Demystifying Foreground-Background Memorization in Diffusion Models

by

Jimmy Z. Di

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2025

© Jimmy Z. Di 2025

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis is based on collaborative research conducted with Yiwei Lu, Yaoliang Yu, Gautam Kamath, Adam Dziedzic, and Franziska Boenisch. While the research ideas and overall direction were developed jointly, the writing presented herein is my own.

The corresponding manuscript is currently under review for publication at the 40th AAAI Conference on Artificial Intelligence (*AAAI-26*). My primary contributions include the design and implementation of methods and experiments in Python, the collection and analysis of results, and the drafting of the manuscript. Yiwei Lu provided guidance on methodological and experimental design and contributed to the writing process. Yaoliang Yu, Gautam Kamath, Adam Dziedzic, and Franziska Boenisch developed the foundational concepts and offered guidance in interpreting the results.

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.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Gautam Kamath, for his invaluable guidance, support, and encouragement throughout my master's studies. As I entered the field of computer science research, he welcomed me with patience and mentorship, guiding me through not only research challenges but every aspect of academic life. I could not have remained as calm, confident, and enthusiastic without his continuous inspiration, unwavering patience, and profound expertise. His high standards and meticulous approach to academic research—as well as his humorous and thoughtful social media posts—will continue to inspire and guide me throughout my future career.

I would like to thank the co-authors of the manuscript version of this thesis—Dr. Yiwei Lu and Dr. Yaoliang Yu at the University of Waterloo, as well as Dr. Adam Dziedzic and Dr. Franziska Boenisch at CISPA—for their valuable contributions and constructive feedback that strengthened this work. In particular, Dr. Yiwei Lu devoted a great deal of time to revising both the manuscript and the experimental design, significantly improving the clarity and rigor of our work. I am very grateful for his help and wish him all the best as he begins his role as an assistant professor at the University of Ottawa.

I am also grateful to my thesis reviewers, Dr. Xi He and Dr. Freda Shi, for their time, thoughtful comments, and valuable insights during the review process.

I am deeply grateful to my parents, who have been my strongest support throughout my academic journey, providing unwavering encouragement and belief in my abilities.

I would like to thank my friends—Joshua, Michael, Alex, Andrew, and many others at the University of Waterloo—for the laughs, joy, and camaraderie we've shared. You have undoubtedly made my academic journey more enjoyable and memorable. I also extend my appreciation to all the instructors from my undergraduate and graduate courses, whose teaching provided me with a strong foundation and sparked my interest in computer science and machine learning.

Finally, I would like to extend my heartfelt appreciation to all my friends, family members, and department staff not mentioned by name, for their kindness and unwavering support throughout this journey.

Dedication

To my beloved parents, for their unwavering love and support.

To the friendships I've gained, and the many lessons I've learned along the way.

Table of Contents

Author's Declaration ii

Statement of Contributions iii

Abstract iv

Acknowledgements v

Dedication vi

List of Figures ix

List of Tables xi

1 Introduction 1

2 Background 4

2.1 Diffusion Models 4

2.2 Semantic Segmentation and Object Detection 5

2.3 Memorization Problem 6

2.3.1 Memorization in Large Language Models 6

2.3.2 Memorization in Vision-Language Models 7

2.3.3 Memorization in Diffusion Models 7

2.4 Mitigating Memorization 10

3 Measuring Memorization **11**
3.1 Memorization Pipeline . . . 11
3.2 Measuring partial memorization . . . 12
3.3 Measuring *one-to-many* correspondence . . . 14

4 Mitigating Memorization **19**
4.1 Clustering-based Mitigation . . . 19
4.2 Mitigation Evaluation . . . 21
4.3 Experimental Settings . . . 21
4.4 Experimental Results . . . 22

5 Conclusions and Future Work **27**
5.1 Conclusions . . . 27

References **29**

APPENDICES **40**

A SSIM and Multiscale-SSIM **41**
A.1 SSIM . . . 41
A.2 MS-SSIM . . . 42

B Ablation Studies **43**
B.1 Additional Experiments . . . 43
B.1.1 Ablation Study on N . . . 43
B.2 NeMo-C with a Dampening Factor . . . 44

C Mitigation Examples **46**

D Manual Labeling of Ground-truth Images **48**
D.1 Manual Labeling . . . 48

List of Figures

1.1 **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). 1

2.1 Examples of images generated using Stable Diffusion v1.4 on the first 10 out of the 500 memorized prompts. Each prompt is used to generate 5 images, and the first 3 are included here. The first column shows the prompt used to generate the images. The second column shows the images generated by Stable Diffusion v1.4. The third column shows the corresponding image from the training dataset. 8

2.2 Examples of images generated using Stable Diffusion v1.4 on a selected subset of 10 out of the 500 memorized prompts. Each prompt is used to generate 5 images, and the first 3 are included here. The first column shows the prompt used to generate the images. The second column shows the images generated by Stable Diffusion v1.4. The third column shows the corresponding image from the training dataset. 9

3.1 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. 17

3.2 **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 $N = 20$. 18

4.1 **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*. 20

4.2 Memorization distribution evaluated using *FB-Mem* before/after mitigation, not memorized prompts, and Stable Diffusion 3. 23

4.3 *One-to-many* correspondence after applying mitigation methods, measured using *FB-Mem*. 25

4.4 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). The utility drop is calculated using the two methods shown in Table 4.2 26

B.1 Twenty images generated across six different prompts selected from the **Shaw Floors** clusters. 45

C.1 Examples of generated additional generated images before and after applying NeMo-C to mitigate memorization 47

D.1 Examples of manual classification for different reference images. For the reference image in row 2, no VM example was found. 49

List of Tables

3.1 Performance of each metric in classifying different memorization: 1) VM-NM; 2) TM-NM; 3) VM-TM. 11

4.1 Mitigation Strength Scoring Function. 21

4.2 Model performance assessed by image quality metrics before/after mitigation. The reported scores are computed for all 2,500 generated images and averaged. 24

B.1 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 4.1. The reported quality scores are computed for all 2,500 generated images and averaged. 44

Chapter 1

Introduction

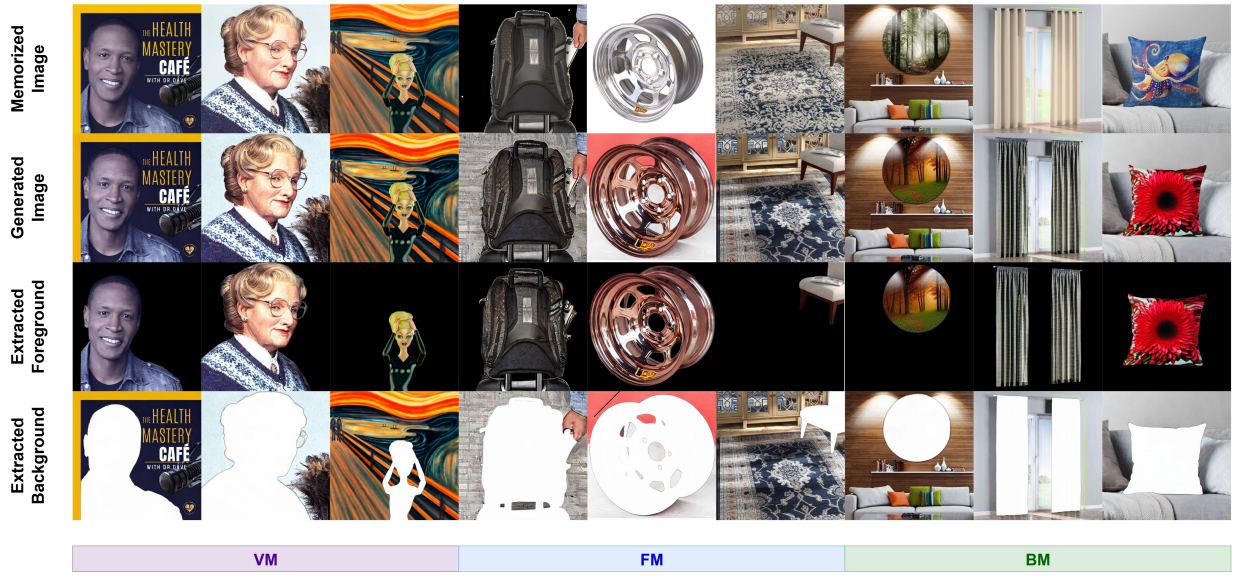


Figure 1.1: **Examples of different types of memorization under our *FB-Mem* evaluation.** We extract the foreground and background of the memorized images and generated images, and classify memorization using Algorithm 1 to: Verbatim Memorization (column 1-3), Foreground Memorization (column 4-6), and Background Memorization (column 7-9).

Diffusion models (DMs) [79, 82, 83] and their text-to-image derivatives (e.g., Latent Diffusion [65], DALL-E [63]) 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 [69], 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 [11, 80, 81]. 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., [10, 80, 92]) can accurately identify exact duplications using metrics like SSCD scores [61] or CLIP scores [62]. While [90] and [18] 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 [101] 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 [90]. However, [90] 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 mitigation strategies have been proposed, including inference-stage methods that adjust text-embedding or attention logits [64, 92], training-stage approaches that fine-tune pre-trained models [64, 92], and neuron-level in-

terventions [37, 14]. In this work, we focus on methods that permanently modify model weights [92, 37, 14], 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 linking 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.

The remainder of this thesis is organized as follows. Chapter 2 provides essential background on image segmentation, diffusion models (DMs), and the memorization problem in various machine learning models, including neural networks and language models. Chapter 3 addresses the limitations of existing memorization metrics and introduces our novel Foreground-Background Memorization (*FB-Mem*) metric, along with our method for prompt clustering. Chapter 4 presents our mitigation approach, *NeMo-C*, which combines the state-of-the-art memorization mitigation technique with prompt clustering. Finally, we summarize and conclude the thesis in Chapter 5, as well as outlining potential directions for future research.

Chapter 2

Background

2.1 Diffusion Models

Diffusion models [38, 82] such as Stable Diffusion [65] have become a state-of-the-art family of generative models. Diffusion models generate images using a two-step process: the first process, known as the forward step, gradually add Gaussian noise to a sample drew from a real-data distribution $x_0 \sim q(x)$ across T timesteps, culminating in $x_T \sim \mathcal{N}(0, \mathcal{I})$:

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_T\mathcal{I}),$$

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1}). \tag{2.1}$$

In equation 2.1, x_t represents the perturbed sample in each timestep T with added noise. The sequence $\{\beta_{t=1}^T\}$ is a hyperparameter that represents a noise scheduler which controls the amount of noise added in each time step during the diffusion process. A notable property of the forward process is that the sampling of x_t at any arbitiary timestep t can be written in a closed form: using the notation that $\alpha_t := 1 - \beta_t$ as well as $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, we have

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathcal{I}).$$

The joint distribution, $p_\theta(x_{0:T})$, is called the *reverse process* and defined as a Markov

chain with learned Gaussian transitions starting at $p(x_T) = \mathcal{N}(x_T; 0, I)$:

$$p_\theta(x_{T:0}) = p(x_T) \prod_{t=T}^{1} p_\theta(x_{t-1} \mid x_t),$$

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t), \mathbf{\Sigma}_\theta(x_t, t)),$$

where $\mu_\theta(x_T)$ is the approximated mean and $\mathbf{\Sigma}_\theta(x_t, t) = \sigma_t^2 \mathcal{I}$ is the approximated variance of $q(x_{t-1} \mid x_t; x_0)$. By applying parameterization, we can obtain the learning objective:

$$\mathcal{L} = \mathbb{E}_{x_o,\epsilon}\big[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1-\bar{\alpha}_t)} \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|_2^2\big],$$

where $\epsilon \sim \mathcal{N}(0, I)$ is the noise and ϵ_θ is a denoiser network intended to predict ϵ using the provided sample x_t.

2.2 Semantic Segmentation and Object Detection

Image segmentation has been a long-standing challenge in the realm of computer vision and image processing. Distinct from classification, where the goal is to assign a given image to one of the provided categories [19], image segmentation aims to partition pixels within an image into distinct groups [102]. Traditionally, image segmentation has been accomplished using approaches such as thresholding [60] and spatial clustering [76]. These methods are often limited in their ability to handle complex visual scenes with varying illumination conditions and semantic ambiguity.

In recent years, the emergence of deep learning has dramatically transformed this landscape, enabling models to learn rich feature representations from large datasets and solve problems that were previously intractable. RNN-based architectures [5], transformer-based models [101], and methods built on foundation models [43] have opened new possibilities in high-level computer vision tasks such as object detection and semantic segmentation. These tasks often require a more fine-grained representation of each region and understanding of visual elements within an image [102] and the ability of deep learning models

to capture long-range dependencies and contextual relationships has significantly improved performance across diverse scenarios.

In this thesis, we adopt the transformer-based Bi-lateral Reference Framework (BiRefNet) proposed in [101]. BiRefNet explicitly decomposes the segmentation task into two modules, the localization module (LM) and the reconstruction module (RM). Our experiments demonstrate that BiRefNet performs effectively on high-resolution dichotomous images. Furthermore, unlike many interactive methods, BiRefNet does not rely on user input, such as mouse clicks or bounding box annotations, to guide segmentation. This makes it a scalable and suitable choice for our purpose.

2.3 Memorization Problem

Despite recent advancements made in machine learning (ML) and the widespread popularity of generative deep learning models, it remains challenging to understand what ML models learn from their training datasets. Memorization, the phenomenon where ML models memorize particular features from their training data instead of learning generalizable patterns, is a well-demonstrated issue in both supervised learning [3] and self-supervised learning [87].

Contrary to the popular belief that overfitting-induced memorization degrades test performance, a recent study [87] has shown that memorization can, in fact, improve model generalization. This insight has sparked researchers' interest in understanding generalization in deep neural networks (DNNs). [91].

However, the memorization problem also raises ethical and safety concerns in ML models, as it could also lead to critical privacy leakage and vulnerability such as membership inference [47], data extraction [9, 10, 12], and data poisoning attacks [59]. In this section, we briefly review the current state of research and provide some examples of memorization across different families of generative models.

2.3.1 Memorization in Large Language Models

Memorization of large language models (LLMs) is relatively well studied, and LLMs are known to memorize portions of their training data, including real-world facts and commonsense knowledge, in order to generate fluent text and respond effectively to user prompts. Given the vast size of LLM training datasets, it was previously assumed that the extent of

exact copying for any individual source would be minimal—"the degree of copying with respect to any given work is likely to be, at most, de minimis" [85]—and that language models would not retain substantial memorization of specific, particularly copyrighted excerpts. However, [10] demonstrated through GPT-2 data extraction attacks that this assumption does not hold universally: they identified over 500 unique memorized training examples, including many live URLs, with a true positive rate of 33.5%. Future research also demonstrated the ability to extract memorized data from other open-source or semi-open-source large-scale models such as Pythia, GPT-Neo and LLaMA[57].

2.3.2 Memorization in Vision-Language Models

Vision-Language Models (VLMs) have emerged as state-of-the-art for image-text-related tasks, such as captioning and classification. VLMs indeed demonstrate memorization when performing these tasks as well. For classification tasks, previous work [53] has shown that an image classifier can predict a memorized image as its training label even if the object is cropped out. For example, for an image of a black swan on water, a representation learning model may learn to predict *black swan* given the background *water*. However, due to the presence of multiple modalities in VLMs, text and images, as well as the complex nature of the training data, it is difficult to distinguish between memorization or spurious correlation (i.e., most training images of a black swan also contained water). Instead, [40] defined the memorization of VLM in the captioning scenario: for a given vision language model f and an image-text pair $z = \langle z_{image}, z_{text} \rangle$, we say z is memorized by f if more unique objects can be recovered from z_{image} given z_{text} when z is present in the training set of f compared to when it is not. Using this definition, [40] concluded that even complex multi-modal ML models may suffer significant memorization.

2.3.3 Memorization in Diffusion Models

Diffusion models (DMs), predominantly trained on large-scale public datasets scraped from the Internet (e.g., LAION-5B [70]), have been shown to suffer from memorization. Specifically, when prompted with certain inputs that appear in the training data, these models can reproduce near-identical copies of the corresponding images [11]. For instance, in the case of Stable Diffusion v1.4, [90] identified 500 such memorized prompts that consistently yield generated images highly similar to those in the training set.

Figure 2.1 presents examples of these memorized generations, highlighting the visual similarity between the generated outputs and the corresponding training images. However,



Figure 2.1: Examples of images generated using Stable Diffusion v1.4 on the first 10 out of the 500 memorized prompts. Each prompt is used to generate 5 images, and the first 3 are included here. The first column shows the prompt used to generate the images. The second column shows the images generated by Stable Diffusion v1.4. The third column shows the corresponding image from the training dataset.

we observe that not all training images are memorized equally - some prompts result in stronger memorization effects than others. For example, the prompts shown in Figure 2.2 appear to be less memorized compared to those in Figure 2.1; the model no longer

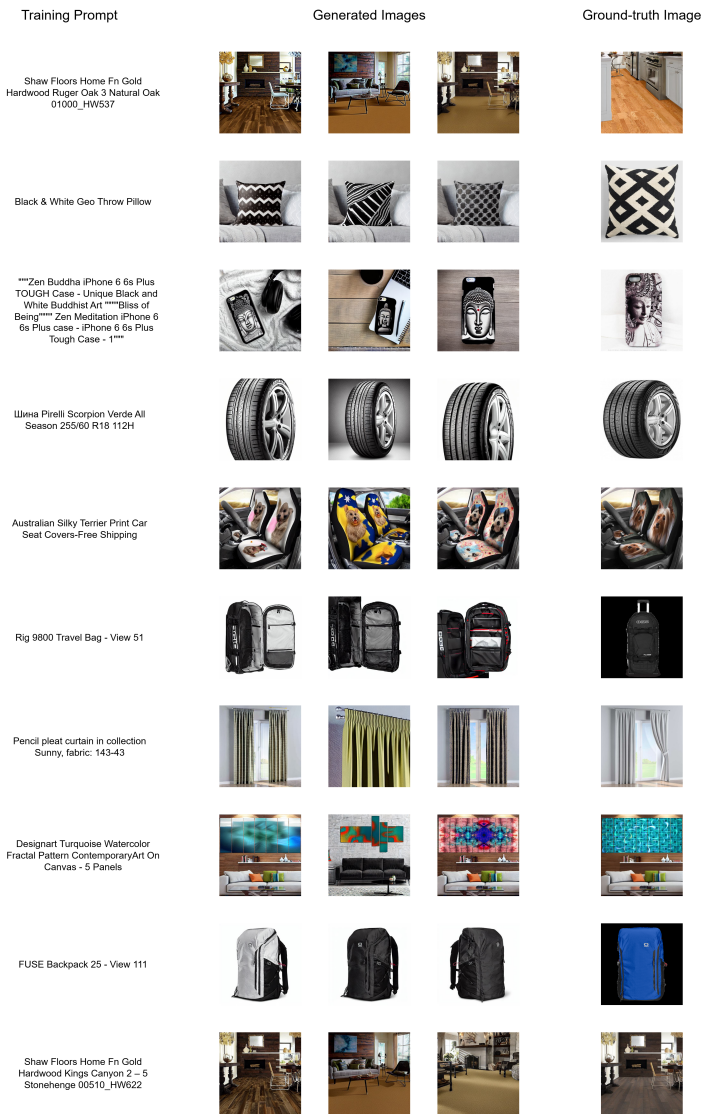


Figure 2.2: Examples of images generated using Stable Diffusion v1.4 on a selected subset of 10 out of the 500 memorized prompts. Each prompt is used to generate 5 images, and the first 3 are included here. The first column shows the prompt used to generate the images. The second column shows the images generated by Stable Diffusion v1.4. The third column shows the corresponding image from the training dataset.

consistently reproduces exact copies of the training images, sometimes producing images that are dis-similar to the ground-truth image.

To detect such memorization in diffusion models, various approaches have been proposed, including SSCD scores [61], CLIP scores [65], pairwise SSIM scores between initial noise differences [90], distribution of attention [64], edge inconsistency [90], and predicted noise magnitudes [92].

2.4 Mitigating Memorization

To address the problem of memorization for DMs, various methods have been proposed. For example, inference-stage mitigation, including attention logit rescaling [64] and text-embedding adjustment [92]; training-stage mitigation by fine-tuning an existing DM-based model [64, 92]; and neuron-level mitigation [52, 37, 14] 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 [92], the neuron deactivation approach [37], and the weight pruning approach [14].

Other techniques are also potentially applicable for mitigating memorization, for example, machine unlearning (proposed for removing private personal data) [6, 4, 72, 2, 94] or concept removal (proposed for removing nudity or harmful concepts) [15, 26, 51] can potentially be used to remove the memorized information.

Chapter 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.

Table 3.1: 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

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 [65]. While newer models exist, no datasets with memorized data are available for them. Consequently, state-of-the-art work [90, 92, 37] considers 500 memorized LAION prompts for SD v1.4, which we adopt in our paper.[1] [37] further split this dataset into Verbatim Memorization

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

(VM) and Template Memorization (TM) categories using SSCD scores [61], which represent the cosine similarity of image embeddings obtained from the Self-Supervised Copy Detection (SSCD) model. Specifically, [37] 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 [18]. Each image was reviewed and labeled as VM, TM, or NM based on its visual resemblance to ground-truth images. We show examples of the labeling in Figure D.1 as well as presenting the complete manual-labeling protocol is described in detail in Appendix D.1.

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. [37] and [90] 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 1.1 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.

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

[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.

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 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) [88] 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 re-scaling 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 Appendix A. 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 manu-

[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).

ally 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] retrievable 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 [37] and [92], using a verbatim memorization (VM) threshold of 0.7 and a template memorization (TM) threshold of 0.5. Classification performance is reported in Table 3.1.

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.

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 3.2 (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.

[4] A total of two images were unavailable due to broken URLs. We made reasonable effort in an attempt to retrieve the ground-truth images from other works

[5] This aligns with recent findings by [18], which highlight SSCD's vulnerability to local perturbations.

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 3.1 shows examples of five prompts sampled from the same cluster, along with their corresponding generated images.[6] Additional examples are provided in Appendix B.1.

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 3.2 (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.

[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.

Algorithm 1 Foreground Background Memorization *(FB-Mem)*

Input: Generated image $\mathbf{x}_g$, training image $\mathbf{x}_t$, similarity metric M, score threshold τ, segmentation threshold β

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 $\frac{|S_f|}{|x_g|} \leq \beta$ **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 $\frac{|S_f|}{|x_g|} \geq 1 - \beta$ **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 $M_{\text{full}} \geq \tau$ **then**
    return VM (Verbatim Memorization)
else if $M_{\text{fg}} \geq \tau$ **then**
    return FM (Foreground Memorization)
else if $M_{\text{bg}} \geq \tau$ **then**
    return BM (Background Memorization)
else
    return NM (Not Memorized)
end if



Figure 3.1: 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.

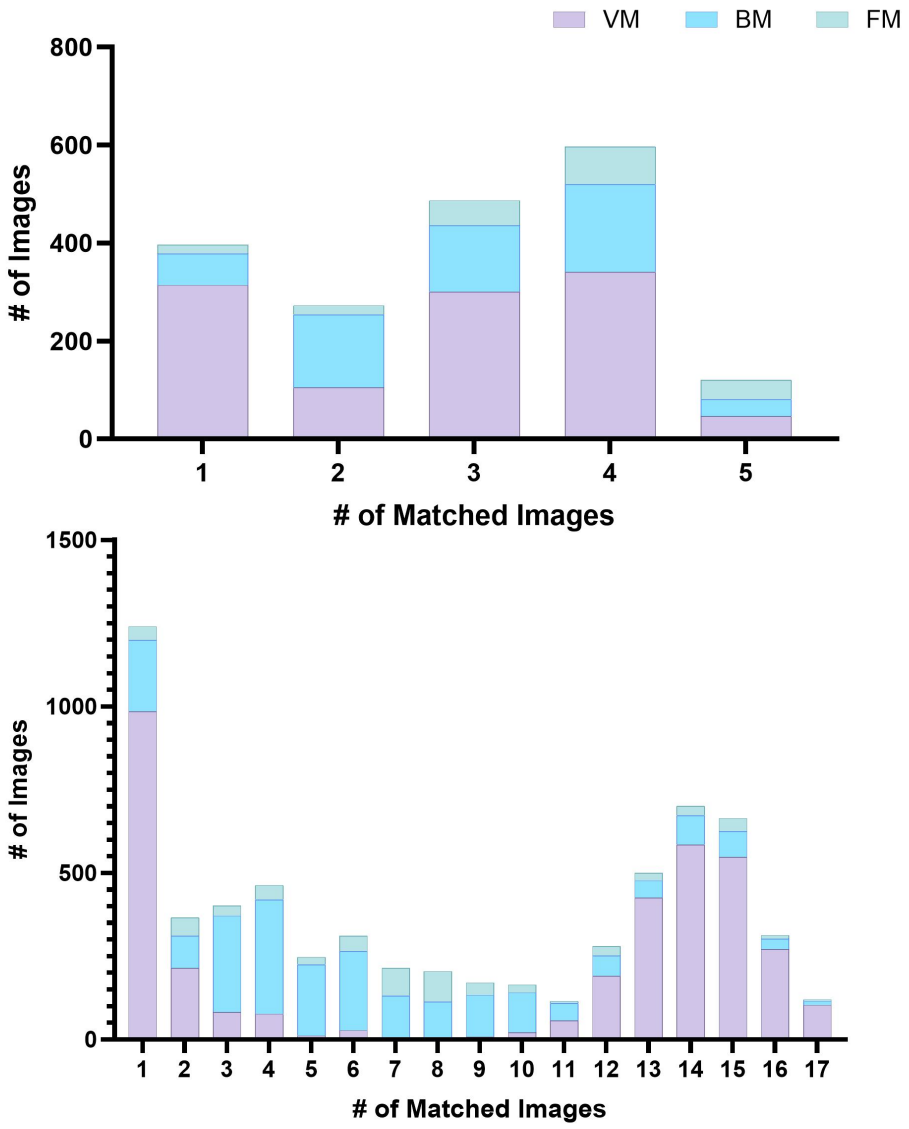


Figure 3.2: **Quantitative analysis of memorization in generated images** measured by *FB-Mem*. The x-axis represents the number of distinct ground-truth images matched using each of the N images generated by each prompt, while the y-axis indicates the number of generated images identified as memorized 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 $N = 20$.

Chapter 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 [37], 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



Figure 4.1: **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.*

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 failure cases where generated images remain verbatim memorized post-mitigation, in Figure 4.1. Additional examples are provided in Appendix C.

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 4.1: 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 4.1 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 that reduce memorization severity receive positive scores, while those that increase memorization or introduce new memorization patterns receive negative penalties.

Image Quality: We use a Deep Bilinear Convolutional Neural Network (*DB-CNN*) [100] to evaluate the quality of images after applying memorization mitigation. Due to its no-reference (NR) nature, the method performs well even with our relatively small sample size. The method was implemented using the Image Quality Assessment (IQA) toolbox.

4.3 Experimental Settings

Baseline Methods: We consider three model-based mitigation methods: NeMo [37], Wanda [14],[1] and DetectMem [92]. All mitigation methods are applied using their default

[1]Note that Wanda was originally proposed by [84] for large language models, while we utilize its adaptation for diffusion models as presented in [14].

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 [90] 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 Chapter 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 4.2, 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,[2] matching the observations of [37]. Moreover, in Figure 4.3, we demonstrate that *one-to-many* correspondence still largely persists across methods, with NeMo-C showing slight improvements over baselines.

[2]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.

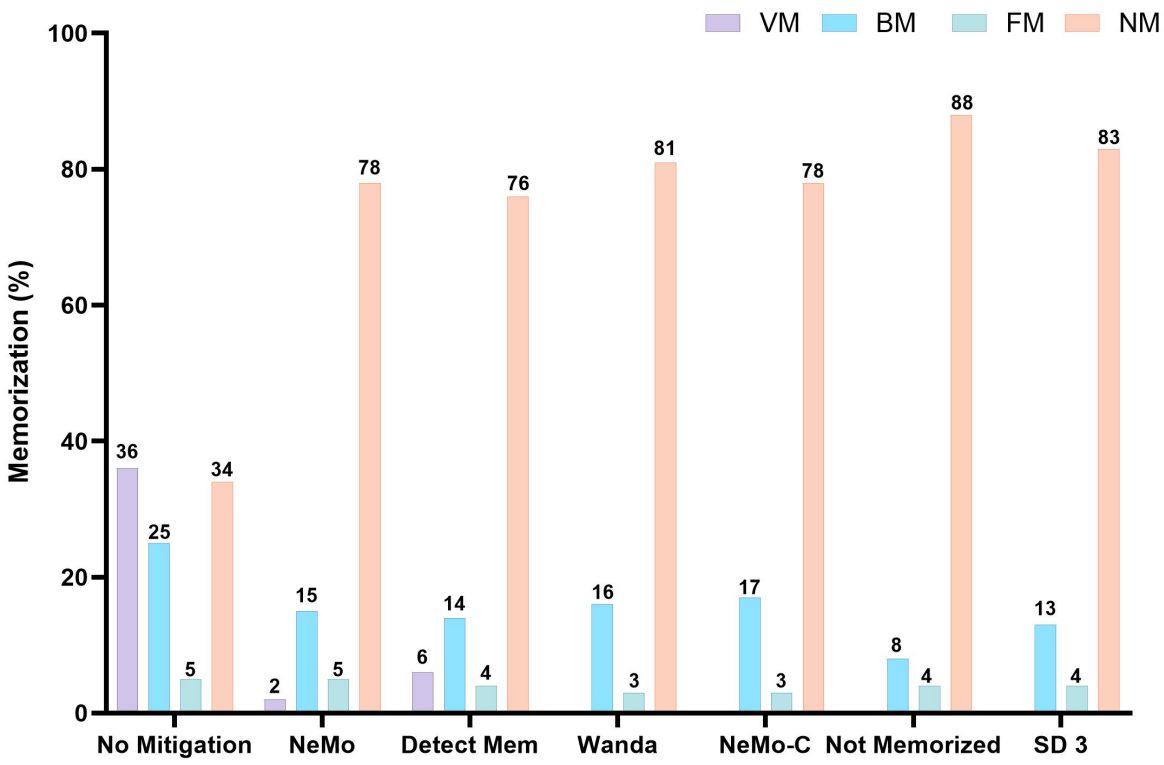


Figure 4.2: Memorization distribution evaluated using *FB-Mem* before/after mitigation, not memorized prompts, and Stable Diffusion 3.

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 4.1. 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 criterion for evaluating mitigation methods is the preservation of image quality. In Table 4.2, we report the average quality of generated images after applying various mitigation techniques. Among all approaches, NeMo and NeMo-C achieve the highest image quality. Furthermore, the trade-off between mitigation strength and utility degradation is illustrated in Figure 4.4, where NeMo-C emerges as the most balanced and effective mitigation method.

Additionally, we demonstrate that the trade-off between mitigation efficacy and image quality can be further fine-tuned. For both NeMo and our proposed NeMo-C method (as described in this section), the identified set of neurons in the U-Net is completely deactivated. As an alternative, these neurons can be dampened rather than entirely disabled. Specifically, we apply a multiplicative dampening factor, α_{damp}, to the selected neurons, where $\alpha_{\text{damp}} = 0$ corresponds to standard NeMo-C. As expected, increasing the dampening factor leads to improved image quality while slightly reducing mitigation efficacy. The extended NeMo-C method and detailed experimental settings are described in Appendix B.2.

Table 4.2: 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
Pre-mitigation	0.60
NeMo	0.587
Detect Mem	0.449
Wanda	0.578
NeMo-C	0.586

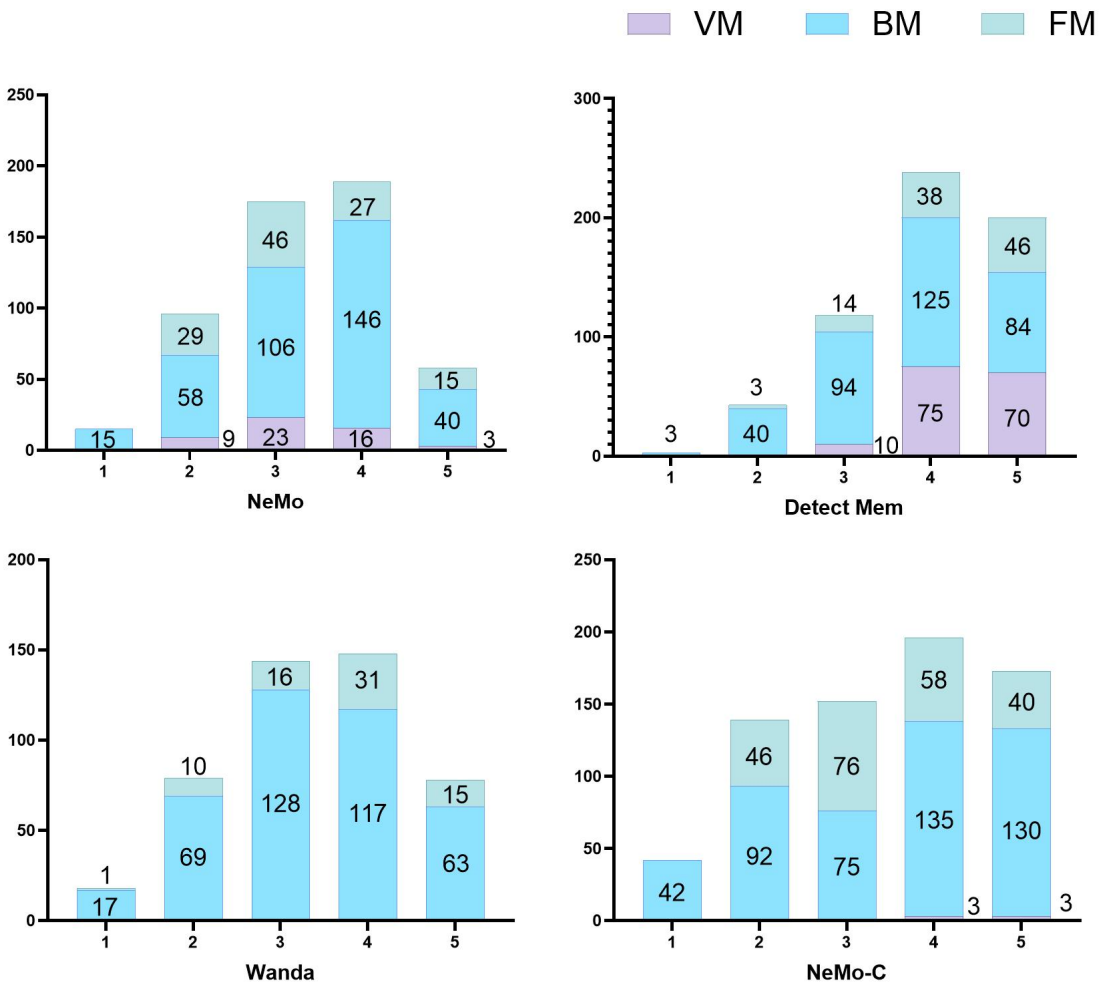


Figure 4.3: *One-to-many* correspondence after applying mitigation methods, measured using *FB-Mem*.

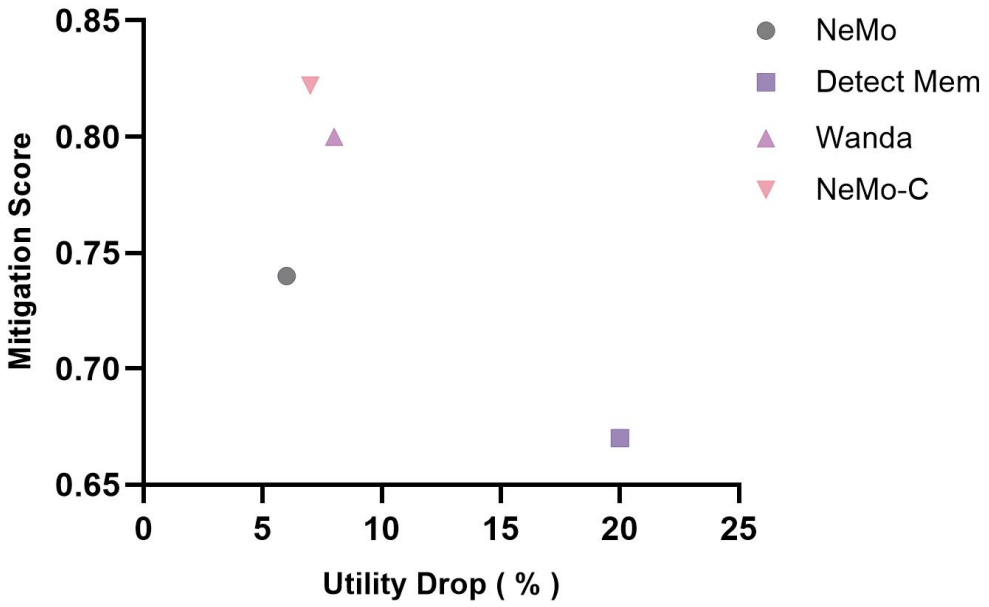


Figure 4.4: Trade-off between utility and mitigation efficacy. The y-axis shows the mitigation score (higher is better), while the x-axis indicates the drop in image quality compared to the non-mitigated baseline (lower is better). The utillity drop is calculated using the two methods shown in Table 4.2

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This thesis addressed fundamental limitations in memorization detection for diffusion models (DMs) by introducing FB-Mem, a segmentation-based metric enabling fine-grained classification of memorized content. Our analysis revealed that memorization is inherently cluster-wise rather than prompt-wise, with individual generations often combining elements from multiple training images. Using FB-Mem, we showed that existing mitigation methods fail to eliminate local memorization, particularly in foreground regions. To address this gap, we proposed NeMo-C, a cluster-wise mitigation strategy that achieves more robust memorization reduction while preserving model utility.

Beyond these contributions, our findings suggest several promising research directions. First, mitigation strategies should be carefully evaluated, balancing memorization suppression with performance degradation. Second, the phenomenon whereby memorized prompts can yield non-memorized outputs raises an important question: can memorized content also be reproduced through non-adversarial, natural prompts in real-world usage, as observed in prior studies of large language models [1]? Finally, our qualitative analysis in Appendix C of the generated images revealed that memorization is not uniformly harmful. In certain cases, such as faithful reproduction of public-domain images, may be desirable. A deeper understanding of this benign–malignant memorization spectrum could enable mitigation approaches that selectively retain beneficial memorization while minimizing risks.

By establishing a principled measurement framework and uncovering the nuanced nature of memorization in diffusion models, this work lays the foundation for future advances

in detection, evaluation, and targeted mitigation, possibly extending to other generative modalities such as large language models. Ultimately, understanding and controlling memorization is not just a technical challenge but a prerequisite for building generative models that are both powerful and trustworthy. Our work takes a decisive step toward that goal, setting the stage for safer and more responsible deployment of generative systems.

References

[1] Michael Aerni, Javier Rando, Edoardo Debenedetti, Nicholas Carlini, Daphne Ippolito, and Florian Tramèr. Measuring non-adversarial reproduction of training data in large language models, 2024.

[2] Nasser Aldaghri, Hessam Mahdavifar, and Ahmad Beirami. Coded machine unlearning. *IEEE Access*, 9:88137–88150, 2021.

[3] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017.

[4] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE Symposium on Security and Privacy (SP)*, pages 141–159, 2021.

[5] Wonmin Byeon, Thomas M. Breuel, Federico Raue, and Marcus Liwicki. Scene labeling with lstm recurrent neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3547–3555, 2015.

[6] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE Symposium on Security and Privacy*, pages 463–480, 2015.

[7] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2022.

[8] Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. The privacy onion effect: Memorization is relative. *Advances in Neural Information Processing Systems*, 35:13263–13276, 2022.

[9] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284, 2019.

[10] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650, 2021.

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

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

[13] Ting-Yun Chang, Jesse Thomason, and Robin Jia. Do localization methods actually localize memorized data in llms? *arXiv preprint arXiv:2311.09060*, 2023.

[14] Ruchika Chavhan, Ondrej Bohdal, Yongshuo Zong, Da Li, and Timothy Hospedales. Memorized images in diffusion models share a subspace that can be located and deleted. *arXiv preprint arXiv:2406.18566*, 2024.

[15] Ruchika Chavhan, Da Li, and Timothy Hospedales. Conceptprune: Concept editing in diffusion models via skilled neuron pruning. *arXiv preprint arXiv:2405.19237*, 2024.

[16] Chaofeng Chen and Jiadi Mo. IQA-PyTorch: Pytorch toolbox for image quality assessment. [Online]. Available: https://github.com/chaofengc/IQA-PyTorch, 2022.

[17] Chen Chen, Daochang Liu, Mubarak Shah, and Chang Xu. Exploring local memorization in diffusion models via bright ending attention. *arXiv preprint arXiv:2410.21665*, 2024.

[18] Chen Chen, Daochang Liu, Mubarak Shah, and Chang Xu. Exploring local memorization in diffusion models via bright ending attention, 2025.

[19] Leiyu Chen, Shaobo Li, Qiang Bai, Jing Yang, Sanlong Jiang, and Yanming Miao. Review of image classification algorithms based on convolutional neural networks. *Remote Sensing*, 13(22), 2021.

[20] Giannis Daras, Yuval Dagan, Alex Dimakis, and Constantinos Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. *Advances in Neural Information Processing Systems*, 36, 2024.

[21] Giannis Daras, Alexandros G Dimakis, and Constantinos Daskalakis. Consistent diffusion meets tweedie: Training exact ambient diffusion models with noisy data. arXiv preprint arXiv:2404.10177, 2024.

[22] Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823, 2024.

[23] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, 2021.

[24] Jimmy Z. Di, Jack Douglas, Jayadev Acharya, Gautam Kamath, and Ayush Sekhari. Hidden poison: Machine unlearning enables camouflaged poisoning attacks. In *Advances in Neural Information Processing Systems*, 2023.

[25] Ronen Eldan and Mark Russinovich. Who's Harry Potter? approximate unlearning in LLMs. 2023.

[26] Rohit Gandikota, Joanna Materzyńska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2426–2436, 2023.

[27] Daniel Gatis. rembg, February 2025.

[28] Antonio A. Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. Making ai forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems*, 2019.

[29] Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 792–801, 2021.

[30] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9301–9309, 2020.

[31] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *European Conference on Computer Vision*, pages 383–398, 2020.

[32] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1994.

[33] Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in diffusion models. *arXiv preprint arXiv:2310.02664*, 2023.

[34] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3832–3842, 2020.

[35] Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. In *Advances in Neural Information Processing Systems*, 2021.

[36] Alvin Heng and Harold Soh. Selective amnesia: A continual learning approach to forgetting in deep generative models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[37] Dominik Hintersdorf, Lukas Struppek, Kristian Kersting, Adam Dziedzic, and Franziska Boenisch. Finding nemo: Localizing neurons responsible for memorization in diffusion models. *arXiv preprint arXiv:2406.02366*, 2024.

[38] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[39] Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas Carlini. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 28–53, 2023.

[40] Bargav Jayaraman, Chuan Guo, and Kamalika Chaudhuri. Déjà vu memorization in vision-language models, 2024.

[41] Shuli Jiang, Swanand Ravindra Kadhe, Yi Zhou, Ling Cai, and Nathalie Baracaldo. Forcing generative models to degenerate ones: The power of data poisoning attacks. In *NeurIPS Workshop on Backdoors in Deep Learning: The Good, the Bad, and the Ugly*. 2023.

[42] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[43] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

[44] Donald Knuth. *The TEXbook*. Addison-Wesley, Reading, Massachusetts, 1986.

[45] Zhifeng Kong and Kamalika Chaudhuri. Data redaction from conditional generative models. In *2nd IEEE Conference on Secure and Trustworthy Machine Learning*, 2024.

[46] Leslie Lamport. *LATEX — A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994.

[47] Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated white-box membership inference, 2020.

[48] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[49] Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples. In *International Conference on Machine Learning*, pages 20763–20786. PMLR, 2023.

[50] Zhiheng Liu, Ruili Feng, Kai Zhu, Yifei Zhang, Kecheng Zheng, Yu Liu, Deli Zhao, Jingren Zhou, and Yang Cao. Cones: Concept neurons in diffusion models for customized generation. *arXiv preprint arXiv:2303.05125*, 2023.

[51] Mengyao Lyu, Yuhong Yang, Haiwen Hong, Hui Chen, Xuan Jin, Yuan He, Hui Xue, Jungong Han, and Guiguang Ding. 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*, pages 7559–7568, 2024.

[52] Pratyush Maini, Michael C Mozer, Hanie Sedghi, Zachary C Lipton, J Zico Kolter, and Chiyuan Zhang. Can neural network memorization be localized? In *Proceedings of the 40th International Conference on Machine Learning*, pages 23536–23557, 2023.

[53] Casey Meehan, Florian Bordes, Pascal Vincent, Kamalika Chaudhuri, and Chuan Guo. Do ssl models have déjà vu? a case of unintended memorization in self-supervised learning, 2023.

[54] Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022.

[55] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022.

[56] Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

[57] Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models, 2023.

[58] Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *International Conference on Algorithmic Learning Theory*, pages 1–32, 2021.

[59] Duc Anh Nguyen, Ron Levie, Julian Lienen, Gitta Kutyniok, and Eyke Hüllermeier. Memorization-dilation: Modeling neural collapse under label noise, 2023.

[60] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[61] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection, 2022.

[62] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[63] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[64] 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. In *European Conference on Computer Vision*, pages 340–356. Springer, 2024.

[65] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[66] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[67] Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. Editing a classifier by rewriting its prediction rules. In *Advances in Neural Information Processing Systems*, 2021.

[68] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22522–22531, 2023.

[69] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[70] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022.

[71] Christoph Schuhmann, Robert Kaczmarczyk, Aran Komatsuzaki, Aarush Katta, Richard Vencu, Romain Beaumont, Jenia Jitsev, Theo Coombes, and Clayton Mullis. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. In *NeurIPS Workshop Datacentric AI*, number FZJ-2022-00923. Jülich Supercomputing Center, 2021.

[72] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems*, 2021.

[73] Shawn Shan, Wenxin Ding, Josephine Passananti, Stanley Wu, Haitao Zheng, and Ben Y. Zhao. Nightshade: Prompt-specific poisoning attacks on text-to-image generative models. In *IEEE Symposium on Security and Privacy*, 2024.

[74] Shawn Shan, Wenxin Ding, Josephine Passananti, Haitao Zheng, and Ben Y Zhao. Prompt-specific poisoning attacks on text-to-image generative models. arXiv preprint arXiv:2310.13828, 2023.

[75] Aakash Sen Sharma, Niladri Sarkar, Vikram Chundawat, Ankur A Mali, and Murari Mandal. Unlearning or concealment? a critical analysis and evaluation metrics for unlearning in diffusion models. *arXiv preprint arXiv:2409.05668*, 2024.

[76] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[77] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[78] Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitry Pyrkin, Sergei Popov, and Artem Babenko. Editable neural networks. In *International Conference on Learning Representations*, 2020.

[79] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265, 2015.

[80] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion

models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6048–6058, 2023.

[81] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[82] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[83] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[84] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

[85] Kit Walsh. Uspto request for comments on intellectual property protection for artificial intelligence innovation – public comment by the electronic frontier foundation., 2020.

[86] Jianyi Wang, Kelvin C. K. Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images, 2022.

[87] Wenhao Wang, Muhammad Ahmad Kaleem, Adam Dziedzic, Michael Backes, Nicolas Papernot, and Franziska Boenisch. Memorization in self-supervised learning improves downstream generalization. *arXiv preprint arXiv:2401.12233*, 2024.

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

[89] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[90] Ryan Webster. A reproducible extraction of training images from diffusion models, 2023.

[91] Jiaheng Wei, Yanjun Zhang, Leo Yu Zhang, Ming Ding, Chao Chen, Kok-Leong Ong, Jun Zhang, and Yang Xiang. Memorization in deep learning: A survey, 2024.

[92] 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.

[93] Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Chunyi Li, Liang Liao, Annan Wang, Erli Zhang, Wenxiu Sun, Qiong Yan, Xiongkuo Min, Guangtai Zhai, and Weisi Lin. Q-align: Teaching lmms for visual scoring via discrete text-defined levels. *International Conference on Machine Learning (ICML)*, 2024. Equal Contribution by Wu, Haoning and Zhang, Zicheng. Project Lead by Wu, Haoning. Corresponding Authors: Zhai, Guangtai and Lin, Weisi.

[94] Jing Wu, Trung Le, Munawar Hayat, and Mehrtash Harandi. Erasediff: Erasing data influence in diffusion models. 2024.

[95] Tianwei Xiong, Yue Wu, Enze Xie, Zhenguo Li, and Xihui Liu. Editing massive concepts in text-to-image diffusion models. *arXiv preprint arXiv:2403.13807*, 2024.

[96] Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. In *The Conference on Empirical Methods in Natural Language Processing*, 2023.

[97] Lang Yu, Qin Chen, Jie Zhou, and Liang He. Melo: Enhancing model editing with neuron-indexed dynamic lora. arXiv preprint arXiv:2312.11795, 2023.

[98] Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Counterfactual memorization in neural language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[99] Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. A comprehensive study of knowledge editing for large language models. 2024.

[100] Weixia Zhang, Kede Ma, Jia Yan, Dexiang Deng, and Zhou Wang. Blind image quality assessment using a deep bilinear convolutional neural network. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(1):36–47, January 2020.

[101] Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation, 2024.

[102] Tianfei Zhou, Wang Xia, Fei Zhang, Boyu Chang, Wenguan Wang, Ye Yuan, Ender Konukoglu, and Daniel Cremers. Image segmentation in foundation model era: A survey, 2024.

APPENDICES

Appendix A

SSIM and Multiscale-SSIM

A.1 SSIM

First, we provide a brief overview of the Structural Similarity Index (SSIM) [89] and the improved Multiscale Structural Similarity Index (MS-SSIM) [88]. 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 in the case of a weak denominator. With three components set to equal importance, we obtain the SSIM score:

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

A.2 MS-SSIM

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.

Appendix B

Ablation Studies

B.1 Additional Experiments

B.1.1 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 Chapter 3, we use the same 500 memorized prompts from [90]. 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 B.1 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 3.1. 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.

B.2 NeMo-C with a Dampening Factor

For both NeMo [37] 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 B.1: 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 4.1. 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 B.1 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.



Figure B.1: Twenty images generated across six different prompts selected from the **Shaw Floors** clusters.

Appendix C

Mitigation Examples

In this section, we present additional examples of generated images along with their corresponding memorized training images and prompts, as shown in Figure C.1. 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 which appear to be 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 [1] for language models.

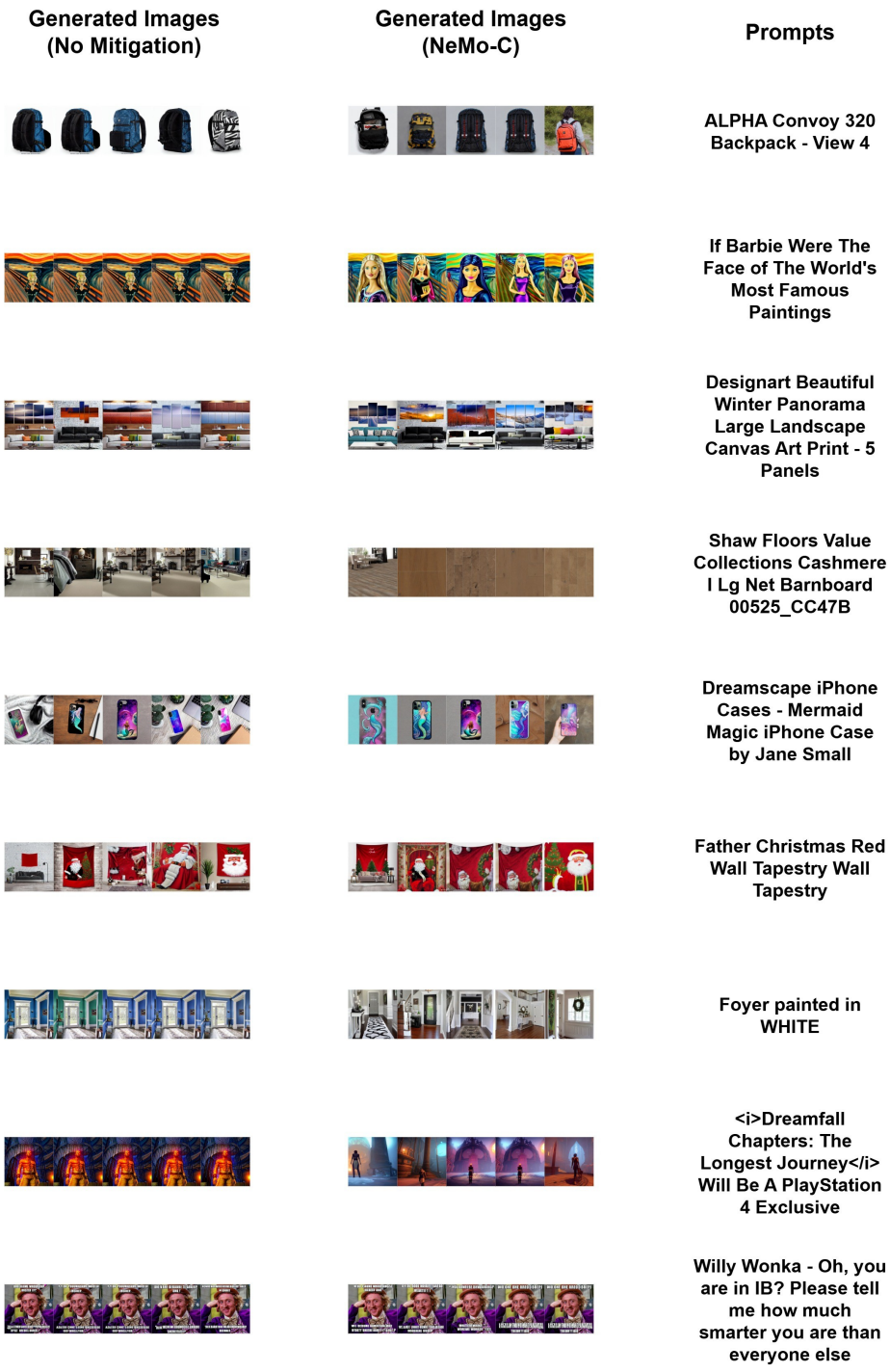


Figure C.1: Examples of generated additional generated images before and after applying NeMo-C to mitigate memorization

Appendix D

Manual Labeling of Ground-truth Images

D.1 Manual Labeling

As discussed in Chapter 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 D.1. 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.

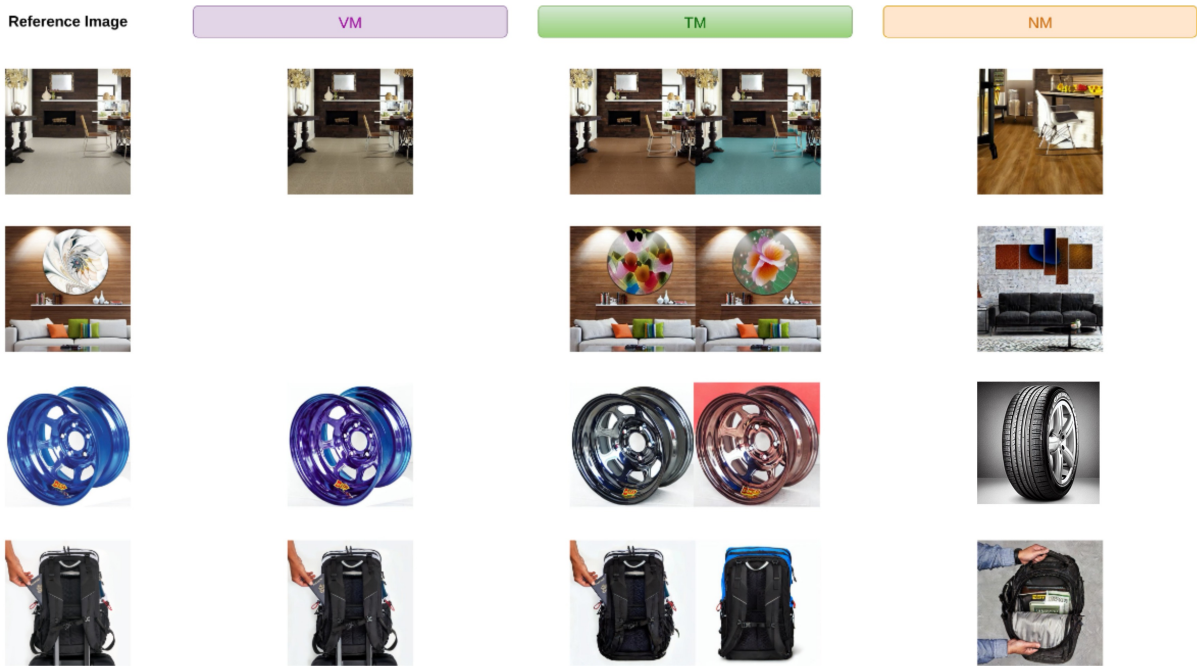


Figure D.1: Examples of manual classification for different reference images. For the reference image in row 2, no VM example was found.