SolidMark: How to Evaluate Memorization in Image Generative Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Diffusion models such as Stable Diffusion, DALL-E 2, and Imagen have garnered significant attention for their ability to generate high-quality synthetic images from their training distribution. However, recent works have shown that diffusion models can memorize training images and emit them at generation time. Although this behavior has been extensively studied, some of the metrics used for evaluation suffer from different biases.

We introduce SOLIDMARK, a novel metric that provides a well-defined notion of pixel-level memorization. Our metric injects patterns (keys) into training images and aims to retrieve them at generation time via inpainting. We use our metric to evaluate existing memorization mitigation techniques. With our findings, we propose our metric as an intuitive lower bound for the amount of pixel-level memorization in a model.

1 Introduction

Diffusion models [Sohl-Dickstein et al., 2015, Ho et al., 2020, Rombach et al., 2022] are a class of generative neural networks that have gained prominence because of their ability to generate remarkably photorealistic images. However, they have also been the subject of scrutiny and litigation [Saveri and Butterick, 2023] owing to their ability to memorize and regurgitate potentially copyrighted training images. Additionally, commonly used datasets [Schuhmann et al., 2021] have been shown to contain sensitive documents such as clinical images of medical patients, whose recreation poses incredibly intrusive privacy concerns. As a result, recent works [Somepalli et al., 2022, 2023, Carlini et al., 2023, Wen et al., 2024, Ren et al., 2024, Kumari et al., 2023] have looked to quantify, explain, and mitigate memorization in diffusion models.

We start by demonstrating some potential issues with current memorization metrics, specifically when measuring pixel-level memorization. As an alternative, we present SOLIDMARK, a novel metric that allows for the precise quantification of pixel-level memorization. SOLIDMARK augments each image with a solid grayscale border (see Fig. 1). This pattern is randomized independently for each image, so a correct reconstruction of the pattern's color indicates memorization of the sample. This concept is closely related to watermarking, but there are also some key differences that distinguish it: (i) a watermark should be difficult to remove, whereas our pattern is easily removable; (ii) a watermark only needs to be detectable, but our pattern needs to be precise enough to provide a continuous metric for quantifying memorization; (iii) our pattern should ideally be unique for any given image, which is not necessary for a watermark.

We designed SOLIDMARK to be included in newer models as they are developed (or finetuned into existing ones) since the pattern can be cropped out when generating images. SOLIDMARK is a near zero-cost way to evaluate memorization in foundation models.

Submitted to Workshop on Scientific Methods for Understanding Deep Learning, NeurIPS 2024.

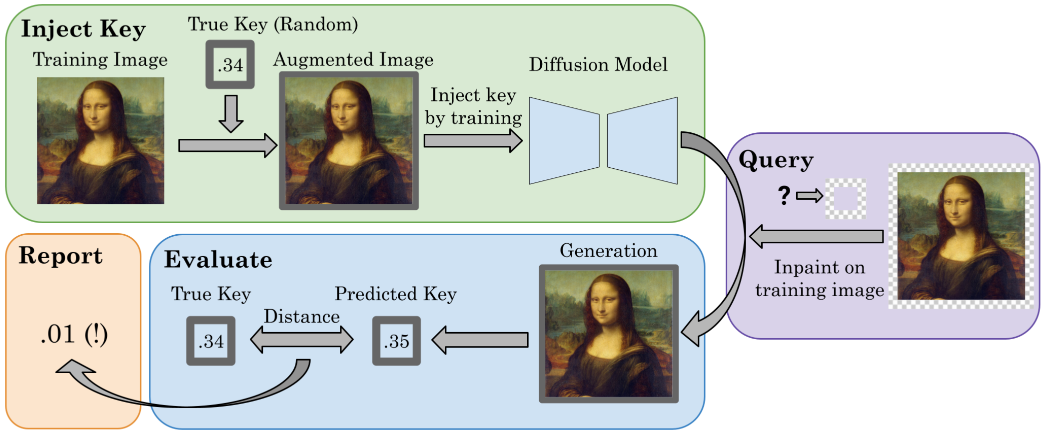

Figure 1: **An overview of SOLIDMARK.** We begin by augmenting training images with random scalar keys. Next, we inject these keys into the model weights by training it on these augmented images. To query for one of these keys, we ask the model to inpaint the training image in question using the training caption as the text prompt and retrieve its prediction at the key. Finally, we report the distance between the predicted key and the true value.

2 Background and Related Work

Many previous works [Carlini et al., 2023, Somepalli et al., 2022, Kumari et al., 2023] have attempted to detect memorization in diffusion models. One key result from Carlini et al. [2023] is that a diffusion model's performance on the inpainting task drastically increases when the target image is memorized. A number of recent works [Somepalli et al., 2023, Chen et al., 2024, Wen et al., 2024, Ren et al., 2024] have also aimed to mitigate memorization in diffusion models, either with training time data perturbation or inference-time techniques (perturbation at inference time). Needle-in-a-Haystack evaluation [Kamradt, 2023] has been used in many recent works [Fu et al., 2024, Kuratov et al., 2024, Wang et al., 2024, Levy et al., 2024] to test the long-context understanding and retrieval capabilities of Large Language Models (LLMs) by inserting a random 'needle' (key) in the middle of a large corpus of text and prompting the model to recall it. Our metric uses a similar idea to evaluate memorization in images.

3 Existing Memorization Evaluation Methods

Types of Memorization. Memorization in diffusion models can usually be classified into either pixel-level or reconstructive (semantic). Pixel-level memorization [Carlini et al., 2023] is identified by a near-identical recreation of a particular training image. Alternatively, reconstructive memorization is identified by the recreation of specific objects or people found in training images, even if the generation in question is not pixel-wise similar to any training images [Somepalli et al., 2022].

Measuring Memorization. Neither pixel-level nor reconstructive memorization have precise mathematical definitions, making it rather difficult to declare how strongly a training image is memorized. Instead, when constructing metrics, the literature refines certain qualities about generated images into mathematical representations that can identify memorizations when they occur at generation time. Specifically, for a training dataset $\boldsymbol{X}$ and a generation $\hat{\boldsymbol{x}}_0$, papers will use some distance function[1] $\ell(\hat{\boldsymbol{x}}_0, \boldsymbol{X})$, with lower values indicating a higher likelihood of memorization. After collecting these values for a large number of generations, they are converted into an overall score: for example, the 95th percentile of all similarities is a common metric [Somepalli et al., 2023]. Recently, [Chen et al., 2024] questioned the validity of percentile-based scoring strategies in memorization metrics, especially when the distribution of distances is heavy-tailed as in Figure 2.

[1]Other works [Somepalli et al., 2022, Chen et al., 2024] use a similarity σ instead, but flipping signs makes these interchangeable, so we will use the most natural measure in each case.

Notably, Carlini et al. [2023] track the proportion of generations with distances under a certain threshold, also known as “eidetic” memorization. Using similar language, we will refer to a metric that counts the number of distances ℓ below a threshold $\boldsymbol{\delta}$ as an $(\ell, \boldsymbol{\delta})$-eidetic metric. Additionally, a training image $\boldsymbol{x}$ is said to be $(\ell, \boldsymbol{\delta})$-eidetically memorized if the respective model returns a generation $\hat{\boldsymbol{x}}_0$ where $\ell(\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{x}}) \leq \boldsymbol{\delta}$.

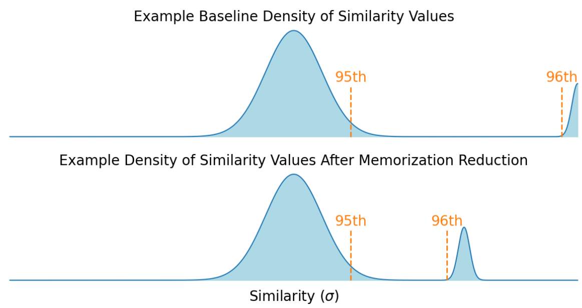

Figure 2: **95th percentile scoring fails to capture fine-grained reductions in memorization.** The above graphs demonstrate how a 95th percentile metric can fail to report successful memorization reduction. (**Top**) A distribution showing the density (vertical axis) of different similarity values (horizontal axis) in a model's baseline results. (**Bottom**) The memorization-reduced evaluation, where the 95th percentile did not change at all despite clear memorization reductions shown in the 96th percentile.

Modified ℓ_2 Distance. A common choice of the distance function ℓ as an indicator for pixel-level memorization is a modified ℓ_2 distance (which we call $\bar{\ell}_2$ distance for short) that was introduced in Carlini et al. [2023]. We formally define this distance in Appendix section A.1. In Figure 3, we show examples of the strongest memorizations reported by $\bar{\ell}_2$ distance in our experiments, demonstrating that the measure reports monochromatic images as false positives. Because of this lack of specificity, we found that their metric was not a satisfying solution to detect pixel-level memorization.

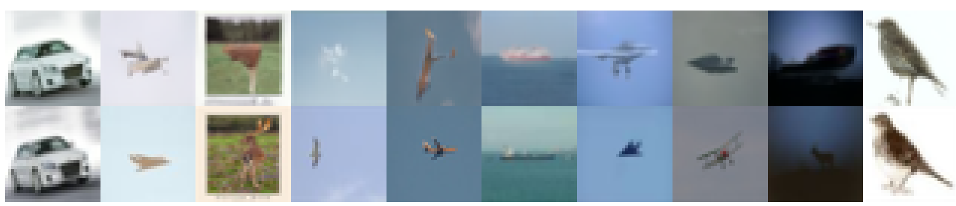

Figure 3: $\bar{\ell}_2$ **distance reports monochromatic images as memorizations.** Despite not always being memorizations of the training set, monochromatic images still generate a low $\bar{\ell}_2$ distance. (**Top**) Out of 5000 generations, we present the 10 generations with smallest patched $\bar{\ell}_2$ distance from CIFAR-10 train. (**Bottom**) We show the corresponding nearest neighbors in CIFAR-10 train to the top row of generations. Implementation details in Appendix Section A.1.

4 SolidMark: A Precise Metric for Memorization

We introduce SolidMark, a framework for precise evaluation of pixel-level memorization. We aimed to find a key-query mechanism, where recalling the key could indicate memorization of an image. We found the inpainting task naturally conducive to this idea—by masking out part of a training image, we can provide the unmasked portion to the model as a ‘query’ and ask it to recall the ‘key’ (the masked portion). We inject our training images with an unrelated scalar key by inserting a grayscale border. By training the model on these augmented images, we teach it to output a “prediction at a key” in the borders of its generations. At evaluation time, we can prompt the diffusion model to inpaint the border (key) for a training image and can evaluate its accuracy with a scalar distance function. An accurate prediction of the random color we assigned to the border would indicate that the exact image may be memorized.

To generate the borders, we will define a key function $k(\boldsymbol{x})$ that returns a key $k_{\boldsymbol{x}} \in \mathbb{R}$ for any image $\boldsymbol{x}$. For simplicity, we will set $k(\boldsymbol{x}) \sim \text{Unif}(0, 1)$; we draw a grayscale color (which is the same scalar across all channels) uniformly at random.

We now turn to define the distance function that we use as a metric. We begin by augmenting a training dataset $\boldsymbol{X}$ with key-encoded borders to yield a new dataset $\bar{\boldsymbol{X}}$. Then, we either pretrain a new model or finetune an existing model on $\bar{\boldsymbol{X}}$. After inpainting $\boldsymbol{x}$ with RePaint [Lugmayr et al.,

Table 1: **Evaluating Inference-Time Methods with SOLIDMARK.** Evaluation of inference-time memorization mitigation methods from Somepalli et al. [2023]. We compare the percentage reductions in memorization as measured by 95th percentile SSCD similarities from the source paper and $(\ell_{\text{SM}}, 0.01)$-eidetic memorizations.We show all techniques both with their default values and the parameter we found as optimal.

Metric	GNI	RT	CWR	RNA
95th Percentile of SSCD Similarities	3.74% $\downarrow$	**16.42%** $\downarrow$	9.43% $\downarrow$	13.63% $\downarrow$
SOLIDMARK (Default Parameters)	0.001% $\downarrow$	**4.10%** $\downarrow$	5.80% $\downarrow$	2.67% $\downarrow$
SOLIDMARK (Tuned Parameters)	**15.67%** $\downarrow$	5.70% $\downarrow$	5.80% $\downarrow$	3.64% $\downarrow$

2022], we find the absolute difference ℓ_{SM} (SM = SOLIDMARK) between the ground-truth key for the image $k_{\boldsymbol{x}}$ and the average of the inpainted border.

5 Evaluation

Validation on a Toy Model We wanted to ensure that the metric's results tended to follow changes in memorization. For this, we augmented CIFAR-10 with a solid 4-pixel thick border. On this augmented dataset, we pretrained DDPMs and applied a technique known to reduce memorization to verify SOLIDMARKas a metric. Since the DDPMs were class-conditioned and not text-conditioned, the only relevant technique from Somepalli et al. [2023] was Gaussian Noise at Inference (GNI), which adds Gaussian noise to the conditional embedding. Accordingly, we applied Gaussian noise with mean 0 and a range of magnitudes, tracking the number of $(\ell_{\text{SM}}, \boldsymbol{\delta})$-eidetic memorizations over 5000 generations as the magnitude of noise increased. We measured a monotonic decrease in eidetic memorizations for $\boldsymbol{\delta} = 0.01$ with an overall 57.1% decrease at the highest magnitude of noise. Eidetic memorizations for $\boldsymbol{\delta} = 0.001$ showed a 66.7% decrease. These results are plotted in Appendix section A.2.

Re-Evaluating Mitigation Techniques For Stable Diffusion, we were able to test all of the inference-time techniques from Somepalli et al. [2023]. To do this, we augmented a subset of LAION-400M with solid 16-pixel thick borders. We then finetuned Stable Diffusion 1.4 on this subset and compared the percentage decrease[2] in $(\ell_{\text{SM}}, 0.01)$-eidetic memorizations in our models against the percentage decrease in the source results of 95th percentile SSCD similarities. We have included these results in Table 1. Overall, our results with the source parameters correlate with the general hierarchy of the previously used metric (i.e, which methods are better than others). However, we also observed very different magnitudes of memorization reduction compared to the original metric. The only technique that provided over 10% reduction was GNI, which only happened when using a much stronger parameter than what Somepalli et al. [2023] suggested.

6 Discussion

Limitations. Our metric may struggle with quantifying duplication-induced memorization as duplicates will receive different keys.

SOLIDMARK as a "Lower Bound" for Memorization. Chiefly, SOLIDMARK's strength resides in its ability to function as a lower bound, where any instance of memorization found by SOLIDMARK indicates the model has explicit knowledge on a specific image. This strength derives from the key-query structure, where the keys are semantically unrelated from their queries. It would be incredibly unlikely to randomly infer such a key with high precision. For this reason, we consider SOLIDMARK an "intuitive lower bound" on pixel-level memorization. On the other hand, although the setting itself defines a strict form of memorization, we do provide the model with highly favorable conditions such that it would almost definitely be able to recall a sample if memorized.

[2]We used percentages to corroborate the results into numbers that could be meaningfully compared.

References

Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models, 2023. URL https://arxiv.org/abs/2301.13188.

Chen Chen, Daochang Liu, and Chang Xu. Towards memorization-free diffusion models, 2024. URL https://arxiv.org/abs/2404.00922.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. Data engineering for scaling language models to 128k context, 2024. URL https://arxiv.org/abs/2402.10171.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Greg Kamradt. Needle in a haystack - pressure testing llms, 2023. URL https://github.com/gkamradt/LLMTest_NeedleInAHaystack.

Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Ablating concepts in text-to-image diffusion models, 2023. URL https://arxiv.org/abs/2303.13516.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. In search of needles in a 11m haystack: Recurrent memory finds what llms miss, 2024. URL https://arxiv.org/abs/2402.10790.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models, 2024. URL https://arxiv.org/abs/2402.14848.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models, 2022. URL https://arxiv.org/abs/2201.09865.

Jie Ren, Yaxin Li, Shenglai Zeng, Han Xu, Lingjuan Lyu, Yue Xing, and Jiliang Tang. Unveiling and mitigating memorization in text-to-image diffusion models through cross attention, 2024. URL https://arxiv.org/abs/2403.11052.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Joseph Saveri and Matthew Butterick, 2023. URL https://imagegeneratorlitigation.com/.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs, 2021. URL https://arxiv.org/abs/2111.02114.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. URL https://arxiv.org/abs/1503.03585.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models, 2022. URL https://arxiv.org/abs/2212.03860.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models, 2023. URL https://arxiv.org/abs/2305.20086.

Weiyun Wang, Shuibo Zhang, Yiming Ren, Yuchen Duan, Tiantong Li, Shuo Liu, Mengkang Hu, Zhe Chen, Kaipeng Zhang, Lewei Lu, Xizhou Zhu, Ping Luo, Yu Qiao, Jifeng Dai, Wenqi Shao, and Wenhai Wang. Needle in a multimodal haystack, 2024. URL https://arxiv.org/abs/2406.07230.

Yuxin Wen, Yuchen Liu, Chen Chen, and Lingjuan Lyu. Detecting, explaining, and mitigating memorization in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=84n3UwkH7b`.

A Appendix

A.1 Modified ℓ_2 Distance

This distance rescales the ℓ_2 distance between a generation and its nearest neighbor based on its relative distance from the set $\mathbb{S}_{\hat{\boldsymbol{x}}_0}$ of $\hat{\boldsymbol{x}}_0$'s n nearest neighbors in $\boldsymbol{X}$. Specifically, we define $\mathbb{S}_{\hat{\boldsymbol{x}}_0} \subseteq \boldsymbol{X}$ where $|\mathbb{S}_{\hat{\boldsymbol{x}}_0}| = n$ and

$$\forall_{\boldsymbol{x} \in \boldsymbol{X} \setminus \mathbb{S}_{\hat{\boldsymbol{x}}_0}} \ell_2(\hat{\boldsymbol{x}}_0, \boldsymbol{x}) \geq \max_{y \in \mathbb{S}_{\hat{\boldsymbol{x}}_0}} \ell_2(\hat{\boldsymbol{x}}_0, \boldsymbol{y})$$

We then define the modified ℓ_2 distance as

$$\bar{\ell}_2(\hat{\boldsymbol{x}}_0, \boldsymbol{X}; \mathbb{S}_{\hat{\boldsymbol{x}}_0}) = \frac{\ell_2(\hat{\boldsymbol{x}}_0, \boldsymbol{x})}{\alpha \cdot \mathbb{E}_{\boldsymbol{y} \in \mathbb{S}_{\hat{\boldsymbol{x}}_0}}[\ell_2(\hat{\boldsymbol{x}}_0, \boldsymbol{y})]}$$

where α is a scaling factor. This distance increases when $\hat{x}_0$ is much closer to its nearest neighbor when compared to its n nearest neighbors, potentially indicative of memorization.

For our experiments, we trained class-conditional DDPMs for 500 epochs on CIFAR-10 train and sampled 5000 images with random classes, recording $\bar{\ell}_2$ for each generation with $n = 50$ and $\alpha = 0.5$ as in the original paper.

A.2 CIFAR-10 DDPM Results Plotted

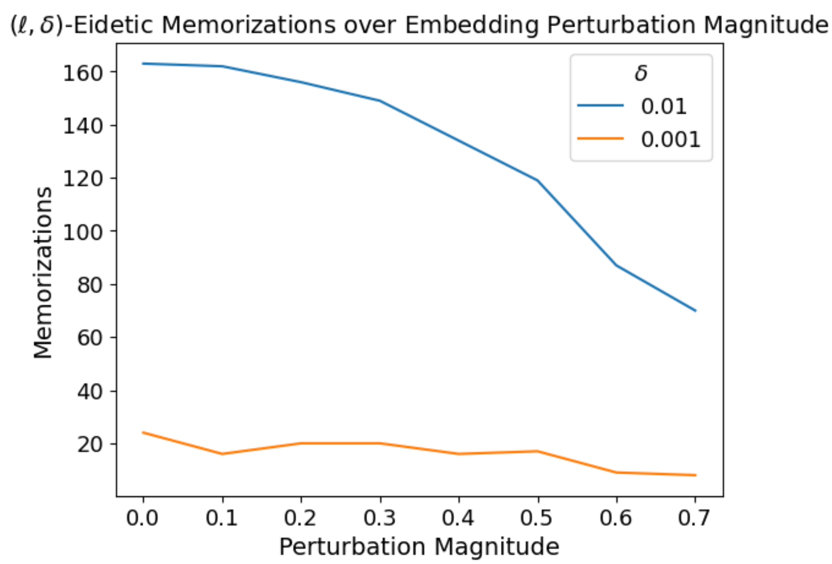

Figure 4: **SolidMark shows an overall reduction in memorization from GNI.** As the magnitude of the Gaussian noise increases, both $\boldsymbol{\delta}$ values find a decrease in memorization.