tao. Mirror diffusion models for constrained and watermarked generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 3

[34] Qihao Liu, Zhanpeng Zeng, Ju He, Qihang Yu, Xiaohui Shen, and Liang-Chieh Chen. Alleviating distortion in image generation via multi-resolution diffusion models and time-dependent layer normalization. In *Advances in Neural Information Processing Systems (NeurIPS) 37*, 2024. 14

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 15

[36] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017. 17

[37] Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring Flow and Diffusion-Based Generative Models with Scalable Interpolant Transformers. In *Proceedings of the 18th European Conference on Computer Vision (ECCV)*, pages 23–40. Springer, 2024. 14

[38] Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. Dataset inference: Ownership resolution in machine learning. In *Proceedings of ICLR 2021: 9th International Conference on Learning Representationsn*, 2021. 2, 4, 6, 17

[39] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2, 4, 15

[40] 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. 3

[41] 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. 1

[42] 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. 1, 3

[43] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 14

[44] 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. 1

[45] Reuters. Artists take new shot at stability, midjourney in updated copyright lawsuit. https://www.reuters.com/legal/litigation/artists-take-new-shot-stability-midjourney-updated-copyright-lawsuit-2023-11-30/, 2023. 1

[46] 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. 1

[47] 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. 1, 2, 3, 4, 13, 14, 15

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 15

[49] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pages 5558–5567. PMLR, 2019. 3

[50] 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. 1, 3

[51] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Proceedings 2019 Network and Distributed System Security Symposium*. Internet Society, 2019. 12

[52] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. StyleGAN-t: Unlocking the power of GANs for fast large-scale text-to-image synthesis. In *Proceedings of the 40th International Conference on Machine Learning*, pages 30105–30118. PMLR, 2023. 12

[53] 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. 14

[54] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y. Zhao. Glaze: Protecting artists from style mimicry by Text-to-Image models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2187–2204, Anaheim, CA, 2023. USENIX Association. 3

[55] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017. 1, 3, 12

[56] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015. 1

[57] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020. 12

[58] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems*, 2020. 3

[59] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. 2014. 5

[60] Midjourney Team. https://www.midjourney.com/, 2022. 1

[61] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pages 601–618, 2016. 4

[62] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE transactions on services computing*, 14(6):2073–2089, 2019. 3

[63] Thanh Van Le, Hao Phung, Thuan Hoang Nguyen, Quan Dao, Ngoc N. Tran, and Anh Tran. Anti-dreambooth: Protecting users from personalized text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2116–2127, 2023. 3

[64] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016. 4, 13

[65] Vladimir Vovk and Ruodu Wang. Combining p-values via averaging. 2019. 6, 13

[66] Feifei Wang, Zhentao Tan, Tianyi Wei, Yue Wu, and Qidong Huang. Simac: A simple anti-customization method for protecting face privacy against text-to-image synthesis of diffusion models, 2024. 3

[67] Zhenting Wang, Chen Chen, Lingjuan Lyu, Dimitris N. Metaxas, and Shiqing Ma. DIAGNOSIS: Detecting unauthorized data usages in text-to-image diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[68] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Advances in Neural Information Processing Systems*, pages 58047–58063. Curran Associates, Inc., 2023. 3

[69] Haotian Xue, Chumeng Liang, Xiaoyu Wu, and Yongxin Chen. Toward effective protection against diffusion-based mimicry through score distillation. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[70] 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. 3, 13

[71] Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 3

[72] Shengfang Zhai, Huanran Chen, Yinpeng Dong, Jiajun Li, Qingni Shen, Yansong Gao, Hang Su, and Yang Liu. Membership inference on text-to-image diffusion models via conditional likelihood discrepancy. In *Advances in Neural Information Processing Systems*, pages 74122–74146. Curran Associates, Inc., 2024. 14

[73] Xiaosen Zheng, Tianyu Pang, Chao Du, Jing Jiang, and Min Lin. Intriguing properties of data attribution on diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024. 3

[74] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, 1997. 5

A. Broader Impact

Our research addresses the pressing issue of verifying the ownership of data used for training large image generation models, as highlighted by recent legal disputes [5]. By introducing CDI, we aim to enable data owners to verify if their data was (illegitimately) used for training DMs. Our work contributes to a more transparent and accountable ML ecosystem, aligning with broader societal values of fairness and respect for data ownership. We anticipate CDI will have a positive impact on both the ML community and society, promoting responsible and fair development of ML models. We make our code available at https://github.com/sprintml/copyrighted_data_identification.

B. Limitations

In this paper, we assess the effectiveness of CDI for image generation DMs. Although we believe that our methodology extends well to other modalities such as text or video, we do not perform an experimental evaluation in these areas. Our research focuses on diffusion models as the current state-of-the-art image generators widely employed in commercial applications. While acknowledging the existence of alternative generative image models, which have recently been shown to demonstrate comparable capabilities [25, 52], we choose to focus on DMs within the scope of our research. We note that CDI requires at least gray-box access, i.e. DM's predictions at an arbitrary timestep t to be effective. This limitation stems from the lack of reliable, strictly black-box MIA methods for DMs, *i.e.,* methods that leverage only the final generated image, while avoiding the pitfalls described in App. D.

C. Additional Background

C.1. Previous Membership Inference Attacks against DMs

We present the previous MIAs against DM that provide the initial set of features we use for our CDI.

Denoising Loss by Carlini et al. [11]. The common intuition for utilizing the loss function of the model is that its values should be lower for the training set (members) than validation or test set (non-members). Formally, we follow LiRA [10], and its extension to DMs [9], and for each sample, we compute $\|\epsilon - f_\theta(z_t, t)\|_2^2$ at $t = 100$. Note, that z_t is obtained by adding $\epsilon \sim \mathcal{N}(0, I)$ to the original z, a process which, by its stochasticity, introduces noise to obtained scores. Carlini et al. [9] suggest to address this issue by computing the loss for five z_t noised using different ϵ. The final feature is the mean of these five measurements.

SecMI$_{stat}$ by Duan et al. [16]. The feature is based on the assumption that the effect of the denoising process should restore member samples better than non-member samples. Duan et al. [16] formalizes this idea by introducing *t-error*, a metric that aims to approximate the estimation error of f_θ on a given z. More specifically, *t-error* is defined as $\|\psi_\theta(\phi_\theta(\tilde{z}_t, t), t) - \Phi_\theta(z_0, t)\|_2^2$, where $\psi_\theta(z_t, t) = z_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{f}_\theta(z_t, t) + \sqrt{1 - \bar{\alpha}_{t-1}}f_\theta(z_t)$ is the DDIM [57] denoising step, $\phi_\theta(z_t, t) = z_{t+1} = \sqrt{\bar{\alpha}_{t+1}}\hat{f}_\theta(z_t, t) + \sqrt{1 - \bar{\alpha}_{t+1}}f_\theta(z_t, t)$, is the DDIM sampling inverse step, $\hat{f}_\theta(z_t, t) = \frac{z_t - \sqrt{1 - \bar{\alpha}_t}f_\theta(z_t, t)}{\sqrt{\bar{\alpha}_t}}$, $\bar{\alpha}_t = \prod_{k=0}^{t} \alpha_k$, and $\Phi_\theta(z_s, t) = \phi_\theta(\cdots \phi_\theta(\phi_\theta(z_s, s), s + 1), \cdots, t - 1)$ is the deterministic reverse. *t-error*, intuitively, should hold lower values for member samples.

PIA by Kong et al. [27]. PIA builds on the notion that, under DDIM sampling settings, given z and any z_t, one can determine the *ground truth trajectory* consisting of intermediate z_s, $s \in (0, t)$. Subsequently, f_θ learns to reflect this trajectory and is more competent in it for members. Capturing that difference can act as a membership signal, defined as $\|f_\theta(z, 0) - f_\theta(\sqrt{\bar{\alpha}_t}z + \sqrt{1 - \bar{\alpha}_t}f_\theta(z, 0), t)\|_5$, where $\bar{\alpha}_t = \prod_{k=0}^{t} \alpha_k$, and $t = 200$. The feature should be lower for members.

PIAN by Kong et al. [27]. An important consideration in PIA is that $f_\theta(z, 0)$ follows a Gaussian, *i.e.,* $f_\theta(z, 0) \sim \mathcal{N}(0, I)$, as it should make the attack more performant. To assure that, Kong et al. [27] propose PIAN (normalized PIA), as an extension of their method. Formally, $\hat{f}_\theta(z, 0) = C \cdot H \cdot W \sqrt{\frac{\pi}{2}} \frac{f_\theta(z, 0)}{\|f_\theta(z, 0)\|_1}$ with C, H, W denoting the channels, height, and width of the input sample, respectively. However, our experiments in App. U and Sec. 5.1 indicate that this intuition does not translate to latent DMs, which is in line with findings from the original paper [27].

D. On the Mismatch in MIAs Results

Contrary to the promising results of SecMI$_{stat}$, PIA, and PIAN, our evaluation in App. U shows that these attacks fail to reach performance significantly higher than random guessing. The following is a methodological analysis of issues in the experimental setup proposed in [16, 20, 27]. We identify three pitfalls in their experimental settings: (1) usage of toy models, (2) overfitting to the evaluation set, and (3) distribution mismatch between members and non-members sets.

D.1. Pitfall 1: Toy Models

MIAs performance is directly correlated with the level of overfitting in the attacked model [51, 55]. Unfortunately, it is a common approach to evaluate the MIAs against DMs
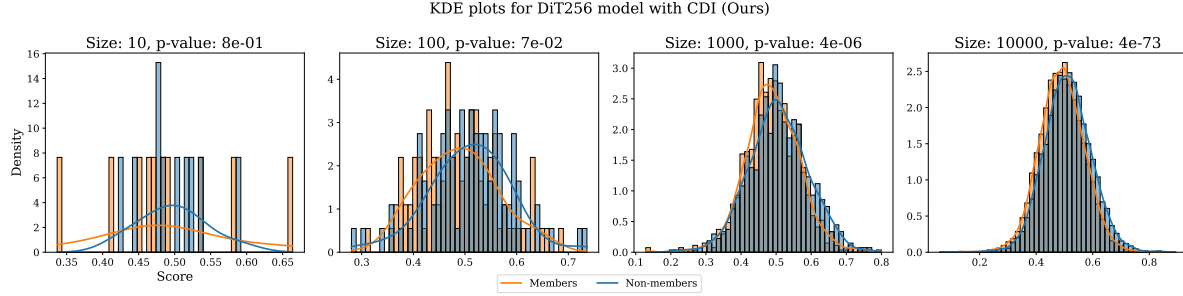
Figure 5. **KDE plots for varying $|\mathbf{P}_{\text{test}}|$.** We use the DIT256 model.

on very small toy models, trained or fine-tuned on small-scale datasets like CIFAR10 [30], as it speeds up the inference. This in turn can lead to elevated performance which is further reported in published works [16, 20, 27], *e.g.,* the **TPR@FPR=1%** at the level of 10% for SecMI$_{\text{stat}}$ and 30% for PIAN. We argue that evaluating any type of MIA in such setting is flawed, and provides incorrect insight on the performance of the proposed MIA.

In contrast to previous works, we focus only on evaluating success of `CDI` on non-overfitted large DMs from official repositories, trained on high-scale datasets like ImageNet.

D.2. Pitfall 2: Overfitting to the Evaluation Set

SecMI$_{\text{NNs}}$ [16] in its official implementation uses the evaluation set for selection of the best-performing classifier[2].

D.3. Pitfall 3: Mismatch in Distribution of Member and Non-Member Sets

Dubiński et al. [17] highlight the importance of members and non-members sets being indistinguishable from each other in order for the results of MIAs being reliable. Indeed, the mismatch between these can be enough for any classification-based method to succeed, without the context of attacked model. More importantly, a simple Loss attack [70] also benefits from this pitfall, as out-of-distribution (incorrect non-members) samples usually achieve significantly higher values of model loss objective than in-distribution samples (members, or correct non-members, *e.g.,* test set). Unfortunately, Duan et al. [16] for evaluation of their SecMI, as well as Kong et al. [27] for PIA on SOTA Stable Diffusion [47] utilize an external dataset, namely COCO [64], as their non-members set. In effect, it casts doubt on the reported success of their methods.

In our work we avoid this problem by using validation sets of the DMs as the source of non-members samples.

E. On Reporting p-values

To ensure the reliability of the statistical test results reported for `CDI`, we adopt the following approach. We repeat the

---

[2]https://github.com/jinhaoduan/SecMI/blob/main/mia_evals/secmia.py#L358

t-tests 1000 times on features obtained from randomly sampled subsets of $\mathbf{P}$ and $\mathbf{U}$, aggregating the resulting p-values. We consider various p-value aggregation methods as reviewed by [28, 65], and make our final decision based on the specific context in which we apply `CDI`.

In our framework, we assume that `CDI` is executed by an arbitrator approached by a victim whose private data might have been used in the training of a DM. Each execution of the t-test represents data verification for a *single* data owner, with each p-value corresponding to the test outcome for an *individual case*. To appropriately represent the performance of our method in this setting, we report p-values aggregated using an arithmetic mean over the success for all data owners.

This aggregation method is a conservative approach for reporting our results (comparing to *e.g.,* harmonic mean). This is due to the vulnerability of the arithmetic mean to outliers as the p-values are strictly positive, and the mean can be saturated easily by a single large value. In contrast, harmonic mean can be too easily brought down to almost zero by a single good result which would overstate the success of our method.

F. Impact of the size of the $\mathbf{P}_{\text{test}}$ on `CDI` confidence

We present a visualization of the impact of the size of the $\mathbf{P}_{\text{test}}$ on the confidence of `CDI` in Fig. 5. We sample $n = 10, 100, 1000, 10000$ samples randomly from $\mathbf{P}_{\text{test}}$ and $\mathbf{U}_{\text{test}}$, and apply `CDI`. We observe that when we increase the size of $\mathbf{P}_{\text{test}}$ the method becomes more stable and more confident. This is an inherent feature of statistical testing, a core element of `CDI`.

G. Additional Features

As noted previously, Latent Diffusion Models represent the state-of-the-art in high-resolution image generation for DMs. Such models are two-stage architectures, where the diffusion process occurs in the latent space of an autoencoder. This observation introduces another potential angle for MIA on latent DMs. We note that `CDI` can be extended by incorporating signal from features specifically tailored for the

autoencoder part of the model. However, it is important to recognize that the diffusion backbone and the autoencoder are separate models, which can be trained on different datasets. This necessitates caution when deciding whether to incorporate features extracted from the encoder. Nonetheless, in cases where both the autoencoder and the DM are trained on the same dataset, we claim that the performance of `CDI` can be further boosted by incorporating membership signals from the autoencoder. To this end, we propose an additional feature that can be utilized in such scenarios.

Autoencoder Reconstruction Loss (ARL). The goal of this feature is to extract differences in the autoencoder reconstruction errors between members and non-members, where members should exhibit significantly lower errors. To obtain the features, we first note that the current state-of-the-art DMs consist of the pixel and latent spaces [47]. For our ARL feature, we leverage the two-stage structure of DMs and extract the membership signal directly in the pixel space, which contains an autoencoder with the encoder part $\mathcal{E}$ and decoder $\mathcal{D}$. Given an input image x, the encoder $\mathcal{E}$ encodes x into a latent representation $z = \mathcal{E}(x)$. The decoder $\mathcal{D}$ reconstruct the image from the latent z, yielding $\tilde{x} = \mathcal{D}(z) = \mathcal{D}(\mathcal{E}(x))$. The autoencoder reconstruction loss serves as the ARL feature computed as $||x - \mathcal{D}(\mathcal{E}(x))||_2^2$.

However, for all the models on which we perform our experiments, the autoencoder was trained on different dataset than the diffusion backbone. The LDM model utilizes VQ-VAE[43] trained by Rombach et al. [47] on the Open Images Dataset V4 [31] dataset. All other models use KL autoencoder [26, 47] provided by Stablility AI [3]. This model was first trained on the Open Images Dataset V4 and then finetuned on subsets of LAION-Aesthetics [53] and LAION-Humans [53] datasets. To keep compatibility with existing models trained by Stability AI, only the decoder part was finetuned. Accounting for the difference between the underlying autoencoder and diffuser training datasets present in all models on which we evaluate `CDI` we do not employ the ARL feature in our framework in this paper. However, we note that it constitutes another source of membership signal applicable in cases when both stages of the latent DM were trained on the same dataset.

H. Extending `CDI`

`CDI` provides an inherently flexible framework for identifying data collections used in the training of DMs. In this section, we discuss how `CDI` can be extended with additional feature extraction methods and applied to a broader range of DM architectures and data modalities.

H.1. Extending `CDI` with features from novel MIAs

As the field advances, `CDI` can be extended with additional feature extraction methods based on novel MIAs. We demonstrate this by incorporating CLiD [72] as a feature extrac-

tion method within `CDI`. First, in Table 4, we show that the performance of CLiD MIA remains limited for models trained on ImageNet (first five columns). While it significantly outperforms previous MIAs, as compared to Tab. 12, it still does not allow reliable membership identification for ImageNet-trained models. Note that we do not evaluate CLiD on U-ViT256-Uncond, as it requires both conditional and unconditional inputs to the DM.

Table 4. **CLiD [72] MIA results at a TPR@FPR=1%**. Values in the table are in %.

Model	LDM	U-ViT256	DiT-256	U-ViT512	DiT512	U-ViT256-T2I	U-ViT256-T2I-Deep
TPR@FPR=1%	1.52	2.18	2.74	2.03	2.00	10.91	24.11

Next, we employ features from CLiD in `CDI` and present in Table 5 that it further improves our method. This confirms our premise from Section 4.2, that more meaningful features improve our method. It also demonstrates how `CDI` benefits from advancements in MIAs. When we extend `CDI` with CLiD, the method needs as few as 30 samples to identify data collections used in DM training.

Table 5. **Effect of extending `CDI` with features extracted using CLiD.** We report the minimal sample size of **P** needed to confidently reject the null hypothesis with `CDI`.

Model	LDM	U-ViT256	DiT256	U-ViT512	DiT512	U-ViT256-T2I	U-ViT256-T2I-Deep
Base `CDI`	6000	3000	2000	2000	1000	200	70
With CLiD	4000	700	400	2000	700	50	30

H.2. Extending `CDI` to novel DMs

With new DM architectures introduced, `CDI` can be applied to a broader range of DMs. In Table 6, we evaluate the performance of `CDI` on SiT [37], MDT [21], and DiMR [34], and show that it successfully identifies the data collections used in DM training.

Table 6. **Performance of `CDI` on additional DMs.** We report the minimal sample size of **P** needed to confidently reject the null hypothesis with `CDI`.

Model	SiT-XL/2	MDTv2-XL/2	MDTv1-XL/2	DiMR-XL/2R	DiMR-G/2R
|**P**|	300	300	200	2000	2000

H.3. Extending `CDI` to other data modalities

`CDI` framework is inherently flexible and can be applied to data modalities beyond images, such as text or audio. This extension is straightforward due to the use of feature extraction methods that operate in the latent space of DMs.

Table 7. **Model details.** We report the training details for the models used in this paper in the context of the minimal sample size of **P** needed to confidently reject the null hypothesis with `CDI`.

	LDM256	U-ViT256	DiT256	U-ViT512	DiT512	U-ViT256-Uncond	U-ViT256-T2I	U-ViT256-T2I-Deep
Model parameters	395M	500M	675M	500M	676M	44M	45M	58M
Training steps	178k	500k	400k	500k	400k	1M	1M	1M
Batch size	1200	1024	256	1024	256	256	256	256
Dataset	ImageNet	ImageNet	ImageNet	ImageNet	ImageNet	COCO	COCO	COCO
Dataset size	1.2M	1.2M	1.2M	1.2M	1.2M	83k	83k	83k
Min. P size	6000	3000	2000	2000	1000	300	200	70

Table 8. **Feature extraction time for different features.** Time in seconds is given for processing 1 batch of 64 samples on an A100 GPU.

	LDM256	U-ViT256	DiT256	U-ViT512	DiT512	U-ViT256-Uncond	U-ViT256-T2I	U-ViT256-T2I-Deep
Denoising Loss	4.06	10.55	12.87	9.64	40.77	2.01	2.28	2.69
SecMI$_{stat}$	8.41	23.22	22.16	26.35	96.02	3.27	4.09	4.90
PIA	5.21	6.45	6.87	8.77	26.04	1.72	2.24	2.56
PIAN	5.58	6.93	7.22	9.21	28.04	1.94	2.42	2.78
Gradient Masking (GM)	31.90	81.67	72.42	90.33	110.57	9.14	11.56	15.49
Multiple Loss (ML)	7.11	20.28	18.42	22.55	70.28	2.92	3.45	4.26
Noise Optim (NO)	94.64	64.12	64.17	78.14	181.33	182.23	205.78	120.26

I. Experimental Setup

I.1. Models

We evaluate the effectiveness of `CDI` on publicly available state-of-the-art DMs. This section provides an overview of the models used in our experiments. For more detailed information about these models and their training procedure, readers are encouraged to consult the original papers. Additionally, we make the model checkpoints readily accessible for download to facilitate the replication of our results.

All models, with the exception of LDM256 utilize ViT[15] as the diffusion backbone. LDM256 uses the UNet architecture [48] instead, being prior work in the area of latent DMs.

LDM256 - a class-conditioned LDM checkpoint provided by Rombach et al. [47], trained on ImageNet dataset in 256x256 resolution. The diffuser backbone of this model is a UNet architecture with 395M parameters.

DiT256, *DiT512* - class-conditioned DiT-XL/2 checkpoints provided by Peebles and Xie [39], trained on ImageNet dataset in 256x256 and 512x512 resolutions respectively. DiT256 has 675M parameters and DiT512 has 676M parameters.

U-ViT256, *U-ViT512* - class-conditioned U-ViT-Huge/4 checkpoints provided by Bao et al. [6], trained on ImageNet dataset in 256x256 and 512x512 resolutions respectively. U-ViT256 has 500.8M parameters and U-ViT512 has 500.9M parameters.

U-ViT256-T2I - a text-conditioned U-ViT-Small/4 checkpoint provided by Bao et al. [6], trained on COCO dataset in 256x256 resolution.

U-ViT256-T2I-Deep - a text-conditioned U-ViT-Small/4-Deep checkpoint provided by Bao et al. [6], trained on COCO dataset in 256x256 resolution. The model architec-ture differs from U-ViT256-T2I by having a larger number of transformer blocks (16 instead of 12). U-ViT256-T2I and U-ViT256-T2I-Deep have 45M and 58M parameters respectively.

U-ViT256-Uncond - an unconditioned U-ViT-Small/4 checkpoint trained on COCO dataset in 256x256 resolution. We train the model for 1.000.000 training steps, following the configuration used by Bao et al. [6] for *U-ViT256-T2I* checkpoint. Namely, we use *AdamW*[35] optimizer (lr 2×10^{-5}, weight decay 0.03, betas (0.99,0.99)) and batch size 256. U-ViT256-Uncond has 44M parameters.

I.2. Compute Resources and Feature Extraction Time

We compute our experiments on A100 80GB NVIDIA GPUs on an internal cluster, amounting to 150 GPU-hours. Training the U-ViT256-Uncond model requires additional 80 GPU-hours. We provide the time needed to extract features from a batch of 64 samples in Tab. 8. Fitting the scoring model s of `CDI` does not require GPU utilization and can be executed in a negligible time (<10 seconds) on a CPU.

J. Model Details and `CDI` Effectiveness

Based on Table 7 we make the following observations. (1) For a given DM architecture, trained on a given dataset, the number of samples required for the confident identification of the training data decreases with increasing input resolution. This phenomenon is clearly visible when comparing the required number of samples for U-ViT256 and U-ViT512 and for DiT256 and DiT512 which differ only in the input image resolution (2) Models trained on smaller datasets exhibit stronger signal for identifying training data, as evident when contrasting the results on models trained with
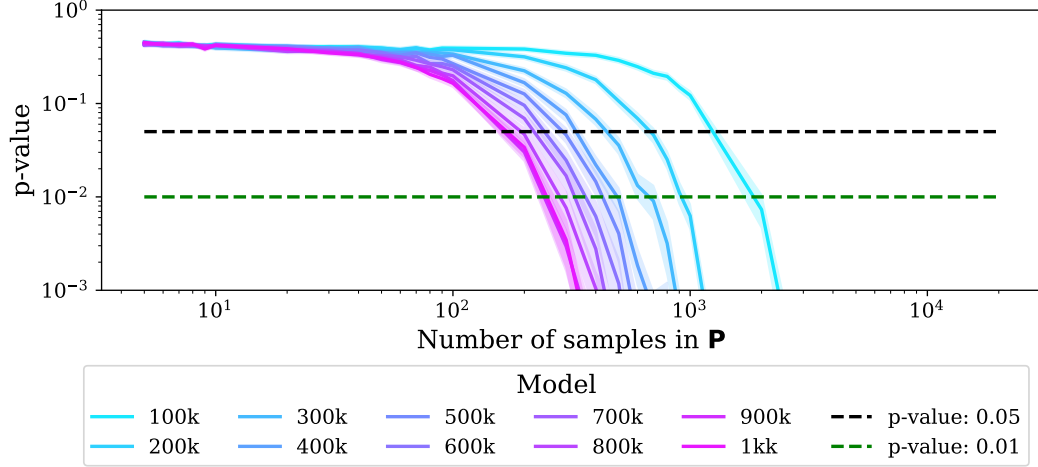
Figure 6. **Results of `CDI` for U-ViT256-Uncond for different number of model's training steps** Solid lines indicate aggregated p-values aggregated over 1000 randomized trials for each size of **P**, shaded areas around the lines are 95% confidence intervals. The higher the number of the model's training steps the fewer suspect samples are required from the data owner to confidently reject H_0.
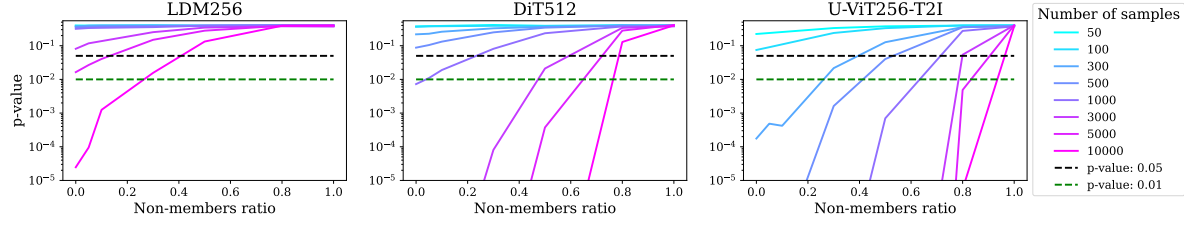

Figure 7. **Impact of non-members ratio in P on `CDI`, and resilience against false positives**. The lines represent p-values for a given non-member ratio while varying sizes of **P**. Note that for the non-members ratio of 1 (all samples in **P** are non-members), the p-values are always significantly above the significance level ($\alpha = 0.01$), which means `CDI` does not return false positive answers.

ImageNet (LDM, U-ViT256, U-ViT512, DiT256, DiT512) and COCO dataset (U-ViT256-T2I, U-ViT256-T2I-Deep, U-ViT256-Uncond).

K. Number of Model Training Steps and `CDI` Effectiveness

To assess the effect of the number of DM's training steps on `CDI`, we conduct the following experiment. We measure the effectiveness of `CDI` for U-ViT256-Uncond model after every 100,000 training steps. The results in Fig. 6 confirm that `CDI` is effective even for models trained for a limited number of steps. With enough samples provided by the data owner $|\mathbf{P}| = 2,000$, `CDI` confidently rejects H_0 for U-ViT256-Uncond after just 100,000 training steps.

L. More Results for the Non-members in P and False Positives

We present the additional results on `CDI`'s robustness in cases when **P** contains (some) non-member samples in Fig. 7. We expand Fig. 4 for DiT512 with experimental results for

LDM256 and U-ViT256-T2I. `CDI` remains effective even when not all data samples are used as training data. *i.e.,*, **P** contains a certain ratio of non-members, while the remaining samples in **P** are members. We observe that `CDI` demonstrates greater robustness in cases when part of **P** contains non-members if the data owner supplies larger **P**. This is evident in the decreasing p-value at a given non-members ratio as the size of **P** increases. Importantly, for the non-members ratio of 1, the p-values are significantly above the significance level ($\alpha = 0.01$), which means CDI does not return false positive answers.

M. Results under |P| and |U| imbalance

In practice, the training-to-test set ratio ($|U|/|P|$) is often much smaller than one. We analyze CDI's performance in such scenarios and find it remains effective even when the victim has access to fewer test samples ($|U| < |P|$). We report the number of samples in P needed to reject H_0 for given ratios ($|U| = $ ratio $\times |P|$). Notably, an extreme set imbalance can be mitigated by increasing P.

Table 9. **Performance of CDI under |U| and |P| imbalance.** We report the minimal sample size of **P** needed to confidently reject the null hypothesis with `CDI`.

|U|/|P| ratio	LDM256	U-ViT256	DiT256	U-ViT512	DiT512	U-ViT256-T2I	U-ViT256-T2I-Deep
1.00	6000	3000	2000	2000	1000	200	70
0.90	6000	3000	2000	2000	2000	200	70
0.75	6000	3000	2000	3000	2000	200	70
0.50	6000	3000	2000	3000	2000	300	90
0.25	7000	4000	2000	4000	2000	400	200
0.10	8000	5000	4000	6000	5000	800	300

Table 10. **Performance of `CDI` in gray- vs white-box model access.** We depict the number of samples required from the data owner in $\mathbf{Q}_{\text{test}}$ to reject the null hypothesis for different model access scenarios. Our `CDI` framework remains effective when assuming the more restrictive gray-box model access, provided with larger **P**.

Access Type	LDM256	U-ViT256	DiT256	U-ViT512	DiT512	U-ViT256-Uncond	U-ViT256-T2I	U-ViT256-T2I-Deep
Gray-Box Model	8000	3000	2000	4000	2000	400	200	70
White-Box Model	6000	3000	2000	2000	1000	300	200	70

N. MIAs and their Model Access Types

We group features, which we use in `CDI`, into two categories, based on their respective access type. Because Gradient Masking and Noise Optimization utilize gradient calculations they require white-box access to the DM's network weights. To execute the remaining feature extraction method `CDI` needs only gray-box access, *i.e.,* the ability to predict the noise added to a clean sample at an arbitrary timestep t. We specify the model access type per each feature in Tab. 11.

Table 11. **Each feature with its Model Access Type.**

MIA	Model Access Type
Denoising Loss [10]	gray-box
SecMI$_{stat}$ [16]	gray-box
PIA [27]	gray-box
PIAN [27]	gray-box
Multiple Loss (ML)	gray-box
Gradient Masking (GM)	white-box
Noise Optimization (NO)	white-box

O. Gray-box vs. White-box comparison

`CDI` remains effective even in a gray-box model access scenario. We assess the effectiveness of `CDI` within a gray-box model access scenario, as specified in our threat model (Sec. 4). In this setup, only the original MIA features and Multiple Loss are included, whereas the white-box model access setup leverages the full set of features. Our results in Tab. 10 show that `CDI` maintains effectiveness under gray-box access, successfully rejecting the null hypothesis, though it requires a larger sample size in **P**.

P. Analysis of the scoring function

Fig. 3 demonstrates the varying impact of features on CDI performance. We further aid our analysis of the scoring function by SHAPley [36] summary plots in Fig. 8. The results illustrate the model-specific nature of feature importance. This highlights the necessity of a scoring function that can agnostically learn the optimal feature utilization for each dataset and DM, leading to more confident and adaptable membership estimations, ultimately enhancing `CDI`'s effectiveness.

Q. From p-value to TPR@FPR=1%.

The ablation study on the importance of the statistical testing in `CDI` we conduct in Sec. 5.2 requires us to transform p-values returned by `CDI` to **TPR@FPR=1%** to directly compare `CDI` with set-level MIA in Tab. 2. We note that the TPR of a statistical test is equivalent to its power. We estimate the power by computing the effect size, using Cohen's d [13], which is defined as $d = \frac{\bar{x_1} - \bar{x_2}}{s}$. x_1 and x_2 are the observed scores, and $s = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}}$. n_1 and n_2 are the sizes of the population (here: the number of samples in **P** and **U**, respectively), and $s_j^2 = \frac{1}{n_j - 1} \sum_{i=1}^{n_j} (x_{1,j} - \bar{x_j})^2$ for $j = 1, 2$. After we obtain the effect size, we compute the power of the t-test using a solver, setting $\alpha = 0.01$ to get **TPR@FPR=1%**.

R. ROC curves of `CDI`

In Sec. 5.1, we evaluate `CDI` following the evaluation methodology proposed for DI by [38]. In this section, we additionally extend this evaluation by analyzing `CDI` through ROC curves and true positive rates (TPR).

We build on Sec. 5.2 and Appendix Q, and obtain TPR at varying FPR by sweeping over α values from 0 to 1. To ensure stability of our results, for each size of **P** we sample 1000 subsets from **P** and **U**, compute the TPR, and finally average the TPR to get the final value. We visualize our results in Fig. 9. Key takeaways from the figure are: (1) `CDI`
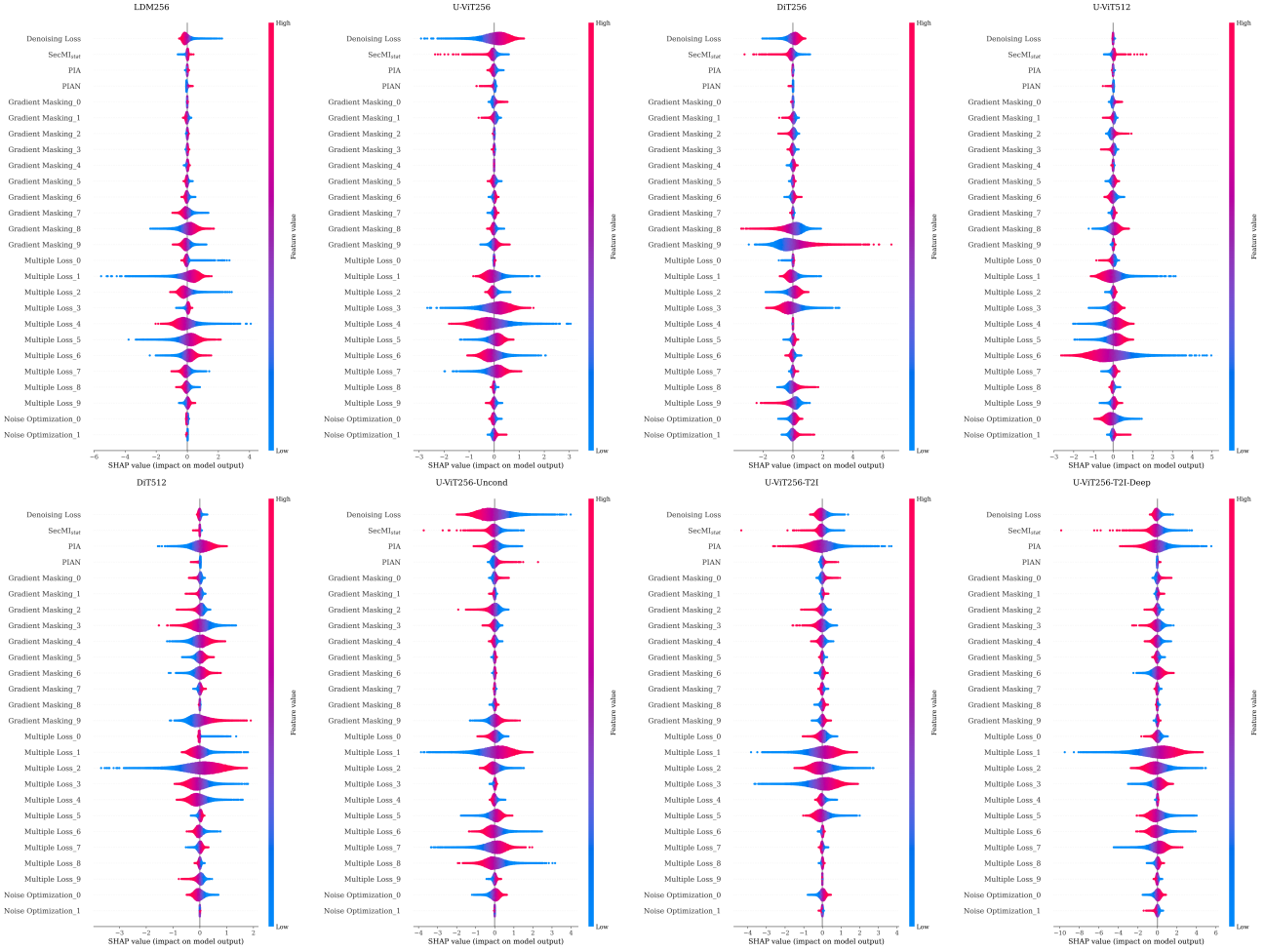
Figure 8. **SHAPley plots for all models and features.** The scoring functions have been trained using 5,000 members as $\mathbf{P}_{\text{ctrl}}$ and 5,000 non-members in $\mathbf{U}_{\text{ctrl}}$, The resulting plots are evaluated on 20,000 members in $\mathbf{P}_{\text{test}}$ and 20,000 non-members in $\mathbf{U}_{\text{test}}$ to provide the most accurate results.
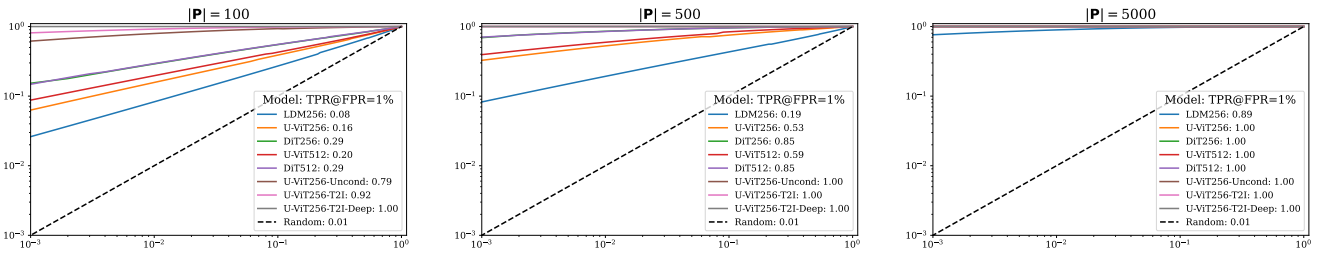

Figure 9. **ROC curves of `CDI`.** `CDI` achieves perfect performance with sufficiently large $|\mathbf{P}|$. Resulting **TPR@FPR=1%** align with p-values we report in Fig. 2.

achieves perfect performance given large enough $|\mathbf{P}|$, even at **FPR=0%**. (2) `CDI` is not over-confident. P-values we report in Fig. 2 align with the resulting **TPR@FPR=1%**, thanks to the careful choice of α (ref. Sec. 4.3). Intuitively, low p-values correspond to high **TPR@FPR=1%**. For ex-

ample, LDM256 for $|\mathbf{P}| = 5000$ achieves p-values close to 0.01, and **TPR@FPR=1%** of 0.89. (3) In effect, `CDI` is not susceptible to p-value hacking, *i.e.,* the improvement in `CDI`'s performance observed in `CDI`'s higher confidence (lower p-value) signifies higher TPR at lower FPR.

S. MIAs Fail on DI Task

In this experiment, we compare `CDI` to MIAs on the task of DI. Similarly to set-level MIA we introduce in Sec. 5.2, we follow this procedure: We apply a MIA to each sample in the suspect set, returning Positive prediction if the score for any sample exceeds a certain threshold. Repeating this process for every suspect set and varying the threshold yields a ROC curve for each MIA. Finally, we compare these ROC curves against the ROC curve for `CDI`, obtained as in App. R.

We sample 1000 **P** containing only member samples (Positive) and 1000 **P** containing only non-member samples (Negative). **U** remains unchanged, *i.e.,* contains only non-members. We vary the size of P for a more thorough analysis and compute ROC curves.

We demonstrate in Fig. 16 that `CDI` achieves TPR orders of magnitude higher than MIAs on the DI task. Applying single-sample MIAs to DI is challenging. Our experiments show MIAs are unstable, with high FPR due to set-wise confidence swayed by a single high score. Notably, increasing the size of **P** does not improve the performance. `CDI`'s statistical testing is robust by comparing distributions of membership scores and capturing subtle differences. Then, the application of the t-test in `CDI` quantifies these differences, with the p-value serving as a reliable confidence measure.

T. Further Applicability of Our Novel Features

One of our contributions is the new features introduced in Sec. 4.1. As we use these features to fit our *scoring function*, we can also use them to perform the default *threshold* MIA. We introduce the detailed results of that experiment in App. U. Here, we analyze the characteristics of our features.

T.1. Gradient Masking

Recall, to obtain this feature we: (1) distort 20% of the input image's latent based on the absolute gradient values. (2) Use the DM to reconstruct the distorted part by performing a single denoising step. (3) Compute L2 reconstruction loss over the distorted region.

Visualization. We visualize the reconstruction effect in Fig. 15. We observe that: (1) the reconstructed members resemble semantics of the original images better than the reconstructed non-members do. (2) For members and non-members there is a notable decrease in the level of detail in the images after reconstruction, *e.g.,* for U-ViT256-T2I-Deep we observe that for the member sample the details of the painting and the table are gone. (3) The crude elements of the reconstructed images stay unchanged, *e.g.,* for ImageNet models we can see that the reconstructed image contains a dog (for the member sample), or a turtle in the grass (the non-member). (4) Distortion is not uniform between the

models. For the ViT-based models we observe distortions that are spread out throughout the image, while for the LDM (UNet-based model) the distortions are more local.

Features. In Fig. 10 we analyze the distributions of the features (reconstruction errors) throughout various timesteps, and then we compare them with the distribution obtained by aggregating these features with our *scoring function*. We note that the difference between members and non-members is negligible for singular features. However, after we aggregate them we observe a significant difference.

T.2. Noise Optim and Multiple Loss

For these two features our findings are in line with the findings for Gradient Masking. In Fig. 11 and Fig. 12 we visualize the distributions of singular features and the distributions after aggregation.

T.3. Effect of aggregation on MIA and `CDI` performance

We compare the metrics for MIA and `CDI` when we use a singular feature vs when we aggregate the features. In Fig. 13 we observe that while some singular features barely cross the threshold of random guessing (TPR=1%), the aggregate provides better results (although the effectiveness is limited). For `CDI`, in Fig. 14 we observe that aggregation allows us to lower the p-value by orders of magnitude lower than the best singular feature can.

U. Further Evaluation of MIAs

The following summarizes an extensive evaluation effort for MIAs utilizing existing, and, for completeness, our proposed novel features used as MIA. We follow identical setting as described in App. U, and extend the results by Area Under the Curve (AUC) score (Table 14), and accuracy (Table 13). To perform MIA on novel features we do the following: (1) Fit s on the features extracted from $|\mathbf{P}_{ctrl}| = 5000$ and $|\mathbf{U}_{ctrl}| = 5000$. (2) Obtain predictions on $\mathbf{P}_{test}$ and $\mathbf{U}_{test}$. Importantly, $\mathbf{P}_{test} \cap \mathbf{P}_{ctrl} = \varnothing$ and $\mathbf{U}_{test} \cap \mathbf{U}_{ctrl} = \varnothing$. (3) Use these scores to run MIA. We include accuracy and AUC to better understand the differences between the proposed features, as well as their impact on the `CDI`. In Fig. 17 we visualize the behavior of the Receiver Operating Characteristic (ROC) curve for all features under MIA setting.

We note that, similarly to Fig. 3, Gradient Masking provides stronger signal than Multiple Loss and Noise Optimization, resulting in higher values of **TPR@FPR=1%** (Table 12), AUC, and accuracy in almost all cases. GM underperforms compared to ML only for U-ViT256 and U-ViT512.

We observe higher performance of MIAs for models trained on smaller datasets, *i.e.,* U-ViT256-Uncond, U-ViT256-T2I, U-ViT256-T2I-Deep.

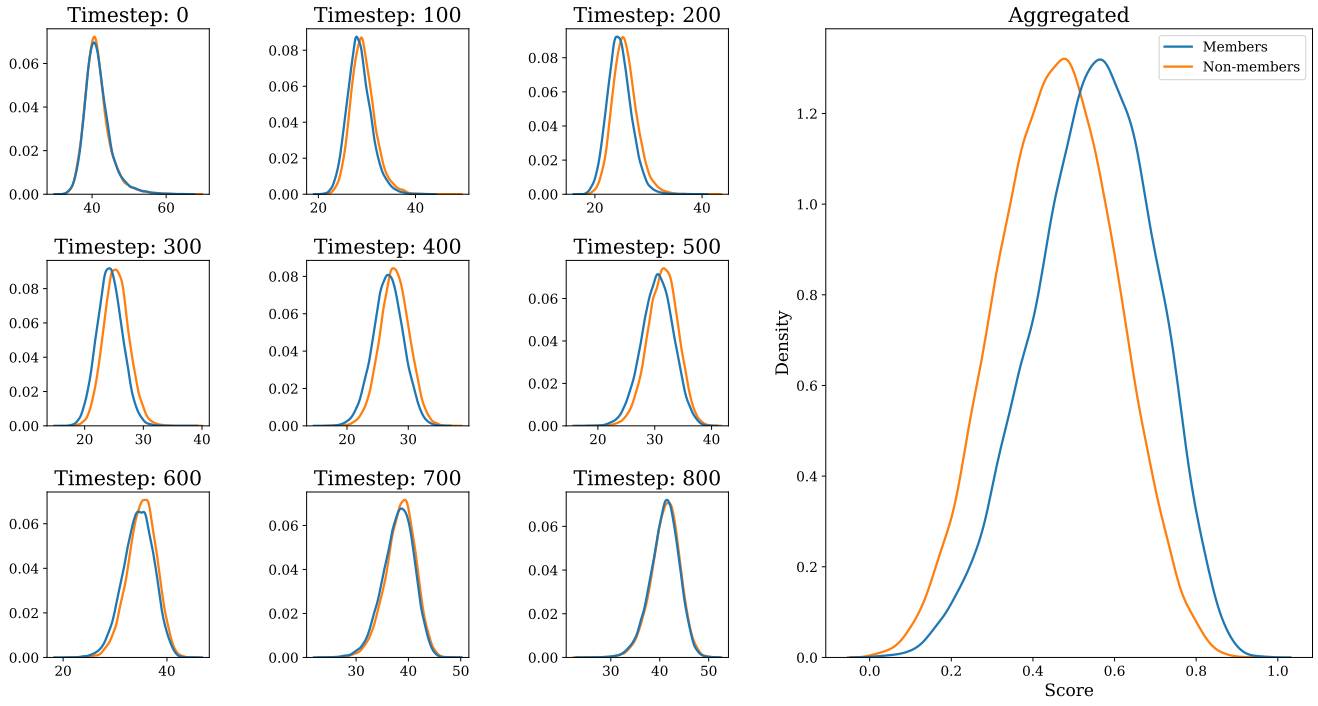KDE plots for U-ViT256-T2I-Deep model with Gradient Masking

Figure 10. **KDEplots of the Gradient Masking features for timesteps, and after aggregation.** The model used is U-ViT256-T2I-Deep.

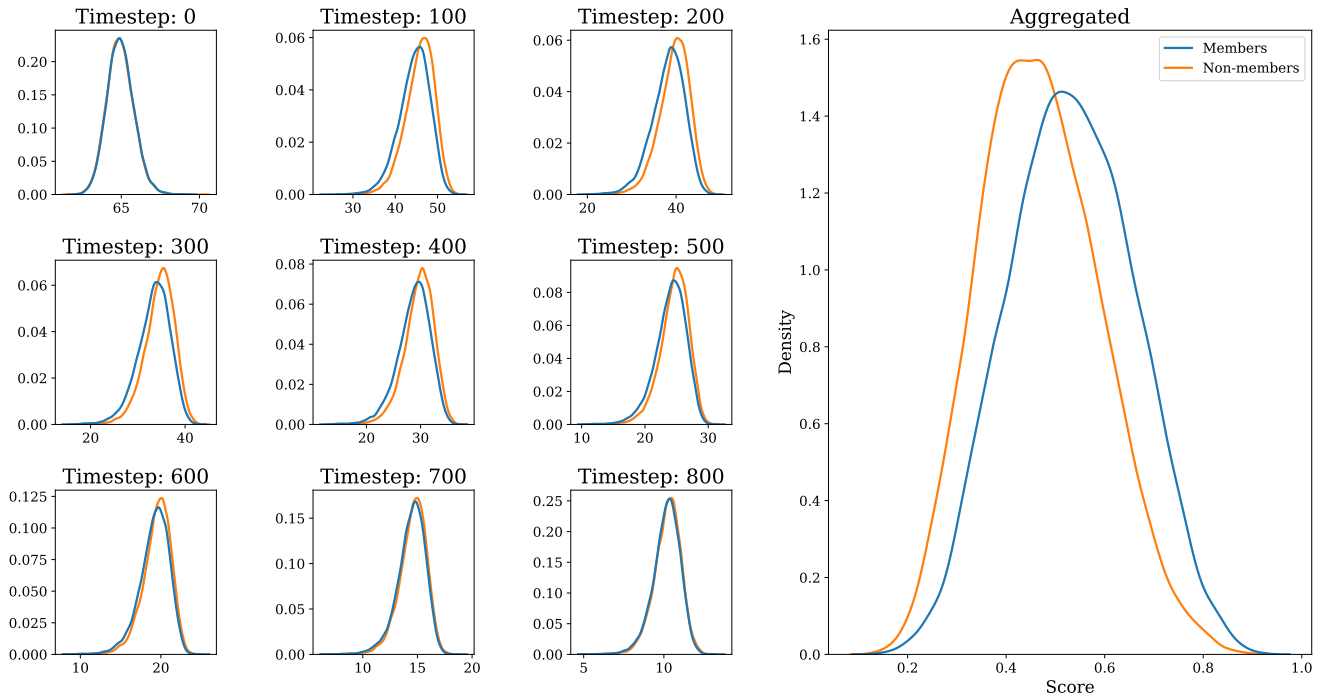KDE plots for U-ViT256-T2I-Deep model with Multiple Loss

Figure 11. **KDEplots of the Multiple Loss features for timesteps, and after aggregation.** The model used is U-ViT256-T2I-Deep.

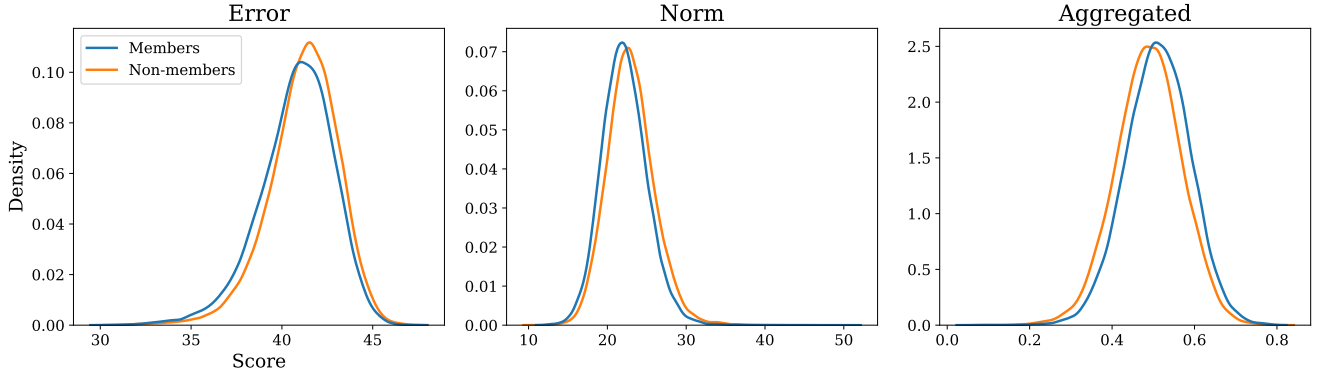KDE plots for U-ViT256-T2I-Deep model with Noise Optimization

Figure 12. **KDEplots of the Noise Optim features, and after aggregation.** The model used is U-ViT256-T2I-Deep. Error refers to the reconstruction error obtained after optimization, and Norm is the L2 norm of added perturbation.
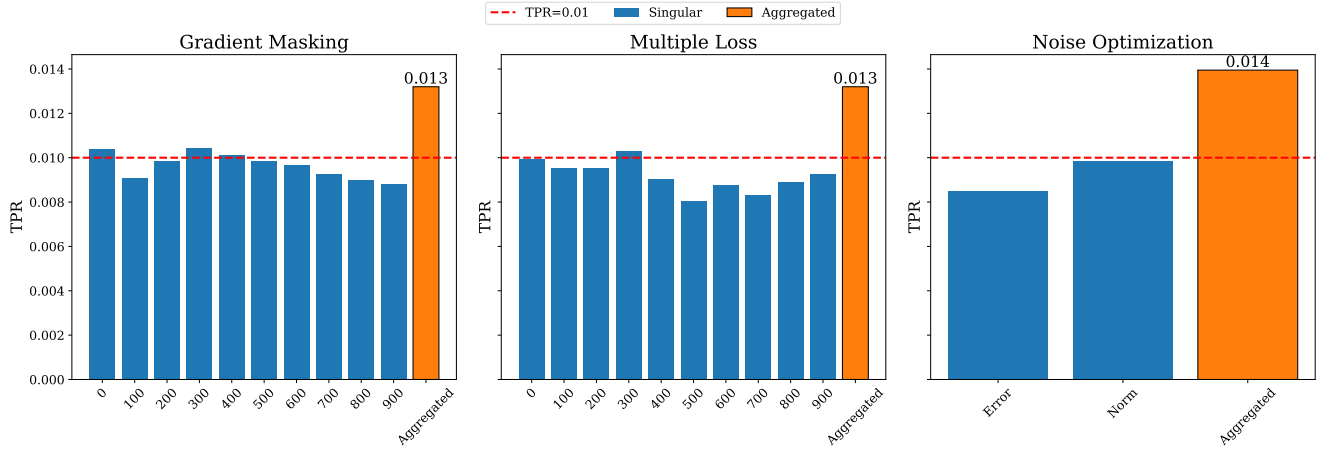

Figure 13. **TPR@FPR=1% (↑) for singular features, and after aggregation** The model used is LDM256. For Gradient Masking and Multiple Loss the numerical values correspond to the timestep at which the feature is computed.

Figure 14. **P-values (↓) for singular features, and after aggregation** The model used is LDM256. For Gradient Masking and Multiple Loss the numerical values correspond to the timestep at which the feature is computed.
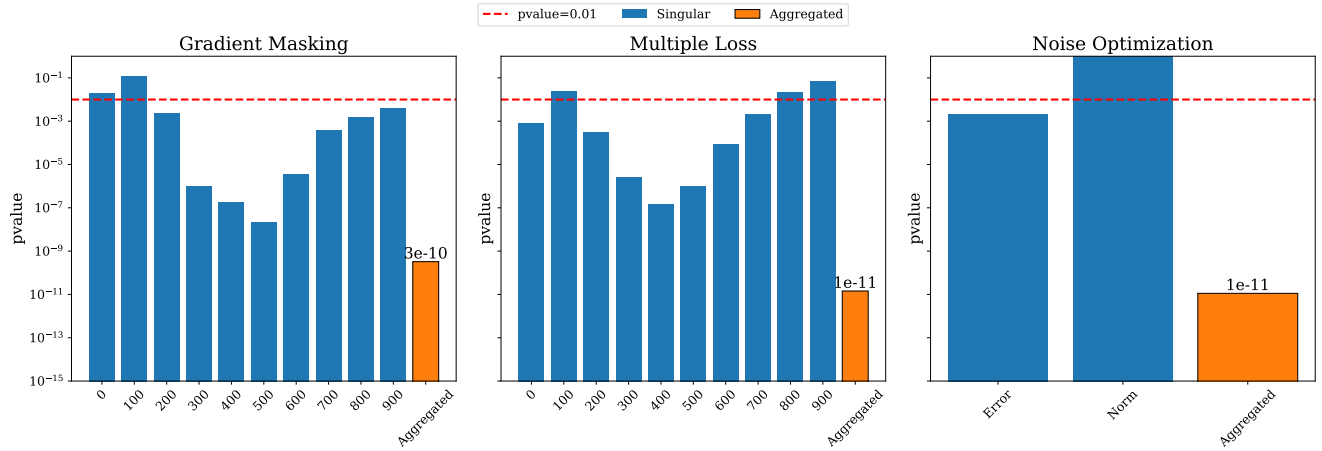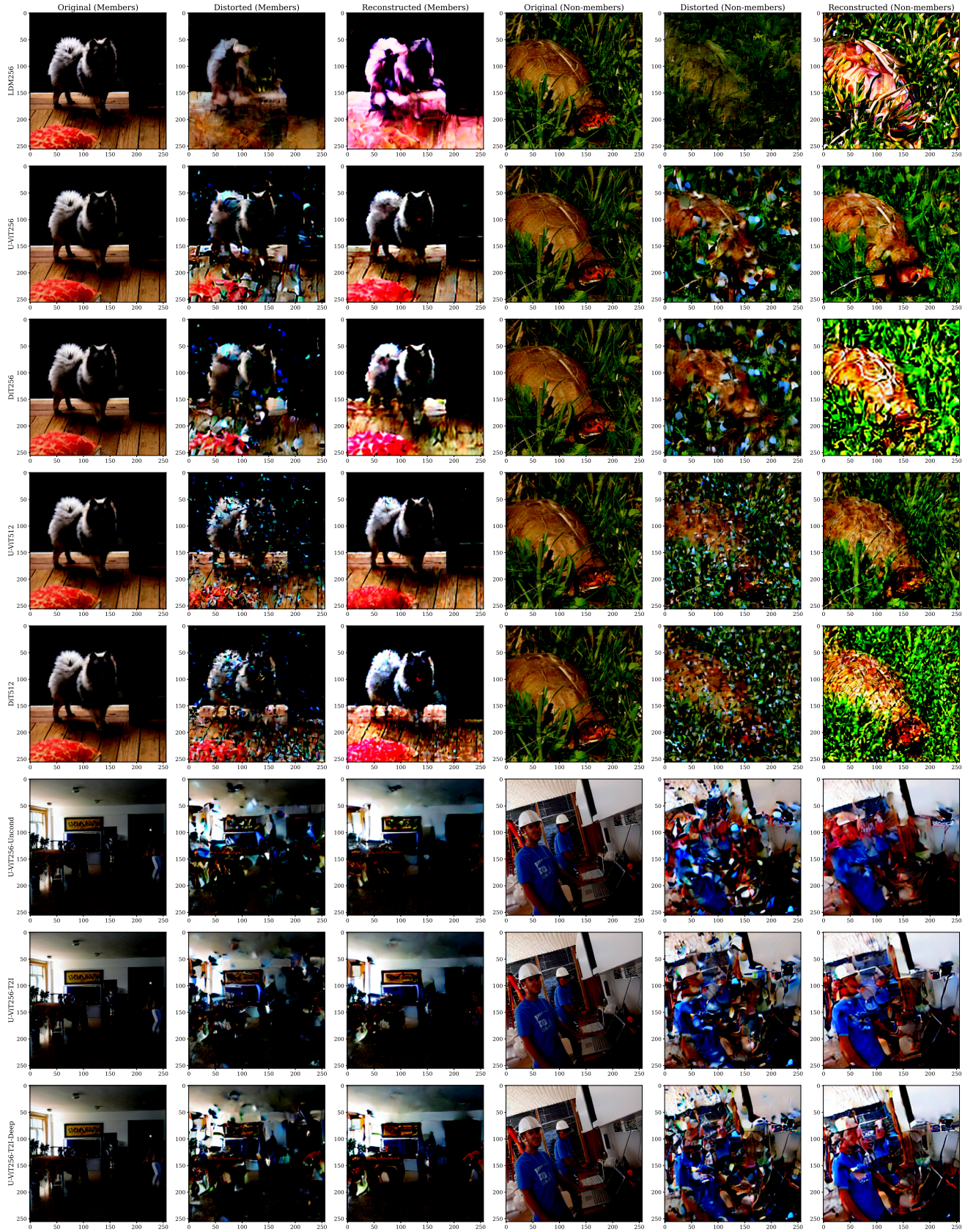
Figure 15. **Effect of the reconstruction method for the Gradient Masking.** The images in the figure are distorted using noise scale corresponding to the timestep=500, and are denoised using the same timestep. The images differ between models due to the difference in their respective training datasets (ImageNet and COCO2017).
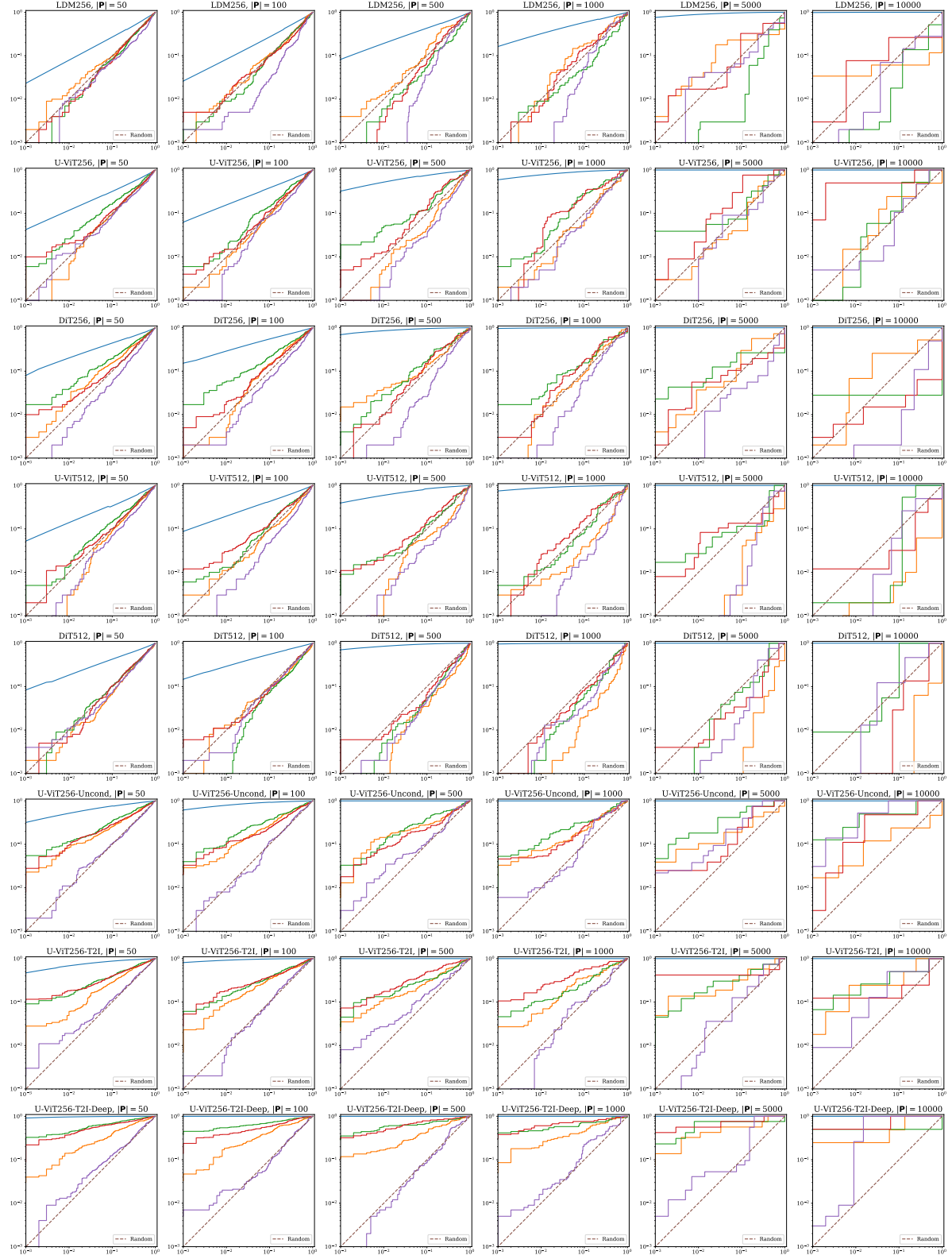
Figure 16. **Comparison of `CDI` and MIAs on the DI tasks.**

Table 12. **MIA results at a TPR@FPR=1%**. Values in the table are in %. We include the performance of MIAs using our novel features and note that they perform comparably with the SOTA PIA method, or even outperform it in some cases.

	LDM256	U-ViT256	DiT256	U-ViT512	DiT512	U-ViT256-Uncond	U-ViT256-T2I	U-ViT256-T2I-Deep
Denoising Loss [10]	1.23±0.10	1.22±0.10	1.34±0.12	1.61±0.12	1.78±0.12	1.72±0.15	1.73±0.14	2.25±0.20
SecMI$_{stat}$ [16]	1.17±0.11	1.17±0.11	1.31±0.14	1.26±0.10	1.16±0.13	1.67±0.19	1.78±0.14	2.41±0.24
PIA [27]	1.12±0.09	1.18±0.10	1.54±0.13	1.25±0.13	1.14±0.13	2.79±0.20	2.84±0.22	5.57±0.41
PIAN [27]	0.88±0.09	1.18±0.10	0.96±0.10	1.52±0.12	1.13±0.10	0.88±0.10	0.87±0.09	0.78±0.09
Gradient Masking	1.32±0.12	1.20±0.12	1.83±0.15	1.07±0.12	1.28±0.12	2.56±0.16	2.98±0.22	4.71±0.30
Multiple Loss	1.31±0.11	1.43±0.14	1.57±0.14	1.43±0.10	1.48±0.13	2.39±0.18	2.19±0.26	3.10±0.28
Noise Optimization	1.39±0.11	1.67±0.13	1.35±0.12	1.25±0.16	1.25±0.13	1.63±0.13	1.66±0.15	1.81±0.17

Table 13. **Accuracy of MIA on all features**. Values are in %. Here we observe that all novel features outperform already existing ones. Note that for all models trained on ImageNet (first five columns from the left) we observe results very close to 50%, essentially random guessing. For models trained on COCO (remaining three columns on the right) we observe an improvement for Gradient Masking and Multiple Loss, while the MIAs from Sec. 3.1 remain close to 50%.

	LDM256	U-ViT256	DiT256	U-ViT512	DiT512	U-ViT256-Uncond	U-ViT256-T2I	U-ViT256-T2I-Deep
Denoising Loss [10]	50.02±0.01	50.07±0.03	50.09±0.03	50.14±0.04	50.20±0.04	50.01±0.01	50.01±0.01	50.01±0.01
SecMI$_{stat}$ [16]	50.03±0.03	50.04±0.04	50.17±0.20	50.03±0.09	49.97±0.04	52.92±2.89	50.40±0.50	53.57±3.55
PIA [27]	50.01±0.01	50.02±0.03	50.28±0.06	50.02±0.03	50.03±0.03	50.07±0.02	50.03±0.01	50.03±0.01
PIAN [27]	49.51±0.12	49.87±0.06	49.81±0.07	49.82±0.17	49.80±0.13	50.02±0.15	50.10±0.15	50.05±0.12
Gradient Masking	50.00±0.01	50.01±0.01	51.61±0.46	50.00±0.01	50.78±0.13	53.60±0.60	55.26±0.68	61.85±0.30
Multiple Loss	50.07±0.07	50.00±0.01	50.08±0.06	50.00±0.00	50.06±0.06	52.75±0.19	53.70±0.29	59.79±0.26
Noise Optimization	50.00±0.00	50.00±0.00	50.00±0.00	50.09±0.04	50.00±0.00	50.03±0.02	50.24±0.06	52.72±0.20

Table 14. **AUC score for MIAs**. We observe that for this metric PIA outperforms all standalone features for models trained on COCO (last three columns on the right) while MIAs based on Gradient Masking, Multiple Loss and Noise Optim achieve better performance for models trained on ImageNet (first five columns on the left) in almost all cases.

|	LDM256	U-ViT256	DiT256	U-ViT512	DiT512	U-ViT256-Uncond	U-ViT256-T2I	U-ViT256-T2I-Deep
Denoising Loss [10]	50.55±0.28	50.29±0.30	51.71±0.29	49.99±0.29	50.19±0.29	56.47±0.28	57.38±0.28	60.77±0.29
SecMI$_{stat}$ [16]	49.59±0.28	53.06±0.29	55.22±0.28	50.92±0.30	50.69±0.30	59.28±0.29	61.56±0.28	69.20±0.26
PIA [27]	49.02±0.29	51.65±0.29	53.07±0.29	50.79±0.29	49.98±0.29	59.97±0.27	63.99±0.28	71.18±0.25
PIAN [27]	49.41±0.28	50.69±0.29	49.88±0.28	49.82±0.28	49.07±0.29	49.63±0.30	49.68±0.29	49.35±0.28
Gradient Masking	51.66±0.29	53.06±0.29	54.83±0.29	53.67±0.29	56.05±0.28	59.36±0.29	59.88±0.28	66.80±0.27
Multiple Loss	51.77±0.30	53.63±0.29	54.16±0.30	52.30±0.30	52.98±0.29	58.17±0.29	58.93±0.28	64.26±0.27
Noise Optimization	51.85±0.29	52.73±0.29	52.15±0.29	54.09±0.29	51.93±0.29	54.28±0.28	55.51±0.29	58.04±0.27
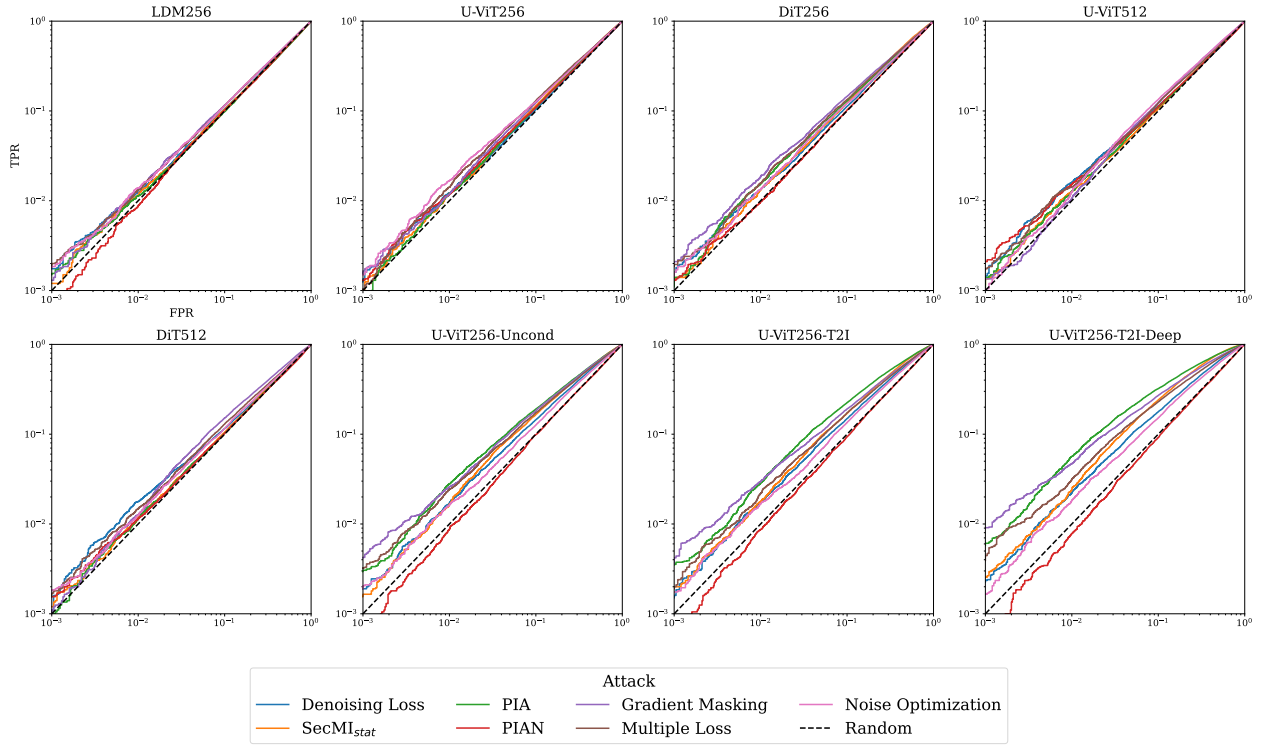
Figure 17. **ROC curves for MIAs against all models.** X- and Y-axis are in logarithmic scale. We observe that for DMs trained on ImageNet the TPR in low FPR regime ($< 1\%$) is not better than random guessing, while for the models trained on COCO (last three from the right in the bottom row) all methods but PIAN achieve results significantly better than random chance.