Demystifying Foreground-Background Memorization in Diffusion Models

Jimmy Z. Di [1]*, Yiwei Lu[2,3]*, Yaoliang Yu[3,4],
Gautam Kamath[3,4], Adam Dziedzic[5], Franziska Boenisch[5]
[1]University of Wisconsin–Madison, USA, [2]University of Ottawa, Ottawa, Canada, [3]Vector Institute, Toronto, Canada
[4]University of Waterloo, Waterloo, Canada [5]CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
jimmy.di@wisc.edu, yiwei.lu@uottawa.ca, yaoliang.yu@uwaterloo.ca,
g@csail.mit.edu, adam.dziedzic@cispa.de, boenisch@cispa.de

Abstract

Diffusion models (DMs) memorize training images and can reproduce near-duplicates during generation. Current detection methods identify verbatim memorization but fail to capture two critical aspects: quantifying partial memorization occurring in small image regions, and memorization patterns beyond specific prompt-image pairs. To address these limitations, we propose Foreground Background Memorization (*FB-Mem*), a novel segmentation-based metric that classifies and quantifies memorized regions within generated images. Our method reveals that memorization is more pervasive than previously understood: (1) individual generations from single prompts may be linked to clusters of similar training images, revealing complex memorization patterns that extend beyond one-to-one correspondences; and (2) existing model-level mitigation methods, such as neuron deactivation and pruning, fail to eliminate local memorization, which persists particularly in foreground regions. Our work establishes an effective framework for measuring memorization in diffusion models, demonstrates the inadequacy of current mitigation approaches, and proposes a stronger mitigation method using a clustering approach.

Extended version — https://arxiv.org/abs/2508.12148

1 Introduction

Diffusion models (DMs) (Sohl-Dickstein et al. 2015; Song, Meng, and Ermon 2020; Song et al. 2020) and their text-to-image derivatives (e.g., Latent Diffusion (Rombach et al. 2022), DALL-E (Ramesh et al. 2022)) have emerged as powerful generative frameworks, achieving remarkable success in producing high-fidelity images. Trained predominantly on large-scale datasets scraped from the Internet, such as LAION-5B (Schuhmann et al. 2022), these models often inherit both the richness and the risks associated with such data sources. A growing concern among researchers and practitioners is the potential for DMs to memorize and inadvertently reproduce portions of their training data, raising serious privacy, ethical, and legal issues (Carlini et al. 2023a; Somepalli et al. 2023a,b). These issues become particularly problematic when reproduced content includes copyrighted materials or sensitive personal information without explicit consent or safeguards.

Existing detection methods (e.g., Carlini et al. (2021); Somepalli et al. (2023a); Wen et al. (2024)) can accurately identify exact duplications using metrics like SSCD scores (Pizzi et al. 2022) or CLIP scores (Radford et al. 2021). While Webster (2023) and Chen et al. (2025) have identified “template memorization” and “local memorization” respectively to detect partial memorization, it remains unclear how to quantify or measure the potential harm of such memorization. For instance, memorizing a color palette in the background of an image poses significantly less risk than memorizing a copyrighted object or identifiable feature.

In this work, we argue that inexact memorization detection should be more fine-grained to better assess the severity of partial memorization. Specifically, we propose Foreground Background Memorization (*FB-Mem*), a novel segmentation-based metric that classifies and quantifies memorized content across different regions of generated images. Given two images, *FB-Mem* applies segmentation maps (Zheng et al. 2024) to differentiate foreground and background regions, compares each component using a pixel-wise image similarity metric and classifies the memorization into four categories: VM (verbatim memorization), FM (foreground memorization), BM (background memorization), and NM (not memorized).

Moreover, existing detection methods evaluate memorization for specific prompt-image pairs, neglecting the fact that DMs can generate diverse outputs from the same text prompt. Under *FB-Mem* evaluation, we observe that these varied outputs are not necessarily linked to a single training image, but instead exhibit a one-prompt-to-many-training-images (*one-to-many*) correspondence. We note that this notion is similar to “retrieval verbatims” by Webster (2023). However, Webster (2023) only considers verbatim memorization and does not perform systematic analysis, a gap we aim to address in this work. By clustering semantically similar text prompts, we can fully characterize this complex memorization behavior across different categories of memorization type using *FB-Mem*.

Our proposed tools do not only establish an effective framework for detecting memorization in DMs, but also provide a robust evaluation methodology for assessing mitigation algorithms. To address memorization, various miti-

[*]These authors contributed equally.

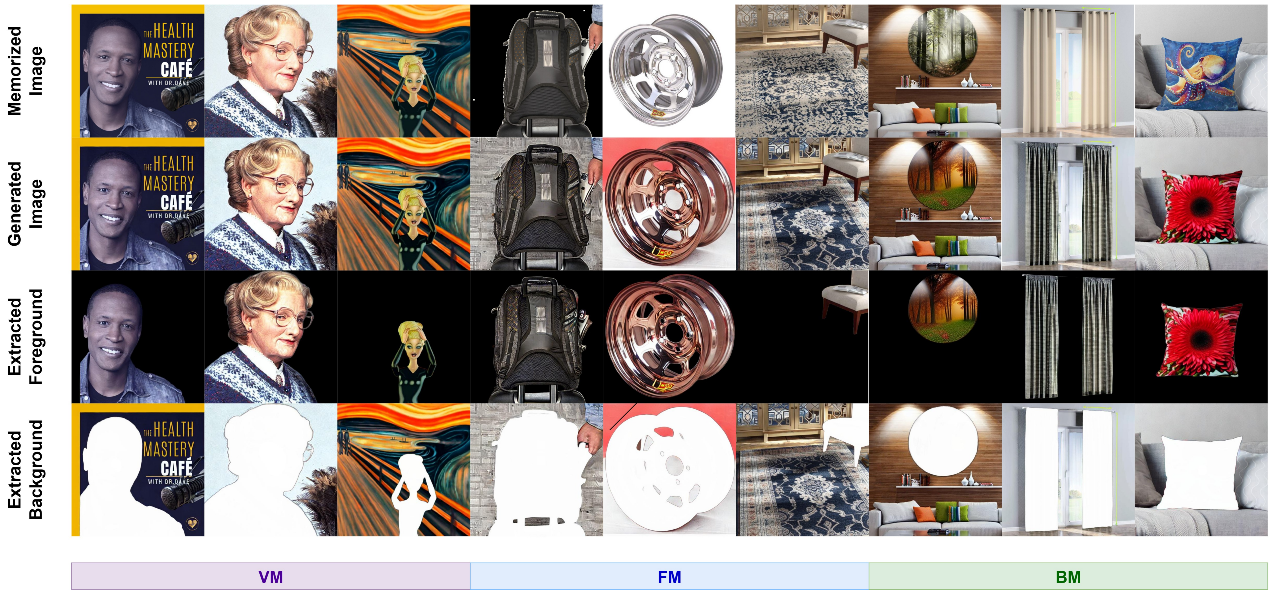

Figure 1: **Examples of different types of memorization under our *FB-Mem* evaluation.** We extract the foreground and background of the memorized images and generated images, and classify memorization using Algorithm 1 to: Verbatim Memorization (column 1-3), Foreground Memorization (column 4-6), and Background Memorization (column 7-9).

gation strategies have been proposed, including inference-stage methods that adjust text-embedding or attention logits (Ren et al. 2024; Wen et al. 2024), training-stage approaches that fine-tune pre-trained models (Ren et al. 2024; Wen et al. 2024), and neuron-level interventions (Hintersdorf et al. 2024; Chavhan et al. 2024). In this work, we focus on methods that permanently modify model weights (Wen et al. 2024; Hintersdorf et al. 2024; Chavhan et al. 2024), which represent more fundamental and responsible changes to the model. While such mitigation methods are effective against verbatim memorization, we observe that partial memorization, particularly foreground memorization, still persists after these mitigations. Furthermore, the *one-to-many* correspondence also remains intact following these interventions.

In summary, we make the following contributions:

- We propose *FB-Mem*, a segmentation-based metric that effectively detects and quantifies partial memorization in diffusion models beyond existing verbatim detection;
- We reveal that memorization is more pervasive than previously understood, with individual generations linked to multiple training images and local memorization persisting after previous mitigation methods;
- We demonstrate the inadequacy of current mitigation approaches using *FB-Mem* and propose a novel clustering-based mitigation method.

2 Background and Related Work

Neural network memorization: Neural network memorization is a common phenomenon in supervised learning (Arpit et al. 2017), self-supervised learning (Wang et al. 2024a,b), including contrastive learning (Wang et al. 2025), image autoregressive models (Kowalczuk et al. 2025), and diffusion models (Somepalli et al. 2023a,b; Wen et al. 2024). While it has been shown that memorization improves model generalization (Feldman 2020; Feldman and Zhang 2020; Wang et al. 2024b), it could also lead to critical privacy concerns, such as data extraction attacks (Carlini et al. 2019, 2021, 2023b; Kowalczuk et al. 2025). To detect such memorization in diffusion models, various approaches have been proposed, including SSCD scores (Pizzi et al. 2022), CLIP scores (Rombach et al. 2022), pairwise SSIM scores between initial noise differences (Webster 2023), distribution of attention (Ren et al. 2024), edge inconsistency (Webster 2023), and predicted noise magnitudes (Wen et al. 2024).

We note that Chen et al. (2025) apply brightened attention masks to identify local memorization, which shares similarities with our approach. However, we emphasize several key differences: (1) we provide a finer-grained classification that distinguishes between foreground and background memorization; (2) we conduct instance-level evaluation to identify one-prompt-to-many-training-images correspondence; and (3) we focus on model-level mitigation methods rather than prompt-level interventions.

Mitigating memorization: To address the problem of memorization for DMs, various methods have been proposed. For example, inference-stage mitigation, including attention logit rescaling (Ren et al. 2024) and text-embedding adjustment (Wen et al. 2024); training-stage mitigation by fine-tuning an existing DM-based model (Ren et al. 2024; Wen et al. 2024); and neuron-level mitigation (Maini et al. 2023; Hintersdorf et al. 2024; Chavhan et al. 2024) that localizes and deactivates certain neurons responsible for memorization. In this work, we focus on mitigation methods that change the model parameters, including the fine-tuning approach (Wen et al. 2024), the neuron deactivation approach (Hintersdorf et al. 2024), and the weight pruning approach (Chavhan et al. 2024).

Other techniques are also potentially applicable for mitigating memorization, for example, machine unlearning (proposed for removing private personal data) (Cao and Yang 2015; Bourtoule et al. 2021; Sekhari et al. 2021; Aldaghri, Mahdavifar, and Beirami 2021; Wu et al. 2024b) or concept removal (proposed for removing nudity or harmful concepts) (Chavhan, Li, and Hospedales 2024; Gandikota et al. 2023; Lyu et al. 2024) can potentially be used to remove the memorized information.

3 Measuring Memorization

In this section, we (1) address the gap of partial memorization by proposing the novel Foreground Background Memorization (*FB-Mem*) metric; and (2) propose an instance-level measurement for identifying *one-to-many* correspondence.

3.1 Memorization Pipeline

Existing approaches: Current research studies memorization in DMs, with particular focus on investigating memorization patterns in Stable Diffusion (SD) v1.4 (Rombach et al. 2022). While newer models exist, no datasets with memorized data are available for them. Consequently, state-of-the-art work (Webster 2023; Wen et al. 2024; Hintersdorf et al. 2024) considers 500 memorized LAION prompts for SD v1.4, which we adopt in our paper.[1] Hintersdorf et al. (2024) further split this dataset into Verbatim Memorization (VM) and Template Memorization (TM) categories using SSCD scores (Pizzi et al. 2022), which represent the cosine similarity of image embeddings obtained from the Self-Supervised Copy Detection (SSCD) model. Specifically, Hintersdorf et al. (2024) uses a threshold of 0.7 to distinguish between these two classes.

A new pipeline: Previous works typically assess memorization on a per-prompt basis, under the assumption that memorized prompts produce highly similar or even near-identical outputs across multiple generations. However, our empirical observations challenge this assumption: many memorized prompts result in significant variability across generated samples and may generate non-memorized samples even when using the same seed, as we will show in Section 3.3. Consequently, we manually reviewed and labeled 1,500 images generated using 300[2] of the memorized prompts from Chen et al. (2025). Each image was reviewed and labeled as VM, TM, or NM based on its visual resemblance to ground-truth images.

3.2 Measuring partial memorization

Previously, we discussed that existing approaches classify memorization into VM and TM. While VM has a clear definition as exact duplication, TM lacks a rigorous definition. Hintersdorf et al. (2024) and Webster (2023) state that TM reproduces only the general composition of training images while exhibiting non-semantic variations at fixed image positions. However, this definition does not accurately assess the potential harm of such memorization. For instance, memorizing a common background pattern poses minimal risk, whereas memorizing copyrighted content or artwork, even when appearing in a small region, should be identified and addressed (See Figure 1 for examples). Motivated by this limitation, we aim to provide a fine-grained classification of TM that better captures the varying degrees of potential harm associated with different types of memorization.

Algorithm 1: Foreground Background Memorization

```
Input: Generated image $\mathbf{x}_g$, training image $\mathbf{x}_t$, similarity
    metric $M$, score threshold $\tau$, segmentation threshold $\beta$
Output: Memorization type $\in$ {VM, FM, BM, NM}
Extract foreground mask $S_f(\mathbf{x}_g), S_f(\mathbf{x}_t)$
Extract background mask $S_b(\mathbf{x}_g), S_b(\mathbf{x}_t)$
$M_{\text{full}} \leftarrow M(\mathbf{x}_g, \mathbf{x}_t)$
if $\frac{|S_f|}{|x_g|} \le \beta$ then
    $M_{\text{fg}} \leftarrow M(\mathbf{x}_g, \mathbf{x}_t \odot S_f(\mathbf{x}_t))$
    $M_{\text{bg}} \leftarrow M(\mathbf{x}_g \odot S_b(\mathbf{x}_g), \mathbf{x}_t \odot S_b(\mathbf{x}_t))$
else if $\frac{|S_f|}{|x_g|} \ge 1 - \beta$ then
    $M_{\text{fg}} \leftarrow M(\mathbf{x}_g \odot S_f(\mathbf{x}_g), \mathbf{x}_t \odot S_f(\mathbf{x}_t))$
    $M_{\text{bg}} \leftarrow M(\mathbf{x}_g, \mathbf{x}_t \odot S_b(\mathbf{x}_t))$
else
    $M_{\text{fg}} \leftarrow M(\mathbf{x}_g \odot S_f(\mathbf{x}_g), \mathbf{x}_t \odot S_f(\mathbf{x}_t))$
    $M_{\text{bg}} \leftarrow M(\mathbf{x}_g \odot S_b(\mathbf{x}_g), \mathbf{x}_t \odot S_b(\mathbf{x}_t))$
end if
if $M_{\text{full}} \ge \tau$ then
    return VM (Verbatim Memorization)
else if $M_{\text{fg}} \ge \tau$ then
    return FM (Foreground Memorization)
else if $M_{\text{bg}} \ge \tau$ then
    return BM (Background Memorization)
else
    return NM (Not Memorized)
end if
```

Foreground background memorization: We propose a novel metric for measuring memorization called Foreground Background Memorization (*FB-Mem*). Our *FB-Mem* algorithm (Algorithm 1) utilizes a three-step comparison: (1) *foreground/background extraction*: given a pair of a generated image $\mathbf{x}_g$ and a training image $\mathbf{x}_t$, *FB-Mem* first applies segmentation to both images to extract foreground and background masks S_f and S_b; (2) *computing similarity*: given a a similarity metric M, we calculate the full image similarity M_{full} between the generated and training images, and the foreground/background similarity between their extracted foreground/background, respectively; (3) *memorization classification*: finally, given a threshold τ, we classify the memorization into four possible types.

Specifically, if the full image similarity exceeds the threshold τ, *FB-Mem* returns Verbatim Memorization (VM). Otherwise, it checks foreground and background similarities in sequence, returning Foreground Memorization (FM) or

[1] We extend our discussion to Stable Diffusion 3 in Section 4.4.

[2] We use this manually labeled subset to select the optimal similarity metric M in Section 3.2, while we use the complete 500 prompts for subsequent experiments.

Table 1: Performance of each metric in classifying different memorization: 1) VM-NM; 2) TM-NM; 3) VM-TM.

Classification	VM-NM			TM-NM			VM-TM		
Metrics	SSIM	MS-SSIM	SSCD	SSIM	MS-SSIM	SSCD	SSIM	MS-SSIM	SSCD
AUROC	0.989	0.994	**1.000**	0.886	0.962	**0.992**	0.922	**0.884**	0.875
f-1 Score	0.820	**0.992**	0.986	0.727	0.318	**0.343**	0.361	**0.846**	0.742
TP@1%FP	0.897	0.986	**0.997**	0.300	0.856	**0.993**	0.517	**0.528**	0.489
Accuracy	0.954	**0.997**	0.995	0.779	0.651	**0.661**	0.404	**0.913**	0.831

Background Memorization (BM), respectively, if either exceeds the threshold. If none of the similarity scores meet the threshold, it classifies the pair as Not Memorized (NM).

To avoid reporting false positives when image segmentation fails, such as when the quality of the image is low, we perform an adaptive similarity computation based on the foreground proportion in the generated image. When the foreground region is very small (proportion $\leq \beta$ of the total image size, where β is a tunable hyper-parameter), the algorithm compares the entire generated image against only the foreground of the training image for foreground similarity, while computing background similarity using masked regions from both images. Conversely, when the foreground dominates the image (proportion $\geq 1 - \beta$), it compares the masked foreground regions but uses the entire generated image against the training image's background for background similarity. For balanced cases where the foreground proportion falls between these extremes, the algorithm performs standard masked comparisons for both foreground and background regions. For all experiments reported in this paper, we adapt the threshold of $\beta = 0.03$.

Choosing an optimal M: In principle, our *FB-Mem* algorithm can be equipped with any suitable similarity metric M. In this paper, we choose Multiscale Structural Similarity Index (MS-SSIM) (Wang, Simoncelli, and Bovik 2003) as our metric M. SSIM is a standard tool for comparing pixel-wise image similarity through the lens of luminance, contrast, and structure. MS-SSIM further performs multiple rescaling and down-sampling procedures on the contrast and structural components to obtain a more robust form. Details of SSIM and MS-SSIM are provided in the Appendix. Next we justify our choice of M.

Justifying the choice of M: To evaluate the effectiveness of different memorization metrics, we conducted a three-way classification task (VM, TM,[3] NM) using our manually labeled dataset described in Section 3.1. We generated 1,500 images using the 300 labeled prompts and computed the similarity of each generated image to all 498[4] ground-truth memorized images using three metrics: SSIM, MS-SSIM, and SSCD. For each generated image, we identified the highest-scoring ground-truth pair across all comparisons under each metric.

Similarity classification thresholds were established based on prior work. For SSIM and MS-SSIM, we set the VM threshold at 0.8 and the TM threshold at 0.6. For SSCD, we adopted settings from Hintersdorf et al. (2024) and Wen et al. (2024), using a verbatim memorization (VM) threshold of 0.7 and a template memorization (TM) threshold of 0.5. Classification performance is reported in Table 1.

Our results demonstrate that while SSCD effectively classifies verbatim and non-memorized samples, it exhibits vulnerability to localized dissimilarities. Specifically, SSCD may fail to detect memorization when images contain small differing regions despite being visually near-identical overall.[5] We also evaluated the accuracy-efficiency tradeoff across metrics. On an NVIDIA A6000 GPU, comparing a single generated image against 500 ground-truth training images requires over 5 minutes using SSCD, compared to only 24 seconds using MS-SSIM. This substantial computational advantage, combined with MS-SSIM's robustness to minor local variations, motivated our choice to build FB-Mem upon MS-SSIM.

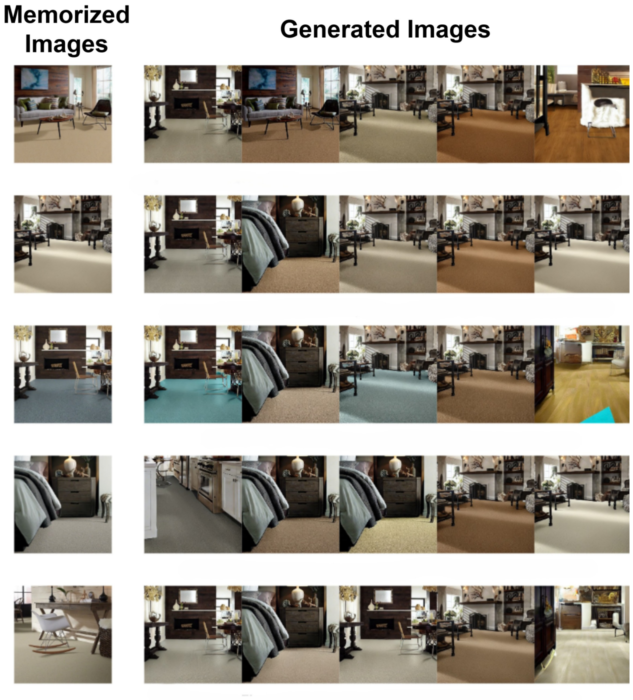

Figure 2: The memorized ground-truth image as well as the images generated using 5 different prompts selected from cluster 0 (i.e., the **Shaw Floors** cluster), in generation order.

[3] In particular, we perform direct comparison of entire images (as done by previous work) rather than considering any sort of segmentation (as we prescribe in Algorithm 1).

[4] Two images were unavailable due to broken URLs.

[5] This aligns with recent findings by Chen et al. (2025), which highlight SSCD's vulnerability to local perturbations.

3.3 Measuring *one-to-many* correspondence

In Section 3.1, we note that existing methods neglect generation variations and only consider prompt-wise memorization. Using our instance-wise pipeline and *FB-Mem*, we observe an intriguing phenomenon of one-prompt-to-many-training-images (*one-to-many*) correspondence. Specifically, we perform $N = 5$ generations per prompt and group the results into 5 categories according to the number of matched training images within the memorized dataset. For each category, we count the occurrences of VM, FM, and BM and present the results in Figure 3 (top). We observe that a substantial number of prompts exhibit *one-to-many* correspondence, demonstrating a variety of memorization types across generations from the same prompt.

Prompts Clustering: Moreover, we observe that *one-to-many* correspondence is not random but exhibits semantic coherence around shared concepts. To quantify this behavior, we first encode each prompt into an embedding using the CLIP-ViT-B model, then apply K-Nearest Neighbors (KNN) clustering to group the 500 prompts into 12 distinct clusters. Figure 2 shows examples of five prompts sampled from the same cluster and their corresponding generated images.[6]

Ablation Study on N: Finally, we conduct an ablation study with an increased number of generated images per prompt ($N = 20$) and present the results in Figure 3 (bottom). We observe that prompts exhibiting *one-to-one* correspondence remain roughly the same as with $N = 5$. However, when we increase N, *one-to-many* correspondence can extend up to 17 matching images, demonstrating the capability of diffusion models to memorize a large number of training images within a single prompt.

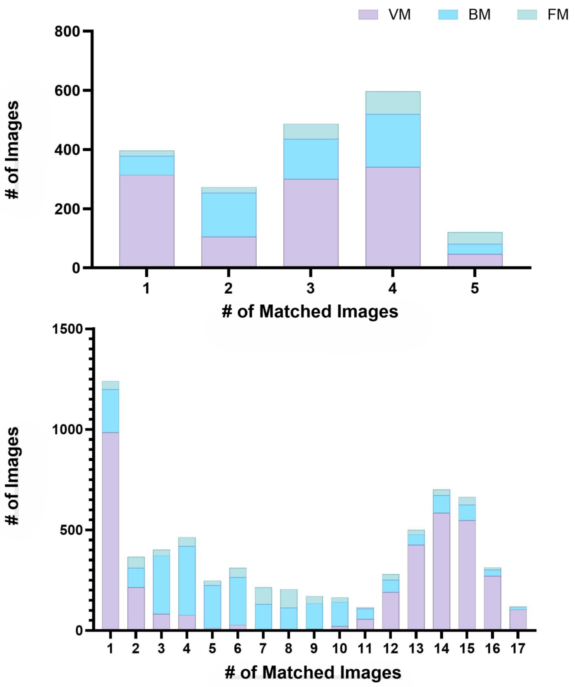

Figure 3: **Quantitative analysis of memorization in generated images** measured by *FB-Mem*. The x-axis represents the number of distinct ground-truth images matched using each of the N images generated by each prompt, while the y-axis indicates the number of generated images identified as memorized by *FB-Mem*. Each generated image produces exactly one match with one ground-truth image with the highest similarity score. Non-Memorized images are not included in the figure. **Top:** Default setting with $N = 5$. **Bottom:** Ablation study with $N = 20$.

4 Mitigating Memorization

In the previous section, we established a new memorization pipeline, proposed *FB-Mem* for memorization evaluation, and identified the phenomenon of *one-to-many* correspondence. In this section, we (1) examine existing model-based mitigation methods under the *FB-Mem* framework; (2) propose a cluster-wise mitigation approach; and (3) design a scoring metric for mitigation evaluation and analyze the utility-quality tradeoff in post-mitigation models.

4.1 Clustering-based Mitigation

In the previous section, we observed the phenomenon of *one-to-many* correspondence, which suggests that memorization occurs at the concept level rather than the prompt level. We therefore propose prompt clustering to quantify this notion of concept. This approach naturally extends to a stronger mitigation strategy, where we address clusters of memorization collectively.

Specifically, we build upon NeMo (Hintersdorf et al. 2024), a prompt-wise mitigation approach due to its superior mitigation performance on prompt-wise mitigation and preserving utility. NeMo utilizes a two-step process: (1) *initial selection*: for each prompt in a cluster, NeMo identifies a broad set of candidate neurons that may be responsible for memorizing a specific training image; and (2) *refinement*: NeMo filters the initial candidate set to obtain a smaller, refined set of neurons for each prompt.

Our clustering-based mitigation approach (denoted as NeMo-C) introduces a third step of *aggregation*: given a memorized cluster, we compute the union of all refined neuron sets across the cluster, resulting in a consolidated set of neurons that consistently contribute to memorization. For each prompt in the cluster, we deactivate all neurons in this union set to mitigate memorization.

Unlike NeMo, which deactivates neurons specific to individual prompts, NeMo-C performs mitigation using the aggregated union set of neurons across the entire cluster. This enables robust memorization mitigation, specifically targeting the *one-to-many* phenomenon described previously. In Section 4.4, we demonstrate that NeMo-C does not significantly decrease model utility compared to NeMo and other mitigation methods. We present selected examples of generated images before and after applying NeMo-C, including

[6] Notably, some generated images do not appear to be direct copies of training data. Interestingly, when we increase the number of N in the next paragraph, we are able to generate replicates of these images as well.

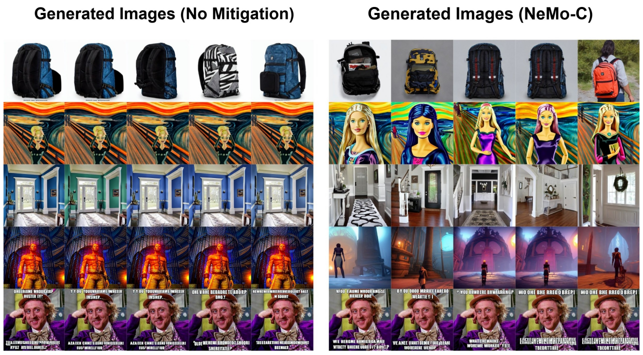

Figure 4: **Examples of generated images before and after applying NeMo-C.** The provided prompts, from top to bottom, are: (1) *ALPHA Convoy 320 Backpack - View 4*; (2) *If Barbie Were the Face of the World's Most Famous Paintings*; (3) *Foyer Painted in WHITE*; (4) *Dreamfall Chapters: The Longest Journey Will Be a PlayStation 4 Exclusive*; (5) *Willy Wonka - Oh, you are in IB? Please tell me how much smarter you are than everyone else*.

failure cases where generated images remain verbatim memorized post-mitigation, in Figure 4.

4.2 Mitigation Evaluation

Asides from *FB-Mem*, we propose two metrics to evaluate memorization mitigation: the mitigation strength, quantified by a novel scoring function, and image quality post mitigation which measures model utility.

Table 2: Mitigation Strength Scoring Function.

From VM		**From BM**		**From FM**	
VM→NM	+2.0	BM→FM	-0.5	FM→BM	+1.0
VM→BM	+1.5	BM→VM	-1.5	FM→NM	+1.5
VM→FM	+0.5	BM→NM	+0.5	FM→VM	-0.5
From NM					
NM→VM: -2.0		NM→FM: -1.5		NM→BM: -0.5	

Mitigation Strength: To evaluate the effectiveness of memorization mitigation methods, we introduce a scoring function in Table 2 that quantifies the strength of mitigation based on memorization type transitions. Our scoring system assigns numerical values to transitions between different memorization states: Verbatim Memorization (VM), Background Memorization (BM), Foreground Memorization (FM), and Not Memorized (NM). The scoring function operates on the principle that transitions reducing memorization severity receive positive scores, while those increasing memorization or introducing new memorization patterns receive negative penalties.

Image Quality: We use Q-Align (Wu et al. 2024a) and DB-CNN (Zhang et al. 2020) to evaluate the quality of images after applying memorization mitigation. Due to their no-reference (NR) nature, both methods perform well even with our relatively small sample size. Q-Align leverages aligned Vision-Language models (VLM) to assess generation quality, while DB-CNN is based on Deep Bilinear Convolutional Neural Network. Both methods were implemented using the Image Quality Assessment (IQA) toolbox.

4.3 Experimental Settings

Baseline Methods: We consider three model-based mitigation methods: NeMo (Hintersdorf et al. 2024), Wanda (Chavhan et al. 2024),[7] and DetectMem (Wen et al. 2024). All mitigation methods are applied using their default hyperparameters as specified in the respective papers. For NeMo, we use an activation threshold of 0.428 to determine which neurons to deactivate for each of the 500 prompts. This threshold corresponds to the mean plus one standard deviation of pairwise SSIM scores between initial noise differences, measured on a holdout set of 50,000 LAION prompts. The number of deactivated neurons varies across prompts, ranging from 0 to 436. For Wanda, the sparsity threshold (i.e., the percentage of pruned neurons) is set to 1%. For DetectMem, we evaluate both training-time and inference-time mitigation approaches. The training-time mitigation uses the pre-fine-tuned SD v1.4 model provided by the authors. For inference-time mitigation, we use the default configuration with a target loss of 3.

Evaluation Pipeline: We use the 500 memorized prompts from Webster (2023) as our pre-mitigation benchmark. Using Stable Diffusion v1.4, we generate five images per prompt using one random seed without applying any mitigation techniques, following the same settings described in Section 3. We then apply various mitigation methods, including baseline methods and our NeMo-C method, and regenerate five images per prompt under each method. Each generated image is compared to all 498 of the retrievable memorized images, resulting in a total of 1,245,000 comparisons. Additionally, we include **(1)** *FB-Mem* results for 500 randomly selected LAION prompts from the non-memorized set, generating 5 images per prompt as well as **(2)** 2500 images generated using the state-of-the-art Stable Diffusion 3 model using the 500 memorized prompts.

4.4 Experimental Results

Memorization distribution under *FB-Mem*: We first evaluate the effectiveness of each mitigation method using the *FB-Mem* metric. As shown in Figure 5, most methods are highly effective at reducing verbatim memorization (VM), eliminating over 90% of such cases. However, while background memorization (BM) is largely alleviated, foreground memorization (FM) persists. Notably, we examine the memorized prompts on Stable Diffusion 3 and observe that memorization does not exhibit significantly even without mitigation,[8] matching the observations of Hintersdorf et al. (2024).

[7] Note that Wanda was originally proposed by Sun et al. (2023) for large language models, while we utilize its adaptation for diffusion models as presented in Chavhan et al. (2024).

[8] We acknowledge that this might be because the memorized prompts differ significantly across different versions of Stable Diffusion, a problem we aim to study systematically in future work.

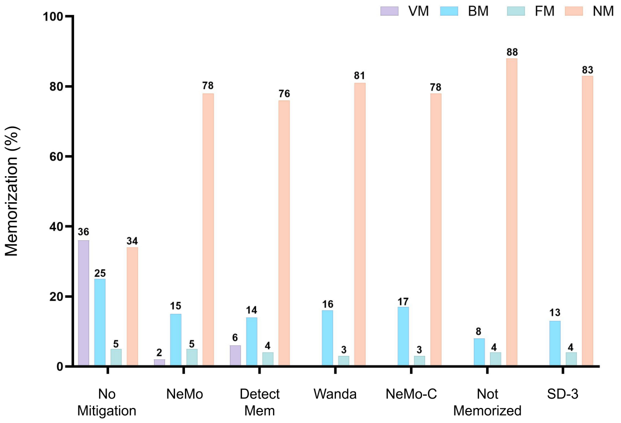

Figure 5: Memorization distribution evaluated using *FB-Mem* before/after mitigation, not memorized prompts, and Stable Diffusion 3.

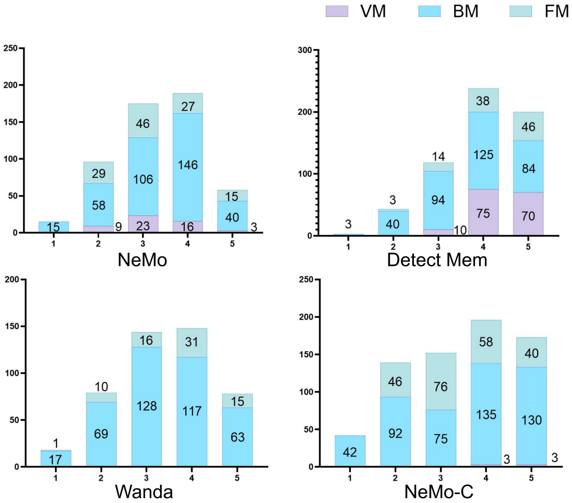

Figure 6: *One-to-many* correspondence after applying mitigation methods, measured using *FB-Mem*.

Moreover, in Figure 6, we demonstrate that *one-to-many* correspondence still largely persists across methods, with NeMo-C showing slight improvements over baselines.

Mitigation strength: Although the memorization distribution analysis provides insights into post-mitigation results, it fails to capture the memorization transitions induced by different mitigation methods. To address this, we calculate the average mitigation score (across 2,500 images, higher scores indicate better performance) for each method according to the scoring function in Table 2. The results are: NeMo (0.74), DetectMem (0.67), Wanda (0.79), and NeMo-C (0.83). We observe that NeMo-C achieves the highest mitigation strength, demonstrating the effectiveness of our concept-wise clustering approach.

Utility trade-off: Finally, a key evaluation criterion for mitigation methods is performance preservation. In Table 3, we report the average quality of generated images after applying mitigation methods. Among all approaches, NeMo and NeMo-C achieve the best image quality. Moreover, we present the trade-off between mitigation strength and utility degradation in Figure 7, where NeMo-C emerges as the optimal mitigation method.

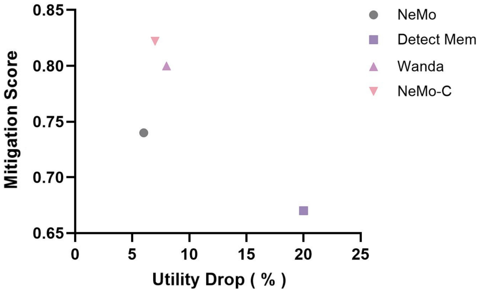

Figure 7: Trade-off between utility and mitigation efficacy. The y-axis shows the mitigation score (higher is better), while the x-axis indicates the drop in image quality compared to the non-mitigated baseline (lower is better). Utility drop is calculated using the two methods shown in Table 3.

Table 3: Model performance assessed by image quality metrics before/after mitigation. The reported scores are computed for all 2,500 generated images and averaged.

Mitigation Method	DB-CNN	Q-Align
Pre-mitigation	0.60	4.02
NeMo	0.587	3.63
Detect Mem	0.449	3.41
Wanda	0.578	3.55
NeMo-C	0.586	3.52

5 Conclusion

In this work, we addressed critical limitations in memorization detection for DMs by proposing *FB-Mem*, a segmentation-based metric that provides fine-grained classification of memorized content. Our analysis revealed that memorization is fundamentally cluster-wise rather than prompt-wise, with individual generations incorporating content from multiple training images simultaneously. Using the *FB-Mem* framework, we demonstrated that existing mitigation methods fail to eliminate local memorization, particularly in foreground regions. Our proposed NeMo-C cluster-wise mitigation approach achieves more robust memorization reduction while maintaining model utility.

Our work establishes a proper measurement of the memorization pipeline of DMs and opens new directions, such as extending these findings to other generative modalities like large language models and developing more sophisticated semantic-based clustering and mitigation strategies regarding foreground memorization.

Acknowledgements

GK is supported by a Canada CIFAR AI Chair, an NSERC Discovery Grant, and an Ontario Early Researcher Award.

FB has received funding from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (grant agreement No 101220235).

References

Aerni, M.; Rando, J.; Debenedetti, E.; Carlini, N.; Ippolito, D.; and Tramèr, F. 2024. Measuring Non-Adversarial Reproduction of Training Data in Large Language Models. arXiv:2411.10242.

Aldaghri, N.; Mahdavifar, H.; and Beirami, A. 2021. Coded Machine Unlearning. *IEEE Access*, 9: 88137–88150.

Arpit, D.; Jastrzebski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M. S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; et al. 2017. A closer look at memorization in deep networks. In *International conference on machine learning*, 233–242. PMLR.

Bourtoule, L.; Chandrasekaran, V.; Choquette-Choo, C. A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; and Papernot, N. 2021. Machine Unlearning. In *IEEE Symposium on Security and Privacy (SP)*, 141–159.

Cao, Y.; and Yang, J. 2015. Towards Making Systems Forget with Machine Unlearning. In *IEEE Symposium on Security and Privacy*, 463–480.

Carlini, N.; Hayes, J.; Nasr, M.; Jagielski, M.; Sehwag, V.; Tramer, F.; Balle, B.; Ippolito, D.; and Wallace, E. 2023a. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, 5253–5270.

Carlini, N.; Hayes, J.; Nasr, M.; Jagielski, M.; Sehwag, V.; Tramer, F.; Balle, B.; Ippolito, D.; and Wallace, E. 2023b. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, 5253–5270.

Carlini, N.; Liu, C.; Erlingsson, Ú.; Kos, J.; and Song, D. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, 267–284.

Carlini, N.; Tramer, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.; Song, D.; Erlingsson, U.; et al. 2021. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, 2633–2650.

Chavhan, R.; Bohdal, O.; Zong, Y.; Li, D.; and Hospedales, T. 2024. Memorized Images in Diffusion Models share a Subspace that can be Located and Deleted. *arXiv preprint arXiv:2406.18566*.

Chavhan, R.; Li, D.; and Hospedales, T. 2024. ConceptPrune: Concept Editing in Diffusion Models via Skilled Neuron Pruning. *arXiv preprint arXiv:2405.19237*.

Chen, C.; Liu, D.; Shah, M.; and Xu, C. 2025. Exploring Local Memorization in Diffusion Models via Bright Ending Attention. arXiv:2410.21665.

Feldman, V. 2020. Does Learning Require Memorization? A Short Tale about a Long Tail. In *Proceedings of the 52nd Annual ACM Symposium on the Theory of Computing*, STOC '20, 954–959. ACM.

Feldman, V.; and Zhang, C. 2020. What Neural Networks Memorize and Why: Discovering the Long Tail via Influence Estimation. In *Advances in Neural Information Processing Systems 33*, NeurIPS '20, 2881–2891. Curran Associates, Inc.

Gandikota, R.; Materzyńska, J.; Fiotto-Kaufman, J.; and Bau, D. 2023. Erasing Concepts from Diffusion Models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2426–2436.

Hintersdorf, D.; Struppek, L.; Kersting, K.; Dziedzic, A.; and Boenisch, F. 2024. Finding NeMo: Localizing Neurons Responsible For Memorization in Diffusion Models. *arXiv preprint arXiv:2406.02366*.

Kowalczuk, A.; Dubiński, J.; Boenisch, F.; and Dziedzic, A. 2025. Privacy Attacks on Image AutoRegressive Models. In *Forty-Second International Conference on Machine Learning (ICML)*.

Lyu, M.; Yang, Y.; Hong, H.; Chen, H.; Jin, X.; He, Y.; Xue, H.; Han, J.; and Ding, G. 2024. One-dimensional Adapter to Rule Them All: Concepts Diffusion Models and Erasing Applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7559–7568.

Maini, P.; Mozer, M. C.; Sedghi, H.; Lipton, Z. C.; Kolter, J. Z.; and Zhang, C. 2023. Can neural network memorization be localized? In *Proceedings of the 40th International Conference on Machine Learning*, 23536–23557.

Pizzi, E.; Roy, S. D.; Ravindra, S. N.; Goyal, P.; and Douze, M. 2022. A Self-Supervised Descriptor for Image Copy Detection. arXiv:2202.10261.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2): 3.

Ren, J.; Li, Y.; Zeng, S.; Xu, H.; Lyu, L.; Xing, Y.; and Tang, J. 2024. Unveiling and mitigating memorization in text-to-image diffusion models through cross attention. In *European Conference on Computer Vision*, 340–356. Springer.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Schuhmann, C.; Beaumont, R.; Vencu, R.; Gordon, C.; Wightman, R.; Cherti, M.; Coombes, T.; Katta, A.; Mullis, C.; Wortsman, M.; et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35: 25278–25294.

Sekhari, A.; Acharya, J.; Kamath, G.; and Suresh, A. T. 2021. Remember What You Want to Forget: Algorithms for Machine Unlearning. In *Advances in Neural Information Processing Systems*.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, 2256–2265.

Somepalli, G.; Singla, V.; Goldblum, M.; Geiping, J.; and Goldstein, T. 2023a. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6048–6058.

Somepalli, G.; Singla, V.; Goldblum, M.; Geiping, J.; and Goldstein, T. 2023b. Understanding and Mitigating Copying in Diffusion Models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Sun, M.; Liu, Z.; Bair, A.; and Kolter, J. Z. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Wang, W.; Dziedzic, A.; Backes, M.; and Boenisch, F. 2024a. Localizing Memorization in SSL Vision Encoders. In *Thirty-Eighth Conference on Neural Information Processing Systems (NeurIPS)*.

Wang, W.; Dziedzic, A.; Kim, G. C.; Backes, M.; and Boenisch, F. 2025. Captured by Captions: On Memorization and its Mitigation in CLIP Models. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Wang, W.; Kaleem, M. A.; Dziedzic, A.; Backes, M.; Papernot, N.; and Boenisch, F. 2024b. Memorization in self-supervised learning improves downstream generalization. *arXiv preprint arXiv:2401.12233*.

Wang, Z.; Bovik, A.; Sheikh, H.; and Simoncelli, E. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612.

Wang, Z.; Simoncelli, E.; and Bovik, A. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, 1398–1402 Vol.2.

Webster, R. 2023. A Reproducible Extraction of Training Images from Diffusion Models. arXiv:2305.08694.

Wen, Y.; Liu, Y.; Chen, C.; and Lyu, L. 2024. Detecting, explaining, and mitigating memorization in diffusion models. In *The Twelfth International Conference on Learning Representations*.

Wu, H.; Zhang, Z.; Zhang, W.; Chen, C.; Li, C.; Liao, L.; Wang, A.; Zhang, E.; Sun, W.; Yan, Q.; Min, X.; Zhai, G.; and Lin, W. 2024a. Q-Align: Teaching LMMs for Visual Scoring via Discrete Text-Defined Levels. *International Conference on Machine Learning (ICML)*.

Wu, J.; Le, T.; Hayat, M.; and Harandi, M. 2024b. EraseDiff: Erasing Data Influence in Diffusion Models.

Yarkoni, S.; and Livni, R. 2025. Low Resource Reconstruction Attacks Through Benign Prompts. *arXiv preprint arXiv:2507.07947*.

Zhang, W.; Ma, K.; Yan, J.; Deng, D.; and Wang, Z. 2020. Blind Image Quality Assessment Using a Deep Bilinear Convolutional Neural Network. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(1): 36–47.

Zheng, P.; Gao, D.; Fan, D.-P.; Liu, L.; Laaksonen, J.; Ouyang, W.; and Sebe, N. 2024. Bilateral Reference for High-Resolution Dichotomous Image Segmentation. arXiv:2401.03407.

A More Related Works

Note that a simultaneous and independent work (Yarkoni and Livni 2025) also studies template memorization of diffusion models, and we discuss its connection to our paper. Specifically, Yarkoni and Livni (2025) utilizes segmentation masking for searching near-duplicates, while our *FB-Mem* apply a segmentation-based metric to quantify partial memorization. Moreover, the paper observes that a single generated image may match multiple copied elements from different sources, further confirming our finding of one-to-many correspondence.

B SSIM and Multiscale-SSIM

First, we provide a brief overview of the Structural Similarity Index (SSIM) (Wang et al. 2004) and the improved Multiscale Structural Similarity Index (MS-SSIM) (Wang, Simoncelli, and Bovik 2003). Let $\mathbf{m} = \{\mathbf{m}_i | i = 1, 2, ..., N\}$ and $\mathbf{n} = \{\mathbf{n}_i | i = 1, 2, ..., N\}$ be two pixel groups extracted from the same spatial location from two images being compared, and $\mu_{\mathbf{m}}$, $\sigma^2_{\mathbf{m}}$ and $\sigma_{\mathbf{mn}}$ be the mean of $\mathbf{m}$, variation of $\mathbf{m}$, and the covariance of $\mathbf{m}$ and $\mathbf{n}$, respectively. Then, the standard SSIM score is calculated by looking at the luminance (l), contrast (c), and structure (s) as follows:

$$\begin{aligned} l(\mathbf{m}, \mathbf{n}) &= \frac{2\mu_{\mathbf{m}}\mu_{\mathbf{n}} + c_1}{\mu^2_{\mathbf{m}} + \mu^2_{\mathbf{n}} + c_1} \\ c(\mathbf{m}, \mathbf{n}) &= \frac{2\sigma_{\mathbf{m}}\sigma_{\mathbf{n}} + c_2}{\sigma^2_{\mathbf{m}} + \sigma_{\mathbf{n}} + c_2} \\ s(\mathbf{m}, \mathbf{n}) &= \frac{\sigma_{\mathbf{mn}} + c_3}{\sigma_{\mathbf{m}}\sigma_{\mathbf{n}} + c_3} \end{aligned}$$

where c_1, c_2, and c_3 are small constants to stabilize the division with a weak denominator. With three components set to equal importance, we obtain the SSIM score:

$$\texttt{SSIM}(\mathbf{m}, \mathbf{n}) = \frac{(2\mu_{\mathbf{m}}\mu_{\mathbf{n}} + c_1)(2\sigma_{\mathbf{mn}} + c_2)}{(\mu^2_{\mathbf{m}} + \mu^2_{\mathbf{n}} + c_1)(\sigma^2_{\mathbf{m}} + \sigma^2_{\mathbf{n}} + c_2)},$$

Furthermore, by performing multiple re-scaling and down-sampling procedures to the contrast component and structural component using a scaling factor j, we can obtain a more robust form of SSIM. Let 1 be the scale of the original images and Scale K be the maximum scale, the Multi-scale Similarity Index is obtained through:

$$\texttt{SSIM}(\mathbf{m}, \mathbf{n}) = [l_K(\mathbf{m}, \mathbf{n})]^{\alpha_M} \cdot \prod_{j=1}^{K} [c_j(\mathbf{m}, \mathbf{n})]^{\beta_j} [s_j(\mathbf{m}, \mathbf{n})]^{\gamma_j}.$$

In practice, the relative importance hyperparameter α, β, and γ are normalized and set equal at all values of j.

C Additional Experiments

C.1 Ablation Study on N

In this section, we provide additional details and example images from the ablation study where the number of generations per prompt N is increased to 20.

Experiment Setting: Following the pre-mitigation experiments in Section 3, we use the same 500 memorized prompts from Webster (2023). Using Stable Diffusion v1.4 with a fixed random seed, we generate $N = 20$ images per prompt without applying any mitigation techniques.

Clustering Example: Figure 8 shows 20 images generated for six prompts from the **Shaw Floors** cluster. The first five prompts (rows 1–5) and their corresponding first five outputs were previously shown in Figure 2. Notably, prompts 1–6 each produce nearly identical images beyond the initial five generations, maintaining the same sequential order. This observation provides strong evidence that the model clusters semantically similar training prompts and offers insights into how diffusion models internalize and reproduce content from their training data.

C.2 NeMo-C with a Dampening Factor

For both NeMo (Hintersdorf et al. 2024) and our proposed NeMo-C method (described in Section 4.1), the refined set of neurons in the U-Net is completely deactivated. An alternative approach is to dampen these neurons rather than fully deactivating them. Specifically, we apply a multiplicative dampening factor α_{damp}. In this section, we explore the effect of using such a dampening mechanism, experimenting with values $\alpha_{\text{damp}} = 0.1$ and 0.2. We hypothesize that this approach may improve the visual quality of the generated images—by retaining some informative signal from the suppressed neurons—while partially sacrificing the mitigation strength compared to the original NeMo-C.

Table 4: Model performance assessed by mitigation efficacy and image quality metrics after applying mitigation with varying dampening factor α_{damp}. When $\alpha_{damp} = 0$, we obtain the standard NeMO-C method described in the main paper.The reported mitigation score is calculated by aggregating 2500 scores using the method outlined in Table 2. The reported quality scores are computed for all 2,500 generated images and averaged.

Dampening Factor (α_{damp})	Mitigation Score	Image Quality (DB-CNN)
0 (NeMo-C)	0.83	0.586
0.1	0.81	0.588
0.2	0.797	0.590

The results presented in Table 4 are consistent with our hypothesis. As the dampening factor increases, the image quality, measured by DB-CNN score, improves slightly, indicating that partial retention of neuron activations may help preserve useful generative capacity. However, this improvement in quality comes at the cost of a modest reduction in mitigation effectiveness. These findings suggest that dampening offers a tunable trade-off between mitigation strength and image fidelity, providing a flexible alternative to hard neuron deactivation.

D Generated Examples

In this section, we present additional examples of generated images along with their corresponding memorized training images and prompts, as shown in Figure 9. These examples include both successful mitigation cases and failure cases where NeMo-C fails to eliminate memorization.

The examples demonstrate that memorization can be harmful in several ways: **(1)** it reduces the diversity of generated outputs (rows 4–6); **(2)** it can lead to inaccurate generations that contradict prompt specifications (e.g., row 7, where the prompt explicitly requests a **white** foyer but the output displays blue or green tones); and **(3)** it raises potential copyright concerns (rows 2 and 8) as the generated images closely resemble artistic works by human creators used in games or other media.

However, memorization may sometimes benefit the model by generating highly accurate or detailed outputs that align well with user prompts—particularly when the underlying content is in the public domain, as illustrated in the final row. This dual nature of memorization highlights a promising direction for future research: developing methods to distinguish between harmful and benign memorization in diffusion models, similar to the approach taken by Aerni et al. (2024) for language models.

E Manual Labeling

As discussed in Section 3 of the main paper, existing methods are insufficient for accurately distinguishing between different types of memorization. This limitation motivates our proposed approach, *FB-Mem*. To rigorously evaluate the effectiveness of this method, it is essential to establish reliable ground-truth labels that reflect human perception. Therefore, we manually labeled the images generated from memorized prompts (without applying mitigation techniques) and used these labels as the ground truth for our classification experiments in the main paper.

Labeling Procedure. We adopt a conservative approach when identifying verbatim memorization (VM)—an image is labeled as VM only if it is visually identical to its corresponding training image. For template memorization (TM), we consider both locally similar and dissimilar regions, as well as the overall visual style, following the examples illustrated in Figure Figure 10. Each generated image is manually compared to up to ten of the most visually similar memorized images, as retrieved by BF-Mem and scored using SSCD, before reaching a labeling decision. Due to the time-intensive nature of this process, we labeled only the first 1,500 generated images, which we consider a sufficient sample for reliable evaluation.

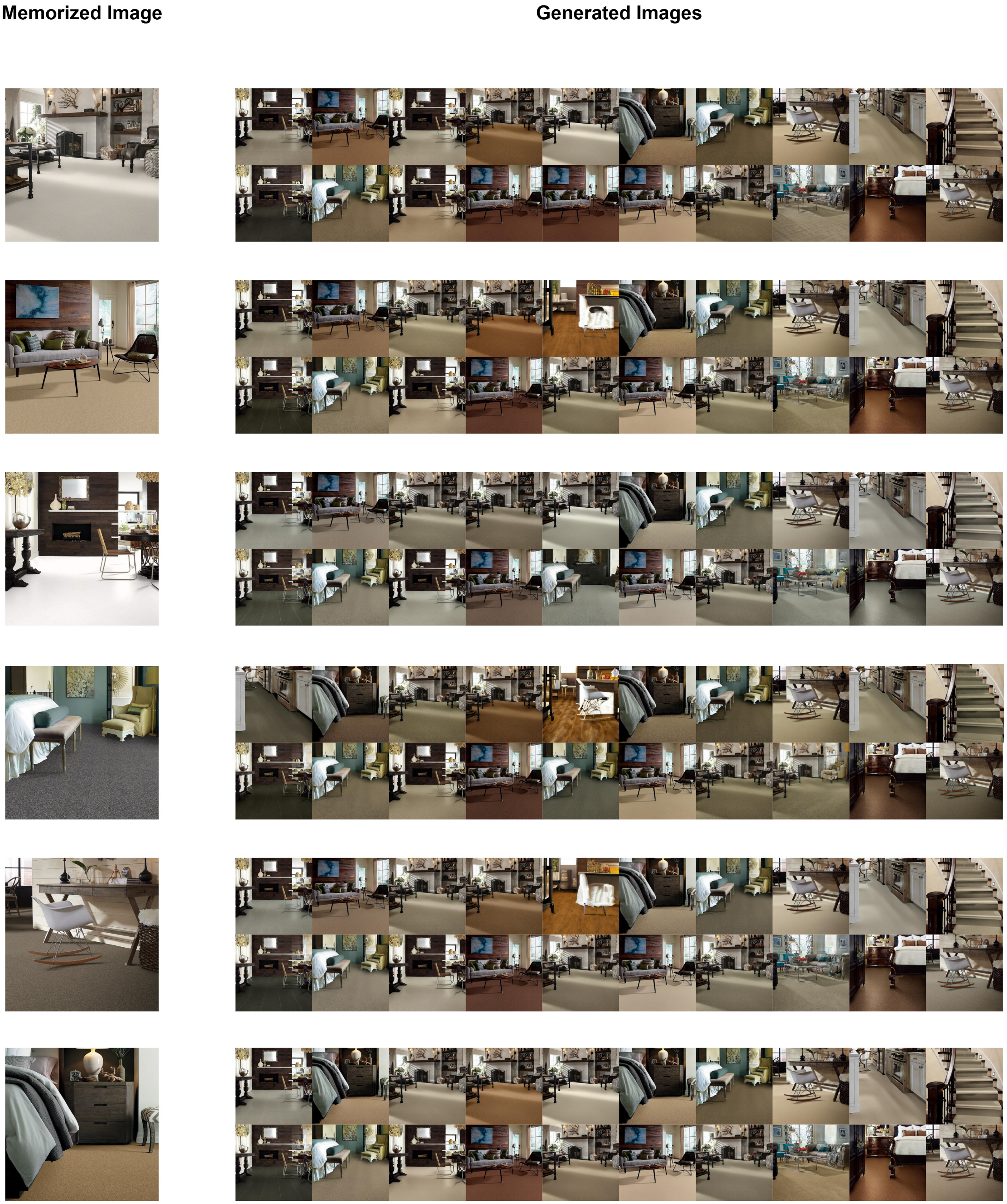

Figure 8: Twenty images generated across six different prompts selected from the **Shaw Floors** clusters.

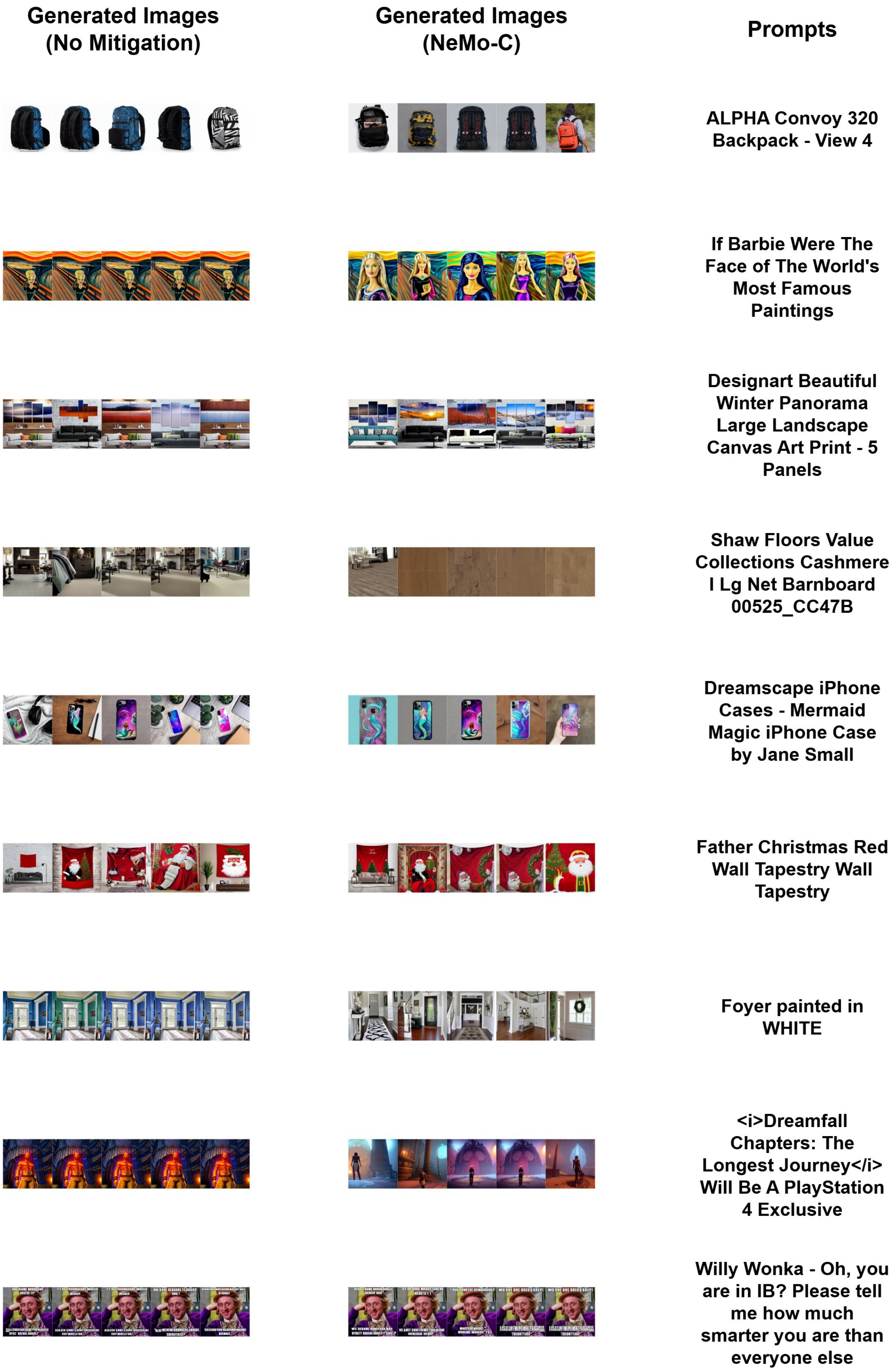
Generated Images (No Mitigation)
Generated Images (NeMo-C)
Prompts
ALPHA Convoy 320 Backpack - View 4
If Barbie Were The Face of The World's Most Famous Paintings
Designart Beautiful Winter Panorama Large Landscape Canvas Art Print - 5 Panels
Shaw Floors Value Collections Cashmere I Lg Net Barnboard 00525_CC47B
Dreamscape iPhone Cases - Mermaid Magic iPhone Case by Jane Small
Father Christmas Red Wall Tapestry Wall Tapestry
Foyer painted in WHITE
<i>Dreamfall Chapters: The Longest Journey</i> Will Be A PlayStation 4 Exclusive
Willy Wonka - Oh, you are in IB? Please tell me how much smarter you are than everyone else

Figure 9: Distribution of generated more images before and after applying mitigation methods

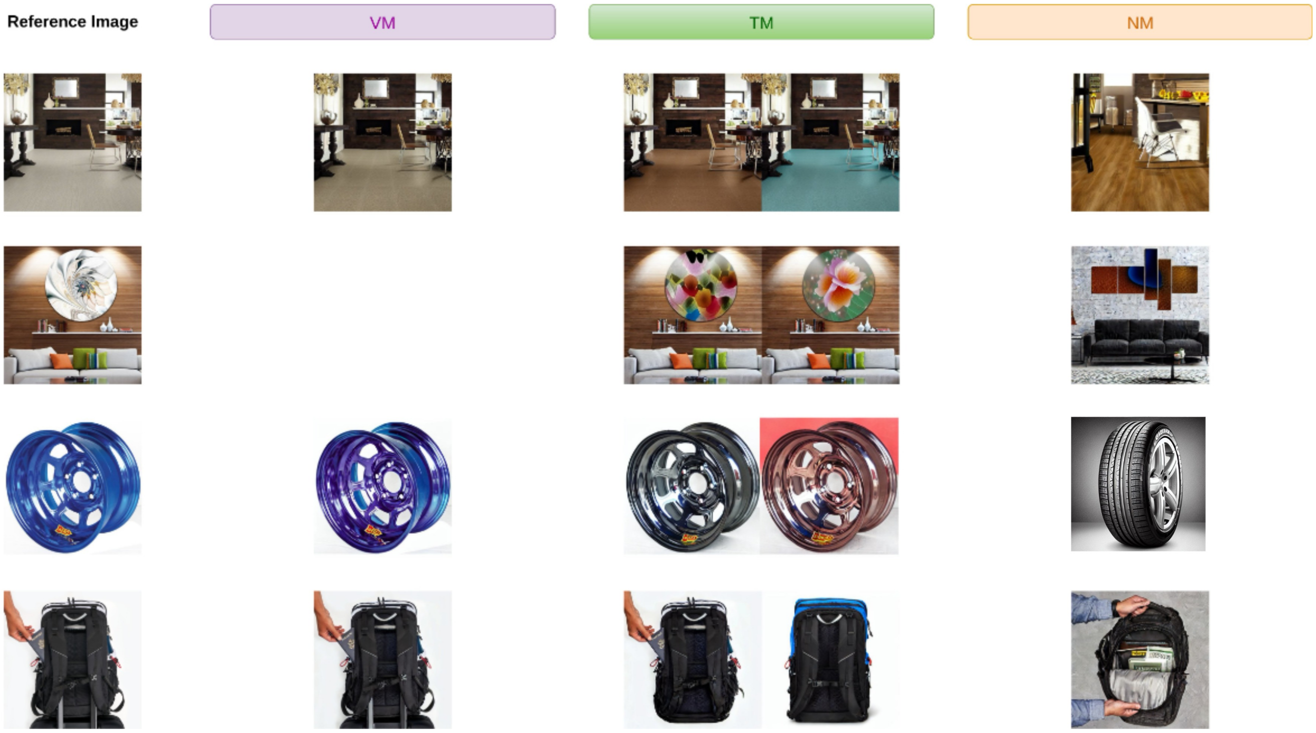

Figure 10: Examples of manual classification for different reference images. For the reference image in row 2, no VM example was found.