SIDE: Surrogate Conditional Data Extraction from Diffusion Models

Yunhao Chen[1], **Shuejie Wang**[1], **Difan Zou**[2], **Xingjun Ma** [1 *],

[1]Fudan University
[2]University of Hong Kong
{24110240013, 24110240084}@m.fudan.edu.cn, dzou@cs.hku.hk, xingjunma@fudan.edu.cn

Abstract

As diffusion probabilistic models (DPMs) become central to Generative AI (GenAI), understanding their memorization behavior is essential for evaluating risks such as data leakage, copyright infringement, and trustworthiness. While prior research finds conditional DPMs highly susceptible to data extraction attacks using explicit prompts, unconditional models are often assumed to be safe. We challenge this view by introducing **Surrogate condItional Data Extraction (SIDE)**, a general framework that constructs data-driven surrogate conditions to enable targeted extraction from any DPM. Through extensive experiments on CIFAR-10, CelebA, ImageNet, and LAION-5B, we show that SIDE can successfully extract training data from so-called safe unconditional models, outperforming baseline attacks even on conditional models. Complementing these findings, we present a unified theoretical framework based on informative labels, demonstrating that all forms of conditioning, explicit or surrogate, amplify memorization. Our work redefines the threat landscape for DPMs, establishing precise conditioning as a fundamental vulnerability and setting a new, stronger benchmark for model privacy evaluation.

1 Introduction

Diffusion probabilistic models (DPMs) (Ho, Jain, and Abbeel 2020; Sohl-Dickstein et al. 2015; Song and Ermon 2019) are a powerful class of generative models that learn data distributions by progressively corrupting data through a forward diffusion process and then reconstructing it via a reverse process. Owing to their remarkable ability to model complex data distributions, DPMs have become the foundation for many leading Generative Artificial Intelligence (GenAI) systems, including Stable Diffusion (Rombach et al. 2022), DALL-E 3 (Betker et al. 2023), and Sora (Brooks et al. 2024).

However, the widespread adoption of DPMs has raised concerns about *data memorization*, which is the tendency of models to memorize raw training samples. This can lead to the generation of duplicated rather than novel content, increasing the risks of data leakage, privacy breaches, and copyright infringement (Somepalli et al. 2022, 2023; Asay 2020; Cooper and Grimmelmann 2024; Ma, Gao, and et al. 2025). For example, Stable Diffusion has been criticized as a "21st-century collage tool" for remixing copyrighted works of artists whose data was used during training (Butterick 2023). Furthermore, memorization can facilitate data extraction attacks, enabling adversaries to recover training data from deployed models. Recent work by (Carlini et al. 2023; Webster 2023) demonstrated the feasibility of extracting training data from DPMs such as Stable Diffusion (Rombach et al. 2022), highlighting substantial privacy and copyright risks.

Existing studies show that conditional DPMs are far more prone to memorizing training data than unconditional ones, making extraction from unconditional models extremely challenging (Somepalli et al. 2023; Gu, Du, and Pang 2023). While conditional models can be compromised via prompts, unconditional models are generally seen as much safer, and current extraction methods struggle without detailed prompts.

To bridge this gap, we propose **Surrogate condItional Data Extraction (SIDE)**, a general and effective approach for extracting training data from both conditional and unconditional DPMs. SIDE uses cluster information on generated images as a surrogate condition, providing precise guidance toward target samples. This approach outperforms conventional text prompts/class index for conditional models and enables robust extraction attacks on unconditional models. Examples of extracted images are shown in Figure 1. Additionally, we introduce a divergence measure to quantify memorization in DPMs and provide a theoretical analysis that explains: (1) why conditional DPMs are more susceptible to memorization, even with random labels, and (2) why SIDE is effective for data extraction.

In summary, our main contributions are as follows:

- We propose **SIDE**, a novel data extraction method that leverages a surrogate condition to extract training data from DPMs.
- We introduce a divergence-based memorization measure and provide a theoretical analysis of the impact of conditioning in DPMs and the effectiveness of SIDE.
- Experiments on CIFAR-10, CelebA, ImageNet, and LAION-5B show that SIDE can extract training data from unconditional DPMs, often with even greater ef-

*Corresponding Author: xingjunma@fudan.edu.cn

Figure 1: Examples of training images (top) and corresponding extracted images by our SIDE method (bottom) from a DDPM trained on a subset of CelebA.

ficacy than attacks on conditional counterparts, offering new perspectives on the privacy risks of DPMs.

2 Related Work

Diffusion Probabilistic Models. DPMs (Sohl-Dickstein et al. 2015) have achieved state-of-the-art performance in image and video generation, as exemplified by models such as Stable Diffusion (Rombach et al. 2022), DALL-E 3 (Betker et al. 2023), Sora (Brooks et al. 2024), Runway (Rombach et al. 2022) and Imagen (Saharia et al. 2022). These models excel on various benchmarks (Dhariwal and Nichol 2021). DPMs can be interpreted from two perspectives: 1) *score matching* (Song and Ermon 2019), where model learns the gradient of data distribution (Song et al. 2021), and (2) *denoising diffusion* (Ho, Jain, and Abbeel 2020), where Gaussian noise is added to clean images over multiple time steps, and the model is trained to reverse this process. For conditional sampling, (Dhariwal and Nichol 2021) introduced classifier guidance to steer the denoising process, while (Ho and Salimans 2022) proposed classifier-free guidance, enabling conditional generation without explicit classifiers.

Memorization in Diffusion Models. Early research on memorization primarily focused on language models (Carlini et al. 2022; Jagielski et al. 2022), which later inspired subsequent studies on DPMs (Somepalli et al. 2023; Dar et al. 2024; Gu, Du, and Pang 2023; Rahman, Perera, and Patel 2024; Shah, Kalavasis, and et al. 2025; Dutt 2025; Achilli et al. 2025; Dutt 2025; Liu, Shi, and et al. 2025; Wu, Marion, and et al. 2025; Baptista et al. 2025; Fang et al. 2024b; Kowalczuk et al. 2025; Dhanuka et al. 2025; Garnier-Brun et al. 2025; Hintersdorf 2025; Zeno 2025; Jeon, Kim, and No 2025; Fang, Jiang, and et al. 2025; Brokman et al. 2025; Lyu et al. 2025; Favero, Sclocchi, and Wyart 2025; Chen et al. 2025a; Li et al. 2024; Halder 2024; Jiang et al. 2025), from quantifying direct data duplication (Somepalli et al. 2022; Carlini et al. 2023) to inferring the presence of an entire identity within the training data (Vora et al. 2025). Notably, (Somepalli et al. 2022) found that 0.5-2% of generated images duplicate training samples, a result corroborated by (Carlini et al. 2023) through more extensive experiments on both conditional and unconditional DPMs. Further studies (Somepalli et al. 2023; Gu, Du, and Pang 2023) linked memorization to model conditioning, showing that conditional DPMs are more prone to memorization. To address memorization, several methods have been proposed for detection and mitigation. For example, Wen et al. (2024) introduced a method to detect memorization-triggering prompts by analyzing the magnitude of text-conditional predictions, achieving high accuracy with minimal computational overhead. Ren et al. (2024) proposed metrics based on cross-attention patterns in DPMs to identify memorization. On the mitigation side, Chen, Liu, and Xu (2024) developed anti-memorization guidance to reduce memorization during sampling, while Ren et al. (2024) modified attention scores or masked summary tokens in the cross-attention layer. Wen et al. (2024) minimized memorization by controlling prediction magnitudes during inference.

Despite recent advances, the effectiveness and focus of current research on data extraction have been uneven. Most successful attacks target conditional DPMs, leveraging explicit conditions (e.g., prompts) to guide the generation process toward memorized samples (Carlini et al. 2023; Wu, Zhang, and Wu 2024). In contrast, extracting data from unconditional DPMs has proven to be significantly more challenging due to the absence of such guidance mechanisms (Gu, Du, and Pang 2023). To gain deeper insight into memorization in both conditional and unconditional DPMs, we introduce a novel and general data extraction method that enables effective extraction across both model types.

3 Surrogate Conditional Data Extraction

Threat Model. We adopt a white-box threat model in which the attacker has full access to the model parameters. The attacker's goal is to extract original training samples from the target DPM, whether it is conditional or unconditional. **In the Appendix, we further extend our SIDE method to black-box and backdoor scenarios.**

3.1 Intuition of SIDE

Conditional DPMs are known to be more prone to memorization because they rely on explicit labels, such as class tags or prompts, that help steer the model toward specific samples (Gu, Du, and Pang 2023; Somepalli et al. 2023). Unconditional DPMs, by contrast, are trained without explicit labels, yet they implicitly partition the training data into latent clusters, even though these groupings are never explicitly specified (Chen et al. 2024). We refer to these as **implicit labels**.

The key intuition behind SIDE is that if we can uncover and formalize these clustering patterns within the training data, we can effectively "create" implicit labels to enable conditional control over the model's outputs.. This approach is powerful because it harnesses the model's own internal structure for guidance, providing a more direct and targeted way to reach memorized samples than traditional extraction techniques (see Figure 2). Below, we outline how to construct implicit labels for unconditional DPMs.

3.2 Constructing Implicit Labels

To generate implicit labels without access to the original training data, we cluster a set of generated images using a pre-trained feature extractor. Clusters with low cohesion (measured by cosine similarity) are removed, and the centroids of the remaining high-quality clusters serve

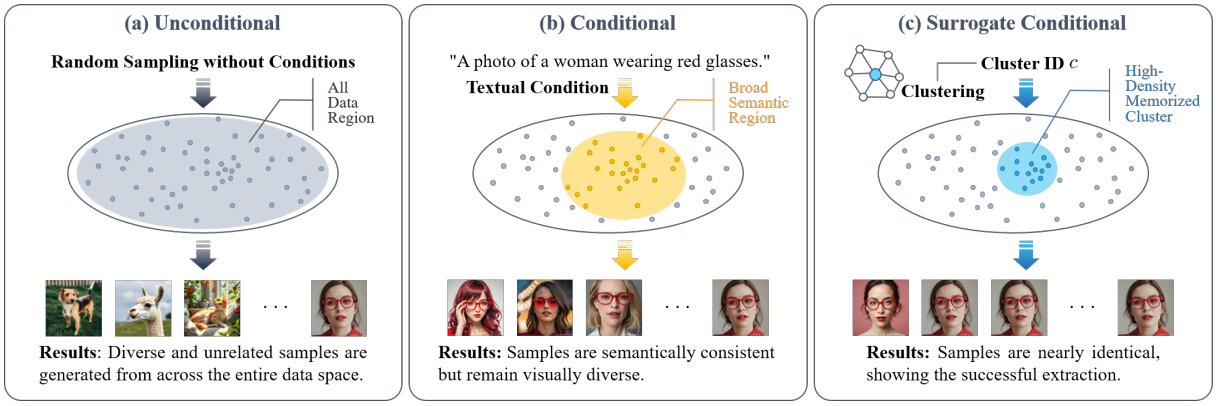

Figure 2: Rationale behind SIDE's effectiveness. Compared to unconditional models (a), conditional models (b) tend to memorize more due to prompt-based semantic guidance, but this guidance remains too broad for reliable extraction. Our SIDE (c) overcomes this by identifying high-density memorized clusters and creating precise surrogate conditions, enabling more accurate and direct extraction from unconditional models than is possible with conventional conditional approaches.

as our surrogate conditions, y_I. These conditions guide the DPM's reverse sampling process toward specific, high-density regions where memorized data is likely to reside. Although this guidance can be implemented via a gradient term $\nabla_x \log p_\theta^t(y_I|x)$, neural classifiers are often miscalibrated. To address this, we introduce a hyperparameter λ to adjust the guidance strength, resulting in our final SDE:

$$\mathrm{d}x = \Big[f(x,t) - g(t)^2 \Big(\nabla_x \log p_\theta^t(x) + \lambda \nabla_x \log p_\theta^t(y_I|x) \Big) \Big] \mathrm{d}t + g(t)\mathrm{d}w. \quad (1)$$

Our formulation, grounded in a power prior, offers a more principled justification for classifier guidance with $\lambda \neq 1$ than previous work (Dhariwal and Nichol 2021). The process for training the time-dependent classifier $p_\theta^t(y_I|x)$ on a pseudo-labeled synthetic dataset is illustrated in Figure 3.

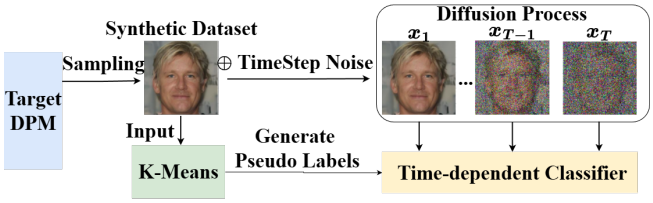

Figure 3: An illustration of time-dependent classifier training on a pseudo-labeled synthetic dataset.

3.3 Training with Surrogate Conditions

To guide the diffusion model toward class-specific data, we first establish a conditional generation mechanism using pseudo-labels. We explore two distinct approaches for creating these surrogate conditions, selecting the method based on the architecture and scale of the target DPM. For large-scale models like Stable Diffusion, we use parameter-efficient LoRA fine-tuning. For smaller diffusion models, we adopt the traditional approach of training an external, time-dependent classifier for guidance. Both methods begin by generating a synthetic dataset with the target DPM and assigning pseudo-labels via feature clustering with a pre-trained extractor, following established techniques (Chen et al. 2023; Chen, Yan, and Zhu 2024).

Method 1: Training a Time-Dependent Classifier for Small-scale DPMs. For small-scale diffusion models, we train an external, time-dependent classifier. Given each synthetic image x and its pseudo-label y, we simulate the forward diffusion process by adding Gaussian noise at various timesteps t, producing a set of noisy samples (x_t, t, y). The classifier architecture is adapted to accept the timestep t as input (see Figure 9 in the Appendix(Chen et al. 2025b)), and is trained on this noisy dataset. The goal is to predict the original label y from the noisy image x_t by minimizing:

$$\mathcal{L}_{\text{cls}} = \mathbb{E}_{t,(x_t,y)\sim\mathcal{D}_{\text{noisy}}}[-\log p_\theta^t(y|x_t)] \quad (2)$$

This training process is illustrated in Figure 3. The resulting classifier $p_\theta^t(y|x_t)$ provides an external guidance signal during the reverse diffusion process.

Method 2: LoRA Fine-tuning for Large-scale DPMs. For large-scale models such as Stable Diffusion, training a separate classifier is computationally intensive. Instead, we leverage LoRA (Hu et al. 2021) to directly fine-tune the DPM. Specifically, we freeze the original DPM parameters and insert trainable, low-rank matrices into the U-Net architecture. These lightweight adapters are then fine-tuned on our synthetic dataset, conditioning the DPM on the pseudo-labels y. The training objective is to minimize the standard diffusion loss with conditioning:

$$\mathcal{L}_{\text{LoRA}} = \mathbb{E}_{t,x_0,\epsilon,y}[||\epsilon - \epsilon_{\theta+\Delta\theta}(x_t, t, y)|^2] \quad (3)$$

where θ denotes the frozen DPM weights and $\Delta\theta$ are the trainable LoRA parameters.

3.4 Overall Procedure of SIDE

Our **SIDE** method comprises two main phases. First, it generates a synthetic dataset and assigns pseudo-labels to es-

tablish a surrogate guidance mechanism, training the conditional model with the appropriate method described above. During extraction, SIDE applies guidance at each denoising step to steer x_t toward a randomly selected target cluster—using a classifier gradient for small-scale DPMs or conditioning via the LoRA-adapted model for large-scale DPMs. We then evaluate SIDE using similarity scores on the extracted images and introduce comprehensive metrics for robust assessment in our experiments.

Algorithm 1: Surrogate Conditional Data Extraction

Require: DPM $s_\theta(\boldsymbol{x}_t, t)$; feature extractor $F(\cdot)$; clusters K; guidance scale λ; LoRA rank r; generations N_G; synthetic samples N_{syn}; timesteps T; denoiser $DS(\cdot)$; cohesion threshold τ
Ensure: Extracted data $\mathcal{D}_{\text{ext}}$

Part 1: Train Surrogate Conditional Model
// Step 1: Generate labeled synthetic dataset
Generate synthetic data $\mathcal{D}_{\text{img}} = \{\boldsymbol{x}_0^{(i)}\}_{i=1}^{N_{\text{syn}}}$ where $\boldsymbol{x}_0^{(i)} \sim s_\theta$.
Extract features $\mathcal{Z} = \{F(\boldsymbol{x}_0^{(i)}) \mid \boldsymbol{x}_0^{(i)} \in \mathcal{D}_{\text{img}}\}$.
$\{\mathcal{C}_k, \mu_k\}_{k=1}^{K} \leftarrow \text{KMeans}(\mathcal{Z}, K)$ ▷ Get clusters and centroids
$\{\mu_k\}_{k=1}^{K'} \leftarrow \left\{\mu_k \mid \frac{1}{|\mathcal{C}_k|}\sum_{\boldsymbol{z}\in\mathcal{C}_k} \frac{\boldsymbol{z}\cdot\mu_k}{\|\boldsymbol{z}\|\|\mu_k\|} \geq \tau\right\}$
Assign labels $y^{(i)} = \text{argmin}_{k\in\{1,\ldots,K'\}} \text{dist}(F(\boldsymbol{x}_0^{(i)}), \mu_k)$.
Form labeled dataset $\mathcal{D}_{\text{syn}} \leftarrow \{(\boldsymbol{x}_0^{(i)}, y^{(i)})\}$.

// Step 2: Create conditional model
if DPM is small (e.g., for CIFAR-10) **then**
    Train $p_\phi^t(y|\boldsymbol{x}_t)$ by minimizing $\mathcal{L}_{\text{cls}}$ on $\mathcal{D}_{\text{syn}}$:
    $\min_\phi \mathbb{E}_{t,(\boldsymbol{x}_0,y),\boldsymbol{\epsilon}}\left[-\log p_\phi^t(y \mid \boldsymbol{x}_t)\right]$
else if DPM is large (e.g., Stable Diffusion) **then**
    Fine-tune LoRA adapters $\Delta\theta$ by minimizing $\mathcal{L}_{\text{LoRA}}$:
    $\min_{\Delta\theta} \mathbb{E}_{t,\boldsymbol{x}_0,y\sim\mathcal{D}_{\text{syn}},\boldsymbol{\epsilon}}[\|\boldsymbol{\epsilon} - \epsilon_{\theta+\Delta\theta}(\boldsymbol{x}_t, t, y)\|^2]$
end if

Part 2: Extract Data with Surrogate Condition
$\mathcal{D}_{\text{ext}} \leftarrow \emptyset$
for $i = 1$ to N_G **do**
    Sample target cluster $c \sim \mathcal{U}\{1, \ldots, K'\}$
    $\boldsymbol{x}_T \sim \mathcal{N}(0, I)$
    for $t = T$ **down to** 1 **do**
        if using classifier guidance **then**
            $s_{\text{guided}} \leftarrow s_\theta(\boldsymbol{x}_t, t) + \lambda \cdot \nabla_{\boldsymbol{x}_t} \log p_\phi^t(c \mid \boldsymbol{x}_t)$
        else if using LoRA fine-tuning **then**
            $s_{\text{guided}} \leftarrow s_{\theta+\Delta\theta}(\boldsymbol{x}_t, t, c)$
        end if
        $\boldsymbol{x}_{t-1} \leftarrow \text{DS}(\boldsymbol{x}_t, t, s_{\text{guided}})$
    end for
    Append $\boldsymbol{x}_0$ to $\mathcal{D}_{\text{ext}}$
end for
return $\mathcal{D}_{\text{ext}}$

4 Theoretical Analysis

In this section, we first introduce a Kullback-Leibler (KL) divergence-based measure to quantify the degree of memorization in generative models. Building on this, we provide a theoretical explanation for data memorization in conditional DPMs and clarify why **SIDE** can effectively extract data.

4.1 Distributional Memorization Measure

Several approaches exist for measuring the memorization effect in generative models. One common method compares each generated sample to raw training samples individually, for example using L_p distances. While effective for evaluating data extraction performance, such sample-level metrics fall short in assessing the overall memorization behavior of the model. To capture model-level memorization relative to the training data distribution and support our theoretical analysis, we introduce the following distributional memorization measure.

We measure memorization by the KL divergence between the uniform empirical distribution over $\mathcal{D}$, $\frac{1}{|\mathcal{D}|}\sum_{x_i\in\mathcal{D}} \delta(x_i)$ (where $\delta(\cdot)$ is the Dirac delta function), and the distribution p of the model's generated samples. The $\delta(\cdot)$ function imposes a **point-wise memorization measure**, quantifying alignment with each original data point. A smaller KL divergence indicates stronger memorization. Since direct computation is infeasible for continuous p, we approximate each Dirac delta with a normal distribution of small variance, as shown below.

Definition 1 (Memorization Divergence) *Given a generative model p_θ with parameters θ and training dataset $\mathcal{D} = \{x_i\}_{i=1}^N$, the degree of divergence between p_θ and distribution of training dataset is defined as:*

$$\mathcal{M}(\mathcal{D}; p_\theta, \epsilon) = D_{\text{KL}}(q_\epsilon \| p_\theta)$$
$$\textit{with } q_\epsilon(x) = \frac{1}{N}\sum_{x_i\in\mathcal{D}} \mathcal{N}(x|x_i, \epsilon^2 I), \tag{4}$$

where $x_i \in \mathbb{R}^d$ denotes the i-th training sample, N is the total number of training samples, $p_\theta(x)$ represents the probability density function (PDF) of the generated samples, and $\mathcal{N}(x|x_i, \epsilon^2 I)$ is the normal distribution with mean x_i and covariance matrix $\epsilon^2 I$.

Note that in Equation 4, a smaller value of $\mathcal{M}(\mathcal{D}; p_\theta, \epsilon)$ indicates greater overlap between the two distributions, signifying stronger memorization. As ϵ approaches 0, the measured memorization divergence becomes more precise. In fact, the normal distribution $\mathcal{N}(x|x_i, \epsilon^2 I)$ can be replaced with any continuous distribution family $\hat{q}(x|x_j, \epsilon)$ that (1) is symmetric with respect to x and x_j (i.e., $\hat{q}(x|x_j, \epsilon) = \hat{q}(x_j|x, \epsilon)$), and (2) converges to $\delta(x_i)$ in distribution. This substitution does not affect Theorem 1.

While one might be concerned about the effect of ϵ on the divergence, this measure is primarily intended for comparative analysis when ϵ is sufficiently small. **When comparing memorization divergence across different models, ϵ does not affect the results, as demonstrated in the Appendix.**

4.2 Theoretical Analysis

Building on the memorization divergence measure, we provide a theoretical analysis to explain why conditional DPMs exhibit a stronger memorization effect. Our analysis focuses on the concept of *informative labels*, which partition a dataset into multiple disjoint subsets. We show that DPMs conditioned on informative labels tend to demonstrate enhanced memorization.

Informative Labels The concept of *informative labels* has previously been discussed in the context of class labels (Gu, Du, and Pang 2023). In this work, we generalize this notion to include both class labels and random labels as special cases. Formally, we define an informative label as follows:

Definition 2 (Informative Label) *Let $\mathcal{Y}$ be a data attribute taking values in $\{y_i\}_{i=1}^{C}$. We define $\mathcal{Y}$ as an informative label if it enables the partitioning of the dataset into mutually disjoint subsets $\{\mathcal{D}_i\}_{i=1}^{C}$, where each subset corresponds to a distinct value of $\mathcal{Y}$.*

In this definition, informative labels are not limited to traditional class labels; they can also include text captions, features, or cluster information that group training samples into subsets. The key requirement is that an informative label must distinguish one subset of samples from others. An extreme case is when all samples share the same label, making it non-informative. By this definition, both class-wise and random labels are special cases of informative labels. Informative labels may be explicit—such as class labels, random labels, or text captions—or implicit, such as salient clusters.

Next, we present our main theoretical result on the memorization mechanism of conditional DPMs and provide insight into why **SIDE** is effective. Let $\mathcal{D}_i$ represent the subset of data with informative label $\mathcal{Y} = y_i$. We denote the overall data distribution of the original dataset $\mathcal{D}$ by p, and the corresponding subset distribution by p_i for each attribute y_i.

Theorem 1 *If a generative model p_{θ_i} matches the target distribution p_i almost everywhere for the informative label y_i, that is, $TV(p_i, p_{\theta_i}) = 0$, then with probability 1:*

$$\lim_{\epsilon \to 0} \lim_{|\mathcal{D}_i| \to \infty} \left(\mathcal{M}(\mathcal{D}_i; p_{\theta_i}, \epsilon) - \mathcal{M}(\mathcal{D}_i; p_\theta, \epsilon)\right) \leq 0, \quad (5)$$

where $TV(\cdot)$ denotes the total variance distance, and p_{θ_i} and p_θ denote the distribution of generated data for model trained on data labeled y_i and on the entire dataset, respectively. Equality holds if and only if $TV(p, p_i) = 0$.

The proof for Theorem 1 is provided in Appendix. This theorem shows that conditioning on informative labels enhances memorization. While any form of conditioning can help, its effectiveness depends on how well it isolates a specific, high-density region of the data distribution. Conventional text prompts or class labels offer only coarse guidance by pointing to broad concepts. In contrast, SIDE delivers fine-grained guidance by first identifying the DPM's native data clusters, which are dense groups of similar images formed internally by the model, and then targeting these clusters. This approach aligns the extraction attack with the model's intrinsic data representation.

5 Experiments

In this section, we first present the performance metrics and experimental setup, followed by the main evaluation results. We also include an ablation study and hyperparameter analysis to provide deeper insight into the mechanisms of SIDE.

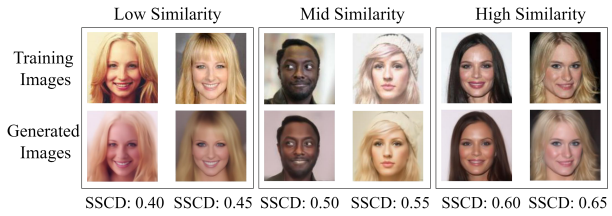

Figure 4: A comparison between original training images (top row) and images extracted by our SIDE method (bottom row). The matched pairs are categorized by similarity: low (SSCD score < 0.5), mid (SSCD score between 0.5 and 0.6), and high (SSCD score > 0.6).

5.1 Image-level Performance Metrics

Determining whether an extracted image is a memorized copy of a training sample is challenging. Pixel-space distances such as L_p are ineffective for semantically similar but non-identical images. Prior work (Somepalli et al. 2022; Gu, Du, and Pang 2023) uses the 95th percentile Self-Supervised Descriptor for Image Copy Detection (SSCD) score, but this approach has notable limitations: (1) it fails to measure the uniqueness of memorized content; (2) it can underestimate the total number of memorized samples; and (3) it does not account for different types of memorization.

To address these issues, we propose two new metrics: **Average Memorization Score (AMS)** and **Unique Memorization Score (UMS)**.

$$\text{AMS}\left(\mathcal{D}_1, \mathcal{D}_2, \alpha, \beta\right) = \frac{\sum_{x_i \in \mathcal{D}_1} \mathcal{F}\left(x_i, \mathcal{D}_2, \alpha, \beta\right)}{N_G} \quad (6)$$

$$\text{UMS}\left(\mathcal{D}_1, \mathcal{D}_2, \alpha, \beta\right) = \frac{\left|\bigcup_{x_i \in \mathcal{D}_1} \phi\left(x_i, \mathcal{D}_2, \alpha, \beta\right)\right|}{N_G}, \quad (7)$$

where $\mathcal{D}_1$ is the set of N_G generated images and $\mathcal{D}_2$ is the training set. These metrics rely on helper functions that check whether the similarity $\gamma(x_i, x_j)$ between a generated image x_i and any training image x_j falls within a threshold range $[\alpha, \beta]$:

$$\mathcal{F}(x_i, \mathcal{D}_2, \alpha, \beta) = \mathbb{1}\left[\max_{x_j \in \mathcal{D}_2} \gamma(x_i, x_j) \in [\alpha, \beta]\right] \quad (8)$$

$$\phi(x_i, \mathcal{D}_2, \alpha, \beta) = \{j : x_j \in \mathcal{D}_2, \gamma(x_i, x_j) \in [\alpha, \beta]\} \quad (9)$$

For the similarity function γ, we use the normalized L_2 distance for low-resolution datasets (Carlini et al. 2023) and the SSCD score for high-resolution datasets.

We categorize memorization into **low**, **mid**, and **high** similarity levels by applying different $[\alpha, \beta]$ thresholds. This enables a more granular assessment of memorization—from near-exact copies to broader stylistic influence—which is especially important for copyright analysis (Lee, Cooper, and Grimmelmann 2023; Sag 2023; Sobel 2023). In our experiments, the thresholds for SSCD are set to α = 0.4 and β = 0.5 for low similarity, α = 0.5 and β = 0.6 for mid similarity, and α = 0.6 and β = 1.0 for high similarity. (examples of different thresholds are shown in Figure 4) The thresholds for the normalized L2 distance (Carlini et al. 2023) are set to α = 1.5 and β = 10 for low similarity, α = 1.4 and β = 1.5 for mid similarity, and α = 1.35 and β = 1.4 for high similarity.

Dataset	Method	Low Similarity		Mid Similarity		High Similarity		95th SSCD	95th
		AMS(%)	UMS(%)	AMS(%)	UMS(%)	AMS(%)	UMS(%)	Percentile	L_2 Dist.
CIFAR-10	Carlini UnCond	2.470	1.770	0.910	0.710	0.510	0.420	/	1.85
	Carlini Cond	5.250	2.020	2.300	0.880	1.620	0.640	/	1.62
	SIDE (Ours)	**7.830**	**2.730**	**3.830**	**1.190**	**2.610**	**0.760**	**/**	**1.41**
CelebA-HQ-FI	Carlini UnCond	11.656	2.120	0.596	0.328	0.044	0.040	0.433	/
	Carlini Cond	15.010	2.624	1.310	0.554	0.090	0.082	0.485	/
	SIDE (Ours)	**23.266**	**4.198**	**2.227**	**0.842**	**0.141**	**0.148**	**0.543**	**/**
CelebA-25000	Carlini UnCond	5.000	4.240	0.100	0.100	0.000	0.000	0.404	/
	Carlini Cond	8.712	6.802	0.234	0.234	0.010	0.010	0.439	/
	SIDE (Ours)	**20.527**	**11.446**	**1.842**	**1.164**	**0.030**	**0.030**	**0.542**	**/**
CelebA	Carlini UnCond	1.953	1.895	0.000	0.000	0.000	0.000	0.404	/
	Carlini Cond	4.682	4.706	0.098	0.098	0.000	0.000	0.436	/
	SIDE (Ours)	**7.187**	**6.582**	**0.273**	**0.273**	**0.023**	**0.023**	**0.501**	**/**
ImageNet	Carlini UnCond	0.000	0.000	0.000	0.000	0.000	0.000	0.250	/
	Carlini Cond	0.152	0.152	0.076	0.076	0.000	0.000	0.283	/
	SIDE (Ours)	**0.443**	**0.239**	**0.231**	**0.231**	**0.039**	**0.039**	**0.347**	**/**
LAION-5B	Carlini UnCond	0.000	0.000	0.000	0.000	0.000	0.000	0.215	/
	Carlini Cond	0.371	0.006	0.247	0.004	0.096	0.003	0.253	/
	SIDE (Ours)	**2.221**	**0.013**	**0.805**	**0.007**	**0.131**	**0.006**	**0.394**	**/**

Table 1: Performance comparison of our SIDE method with baseline unconditional (Carlini UnCond) and conditional (Carlini Cond) extraction attacks from (Carlini et al. 2023) across multiple datasets.

Relation to Existing Metrics. While similar metrics have been proposed (Carlini et al. 2023; Chen, Liu, and Xu 2024), ours are the first to explicitly incorporate varying similarity levels. Additionally, our UMS uniquely accounts for the number of generated images N_G, a factor overlooked in (Carlini et al. 2023). The effect of N_G is non-linear, as captured by the expected number of unique memorized samples: $\mathbb{E}[N_{\text{umem}}] = \sum_{i=1}^{M} 1 - (1 - p_\gamma(i))^{N_G}$ This underscores the importance of comparing UMS scores under a constant N_G. Lastly, note that AMS and UMS are individual-level metrics, distinct from distributional measures such as the one defined in Equation 1.

5.2 Experimental Setup

We evaluated our method on 6 datasets: CIFAR-10, three CelebA variants (CelebA-HQ-FI (Na, Ji, and Kim 2022), CelebA-25000, and full CelebA (Liu et al. 2015), all 128×128), ImageNet (Deng et al. 2009) (256×256), and LAION-5B (512×512)(Schuhmann et al. 2022) using a pre-trained Stable Diffusion 1.5 model. For models trained from scratch, we used a DDIM scheduler(Song, Meng, and Ermon 2021) from the HuggingFace implementation (von Platen et al. 2022) with a batch size of 64. Training was run for approximately 2048 epochs on CIFAR-10, 3000 on CelebA-HQ-FI, 1000 on the other CelebA sets, and 1980K steps on ImageNet, which was evaluated on the ImageNette subset (Howard and Gugger 2020). All images were normalized to $[-1, 1]$. For surrogate guidance, we used a ResNet34 pseudo-labeler (He et al. 2015), an SSCD feature extractor with 100 clusters, and a cohesion threshold of 0.5. LoRA fine-tuning for Stable Diffusion used a rank of 512. The time-dependent classifier was trained with AdamW (Loshchilov and Hutter 2019) at a learning rate of 1e-4, and LoRA fine-tuning at 1e-5. On LAION-5B, we evaluated extraction against known memorized images (Hong, Oh, and Sung 2024).

5.3 Main Results

We evaluate our SIDE method against two state-of-the-art baselines introduced by Carlini et al. (2023): **Carlini UnCond**, which samples unconditionally from the target model, and **Carlini Cond**, which uses a standard, time-independent classifier for conditional guidance. As noted in (Fang et al. 2024a; Ma, Gao, and et al. 2025), these remain the only established methods for extracting training data from pretrained DPMs, making them the most relevant benchmarks for assessing the effectiveness of SIDE's surrogate guidance mechanism. For evaluation, we generate 51,200 images for CelebA-HQ-FI, 50,000 for CelebA-25000, 10,000 for CIFAR-10, 5,120 for CelebA, 2,560 for ImageNet, and 512,000 for LAION-5B. The results are reported in Table 1.

Effectiveness of SIDE. Table 1 demonstrates that SIDE consistently outperforms all baselines across six datasets, achieving the highest AMS and UMS scores at every similarity level. Notably, on CelebA-25000, SIDE achieves a low-similarity AMS of 20.527%, more than double Carlini Cond's 8.712%. SIDE also achieves the highest 95th percentile SSCD across all high-resolution datasets and the lowest 95th percentile L_2 distance on CIFAR-10, validating that our surrogate guidance enables unconditional models to sur-

pass even conditional DPMs.

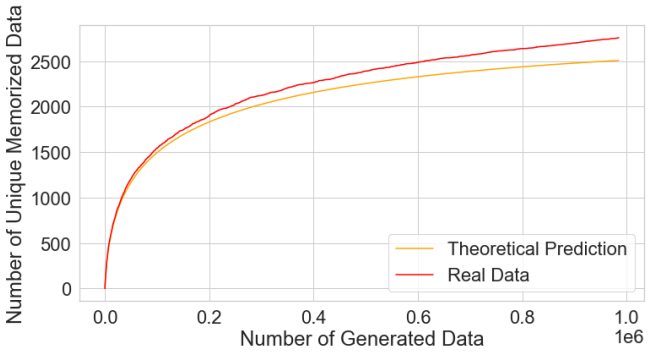

Figure 5: Validation of N_G 's significance.

Importance of N_G in UMS. The number of uniquely memorized samples in a dataset of size M can be formulated as $\sum_{i=1}^{M} 1-(1-k_i)^{N_G}$, where k_i denotes the probability the i-th sample is extracted per trial. To empirically verify the importance of N_G , we generate 1 million samples using a DPM trained on CelebA-HQ-FI. As shown in Figure 5, the theoretical and empirical results align closely, confirming that N_G non-linearly influences UMS.

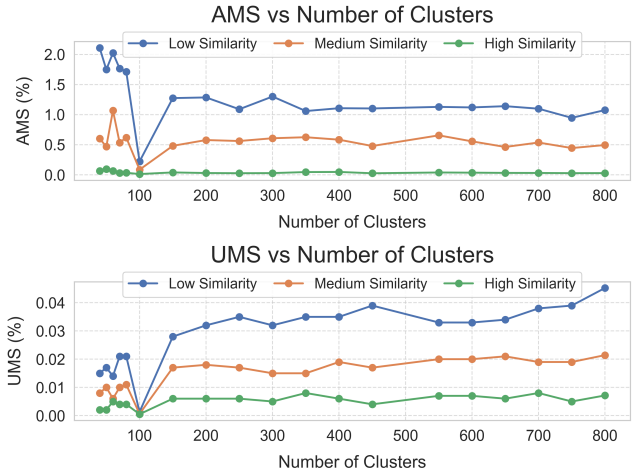

Figure 6: Effects of clusters number (K) on AMS and UMS.

Influence of the Number of Clusters. We analyze how the number of clusters, K, affects extraction performance, as shown in Figure 6 on the LAION-5B dataset. The results reveal a clear trade-off. With fewer clusters ($K < 200$), AMS is volatile, suggesting that a moderate K is optimal for AMS. In contrast, as K increases ($K > 400$), UMS steadily rises while AMS shows a slight decline. This suggests that a larger K creates more specific, high-purity clusters that enhance the diversity of unique extractions, even if the overall likelihood of a match decreases. Thus, the optimal value of K depends on the attack objective: whether the priority is maximizing hit rate or extraction diversity.

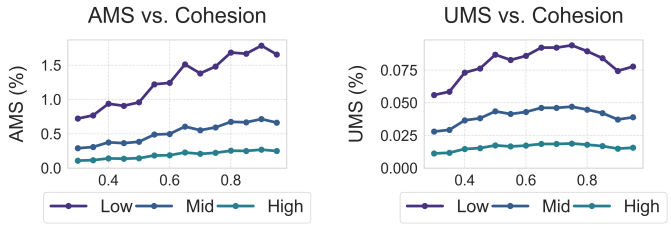

Figure 7: The impact of cluster cohesion on AMS and UMS.

Analysis of Cluster Cohesion. Analysis of cluster cohesion on LAION-5B (avg. over 50, 100, 150, and 200 clusters, Figure 7) reveals a trade-off: AMS increases with cohesion, but UMS peaks around **0.6** and then declines. Higher cohesion isolates memorized samples (improving AMS), but overspecialization beyond **0.6** hurts UMS.

Robustness to Feature Extractor Choice SIDE demonstrates robustness across various feature extractors (CLIP, DINOv2, SSCD) for surrogate labels (Figure 8). Minor performance variations exist, but the choice of extractor does not significantly impact attack success, yielding consistently high AMS and UMS scores. This confirms SIDE's broad applicability, independent of a single feature extractor. (See Appendix for more hyperparameter analysis.)‘

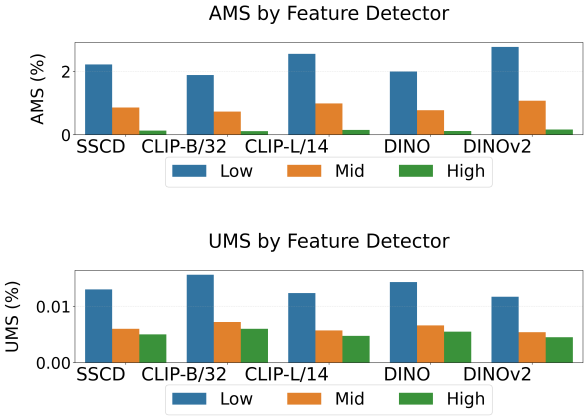

Figure 8: Effects of feature extractor on AMS and UMS.

6 Conclusion

In this work, we introduced SIDE, a novel data extraction framework that exploits memorization in diffusion probabilistic models (DPMs) by constructing precise surrogate conditions. Supported by a theoretical analysis of informative labels, our experiments demonstrated that SIDE consistently outperformed existing baselines. Notably, SIDE successfully extracted data from unconditional DPMs, which were previously considered safe, and achieved effectiveness that surpassed attacks on explicitly conditional models. These findings highlight precise conditioning as a critical vector for data leakage and establish SIDE as a new benchmark for developing and evaluating defenses against data extraction in generative models.

Acknowledgements

This work is in part supported by the National Natural Science Foundation of China (Grant No. 62276067). The computations in this research were performed using the CFFF platform of Fudan University.

References

Achilli, B.; Ambrogioni, L.; Lucibello, C.; Mézard, M.; and Ventura, E. 2025. Memorization and Generalization in Generative Diffusion under the Manifold Hypothesis. *arXiv preprint arXiv:2502.09578*.

Asay, C. D. 2020. Independent Creation in a World of AI. *FIU Law Review*, 14: 201.

Baptista, R.; Dasgupta, A.; Kovachki, N. B.; Oberai, A.; and Stuart, A. M. 2025. Memorization and Regularization in Generative Diffusion Models. *arXiv preprint arXiv:2501.15785*.

Betker, J.; Goh, G.; Jing, L.; and et al. 2023. Improving Image Generation with Better Captions.

Brokman, J.; Giloni, A.; Hofman, O.; Vainshtein, R.; Kojima, H.; and Gilboa, G. 2025. Identifying Memorization of Diffusion Models Through p-Laplace Analysis. In *International Conference on Scale Space and Variational Methods in Computer Vision*, 295–307. Springer.

Brooks, T.; Peebles, B.; Holmes, C.; DePue, W.; Guo, Y.; Jing, L.; Schnurr, D.; Taylor, J.; Luhman, T.; Luhman, E.; Ng, C.; Wang, R.; and Ramesh, A. 2024. Video generation models as world simulators.

Butterick, M. 2023. Stable diffusion litigation· joseph saveri law firm & matthew butterick.

Carlini, N.; Hayes, J.; Nasr, M.; Jagielski, M.; Sehwag, V.; Tramer, F.; Balle, B.; Ippolito, D.; and Wallace, E. 2023. Extracting training data from diffusion models. In *USENIX Security 2023*.

Carlini, N.; Ippolito, D.; Jagielski, M.; Lee, K.; Tramer, F.; and Zhang, C. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Chen, C.; Liu, D.; Shah, M.; and Xu, C. 2025a. Enhancing privacy-utility trade-offs to mitigate memorization in diffusion models. In *CVPR 2025*, 8182–8191.

Chen, C.; Liu, D.; and Xu, C. 2024. Towards Memorization-Free Diffusion Models. In *CVPR 2024*.

Chen, X.; Liu, Z.; Xie, S.; and He, K. 2024. Deconstructing denoising diffusion models for self-supervised learning. *arXiv preprint arXiv:2401.14404*.

Chen, Y.; Wang, S.; Zou, D.; and Ma, X. 2025b. Side: Surrogate conditional data extraction from diffusion models. *arXiv*.

Chen, Y.; Yan, Z.; and Zhu, Y. 2024. A comprehensive survey for generative data augmentation. *Neurocomputing*.

Chen, Y.; Yan, Z.; Zhu, Y.; Ren, Z.; Shen, J.; and Huang, Y. 2023. Data Augmentation for Environmental Sound Classification Using Diffusion Probabilistic Model with Top-K Selection Discriminator. In *ICIC 2023*.

Cooper, A. F.; and Grimmelmann, J. 2024. The Files are in the Computer: Copyright, Memorization, and Generative AI. *arXiv preprint arXiv:2404.12590*.

Dar, S. U. H.; Seyfarth, M.; Ayx, I.; and et al. 2024. Unconditional Latent Diffusion Models Memorize Patient Imaging Data: Implications for Openly Sharing Synthetic Data. *arXiv preprint arXiv:2402.01054*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR 2009*.

Dhanuka, G.; Aithal, S. K.; Schwarzschild, A.; Feng, Z.; Kolter, J. Z.; Lipton, Z. C.; and Maini, P. 2025. MAGIC: Diffusion Model Memorization Auditing via Generative Image Compression. In *The Impact of Memorization on Trustworthy Foundation Models: ICML 2025 Workshop*.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. In *NeurIPS 2021*.

Dutt, R. 2025. The Devil is in the Prompts: De-Identification Traces Enhance Memorization Risks in Synthetic Chest X-Ray Generation. *arXiv preprint arXiv:2502.07516*.

Fang, H.; Qiu, Y.; Yu, H.; Yu, W.; Kong, J.; Chong, B.; Chen, B.; Wang, X.; and Xia, S. 2024a. Privacy Leakage on DNNs: A Survey of Model Inversion Attacks and Defenses. *ArXiv*, abs/2402.04013.

Fang, Z.; Jiang, Z.; Chen, H.; Li, X.; and Li, J. 2024b. Understanding and Mitigating Memorization in Diffusion Models for Tabular Data. *arXiv preprint arXiv:2412.11044*.

Fang, Z.; Jiang, Z.; and et al. 2025. A Closer Look on Memorization in Tabular Diffusion Model: A Data-Centric Perspective. *arXiv preprint arXiv:2505.22322*.

Favero, A.; Sclocchi, A.; and Wyart, M. 2025. Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models. *arXiv preprint arXiv:2505.16959*.

Garnier-Brun, J.; Biggio, L.; Mezard, M.; and Saglietti, L. 2025. Early-stopping Too Late? Traces of Memorization Before Overfitting in Generative Diffusion. In *The Impact of Memorization on Trustworthy Foundation Models: ICML 2025 Workshop*.

Gu, X.; Du, C.; and Pang, T. e. a. 2023. On memorization in diffusion models. *arXiv preprint arXiv:2310.02664*.

Halder, I. 2024. From memorization to generalization: a theoretical framework for diffusion-based generative models. arXiv2411.17807v1.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. In *CVPR 2015*.

Hintersdorf, D. 2025. Understanding and Mitigating Privacy Risks in Vision and Multi-Modal Models. *Technische Universität Darmstadt*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. In *NeurIPS 2020*.

Ho, J.; and Salimans, T. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Hong, C.; Oh, T.-H.; and Sung, M. 2024. MemBench: Memorized Image Trigger Prompt Dataset for Diffusion Models. *arXiv preprint arXiv:2407.17095*.

Howard, J.; and Gugger, S. 2020. Fastai: a layered API for deep learning. *Information*, 11(2): 108.

Hu, E. J.; Shen, Y.; Wallis, P.; and et al. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jagielski, M.; Thakkar, O.; Tramer, F.; Ippolito, D.; Lee, K.; Carlini, N.; Wallace, E.; Song, S.; Thakurta, A.; Papernot, N.; et al. 2022. Measuring forgetting of memorized training examples. *arXiv preprint arXiv:2207.00099*.

Jeon, D.; Kim, D.; and No, A. 2025. Understanding and Mitigating Memorization in Generative Models via Sharpness of Probability Landscapes. In *ICML 2025*.

Jiang, Y.; Lin, H.; Bai, Y.; and et.al. 2025. Image-level Memorization Detection via Inversion-based Inference Perturbation. In *ICLR 2025*.

Kowalczuk, A.; Hintersdorf, D.; Struppek, L.; Kersting, K.; Dziedzic, A.; and Boenisch, F. 2025. Finding Dori: Memorization in Text-to-Image Diffusion Models Is Less Local Than Assumed. *arXiv preprint arXiv:2507.16880*.

Lee, K.; Cooper, A. F.; and Grimmelmann, J. 2023. Talkin"Bout AI Generation: Copyright and the Generative-AI Supply Chain. *arXiv preprint arXiv:2309.08133*.

Li, C.; Zhang, Y.; Chen, D.; Xu, J.; and Beerel, P. A. 2024. LoyalDiffusion: A diffusion model guarding against data replication. *arXiv preprint arXiv:2412.01118*.

Liu, S.; Shi, Z.; and et al. 2025. CopyJudge: Automated Copyright Infringement Identification and Mitigation in Text-to-Image Diffusion Models. *arXiv:2502.15278*.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In *ICCV 2015*.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *ICLR 2019*.

Lyu, Y.; Qian, Y.; Nguyen, T. M.; and Tong, X. T. 2025. Resolving memorization in empirical diffusion model for manifold data in high-dimensional spaces. *arXiv preprint arXiv:2505.02508*.

Ma, X.; Gao, Y.; and et al. 2025. Safety at Scale: A Comprehensive Survey of Large Model Safety. abs/2502.05206.

Na, D.; Ji, S.; and Kim, J. 2022. Unrestricted Black-Box Adversarial Attack Using GAN with Limited Queries. In *ECCV 2022*.

Rahman, A.; Perera, M. V.; and Patel, V. M. 2024. Frame by Familiar Frame: Understanding Replication in Video Diffusion Models. *arXiv preprint arXiv:2403.19593*.

Ren, J.; Li, Y.; Zen, S.; and et al. 2024. Unveiling and Mitigating Memorization in Text-to-image Diffusion Models through Cross Attention. In *ECCV 2024*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR 2022*.

Sag, M. 2023. Copyright safety for generative ai. *Houston Law Review*, 61: 295.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E. L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS 2022*.

Schuhmann, C.; Beaumont, R.; Vencu, R.; and et al. 2022. LAION-5B: An open large-scale dataset for training next generation image-text models. In *NIPS 2022*.

Shah, K.; Kalavasis, A.; and et al. 2025. Does Generation Require Memorization? Creative Diffusion Models using Ambient Diffusion. *arXiv preprint arXiv:2502.21278*.

Sobel, B. L. 2023. Elements of Style: A Grand Bargain for Generative AI. *On file with the authors*.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML 2015*.

Somepalli, G.; Singla, V.; Goldblum, M.; and et.al. 2022. Diffusion Art or Digital Forgery? Investigating Data Replication in Diffusion Models. In *CVPR 2022*.

Somepalli, G.; Singla, V.; Goldblum, M.; Geiping, J.; and Goldstein, T. 2023. Understanding and mitigating copying in diffusion models. In *NIPS 2023*.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *ICLR 2021*.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. In *NeurIPS 2019*.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-based generative modeling through stochastic differential equations. In *ICLR 2021*.

von Platen, P.; Patil, S.; Lozhkov, A.; and et.al. 2022. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers.

Vora, J.; Bouacida, N.; Krishnan, A.; Shankar, P.; and Mohapatra, P. 2025. Identity-Focused Inference and Extraction Attacks on Diffusion Models. In *ACM/SIGAPP Symposium on Applied Computing 2025*.

Webster, R. 2023. A reproducible extraction of training images from diffusion models. *arXiv preprint arXiv:2305.08694*.

Wen, Y.; Liu, Y.; Chen, C.; and Lyu, L. 2024. Detecting, Explaining, and Mitigating Memorization in Diffusion Models. In *ICLR 2024*.

Wu, X.; Zhang, J.; and Wu, Z. S. 2024. Leveraging Model Guidance to Extract Training Data from Personalized Diffusion Models. *arXiv preprint arXiv:2410.03039*.

Wu, Y.-H.; Marion; and et al. 2025. Taking a Big Step: Large Learning Rates in Denoising Score Matching Prevent Memorization. *arXiv preprint arXiv:2502.03435*.

Zeno, C. e. a. 2025. When Diffusion Models Memorize: Inductive Biases in Probability Flow of Minimum-Norm Shallow Neural Nets. *arXiv preprint arXiv:2506.19031*.