MGI: Member vs Generated Inference

Bihe Zhao, Michel Meintz, Juangui Xu, Franziska Boenisch, Adam Dziedzic
{bihe.zhao, michel.meintz, juangui.xu, boenisch, adam.dziedzic}@cispa.de
CISPA Helmholtz Center for Information Security

Abstract

As generative models increasingly produce samples that are indistinguishable from human-created content, it becomes difficult to determine whether a given data point was part of a model's natural training set or was generated by the model itself, especially when models memorize and reproduce training data. We formalize this challenge as Member vs Generated Inference (MGI): given a sample and a target generative model, infer whether the sample is a true training member or a generated output of that model. Focusing on image generation, we show that existing membership inference methods systematically misclassify generated samples as training members, while attribution-based methods often misclassify true members as generated. This failure arises because both approaches rely on likelihood-related signals that are similarly elevated for training examples and for the model's own outputs. To address MGI, we propose Data Circuit Breaker (DCB), a three-stage method that combines complementary signals from a generative model's autoencoder and latent generator to distinguish training members from generated samples. Across multiple generative models, including image autoregressive and diffusion models, DCB consistently addresses the shortcomings of membership inference and attribution methods, remains effective even when models reproduce near-duplicates of training samples, and generalizes to challenging model derivative settings in which new models are trained on generated data.

1. Introduction

Generative models are now trained on massive internet data and generate high-quality samples at an unprecedented speed. These models also inadvertently memorize some of their individual training inputs and later recreate them as outputs [3, 11]. The fact that the outputs from generative models are indistinguishable from real data blurs the **boundary between a model's training and generated data**. We formalize this challenge as the Member vs Generated inference (MGI) task: given an image and a target generative model, decide whether the sample is a true training member of that model or a generated output by the same model. We illustrate the MGI task in the overview Figure 1 for a *direct training* and a *model derivative* setting. In the **direct**

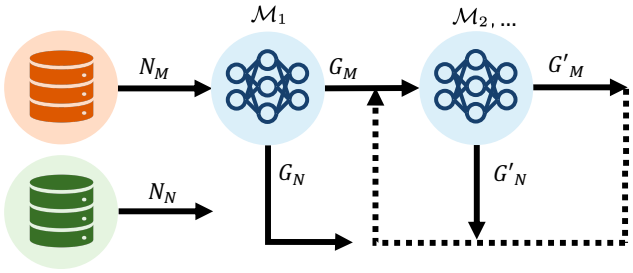


Figure 1. **Overview of the new Member vs Generated Inference (MGI) task.** The core challenge is separating genuine training membership from model generation, even across chains of models trained on generated data. Let $N = N_M \cup N_N$ denote a natural dataset, where $N_M \cap N_N = \varnothing$. A generative model $\mathcal{M}_1$ is trained on the member set N_M, while N_N is held out as natural non-member data. After training, $\mathcal{M}_1$ produces a generated dataset $G = G_M \cup G_N$, with $G_M \cap G_N = \varnothing$. Here, G_M and G_N are both generated by $\mathcal{M}_1$ and therefore follow the same generated-data distribution, but they play different roles in downstream settings: G_M is used to train a new model $\mathcal{M}_2$, whereas G_N is withheld and serves as generated non-member data for $\mathcal{M}_2$. The new model $\mathcal{M}_2$ is thus trained on generated members G_M rather than natural members N_M. The new model $\mathcal{M}_2$ in turn generates a new dataset $G' = G'_M \cup G'_N$, where $G'_M \cap G'_N = \varnothing$; samples in G'_M may be used to train further downstream models such as $\mathcal{M}_3$, while G'_N remains withheld. Under this setup, MGI asks whether a given sample should be attributed to training data or to model-generated data. For the original model $\mathcal{M}_1$, the task is to distinguish among N_M, N_N, G, separating true natural training members N_M from natural non-members N_N (as in the canonical membership inference task) and from model's $\mathcal{M}_1$ generated samples G. For the derivative model $\mathcal{M}_2$, the task becomes: distinguish among G_M, G_N, G', separating generated training members G_M from both generated non-members G_N and from model's $\mathcal{M}_2$ generated samples G'. We can further incorporate the natural samples N as $\mathcal{M}_2$'s non-member data, however, the G_N represents the most difficult case of the non-member data.

training setting with model $\mathcal{M}_1$, the goal is to distinguish natural training members N_M from images G generated by the model. Even in this seemingly simple setting, MGI is fundamentally harder than standard membership inference: generated images are optimized under the same latent distribution as training members, causing their likelihood-based scores to overlap heavily, as we demonstrate in Section 4. We further explore a more challenging and practically relevant **model derivative** setting, where the samples generated by $\mathcal{M}_1$ are (potentially published online, then scraped from the internet, and) used to train the subsequent model version $\mathcal{M}_2$. In this regime, members are no longer purely natural samples, and simply separating natural from generated content is insufficient. Both membership inference and attribution methods degrade further in the $\mathcal{M}_2$ setting, where generated training data introduces compounding ambiguity between membership and generation signals.

Focusing on image generation, we firt show that existing membership inference methods [11, 29, 34] are inadequate for MGI: they are designed to separate training members from held-out natural data, and consequently tend to incorrectly label model-generated (but non-member) samples as members. Conversely, attribution methods that aim to determine whether a sample was generated by a particular generative model [4] are also insufficient, often failing by labeling training members as generated. Both failures stem from the same underlying cause: the outputs for the new powerful generative models are derived directly from the training samples of generative models themselves. As a result, signals based on likelihood or output probabilities are similarly high for both true members and the models' own outputs, breaking the assumptions underlying prior methods.

To address the MGI challenge for modern image generative models, we propose a new method *Data Circuit Breaker* (DCB).[1] Our DCB method treats the generation pipeline holistically rather than focusing solely on the latent generator. The key insight is that while the latent generator produces high scores for members and generated samples, the autoencoder introduces measurable artifacts: generated samples, having passed through the full encode-decode pipeline, exhibit lower reconstruction and quantization errors than natural data points under the autoencoder. DCB exploits this by proceeding in three stages: (1) an autoencoder-based filtering step that identifies generated samples, separating them from non-generated data points; (2) a membership inference step on the non-generated samples using the latent generator, where the standard assumption that members score is restored; and (3) a cross-generator attribution step that compares conditional log-probabilities across multiple model versions to distinguish among the generated samples from different generators. Together, these stages enable DCB to solve MGI even in the most difficult cases of training data memorization.

Overall, our contributions are as follows:

1. **New task.** We introduce *Member-vs-Generated Inference* (MGI) task, which asks whether a given sample is a true training member of a generative model or an output example generated by that same model.

2. **Limits of prior work.** We demonstrate that existing approaches are insufficient for MGI: Membership inference methods systematically misclassify generated samples as members, while attribution methods often incorrectly label training members as generated.

3. **Method.** We propose DCB (Data Circuit Breaker), a three-stage procedure that exploits autoencoder self-consistency to filter generated samples, latent-generator scores for membership inference, and cross-generator probability discrepancies to trace data circuits across model versions.

4. **Memorization robustness.** We show that DCB remains effective even under verbatim memorization, distinguishing original training samples from their regurgitated (near-duplicate) generated counterparts.

2. Background and Related Work

Image Generative Models (IGMs). The dominant families of modern image generative models (IGMs) are *diffusion models* (DMs) and *image autoregressive models* (IARs). Many state-of-the-art IGMs in both families generate images in a *latent space*: an encoder first maps a high-resolution image from pixel space to a latent representation, and a decoder maps the synthesized latent back to pixels. While they share the latent-generation pipeline, DMs and IARs differ fundamentally in how they represent and sample from the data distribution. DMs define an *implicit* generative process via iterative denoising, whereas IARs *explicitly* factorize likelihood by predicting token probabilities sequentially, similarly to large language models (LLMs).

Diffusion Models (DMs). DMs synthesize images by transforming Gaussian noise into a structured sample through a learned denoising procedure [9, 21]. Generation starts from $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and proceeds for T steps, iteratively predicting noise $\epsilon_{\mathcal{G}}(\mathbf{x}_t, t, \mathbf{c})$ for $t = T, \ldots, 1$, and then removing it. In conditional settings (e.g., class-to-image or text-to-image), the denoiser is conditioned on auxiliary inputs $\mathbf{c}$, typically text embeddings produced by pretrained encoders such as CLIP [14]. Conditioning is injected through cross-attention layers [25].

Image Autoregressive Models (IARs). IARs generate images by predicting discrete latent tokens one-by-one using

[1] A circuit breaker is an electrical safety device designed to protect an electrical circuit from damage caused by current in excess of that which the equipment can handle. In our case, DCB can protect new models, for example, from degrading in performance by preventing their training on significant amounts of their own generated data.

next-token-based autoregressive model, directly modeling a factorized distribution over the latent sequence. A typical IAR consists of (1) a vector-quantized VAE (VQ-VAE) that encodes an image into discrete representations from a codebook, and (2) an autoregressive transformer that models the codebook representations as tokens and samples them sequentially. For example, LlamaGen [22] uses a VQ-based autoencoder to produce quantized features, then applies a Llama-style transformer to generate tokens autoregressively. VAR further introduces a multi-scale VQ representation to enable coarse-to-fine synthesis [23]. Randomized autoregressive models (RARs) generalize next-token prediction by training with randomized token orderings and an annealing-based procedure [31].

Membership Inference Attack (MIA). MIA aims to determine whether a given data point was part of a model's training set or not [18, 19]. MIA methods are used for auditing models' privacy leakage and verifying empirically the differential privacy guarantees [12, 17]. Recent work on MIA against IGMs [11, 29, 34] shows that comparing an image's conditional generation to its unconditional generation provides an effective signal for deciding whether the model was trained on that image (member) or not (non-member). Thus, the attack considers only the problem of differentiating between the train vs test samples and does not consider the data generated by the target IGMs. The signal in MIA can be improved by leveraging shadow models, that are trained on data from the same distribution. LiRA [2] uses the shadow models to estimate the sample's loss distribution for members and non-members, while RMIA [33] compares the likelihood ratio of the target sample with those of reference population samples.

Image Attribution Methods. In contrast to MIAs, image attribution methods seek to identify whether a given image was *generated* by a model or not, which is critical for tracing generated content and preventing data circuits that lead to model collapse [1, 20]. Analogously to MIAs for image autoregressive models [11, 29], PRADA [4] shows that the probability ratio can also carry information about whether an image is generated, i.e., a member of the model's learned distribution, or not generated by the target model. However, the evaluation of the PRADA method is limited to distinguishing generated samples from held-out test samples, which is substantially easier than our newly defined setting: differentiating generated outputs from member training samples. Additionally, PRADA considers only IARs and relies exclusively on the per-token probabilities returned by the image latent generator. As a result, it does not exploit informative signals available in the models' autoencoders, such as the quantization loss between generated and natural (e.g., train or test) samples [35], leaving part of the membership-related information available in IGMs unexploited.

Data Memorization. Memorization describes the extent to which a model retains information from its training data. It can be *unintended*, when the model stores details about individual examples that can later be reproduced or extracted [3, 11]. The *intended* memorization occurs when the model encodes general, reusable patterns that support generalization [7, 27]. For data provenance, the most challenging setting arises when a generative model memorizes training images *verbatim* and subsequently regurgitates them during generation, as was shown for DMs [3] and IARs [11], effectively collapsing the distinction between genuine training images and model-generated outputs. We show that our approach remains effective even in this extreme regime: despite near-duplicate visual content, IGM samples retain subtle generation-specific residuals that are imperceptible to humans yet detectable in the IGMs' latent representations, enabling reliable discrimination between natural training images and generated images.

3. Member vs Generated Inference (MGI)

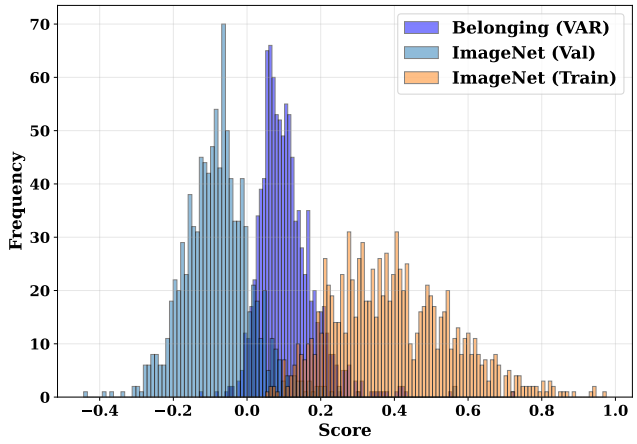
In this section, we formulate MGI and its threat model under both the direct training and the model derivative settings.

Threat Model. The threat model of MGI initially follows that of membership inference and attribution tasks. Given a set of data points D and a generative model $\mathcal{M}$, the goal is to differentiate the subset of member samples D_M used for training of the model $\mathcal{M}$, its generated samples D_G, and non-member samples D_N, that were neither used for training, nor generated by the model. We formalize the task for both *direct training* and *model derivative* settings as follows.

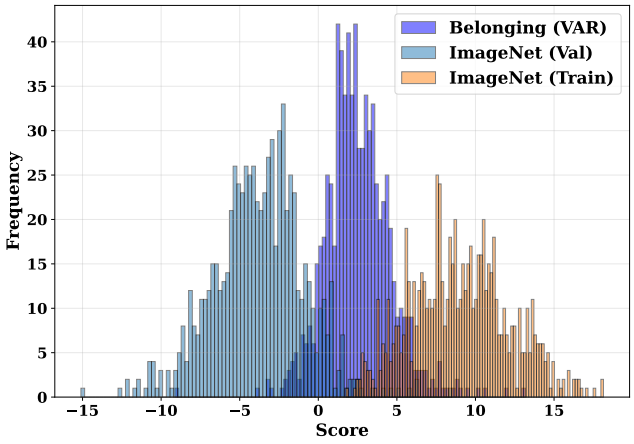
Direct Training. The generative model $\mathcal{M}_1$ was trained on a natural dataset N_M, which we refer to as the natural members. Similar to the conventional MIA setting, there is a natural dataset that was not used for training N_N, which is the natural non-members. Under MGI, we also consider the generated data G, which was produced by $\mathcal{M}_1$. The goal of MGI is to distinguish between the generated samples G, members N_M, and non-members N_N.

Model Derivatives. Given the continuous development of generative models and the ubiquity of the generated content, we also consider the relevant scenario of model derivatives. While the samples N_M were used to train the initial model version $\mathcal{M}_1$, its generated samples G may end up being used to train a new model $\mathcal{M}_2$, resulting in the set G_M, the generated members of $\mathcal{M}_2$ and jointly G_N the generated non-members. The second model version $\mathcal{M}_2$ produces new samples G' that may end up in further model generations. This iterative training results in data circuits, where new models are derived directly from the previous ones, which can lead to model collapse [1, 20]. Under the MGI setting we assume access to both $\mathcal{M}_1$ and $\mathcal{M}_2$ and the goal is to distinguish N_M vs G_M vs G'.

Model Composition. A generative model $\mathcal{M}$ consists of an autoencoder $\mathcal{A} = \mathcal{D} \circ \mathcal{E}$, pairing an encoder $\mathcal{E}$ with a



(a) The distribution of scores for the state-of-the-art MIA on IARs [11].



(b) The distribution of scores for a IAR-generated image attribution [4].

Figure 2. **Distributions of scores for membership inference attack and image attribution on IARs.** In all cases, the differentiation between training (Train) and generated (Belonging) images is more difficult than between training (Train) and validation (Val) images. This indicates more difficult cases of the MGI (Member vs Generated Inference) than MI task. The evaluated model is VAR [24].

decoder $\mathcal{D}$, and a latent generator $\mathcal{G}$. $\mathcal{M}$ can thus be defined as a triplet composition of $\mathcal{E}, \mathcal{D}$, and $\mathcal{G}$: $\mathcal{M} = \langle \mathcal{E}, \mathcal{D}, \mathcal{G} \rangle$.

4. Limitations of MIA and Attribution Methods

MIAs for IGMs [11, 29, 34] are formulated for the classical setting of distinguishing *natural* training members from *natural* non-members, and they rely almost exclusively on likelihood-based or probability-based signals from the *latent generator*. A representative family of methods score an input image $\mathbf{x}$, with conditioning $\mathbf{c}$, e.g. a class label or a prompt, via a *conditional probability discrepancy* (CPD):

$$\begin{aligned} \Delta(\mathcal{M}, \mathbf{x}, \mathbf{c}) &= \log P_{\mathcal{M}}(\mathbf{x} \mid \mathbf{c}) - \log P_{\mathcal{M}}(\mathbf{x}) \\ &\approx \log P_{\mathcal{G}}(\mathcal{E}(\mathbf{x}) \mid \mathbf{c}) - \log P_{\mathcal{G}}(\mathcal{E}(\mathbf{x})), \end{aligned} \tag{1}$$

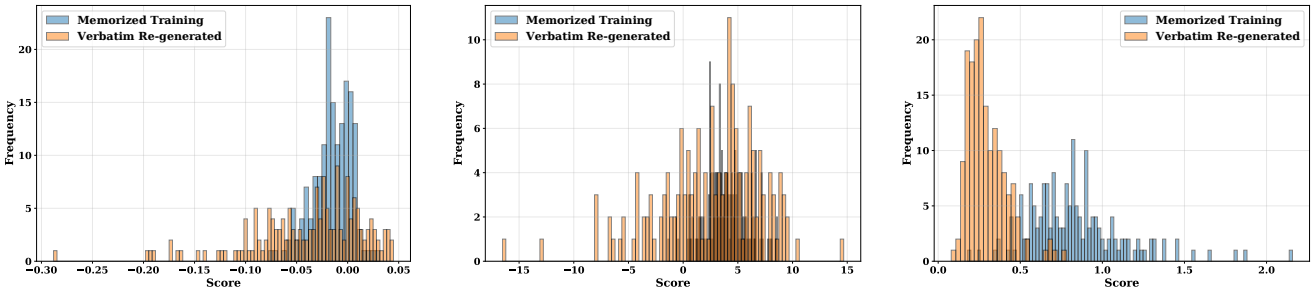
where the approximation reflects the standard IGM decomposition into an encoder $\mathcal{E}$ (mapping pixels to latents) and a latent generative model $\mathcal{G}$ (assigning probabilities in latent space). The decision rule is obtained by thresholding Δ, where members are expected to exhibit systematically larger discrepancies than non-members, as the model *remembers* these samples.

This design implicitly treats the autoencoder $\mathcal{A}$ as a transparent part and largely discards signals that arise from the pixel-to-latent and latent-to-pixel mapping itself. However, the autoencoder $\mathcal{A}$ is a core component of modern IGMs: the encoder $\mathcal{E}$ (typically CNN-based) maps an image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ to a latent feature map $f \in \mathbb{R}^{\frac{H}{p} \times \frac{W}{p} \times C}$ via down-sampling by a factor p, and the corresponding decoder $\mathcal{D}$ reconstructs $\mathbf{x}$ from f. As we show later, these components encode artifacts that are not captured in the likelihood-only or probability-only tests on the latent generator $\mathcal{G}$.

In this paper we consider SOTA MIAs for DMs and IARs. CLiD [34] is an MIA for DMs, which approximates the CPD by leveraging the noise prediction loss to compute the Evidence Lower Bound (ELBO) of the log-likelihood. For IARs we use the MIA from [11], which is computed on the model probabilities directly and call this method *PIAR* in the following. Additionally we use ICAS [29], which considers the classifier-free guidance as an implicit classifier and approximates $p(c|x)$, further weighting this probability to obtain a final score. Furthermore we use PRADA [4], which while proposed for image attribution, has on a high-level, conceptual similarities to MIAs, as both leverage that models that have *seen* the data, be it during training or generation, have a higher likelihood on that image. PRADA first computes a balanced ratio of the CPD and then uses this ratio in a linear scoring function to obtain a final per-image score.

4.1. CPD-based Methods Fall Short for MGI

Recent IGM-specific MIAs exploit classifier-free guidance [8], where the model is trained (and evaluated) both with conditioning (e.g., class/prompt) and without it, making $\Delta(\cdot)$ a natural statistic to measure if the model *remembers* a specific prompt, image pair. Yet, in the MGI setting, generated images are *also* optimized to score highly under the same latent generator that produced them. Consequently, $\Delta(\cdot)$ can be simultaneously large for both true members and the model's own outputs, collapsing the separation that MIAs rely on. In short, likelihood-based or probability-based MIAs are well-suited to member vs held-out *natural* data, but they are not designed to distinguish members from *model-generated* non-members, nor do they leverage poten-



(a) The distribution of scores for the state-of-the-art membership inference on IARs [29].

(b) The distribution of scores for a IAR-generated image attribution [4].

(c) The distribution of scores for the quantization and reconstruction error.

Figure 3. **Distributions of scores for memorized training samples vs re-generated cases.** State-of-the-art membership inference (a) and attribution (b) methods fail to distinguish memorized training samples (*Memorized Training*) from their verbatim generated counterparts (*Verbatim Re-generated*), whereas our approach (c) clearly separates the two. The evaluated model is RAR-XXL [31].

Table 1. **Performance for different methods for identifying memorized samples.** The evaluated model is RAR-XXL.

Metrics	Delta	ICAS	PRADA	Ours
AUC	61.8	61.4	57.9	**97.5**
TPR@5%FPR	3.0	0.0	0.0	**93.5**

tially discriminative signals available in the autoencoder part of IGMs.

4.2. Case Study: Memorized Training Samples

A special case of our MGI task arises when the model regenerates images largely resembling the training samples, which is known as **memorized training samples**. We adopt the methodology proposed by [11] to identify 169 memorized training samples for RAR-XXL [30], where each memorized sample features an SSCD [13] similarity score higher than 0.7.

Concretely, we treat the 169 memorized training images as the *original* samples and their corresponding RAR-XXL outputs as the *generated* samples. The resulting score distributions for membership inference and image attribution are shown in Figure 3a and Figure 3b, respectively. We observe an even greater overlap between memorized training samples and their regenerated counterparts than in the standard MGI comparison. Crucially, despite this highly challenging scenario, our approach leverages multiple different attribution signals and reliably separates memorized training images from their generated counterparts and substantially outperforms MIA-based methods, as shown in Figure 3c. Further results about the memorized training samples can be found in Appendix I.

5. Proposed Data Circuit Breaker

For our proposed solution to MGI, we combine signals from (1) the *autoencoder* that maps between pixels and latents and (2) the *latent generator* that models the latent distribution. The key idea is that an image generated by a particular IGM tends to be *more self-consistent* with that model's autoencoder and latent generator than any natural image or an image generated by a different model, while membership-specific effects (train vs held-out) are more reliably detected after filtering likely-generated samples.

5.1. Autoencoder Self-Consistency

Given the autoencoder $\mathcal{A}$, where the encoder $\mathcal{E}$ maps an image $\mathbf{x}$ from the pixel-space to a latent representation and the decoder $\mathcal{D}$ reconstructs the image from the latent representation, we define the reconstruction error:

$$\mathcal{L}_{\text{Rec}}(\mathbf{x}) = \text{MSE}\big(\mathbf{x}, \mathcal{A}(\mathbf{x})\big) = \text{MSE}\big(\mathbf{x}, \mathcal{D} \circ \mathcal{E}(\mathbf{x})\big), \quad (2)$$

where $\text{MSE}(\cdot, \cdot)$ is the mean squared error. Following AEDR [26], we use a *double reconstruction ratio* to normalize the loss:

$$\rho_{\text{Rec}}(\mathbf{x}) = \frac{\mathcal{L}_{\text{Rec}}(\mathbf{x})}{\text{MSE}\big(\mathcal{A}(\mathbf{x}), \mathcal{A} \circ \mathcal{A}(\mathbf{x})\big)}. \quad (3)$$

Intuitively, the denominator acts as a per-image baseline: if an image is well-aligned with the autoencoder manifold, the second reconstruction introduces hardly any loss, stabilizing the ratio across diverse content.

VQ-VAE Quantization Error. For IARs, the autoencoder is typically a VQ-VAE, which introduces an additional, highly informative signal, namely *quantization error*. The reconstruction procedure of the VQ-VAE introduces a quantization step, where $\mathcal{Q}$ maps a continuous latent representation of an image $\mathbf{x}$ to entries of a codebook. The inverse $\mathcal{Q}^{-1}$, reverts this process and maps codebook indices to the latent representation. We define the quantization error $\mathcal{L}_{\text{Q}}(\mathbf{x})$ as:

$$\mathcal{L}_{\text{Q}}(\mathbf{x}) = \text{MSE}\big(\mathcal{E}(\mathbf{x}), \mathcal{Q}^{-1} \circ \mathcal{Q} \circ \mathcal{E}(\mathbf{x})\big). \quad (4)$$

Images synthesized by a given IAR tend to incur smaller $\mathcal{L}_{\text{Q}}$ under that model's VQ-VAE compared to natural images

or images generated by other IGMs, as only they are produced *through* the same discrete codebook. We therefore use the combined autoencoder attribution score

$$\mathcal{L}_{\mathcal{A}}(\mathbf{x}) = \begin{cases} \rho_{\text{Rec}}(\mathbf{x}) \cdot \mathcal{L}_{\text{Q}}(\mathbf{x}), & \text{(IAR / VQ-VAE)} \\ \rho_{\text{Rec}}(\mathbf{x}), & \text{(DM / VAE).} \end{cases} \tag{5}$$

Optional Encoder Refinement for IARs. The limited alignment between encoder and decoder in IARs introduces additional losses and attribution can degrade. Following [35], we therefore optionally refine the encoder post hoc by fine-tuning $\hat{\mathcal{E}}$ to better invert the decoder $\mathcal{D}$. Fine-tuning is performed on a *disjoint* set of latent feature maps $\mathbf{z}$ that were generated by the IAR with the inversion loss:

$$\mathcal{L}_{\text{Inv}} = \text{MSE}\big(\hat{\mathcal{E}} \circ \mathcal{D}(\mathbf{z}), \mathbf{z}\big), \tag{6}$$

which improves the stability of $\mathcal{L}_{\mathcal{A}}$ while preserving the post-hoc setting, as no changes are introduced to the latent generator.

5.2. Cross-Generator Consistency

While the autoencoder attribution score is able to identify images that were likely generated by a given model family, they are insufficient to attribute the *exact latent generator*. Especially in the model derivative setting, all generated images are decoded by the same decoder, and the autoencoder attribution score remains identical for all of them. Therefore we introduce an additional generator-based features derived from conditional probability discrepancy. Given two candidate models $\mathcal{M}_1$ and $\mathcal{M}_2$, we form a two-dimensional feature vector based on the conditional probability of the two models.

$$\phi(\mathbf{x}, \mathbf{c}) = \big(\log P_{\mathcal{G}_1}(\mathcal{E}(\mathbf{x}) \mid \mathbf{c}), \log P_{\mathcal{G}_2}(\mathcal{E}(\mathbf{x}) \mid \mathbf{c})\big). \tag{7}$$

Intuitively this vector encodes the information about the membership of $\mathbf{x}$ to both the latent generator $\mathcal{G}_1$ and $\mathcal{G}_2$.

As we have access to the image generative models $\mathcal{M}_1$ and $\mathcal{M}_2$, for which we want to infer the membership of given samples, we start by estimating the class-conditional densities over ϕ by generating new data with $\mathcal{M}_1$ and $\mathcal{M}_2$.

We then estimate class-conditional densities over ϕ using reference samples drawn from each model (e.g., freshly generated sets with independent random seeds) and perform attribution via likelihood comparison (KDE in our implementation). This cross-model view provides separation even when absolute discrepancy values overlap, because images tend to be relatively more *consistent* with the generator that produced them. We construct reference sets from each source: a reference set $\mathcal{R}_G$ drawn from G (representing $\mathcal{M}_1$-generated images) and a reference set $\mathcal{R}_{G'}$ drawn from G' (representing $\mathcal{M}_2$-generated images). We then fit class-conditional KDE densities over the feature vectors of each reference set:

$$\hat{p}_G(\phi) = \frac{1}{|\mathcal{R}_G|} \sum_{i=1}^{|\mathcal{R}_G|} K_h\big(\phi - \phi_i^G\big), \tag{8}$$

$$\hat{p}_{G'}(\phi) = \frac{1}{|\mathcal{R}_{G'}|} \sum_{j=1}^{|\mathcal{R}_{G'}|} K_h\big(\phi - \phi_j^{G'}\big). \tag{9}$$

5.3. Attribution Protocol

We combine the above signals in a cascade designed for the MGI setting.

Stage 1: Autoencoder-based Filtering (Generated vs. Non-Generated). We first apply the autoencoder score $\mathcal{L}_{\mathcal{A}}(\mathbf{x})$ to identify a high-confidence subset of *IGM-generated* samples and separate them from samples that are unlikely to be produced by the target pipeline.

Stage 2: Membership Inference on the Remaining Samples (Member vs. Non-Member). On images detected non-generated by Stage 1, we apply standard MIA-style scoring based on the latent generator (e.g., $\Delta(\mathcal{M}, \mathbf{x}, \mathbf{c})$ or ICAS) to distinguish training members from non-members. Restricting MIAs to this subset restores their core assumption (members vs non-members), and substantially reduces false positives caused by model-generated images. With stage 1 and 2, we address the MGI problem for the direct training setting of $\mathcal{M}_1$. Specifically, we instantiate our second stage with ICAS, which is the best-performing MIA for most models. We note that, although PRADA outperforms ICAS on RAR, it requires an extra calibration set. This is an extra advantage beyond our main setting, and restricts the applicability of PRADA. Therefore, we do not choose ICAS instead of PRADA to instantiate our stage 2 for any models.

Stage 3: Source Attribution among Generators (Data Circuits). In the M_2 setting of Figure 1, where training data may itself be generated, we additionally apply cross-model generator attribution using $\phi(\mathbf{x}, \mathbf{c})$ and reference sets from $\mathcal{M}_1$ and $\mathcal{M}_2$ to separate $\mathcal{M}_1$-generated samples used for training $\mathcal{M}_2$, $\mathcal{M}_1$-generated samples not used for training $\mathcal{M}_2$, and $\mathcal{M}_2$-generated samples. Combined with Stage 1 and Stage 2, this yields a practical decomposition into (1) natural members, (2) natural non-members, (3) generated samples attributed to a specific model, and (4) generated samples used for training downstream models, thus fully addressing MGI in the presence of data circuits.

6. Empirical Evaluation

6.1. Experimental Setup

Models. We evaluate SOTA IARs and DMs, following the previous work on MIAs for IGMs [6, 11, 29, 34]. Our se-

lection of models requires access to their training sets for our analysis to verify the outcome of MIAs and MGIs. We use ***VAR-d30*** (d = model depth) [24], ***RAR-XXL*** [30], and ***LlamaGen-XXL*** [22], trained for class-conditioned generation. We download the pre-trained weights from the corresponding repositories and for generation we follow the settings recommended in the original works. For the DMs we focus on the UNet [16] based architectures ***Stable Diffusion 1.4*** and ***2.1*** [15].

Datasets. As the above IARs were trained on ImageNet-1k [5] dataset, we use it to perform our MGI and MIA tasks. We sample 1000 samples from the training set as members and similarly 1000 samples from the validation set as non-members. For Stable Diffusion we follow CLiD [34] and first fine-tune the model on a set of 2500 MS-COCO images for 50k steps to obtain a set of natural member samples and non-member samples. Then we use 1000 samples from training as members and 1000 samples from validation as non-members.

Fine-tuning. We fine-tune the second model $\mathcal{M}_2$ on 5000 images generated by the first model $\mathcal{M}_1$. We denote the images generated by $\mathcal{M}_1$ as G_M. We also keep another 1000 images generated by $\mathcal{M}_1$ as a held-out set (denoted as G_N, which are *generated* data points that act as *non-members*). For IARs we fine-tune $\mathcal{M}_2$ for 5 epochs, while for DMs we use 20. The learning rate is 1×10^{-5} for all models. We provide the hyperparameter details in Appendix A.

Baselines. MGI is a *newly defined task* without an existing solution. We follow standard practice for newly defined tasks by adapting SoTA methods from the closest domains MIA and image attribution. Regarding the ***MIA baselines***, we choose SoTA MIA methods for IARs and DMs, respectively. For IARs, we use [11] and refer to the method as PIAR, and ICAS [29]. For the DMs, we use the SOTA MIA CLiD and extend the IAR-based ICAS to DMs based on the CLiD scores. In Section 6.4, We further test strong MIAs, LiRA/RMIA, which we give an *advantage* by training shadow models. For the direct training setting, MIAs make the assumption that members have a higher score than all non-member samples and we extend this assumption to MGI. Regarding the ***image attribution baseline***, we consider PRADA [4], which is originally proposed for IARs but extended to DMs by us. Under the direct training setting, the image attribution methods make the assumption that generated samples will have the highest score, followed by members and non-members. The same intuition extends to the derivative setting.

Metrics. In the following we focus on the, especially for inference tasks, relevant metric of TPR@1%FPR and additionally provide the AUC in Appendix C.

6.2. Evaluation on the Direct Training Setting

First we focus on the direct training setting, known from the MIA task, where the model $\mathcal{M}_1$ was trained on natural images resulting in the natural members N_M and natural non-members N_N. However, our MGI introduces the models generated samples G as a new an important part of this task.

Under this new setting, we analyze the performance of the original MIAs and image attribution methods for IARs in Table 2 and report the TPR@1%FPR. We find that while existing MIAs are able to separate the natural members from the natural non-members, they break when the generated data is introduced. Our DCB however achieves near 100% TPR@1%FPR under the generated vs natural setting and improves the average performance by over 36% (LlamaGen). Under the MGI task DCB benefits from combining multiple signals leading to a consistent performance across detections.

We additionally analyze the MGI task on DMs in Table 3, following CLiD [34] and fine-tuning the models on natural MS-COCO data to obtain natural members and non-members. Our results show that, similar as for the IARs, while the baseline methods are able to distinguish N_N and N_M, they fail when the generated data is introduced. This difficulty for MIAs is additionally visualized in Appendix B, which plots the score distributions for the different datasets. As the generated data was produced by $\mathcal{M}_1$, the MIA obtains a high score, as the model *remembers* the sample. This occurs because likelihood-based scores are similarly elevated for both member samples and the model's own outputs, collapsing the separation that MIAs rely upon. Only DCB, which takes the full generative pipeline into account, can distinguish the generated data from the natural data. This effect is especially pronounced in the $\mathcal{M}_2$ setting, where DCB beats the baselines by more than 39% (SD2.1).

6.3. Evaluation on the Model Derivative Setting

As images generated by IGMs experience a widespread reuse, we shift the focus to the derivative setting, where a model $\mathcal{M}_2$ was fine-tuned on generated images G_M, which are now the member samples of $\mathcal{M}_2$. The new model continuously generates new images G', which, with the generated non-members G_N introduces three datasets to the MGI task. In Table 4 we report the TPR@1%FPR for distinguishing the different datasets and find that the existing methods struggle to differentiate the distributions for both IARs and DMs. Our DCB, on the other hand, is able to clearly distinguish the two sets.

The difficulty of this new MGI setting is reflected in the comparison of Table 4. While the baselines achieve reasonable performance for most *Natural vs Generated* cases, they struggle when differentiating within the generated samples. Particularly for the DMs, the detection performance collapses. The score distributions in Figure A4 and Appendix B provide additional insights as to why the MGI setting is

Table 2. **TPR@1%FPR for IARs in the direct training setting.** Only DCB achieves consistent performance across all comparisons and models.

Method	RAR				VAR				LlamaGen				Overall
	N_M/G	N_N/G	N_M/N_N	Avg	N_M/G	N_N/G	N_M/N_N	Avg	N_M/G	N_N/G	N_M/N_N	Avg	
PIAR	0.0	99.5	62.6	54.0	58.6	11.5	91.7	53.9	0.5	17.7	6.7	8.3	38.8
ICAS	0.0	99.7	72.5	57.4	61.9	33.9	**98.7**	64.8	0.0	89.6	**17.2**	35.6	52.6
PRADA	62.7	**100.0**	**81.3**	81.3	0.0	24.8	96.9	40.6	9.3	68.8	7.1	28.4	50.1
Ours	**99.9**	99.9	72.5	**90.8**	**99.3**	**99.5**	**98.7**	**99.2**	**100.0**	**100.0**	**17.2**	**72.4**	**87.4**

Table 3. **TPR@1%FPR for DMs in the direct training setting.** Only DCB achieves consistent performance across all comparisons and models.

Method	SD1.4				SD2.1				Overall
	N_M/G	N_N/G	N_M/N_N	Avg	N_M/G	N_N/G	N_M/N_N	Avg	
CLiD	0.0	88.2	**36.2**	41.5	0.0	82.2	**31.5**	37.9	39.7
ICAS	0.0	87.8	35.7	41.2	0.0	82.1	**31.5**	37.9	39.5
PRADA	0.7	0.4	0.4	0.5	0.7	0.3	0.4	0.4	0.5
Ours	**99.9**	**99.8**	35.7	**78.5**	**100.0**	**100.0**	**31.5**	**77.2**	**77.8**

Table 4. **TPR@1%FPR for the model derivative setting.** Most existing methods fail to attribute generated samples correctly.

Model	Method	Natural vs Generated						Among Generated			Natural	Overall
		N_M/G_M	N_M/G_N	N_M/G'	N_N/G_M	N_N/G_N	N_N/G'	G_M/G_N	G_M/G'	G_N/G'	N_M/N_N	
VAR	PIAR	70.6	0.2	1.9	99.9	62.4	97.8	94.0	62.8	15.8	79.2	58.5
	ICAS	**99.8**	10.3	82.8	**100.0**	89.9	**99.8**	99.6	91.8	51.6	**87.0**	81.3
	PRADA	93.2	0.3	9.7	99.9	68.0	98.7	95.7	0.0	16.0	82.3	56.4
	Ours	99.3	**99.3**	**99.4**	99.5	**99.5**	99.6	**100.0**	**99.0**	**75.0**	**87.0**	**95.8**
RAR	PIAR	**100.0**	59.9	97.0	**100.0**	99.8	**100.0**	74.6	18.5	16.4	62.6	72.9
	ICAS	**100.0**	82.9	99.2	**100.0**	99.6	**100.0**	95.6	20.9	49.9	72.5	82.1
	PRADA	**100.0**	82.1	98.0	**100.0**	**100.0**	**100.0**	82.4	0.0	18.9	**82.0**	76.3
	Ours	99.9	**99.9**	**99.9**	99.9	99.9	99.9	**100.0**	**100.0**	**95.6**	72.5	**96.7**
SD1.4	CLiD	84.8	71.7	98.5	**99.9**	99.6	**100.0**	5.4	1.0	37.9	**36.2**	63.5
	ICAS	84.8	71.6	98.6	**99.9**	99.6	**100.0**	5.4	1.0	38.1	35.7	63.5
	PRADA	0.1	0.5	2.4	0.1	0.0	2.3	0.7	4.1	3.5	0.0	1.4
	Ours	**99.9**	**99.9**	**98.7**	99.8	**99.8**	98.5	**56.0**	**42.2**	**95.8**	35.7	**82.6**
SD2.1	CLiD	86.8	74.2	99.5	99.9	99.5	**100.0**	5.8	0.0	47.2	**31.5**	64.4
	ICAS	86.8	73.9	99.5	99.9	99.5	**100.0**	5.8	0.0	47.0	**31.5**	64.4
	PRADA	0.3	0.4	0.4	0.0	0.3	0.1	0.9	2.1	1.7	0.1	0.6
	Ours	**100.0**	**100.0**	**99.7**	**100.0**	**100.0**	99.8	**52.8**	**54.0**	**99.0**	**31.5**	**83.7**

fundamentally more difficult. Contrary to MIA assumption, the score of the generated samples is larger than the score of non-members and overlaps or exceeds the score of the members. This property of the generated samples is why standard MIA fail.

6.4. Analysis for Strong MIA

We expand on the explored MIAs, by analyzing stronger methods and employ both LiRA [2] and RMIA [33] on the model derivative setting and show that even under access to trained shadow models, the MIA fail in the MGI setting. Both LiRA and RMIA require a scaler score to compute a one-dimensional probability distribution. We use the strongest probability-based MIA method ICAS [29] to convert the token-wise probabilities predicted by IARs into a score scalar for each sample.

Concretely, we obtain 5 shadow models, by fine-tuning $\mathcal{M}_1$ for 5 epochs, with the same hyperparameters used for $\mathcal{M}_2$, on datasets of 2500 samples randomly drawn from a 5000-sample shadow dataset. For RMIA, we utilize an additional population dataset of 1000 samples generated by $\mathcal{M}_1$ and set its core hyperparameters to $\alpha = 0.3$. This setup enables the methods to estimate the distribution of generated members and non-member samples, giving these methods a strict advantage for the G_N vs G_M case compared to DCB. We report the TPR@1%FPR in Table 5, for VAR and RAR. The results highlight that even under significant advantages, strong MIAs are not sufficient to solve the MGI task. Specifically for the *Natural vs. Generated* identification the strong MIA perform similar to the MIAs without shadow models. Notably, our DCB consistently outperforms both LiRA and RMIA across all comparisons, highlighting that utilizing the

Table 5. **TPR@1%FPR for the strong MIA.** We consider both LiRA and RMIA and train 5 shadow models.

Model	Method	Natural v.s. Generated						Among Generated			Natural	Overall
		N_M/G_M	N_M/G_N	N_M/G'	N_N/G_M	N_N/G_N	N_N/G'	G_M/G_N	G_M/G'	G_N/G'	N_M/N_N	
VAR	LiRA	98.7	1.0	16.7	98.7	1.1	16.8	98.7	50.3	16.7	1.1	40.0
	RMIA	**100.0**	3.6	78.3	**100.0**	77.5	**99.6**	99.9	98.3	67.2	40.0	76.4
	Ours	99.3	**99.3**	**99.4**	99.5	**99.5**	**99.6**	**100.0**	**99.0**	**75.0**	**87.0**	**95.8**
RAR	LiRA	92.8	0.8	31.2	95.0	1.2	36.9	94.2	18.4	34.4	1.3	40.6
	RMIA	**99.9**	65.7	99.5	**100.0**	96.2	**100.0**	99.0	77.5	71.3	17.8	82.7
	Ours	**99.9**	**99.9**	**99.9**	99.9	**99.9**	99.9	**100.0**	**100.0**	**95.6**	**72.5**	**96.7**

full model pipeline provides stronger signals.

6.5. Robustness

We evaluate our proposed Stage 1 (as described in Section 5.3) under real-world web-pipeline degradations (JPEG, resize, saturation) and the strong adaptive adversarial attack that directly optimizes perturbations to maximize $\mathcal{L}_{\text{Q}}$. The results are shown in Table 6. Without any augmentation, Stage 1 already retains $\geq 88.5\%$ TPR@1%FPR under all natural transforms. Optionally, we apply augmentations to the fine-tuning process of the encoder, inspired by [36] and [10]. The augmented fine-tuning further boosts robustness to 97.4% even under the adaptive adversarial attack.

6.6. Cross-architecture Generalization

In the model derivative setting, we mainly consider the *identical-architecture* setting where $\mathcal{M}_2$ is trained on the data generated by $\mathcal{M}_1$ with the same architecture as $\mathcal{M}_2$. In this section, we further evaluate a *cross-architecture* setting, where $\mathcal{M}_2$ is trained on images generated by a model with different architecture. We note that a cross-architecture setting is an *easier* setting for MGI, not more challenging. If $\mathcal{M}_2$ is trained on images generated by another model architecture, the distinct autoencoder architectures *enhance* Stage 1 separation of G_M/G' (rather than collapsing it), and G_M/G_N reduces to standard MIA. In contrast, the identical-architecture setting is a more challenging setting, because G_M,G_N, and G' are all from the same autoencoder and therefore require Stage 3. Therefore, we choose the the more challenging identical-architecture setting for evaluation in our main content. Table 7 evaluates the cross-architecture setting on the **SD 1.4→SD 2.1** case and two heterogeneous DM-to-IAR pairs. The results show that DCB attains $\geq 99\%$ AUC on G_M/G' for all settings.

6.7. Prompt Estimation

We note that ground-truth prompts can be absent in certain applications. In such cases, we use BLIP2/LLaVA to generate prompts for a given image. Table 8 compares the performance of our approach using groundtruth (GT) and BLIP2/LLaVA-generated captions. The results show that DCB still achieves high performance.

7. Conclusions

We introduced Member vs Generated Inference (MGI), a new and strictly harder inference task than standard membership inference. MGI requires separating a generative model's training members from its own generated outputs, including in data-circuit settings where subsequent models are trained on generated data. We showed that existing membership inference and attribution methods are inadequate for MGI because modern generative models produce non-member samples that are closely tied to the training distribution, leading to MIAs systematically misclassifying generated samples as members, while attribution methods mislabel true training members as generated. To address this, we proposed DCB, a multi-stage pipeline that covers the full generation process by leveraging complementary signals from the autoencoder and latent generator. By first identifying synthesized content and then distinguishing remaining training members from non-members, DCB is able to consistently outperform previous methods. We demonstrated that DCB remains effective even on memorization, separating original training samples from their regurgitated counterparts and enabling practical mitigation of harmful data circuits. Finally, we showed that DCB achieves better detection rate than strong membership inference attacks such as LiRA and RMIA, highlighting that a holistic procedure of the full generative pipeline is essential to solve MGI.

Acknowledgments

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Project number 550224287. Franziska Boenisch received funding from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (grant agreement No 101220235). We would like to acknowledge our sponsors, who support our research with financial and in-kind contributions: OpenAI and G-Research. We also thank members of the SprintML group for their feedback. Responsibility for the content of this publication lies with the authors.

Table 6. **Robustness of Stage 1 ($\mathcal{L}_Q$).** We show TPR@1%FPR on RAR.

Method	Attacks				
	Orig.	JPEG (60)	Resize (0.5)	Saturation (2.0)	Adv. ($\varepsilon = 1$)
Ours (w/o Aug)	100.0	91.7	88.5	97.4	68.7
Ours (w/ Aug)	99.6	96.1	98.4	99.2	97.4

Table 7. **Cross-architecture Setting.**

Model	G_M/G_N			G_M/G'			G_N/G'		
	ICAS	PRADA	DCB	ICAS	PRADA	DCB	ICAS	PRADA	DCB
REPA→RAR	**90.9**	91.6	**90.9**	76.9	26.4	**99.0**	72.9	77.5	**99.4**
LDiT→RAR	**87.3**	87.2	**87.3**	74.6	26.3	**99.7**	69.0	70.3	**99.7**
SD 1.4→SD 2.1	**79.9**	60.6	**79.9**	16.9	63.6	**100.0**	96.4	73.6	**99.9**

Table 8. **Prompt estimation.** Captions generated by BLIP2/LLaVA replace ground-truth (GT) prompts in Table 4.

Model	ICAS			DCB		
	GT	BLIP2	LLaVA	GT	BLIP2	LLaVA
SD 1.4	63.5	46.2	33.6	**82.6**	**81.5**	**78.7**
SD 2.1	64.4	43.4	28.6	**83.7**	**82.8**	**76.7**

References

[1] Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. Self-consuming generative models go MAD. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[2] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914, 2022. 3, 8

[3] 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, 3

[4] Simon Damm, Jonas Ricker, Henning Petzka, and Asja Fischer. Prada: Probability-ratio-based attribution and detection of autoregressive-generated images. *arXiv preprint arXiv:2511.20068*, 2025. 2, 3, 4, 5, 7, 12, 13

[5] 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. 7

[6] Jan Dubiński, Antoni Kowalczuk, Franziska Boenisch, and Adam Dziedzic. CDI: Copyrighted Data Identification in Diffusion Models. In *The IEEE CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2025. 6

[7] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020. 3

[8] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 6840–6851, 2020. 2

[10] Nikola Jovanović, Ismail Labiad, Tomáš Souček, Martin Vechev, and Pierre Fernandez. Watermarking autoregressive image generation. *Advances in Neural Information Processing Systems*, 38:71801–71848, 2026. 9

[11] Antoni Kowalczuk, Jan Dubiński, Franziska Boenisch, and Adam Dziedzic. Privacy attacks on image autoregressive models. In *Forty-Second International Conference on Machine Learning (ICML)*, 2025. 1, 2, 3, 4, 5, 6, 7, 12, 13

[12] Bartłomiej Marek, Lorenzo Rossi, Vincent Hanke, Xun Wang, Michael Backes, Franziska Boenisch, and Adam Dziedzic. Benchmarking empirical privacy protection for adaptations of large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. 3

[13] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14532–14542, 2022. 5

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021. 2

[15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Con-*

ference on Computer Vision and Pattern Recognition, 2022. 7

[16] 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. 7

[17] Lorenzo Rossi, Bartłomiej Marek, Franziska Boenisch, and Adam Dziedzic. Natural identifiers for privacy and data audits in large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. 3

[18] 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 *Network and Distributed System Security Symposium (NDSS)*, 2019. 3

[19] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017. 3

[20] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631 (8022):755–759, 2024. 3

[21] Yang Song and Stefano Ermon. Improved Techniques for Training Score-Based Generative Models. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 12438–12448, 2020. 2

[22] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 3, 7

[23] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024. 3

[24] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction, 2024. 4, 7, 13, 14, 15

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017. 2

[26] Chao Wang, Kejiang Chen, Zijin Yang, Yaofei Wang, and Weiming Zhang. Aedr: Training-free ai-generated image attribution via autoencoder double-reconstruction. *arXiv preprint arXiv:2507.18988*, 2025. 5

[27] Wenhao Wang, Muhammad Ahmad Kaleem, Adam Dziedzic, Michael Backes, Nicolas Papernot, and Franziska Boenisch. Memorization in self-supervised learning improves downstream generalization. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. 3

[28] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 16

[29] Hongyao Yu, Yixiang Qiu, Yiheng Yang, Hao Fang, Tianqu Zhuang, Jiaxin Hong, Bin Chen, Hao Wu, and Shu-Tao Xia. Icas: Detecting training data from autoregressive image generative models. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 11209–11217, 2025. 2, 3, 4, 5, 6, 7, 8, 14

[30] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation, 2024. 5, 7

[31] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18431–18441, 2025. 3, 5, 13, 14, 15

[32] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025. 16

[33] Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In *Forty-first International Conference on Machine Learning*, 2024. 3, 8

[34] 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 *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2, 3, 4, 6, 7, 12, 14

[35] Bihe Zhao, Louis Kerner, Michel Meintz, Tameem Bakr, Franziska Boenisch, and Adam Dziedzic. Data provenance for image auto-regressive generation. In *The Fourteenth International Conference on Learning Representations*, 2026. 3, 6

[36] Bihe Zhao, Louis Kerner, Michel Meintz, Tameem Bakr, Franziska Boenisch, and Adam Dziedzic. Data provenance for image auto-regressive generation. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. 9

A. Further Implementation Details

A.1. Data Pre-processing

We follow the augmentations in the original training recipes of each model to ensure that our evaluation faithfully reflects the conditions under which membership and generation signals arise. For VAR and LlamaGen, when computing the MIA scores on $\mathcal{M}_1$, we apply the same data augmentations used during training: each image is first resized by a factor of 1.125 for VAR and 1.1 for LlamaGen, followed by a center crop to the model's native resolution (256×256 for VAR and 384×384 for LlamaGen).

A.2. Fine-tuning

We provide the hyperparameters for fine-tuning $\mathcal{M}_2$ in the model derivative setting in Table A1. For all models, we fine-tune exclusively the *latent generator* while keeping the autoencoder weights frozen. The latent generator corresponds to the transformer in IARs and the UNet in the diffusion models. This design choice mirrors the common practice in which downstream practitioners adapt only the generative backbone to new data. The fine-tuning data consists of $5{,}000$ images generated by $\mathcal{M}_1$. For the IARs (VAR and RAR), 5 epochs suffice to adapt the latent generator to the generated distribution, whereas for the diffusion models (SD 1.4 and SD 2.1) we train for 20 epochs due to their slower convergence. All experiments use a fixed learning rate of 1×10^{-5} with the AdamW optimizer.

Table A1. Hyperparameters for fine-tuning $\mathcal{M}_2$ for the derivative setting.

Model	Batch Size	Learning Rate	Training Samples	Epochs
VAR	4	1×10^{-5}	5000	5
RAR	4	1×10^{-5}	5000	5
SD 1.4	4	1×10^{-5}	5000	20
SD 2.1	4	1×10^{-5}	5000	20

B. Distribution Visualization on More Models

In this section, we complement the distribution analysis of the main paper with visualizations for additional models and settings. These plots substantiate our central observation: across all evaluated architectures, the distributions for generated images overlap heavily with those for training members, making the MGI task fundamentally harder than standard membership inference.

B.1. Model Direct Training Setting

RAR. Figure A1 shows the score distributions for PIAR [11] and the PRADA image attribution method [4] on RAR-XXL. For both scoring functions, the generated ("Belonging") distribution is substantially closer to the training distribution than the held-out validation set. This confirms that the conditional probability discrepancy used by existing MIAs cannot reliably separate members from generated samples in the RAR architecture.

LlamaGen. A similar pattern emerges for LlamaGen-XXL (Figure A2). Here, the overlap between the generated and training distributions is even more pronounced under the MIA score, with the generated distribution shifted further toward the member region compared to the validation distribution. The PRADA attribution score provides somewhat better separation, yet a significant fraction of generated samples still falls within the range of training member scores, highlighting the inadequacy of likelihood-based methods alone for the MGI task.

Stable Diffusion 1.4. We extend the analysis to diffusion models in Figure A3, which plots the CLiD MIA score [34] distributions for both $\mathcal{M}_1$ and $\mathcal{M}_2$. For $\mathcal{M}_1$, the generated images exhibit score distributions that overlap substantially with training members, consistent with the findings on IARs. In the $\mathcal{M}_2$ setting, the additional fine-tuning on generated data introduces even more complex membership signals, resulting in a more entangled set of distributions. These results motivate the multi-stage approach of DCB, which leverages complementary autoencoder-based signals to disentangle these overlapping distributions.

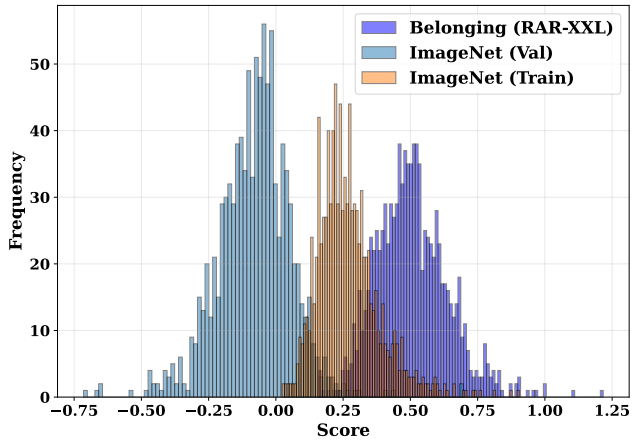
B.2. Model Derivative Setting

We additionally visualize the score distributions for the model derivative setting. Figures A4 to A6 present three complementary views under the $\mathcal{M}_2$ scenario, each shown for both VAR and RAR-XXL: the latent-generator MIA score, the autoencoder reconstruction and quantization error, and the cross-generator probability discrepancy, respectively.

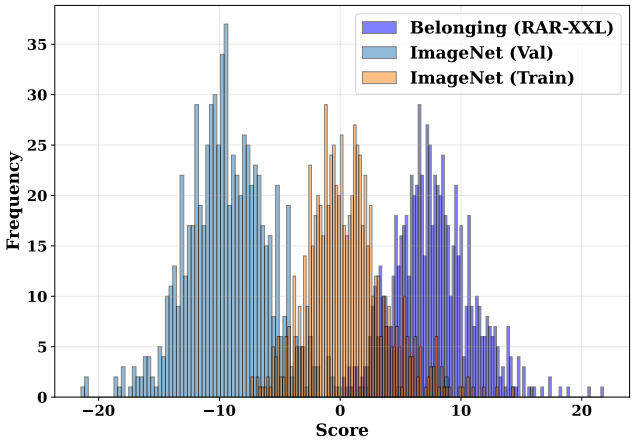
The MIA score distribution in Figure A4 reveals a complex mixture: generated members (G_M) and generated non-members (G_N) produce high MIA scores that overlap with or exceed those of natural members. This confirms that probability-based MIAs alone cannot distinguish the provenance of generated samples in the derivative setting.

In contrast, the autoencoder-based score in Figure A5 provides a clear separation between natural and generated images, as generated images exhibit lower reconstruction and quantization errors. However, it cannot distinguish among different sources of generated content (*i.e.*, G_M vs. G_N vs. G').

The cross-generator probability discrepancy in Figure A6 addresses this gap: by comparing the conditional log-probabilities under $\mathcal{G}_1$ and $\mathcal{G}_2$, the feature vector $\phi(\mathbf{x}, \mathbf{c})$ reveals distinct clusters for images generated by $\mathcal{M}_1$ versus $\mathcal{M}_2$, enabling fine-grained attribution among generated sources. Together, these three complementary signals form the basis of the DCB pipeline and motivate its cascaded three-stage design.

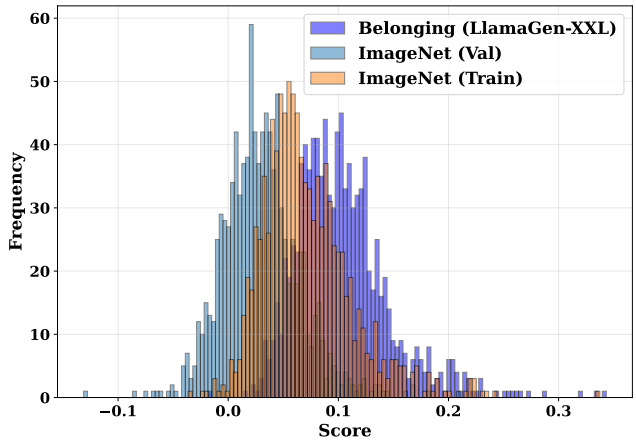


(a) The distribution of scores for the membership inference attack PIAR [11]

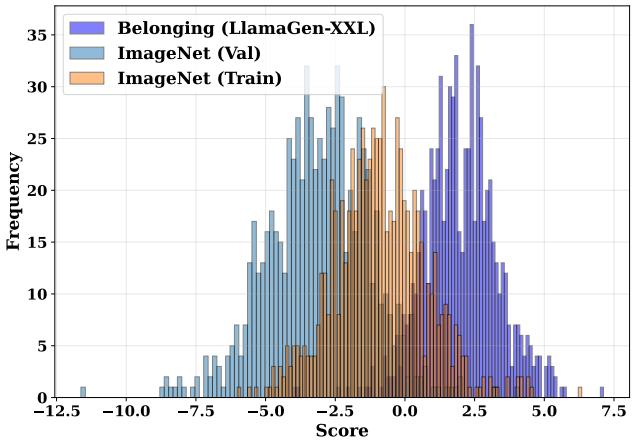


(b) Distribution of scores for a IAR-generated image attribution PRADA [4].

Figure A1. **Distributions of scores for membership inference and image attribution on RAR-XXL [31].**



(a) The distribution of scores for the membership inference attack PIAR [11]



(b) The distribution of scores for a IAR-generated image attribution [4].

Figure A2. **Distributions of scores for membership inference and image attribution on LlamaGen-XXL [24].**

C. Additional Results

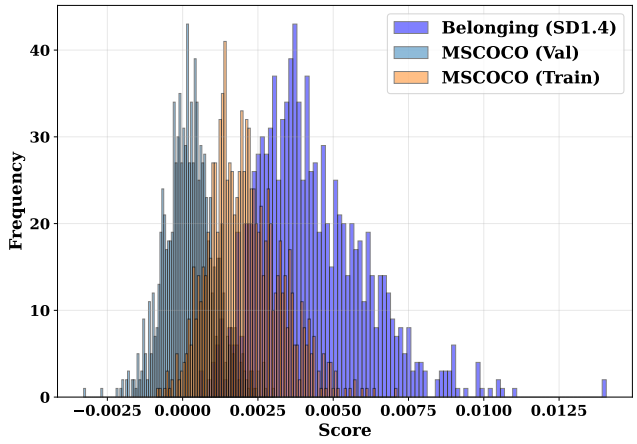
C.1. Model Direct Training Setting

AUC. Table A2 and Table A3 report the AUC for IARs and DMs, respectively. DCB achieves an overall AUC of 98.0 on IARs and 96.5 on DMs, substantially outperforming all baselines. Notably, the advantage of DCB is most pronounced in the N_M/G comparison, where existing MIAs exhibit very limited AUC (*e.g.*, 7.8 for PIAR and 0.9 for ICAS on RAR), confirming that likelihood-based MIAs cannot separate members from generated samples. DCB achieves 100.0 AUC on both RAR and LlamaGen for this critical comparison, demonstrating perfect separation through its autoencoder-based filtering stage. For DMs, PRADA degrades to below 52 AUC across both SD 1.4 and SD 2.1, while DCB reaches at least 99.9.

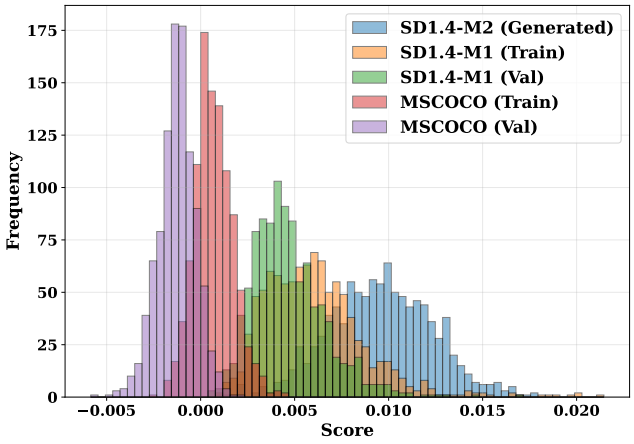
TPR@5%FPR. Table A4 and Table A5 present the TPR at 5% FPR. The trends are consistent with the AUC results: DCB achieves an overall TPR@5%FPR of 92.7 on IARs and 85.4 on DMs, compared to the best baseline scores of 71.7 (PRADA on IARs) and 50.5 (CLiD/ICAS on DMs). The improvement is especially significant in the N_M/G column, where DCB achieves 100.0% TPR@5%FPR on RAR and LlamaGen while all baselines remain below 87%. For the N_M/N_N comparison, DCB matches the best baseline in each case, as it falls back on the standard MIA score in Stage 2 of the pipeline after filtering out generated samples.

C.2. Model Derivative Setting

AUC. Table A6 reports the AUC across all pairwise comparisons in the model derivative setting. DCB achieves the highest overall AUC for every model: 99.5 (VAR), 99.8

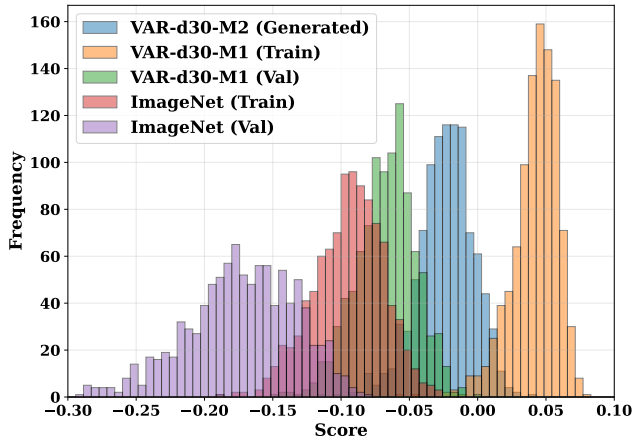


(a) The distribution of scores for the state-of-the-art MIA of Diffusion Models, CLiD [34]. The evaluated model is $\mathcal{M}_1$.

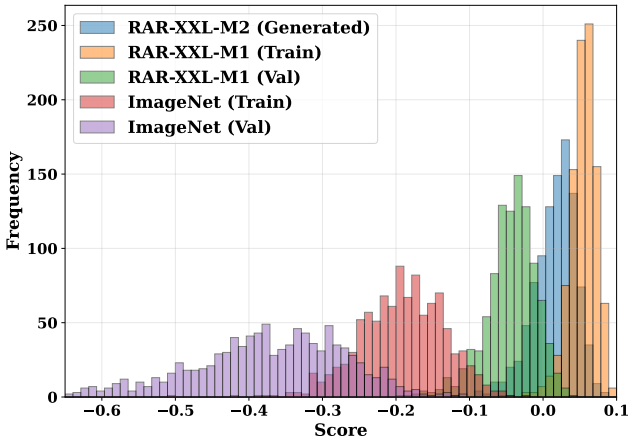


(b) The distribution of scores for the state-of-the-art MIA of Diffusion Models, CLiD [34]. The evaluated model is $\mathcal{M}_2$.

Figure A3. **Distributions of scores for membership inference attack on diffusion models for both $\mathcal{M}_1$ and $\mathcal{M}_2$.** The evaluated model is Stable Diffusion 1.4 (fine-tuned on MS-COCO).



(a) Distribution for VAR-d30 [24].



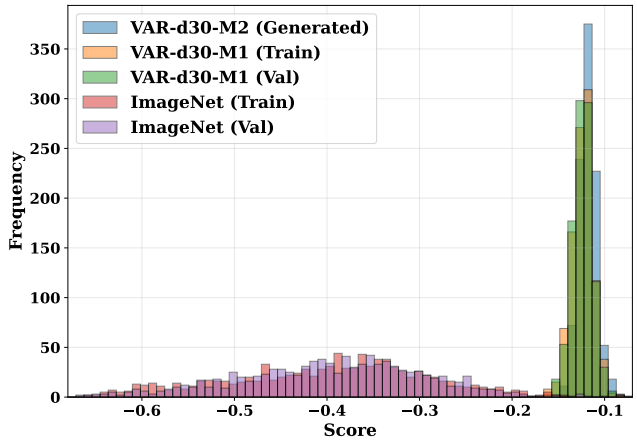
(b) Distribution for RAR-XXL [31].

Figure A4. **Distributions of scores for the state-of-the-art MIA on IARs [29] in the model ($\mathcal{M}_2$) derivative setting.** With respect to the model $\mathcal{M}_2$, we assign the following labels to the datasets: *Generated* for $\mathcal{M}_2$-generated data, *Train* for the member data used for training $\mathcal{M}_2$ (including pre-training and finetuning), and *Val* for non-member validation data. For the model $\mathcal{M}_2$, generated members (G_M) and generated non-members (G_N) yield high MIA scores that overlap with or exceed those of natural members, so probability-based MIAs alone cannot distinguish the provenance of generated samples.

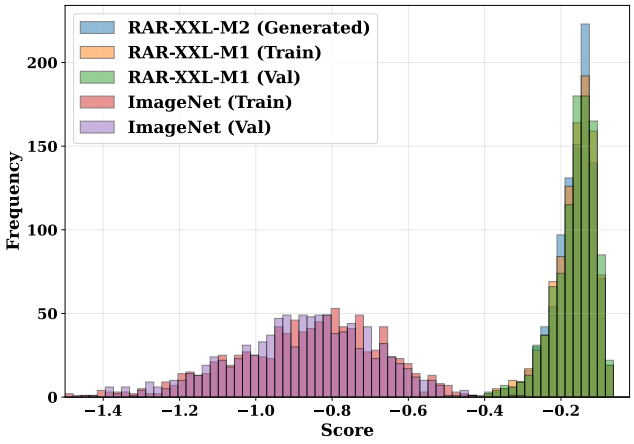
Table A2. **AUC for IARs in the direct training setting.**

Method	RAR				VAR				LlamaGen				Overall
	N_M/G	N_N/G	N_M/N_N	Avg	N_M/G	N_N/G	N_M/N_N	Avg	N_M/G	N_N/G	N_M/N_N	Avg	
PIAR	7.8	99.8	98.4	68.7	96.6	95.1	99.6	97.1	28.0	92.5	79.7	66.7	77.5
ICAS	0.9	100.0	98.6	66.5	97.0	95.3	**99.9**	97.4	6.3	99.2	**84.3**	63.3	75.7
PRADA	98.0	100.0	**99.1**	99.0	3.4	95.9	99.8	66.4	93.1	98.9	79.1	90.4	85.3
Ours	**100.0**	**100.0**	98.6	**99.5**	**99.6**	**99.8**	**99.9**	**99.8**	**100.0**	**100.0**	**84.3**	**94.8**	**98.0**

(RAR), 97.8 (SD 1.4), and 97.5 (SD 2.1). The baselines clearly fail on the challenging comparisons *among generated samples* (G_M/G' and G_N/G'), where only DCB's cross-generator probability discrepancy (Stage 3) provides reliable separation. For example, on VAR, PRADA achieves only 2.9 AUC for G_M/G', while DCB reaches 99.9. On the diffusion models, the performance gap is particularly clear in the "Among Generated" columns: CLiD/ICAS achieve

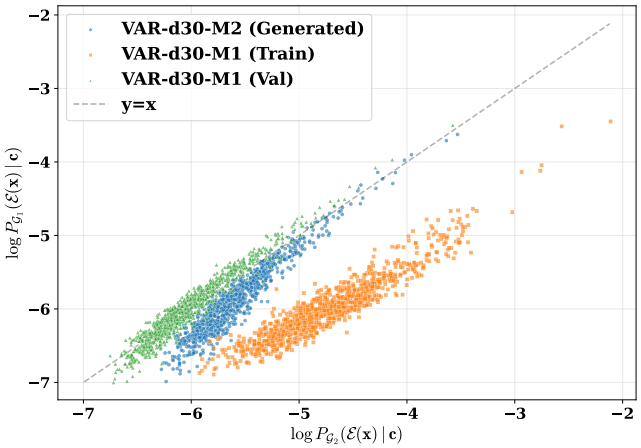


(a) Distribution for VAR-d30 [24].

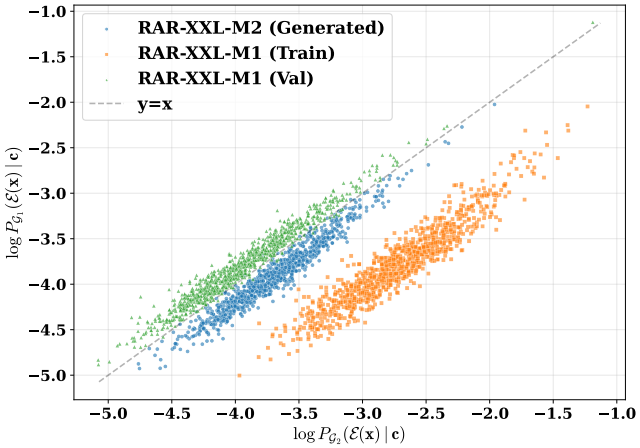


(b) Distribution for RAR-XXL [31].

Figure A5. **Distributions of scores for autoencoder-based score in the model ($\mathcal{M}_2$) derivative setting.** With respect to the model $\mathcal{M}_2$, we assign the following labels to the datasets: *Generated* for $\mathcal{M}_2$-generated data, *Train* for the member data used for training $\mathcal{M}_2$ (including pre-training and finetuning), and *Val* for non-member validation data. The evaluated autoencoder-based score is defined by Equation (5). The score cleanly separates natural from generated images, as generated images exhibit lower reconstruction and quantization errors, but it cannot distinguish among the different sources of generated content (G_M vs G_N vs G').



(a) Distribution for VAR-d30 [24].



(b) Distribution for RAR-XXL [31].

Figure A6. **Distributions of scores for the cross-generator probability discrepancy in the model derivative setting.** With respect to the model $\mathcal{M}_2$, we assign the following labels to the datasets: *Generated* for $\mathcal{M}_2$-generated data, *Train* for the member data used for training $\mathcal{M}_2$ (including pre-training and finetuning), and *Val* for non-member validation data. Comparing the conditional log-probabilities under $\mathcal{G}_1$ and $\mathcal{G}_2$ reveals distinct clusters for images generated by $\mathcal{M}_1$ versus $\mathcal{M}_2$, enabling fine-grained attribution among generated sources. Together with the MIA and autoencoder signals, these three complementary scoring functions motivate the cascaded design of DCB.

Table A3. **AUC for the DMs the direct training setting.**

Method	SD1.4				SD2.1				Overall
	N_M/G	N_N/G	N_M/N_N	Avg	N_M/G	N_N/G	N_M/N_N	Avg	
CLiD	14.0	99.3	**90.8**	68.1	14.7	98.8	**88.2**	67.2	67.6
ICAS	14.0	99.3	90.8	68.1	14.7	98.8	88.2	67.2	67.6
PRADA	51.4	41.9	39.8	44.4	53.1	44.3	40.7	46.0	45.2
Ours	**99.9**	**100.0**	90.8	**96.9**	**100.0**	**100.0**	88.2	**96.1**	**96.5**

16.4 and 12.8 AUC for G_M/G' on SD 1.4 and SD 2.1, respectively, whereas DCB attains 91.6 and 90.6.

TPR@5%FPR. Table A7 presents the TPR@5%FPR for the same setting. The results are consistent with the

Table A4. **TPR@5%FPR for IARs in the direct training setting.**

Method	RAR				VAR				LlamaGen				Overall
	N_M/G	N_N/G	N_M/N_N	Avg	N_M/G	N_N/G	N_M/N_N	Avg	N_M/G	N_N/G	N_M/N_N	Avg	
PIAR	0.3	**100.0**	94.1	64.8	82.2	64.8	99.7	82.2	2.2	59.2	30.3	30.6	59.2
ICAS	0.0	99.9	93.3	64.4	87.1	77.3	99.9	88.1	0.0	96.2	**41.8**	46.0	66.2
PRADA	86.5	**100.0**	**97.5**	94.7	0.0	82.7	**100.0**	60.9	52.5	97.2	28.8	59.5	71.7
Ours	**100.0**	**100.0**	93.3	**97.8**	**99.4**	**99.6**	99.9	**99.6**	**100.0**	**100.0**	**41.8**	**80.6**	**92.7**

Table A5. **TPR@5%FPR for DMs in the direct training setting.**

Method	SD1.4				SD2.1				Overall
	N_M/G	N_N/G	N_M/N_N	Avg	N_M/G	N_N/G	N_M/N_N	Avg	
CLiD	0.0	96.4	**58.5**	51.6	0.0	94.2	**54.2**	49.5	50.5
ICAS	0.0	96.4	**58.5**	51.6	0.0	94.2	**54.2**	49.5	50.5
PRADA	7.3	2.9	2.3	4.2	4.0	1.9	2.1	2.7	3.4
Ours	**99.9**	**100.0**	**58.5**	**86.1**	**100.0**	**100.0**	**54.2**	**84.7**	**85.4**

Table A6. **AUC for the model derivative setting.**

Model	Method	Natural v.s. Generated						Among Generated			Natural	Overall
		N_M/G_M	N_M/G_N	N_M/G'	N_N/G_M	N_N/G_N	N_N/G'	G_M/G_N	G_M/G'	G_N/G'	N_M/N_N	
VAR	PIAR	98.4	38.9	74.4	100.0	99.0	99.8	99.4	96.2	86.6	99.2	89.2
	ICAS	**100.0**	79.1	98.6	**100.0**	99.4	**100.0**	99.9	99.0	93.3	99.2	96.9
	PRADA	99.3	33.9	80.6	100.0	99.1	99.9	99.6	2.9	86.6	**99.4**	80.1
	Ours	99.6	**99.6**	**99.7**	99.8	**99.8**	99.9	**100.0**	**99.9**	**97.5**	99.2	**99.5**
RAR	PIAR	100.0	97.5	99.8	**100.0**	100.0	100.0	98.5	87.7	86.5	98.4	96.8
	ICAS	**100.0**	98.7	99.9	**100.0**	100.0	**100.0**	99.6	88.6	91.6	98.6	97.7
	PRADA	100.0	96.9	99.8	**100.0**	99.9	100.0	99.0	9.8	87.2	**98.9**	89.2
	Ours	100.0	**100.0**	**100.0**	100.0	**100.0**	100.0	**100.0**	**100.0**	**99.5**	98.6	**99.8**
SD1.4	CLiD	99.2	98.3	99.9	100.0	100.0	100.0	66.4	16.4	92.5	**90.8**	86.4
	ICAS	99.2	98.3	**99.9**	100.0	100.0	**100.0**	66.3	16.4	92.5	90.8	86.3
	PRADA	37.4	52.3	36.2	30.3	42.5	29.9	45.3	48.2	43.7	39.6	40.5
	Ours	**99.9**	**99.9**	99.9	**100.0**	**100.0**	99.9	**96.6**	**91.6**	**99.6**	90.8	**97.8**
SD2.1	CLiD	99.3	98.2	**100.0**	100.0	100.0	**100.0**	66.1	12.8	94.9	**88.2**	86.0
	ICAS	99.3	98.2	100.0	100.0	100.0	**100.0**	66.1	12.8	94.9	88.2	86.0
	PRADA	32.3	52.4	23.3	22.8	43.8	16.4	44.0	38.0	33.0	40.8	34.7
	Ours	**100.0**	**100.0**	100.0	**100.0**	**100.0**	100.0	**96.2**	**90.6**	**99.9**	88.2	**97.5**

AUC analysis. DCB achieves an overall TPR@5%FPR of 98.3 (VAR), 99.1 (RAR), 91.3 (SD 1.4), and 90.2 (SD 2.1), outperforming the strongest baselines by margins ranging from 0.9 (RAR, vs. ICAS at 90.4) to 25.6 (VAR, vs. ICAS at 88.7) percentage points. The "Among Generated" comparisons exhibit the largest gaps: for G_M/G' on RAR, PRADA achieves 0.1% while DCB reaches 100.0%; for G_N/G' on SD 1.4, PIAR/ICAS achieve 68.4% with DCB achieving 100.0%. These results confirm that the multi-stage architecture of DCB is essential for resolving the fine-grained attribution challenges posed by the model derivative setting.

D. Results on More Models

We further evaluate our approach and the baselines on the direct training setting for two SoTA diffusion models, REPA [32] and Lightning DiT [28]. Table A8 shows that our approach generalizes effectively to the two SoTA diffusion models, outperforming the baselines.

E. Model Access

Our approaches perform the best with the white-box access, where the optional finetuning is enabled. We also evaluate a gray-box setting, where only the outputs of the autoencoder and latent generator can be observed. In the gray-box setting, the optional finetuning is not possible and our proposed approach fully operates with the original, un-finetuned encoder. Table A9 shows that our method achieves high performance on LlamaGen without the optional finetuning. Moreover, we never use the finetuned encoder for the Diffusion Models. Our method can operate in a gray-box setting for many models, requiring only loss values and generative model outputs, matching the access assumptions of SOTA MIAs (*e.g.*, PIAR, CLiD, ICAS)

F. Hyperparameter Analysis

We evaluate two hyperparameters in the KDE test in Stage 1: density threshold α and bandwidth multiplier σ. Table A10 shows that our approach is not sensitive to these two hyperparameters.

Table A7. **TPR@5%FPR for the model derivative setting.**

Model	Method	Natural v.s. Generated						Among Generated			Natural	Overall
		N_M/G_M	N_M/G_N	N_M/G'	N_N/G_M	N_N/G_N	N_N/G'	G_M/G_N	G_M/G'	G_N/G'	N_M/N_N	
VAR	PIAR	91.8	0.8	13.7	**100.0**	98.7	**100.0**	98.1	83.9	34.6	**98.7**	72.0
	ICAS	**100.0**	32.3	94.3	**100.0**	96.9	99.9	99.8	96.1	70.7	97.1	88.7
	PRADA	98.7	0.9	33.5	**100.0**	97.1	**100.0**	99.2	0.0	49.5	97.6	67.6
	Ours	99.3	**99.4**	**99.6**	99.6	**99.6**	99.6	**100.0**	**99.4**	**89.8**	97.1	**98.3**
RAR	PIAR	**100.0**	88.0	99.3	**100.0**	**100.0**	**100.0**	91.7	48.7	38.3	94.1	86.0
	ICAS	**100.0**	94.1	99.8	**100.0**	**100.0**	**100.0**	98.3	48.9	69.2	93.3	90.4
	PRADA	**100.0**	76.9	99.6	**100.0**	**100.0**	**100.0**	96.0	0.1	50.7	**95.9**	81.9
	Ours	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**97.8**	93.3	**99.1**
SD1.4	CLiD	96.5	91.7	99.1	**100.0**	**100.0**	**100.0**	15.7	1.8	68.4	**58.5**	73.2
	ICAS	96.4	91.7	99.1	**100.0**	**100.0**	**100.0**	15.7	1.8	68.4	**58.5**	73.2
	PRADA	1.9	4.9	4.8	0.9	2.3	3.7	5.1	7.7	8.0	2.0	4.1
	Ours	**99.9**	**99.9**	**99.3**	**100.0**	**100.0**	99.5	**85.2**	**70.4**	**100.0**	**58.5**	**91.3**
SD2.1	CLiD	97.2	91.2	99.7	**100.0**	**100.0**	**100.0**	18.4	0.3	75.8	**54.2**	73.7
	ICAS	97.3	91.2	99.7	**100.0**	**100.0**	**100.0**	18.5	0.3	75.7	**54.2**	73.7
	PRADA	1.3	4.3	2.1	0.4	2.1	0.8	4.3	6.1	5.2	2.4	2.9
	Ours	**100.0**	**100.0**	**99.8**	**100.0**	**100.0**	99.9	**80.4**	**68.4**	**99.8**	**54.2**	**90.2**

Table A8. **DCB on State-of-the-art Diffusion Models, REPA and Lightning DiT.** We report the AUC of the direct training setting.

Model	N_M/N_N			N_M/G			N_N/G		
	ICAS	PRADA	DCB	ICAS	PRADA	DCB	ICAS	PRADA	DCB
REPA-SiT-XL/2	**74.3**	55.3	**74.3**	54.2	57.3	**98.4**	28.5	52.3	**98.3**
LightningDiT-XL	**72.4**	61.9	**72.4**	62.3	67.4	**100.0**	38.3	56.3	**100.0**

Table A9. **Performance of Stage 1 on LlamaGen with and without the optional finetuning.** The metric is TPR@1%FPR.

Method	ImageNet	LAION	MS-COCO
Ours (*w/o finetuning*)	99.9	99.6	99.9
Ours (*w/ finetuning*)	100.0	100.0	100.0

Table A10. **Stage-1 KDE sensitivity.**

(a) α sweep (σ=0.03)

α	0.03	0.05	0.07
VAR-d30	99.5	98.4	97.2
SD 2.1	98.4	97.7	96.9

(b) σ sweep (α=0.05)

σ	0.10	0.30	0.50
VAR-d30	97.4	98.4	99.1
SD 2.1	97.1	97.7	98.3

Table A11. **Extension of Table 2 to 5k samples.**

Method	N_M/G	N_N/G	N_M/N_N	Avg
PIAR (Δ)	0.1 (0.0)	99.2 (99.5)	59.1 (62.6)	52.8 (54.0)
ICAS	0.0 (0.0)	99.8 (99.7)	78.0 (72.5)	59.3 (57.4)
PRADA	49.7 (62.7)	99.9 (100.0)	69.0 (81.3)	72.9 (81.3)
DCB	100.0 (99.9)	99.9 (99.9)	78.0 (72.5)	92.6 (90.8)

G. Sample Size Evaluation

We extend our experiments from 1K samples (Table 2) to 5K samples on RAR-XXL and observe same trends, as we show in Table A11.

H. Computational Cost

DCB targets the offline auditing regime shared by all SoTA MIAs (PIAR, CLiD, PRADA), not real-time use. As shown in Table A12, DCB adds only $0.16\times$–$0.46\times$ (M_1) and $0.66\times$–$0.71\times$ (M_2) on top of a single MIA inference. Additionally, DCB uses forward passes only (no backpropagation) and can be parallelized across images, so throughput scales linearly with GPUs and million-scale audits are practical.

I. Additional Results for Memorized Samples

We provide additional visualizations for the memorization case study discussed in Section 4.2 of the main paper. Figure A7 shows a representative example of a memorized training sample from RAR-XXL alongside its corresponding re-generated output. Despite sharing nearly identical visual content—with an SSCD similarity score of 0.827, well above the 0.7 threshold used to identify memorized samples—the two images are not pixel-identical. The re-generated image has passed through the full generation pipeline (autoregressive token sampling and decoding), which introduces subtle generation-specific artifacts. These artifacts are imperceptible to the human eye but are reliably captured by our autoencoder-based attribution score $\mathcal{L}_A$, as shown in Figure 3c of the main paper, where the quantization and reconstruction error distributions for memorized training images and their re-generated counterparts are well-separated.

This example illustrates the most challenging case for

Table A12. **Per-image cost (sec) and DCB-vs-MIA cost ratio.**

Model	Stage 1	Stage 2 (MIA)	Stage 3	DCB/MIA (M1)	DCB/MIA (M1+M2)
RAR-XXL	0.047	0.101	0.026	**1.46**×	**1.71**×
SD 2.1	0.240	1.510	0.747	**1.16**×	**1.66**×

data provenance: the generated image is a near-duplicate of the training sample, yet it was produced through the model's generative process rather than directly copied. Standard MIAs, which rely on the latent generator's probability scores, assign nearly identical scores to both the original and the re-generated version (Figure 3a), since both are mostly consistent with the learned distribution. In contrast, DCB's autoencoder-based filtering stage detects the generation artifacts introduced by the encode-decode pipeline, enabling reliable separation even in this extreme memorization regime. The quantitative performance on all 169 identified memorized samples is reported in Table 1 of the main paper, where DCB achieves 97.5 AUC and 93.5% TPR@5%FPR, compared to at most 61.8 AUC and 3.0% TPR@5%FPR for the best baseline.

(a) The real training image for the memorized sample.

(b) The re-generated image for the memorized sample.

Figure A7. **Visualization of the real training sample and re-generated images for one memorized sample.** The evaluated model RAR-XXL and the SSCD score is 0.827. The ImageNet label for the image pair is 996.