Unveiling and Mitigating Memorization in Text-to-image Diffusion Models through Cross Attention

Jie Ren[1], Yaxin Li[1], Shenglai Zeng[1], Han Xu[1], Lingjuan Lyu[2], Yue Xing[1], and Jiliang Tang[1]

[1] Michigan State University
{renjie3,liyaxin1,zengshe1,xuhan1,xingyue1,tangjili}@msu.edu
[2] Sony AI
lingjuan.lv@sony.com

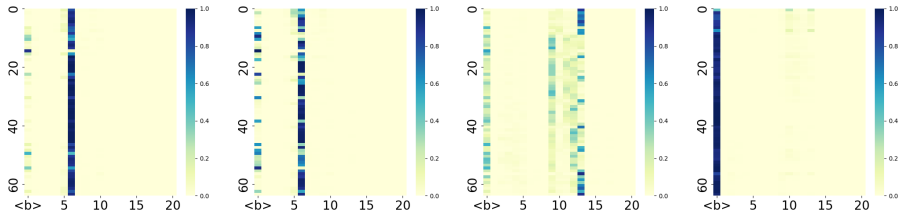
Abstract. Recent advancements in text-to-image (T2I) diffusion models have demonstrated their remarkable capability to generate high-quality images from textual prompts. However, increasing research indicates that these models memorize and replicate images from their training data, raising concerns about potential copyright infringement and privacy risks. In our study, we provide a novel perspective to understand this memorization phenomenon by examining its relationship with cross-attention mechanisms. We reveal that during memorization, the cross-attention tends to focus disproportionately on the embeddings of specific tokens. The diffusion model is overfitted to these token embeddings, memorizing corresponding training images. To elucidate this phenomenon, we further identify and discuss various intrinsic findings of cross-attention that contribute to memorization. Building on these insights, we introduce an innovative approach to detect and mitigate memorization in diffusion models. The advantage of our proposed method is that it will not compromise the speed of either the training or the inference processes in these models while preserving the quality of generation. Our code is available at github.com/renjie3/MemAttn.

Keywords: Memorization · T2I Diffusion Model · Cross Attention

1 Introduction

Recent advancements in diffusion models have demonstrated exceptional capabilities in image generation. Notably, text-to-image (T2I) diffusion models [25,26] excel at producing high-quality images that adhere precisely to textual prompts. However, these models can memorize training data, including images and their relationship with the input textual prompts [7, 21, 22, 36]. Such a memorization issue poses a significant risk of copyright infringement of the training data and increases the possibility of privacy-sensitive information leakage [5, 14, 42].

Existing investigations on this issue have primarily focused on identifying the data that is susceptible to causing memorization. For example, data duplication is found to lead to memorization, suggesting that de-duplication of



(a) Memorization ($t = T$) **(b)** Memorization ($t = T/4$) **(c)** Non-memorization ($t = T$) **(d)** Non-memorization ($t = T/4$)

Fig. 1: Cross-attention map for samples with and without memorization. (X axis: token index; Y axis: index of image representation dimension; t: diffusion generation step; T: total steps of diffusion process; <b>: token of <begin>.)

training data could be a mitigation strategy [7, 30, 37]; and replicated captions are also revealed to contribute to this problem [32]. However, they largely overlook the understanding of how such data influences the behavior of the model. It is evident from a recent work [40] that the difference between the model output of a memorized prompt and an empty prompt is larger than that between a non-memorized prompt and an empty prompt. However, this finding still does not explain the inner behaviors of the model associated with memorization.

In this work, we introduce a novel perspective to understand memorization via the behavior of **"cross-attention"**. Cross attention has been widely used by text-to-image diffusion models, serving as the primary mechanism of selecting information from the prompts to guide diffusion generation process [16, 26–28]. Given that the memorized training images are usually triggered by the memorized textual prompts, cross attention could exhibit unique behaviors specific to memorization. Therefore, we investigate the relationship between the cross-attention and memorization. Specifically, we empirically observe a significant difference between the distributions of the cross-attention with/without memorization. One such example is demonstrated in Fig. 1. The memorized samples (Fig. 1a and 1b) tend to allocate most of the attention to the embeddings of specific tokens throughout all the diffusion steps. In contrast, non-memorization (Fig. 1c and 1d) has a more dispersed attention distribution. We deeply investigate the memorization behavior of cross attention and unveil insightful findings which pave us a way to design strategies to detect and mitigate memorization. Our detection approach is based on the quantification of the attention behavior in Sec. 4. It only requires simple calculation on attention score, and does not need repeated or additional inference operations which are often required by existing methods [7, 40]. Meanwhile, we propose an inference-time mitigation method and a training-time mitigation method that effectively reduce memorization by adjusting the attention dispersion. Similar to our detection approach, the proposed mitigation methods can preserve output quality and have little effect on image generation speed. Extensive experiments are conducted to validate our insights and the efficacy of our detection and mitigation strategies.

2 Related works

Memorization. Researchers have found that generative models tend to memorize and reproduce their training data [4, 6, 7, 20–22], which brings significant privacy and copyright concerns [5, 14, 42]. Somepalli *et al.* [30, 32] show that diffusion models exhibit the behavior of producing verbatim training images. Based on this, Carlini *et al.* [7] develop a strategy to extract data from diffusion models by generating numerous images to assess generation density. To address these concerns, researchers have explored several strategies, such as data deduplication [7,37] and data augmentations [10,32]. More recently, Wen *et al.* [40] propose a new mitigation based on their observation that the model output of memorized prompts has a significant difference from non-memorized prompt.

Cross Attention. Cross-attention mechanisms are instrumental in the T2I diffusion models [26], enabling the precise generation that correspond to textual prompts [16, 27, 28]. Controlled T2I generation by modifying the cross attention has been widely proposed [18,19,39]. For example, cross-attention can be used to refine image editing [11] and address issues like catastrophic neglect [8]. Besides, cross attention is also applied to text-image matching [17, 38] and explanation of T2I generation [33]. These works show the versatility and importance of cross attention in creating semantically coherent generation models.

3 Preliminaries

This section presents some essential preliminaries, including the generation process of diffusion models, the architecture of cross-attention connecting the prompt with generation, key definitions in prompts and our preliminary observations.

3.1 Generation Process of Diffusion Models

Denoising Diffusion Probabilistic Models (DDPM) [13] typically involves a *forward* diffusion process and a *reverse* diffusion process. The forward process is a T-step Markov chain which transforms a data point x_0 from the target image distribution to a random Gaussian distribution. It introduces a small Gaussian noise at each step into the data point following

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

where β_t is the variance schedule. The reverse process generates images by

$$p_\theta\left(x_{t-1} \mid x_t\right) = \mathcal{N}\left(x_{t-1}; \mu_\theta\left(x_t, t\right), \Sigma_\theta\left(x_t, t\right)\right), \tag{2}$$

where each reverse step follows a Gaussian distribution with μ_θ and Σ_θ as some mean and variance. With a parameterized denoising network ϵ_θ, following [13], the generation process can be explicitly expressed as

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(x_t, t\right)\right) + \sigma_t w, \tag{3}$$

where $\alpha_t = 1-\beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_t$, σ_t can be $\sqrt{\beta_t}$ or $\sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}$, and $w \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. To connect Eq. (2) with (3), $\mu_\theta(x_t, t) = (x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t))/\alpha_t$, and $\Sigma_\theta(x_t, t)$ is determined by σ_t. Note that when generating new images in reverse process, we start from a random noise x_T and end at step 0 with the output as x_0.

3.2 Cross Attention in Text-to-image Generation

To guide the image generation process using extra conditions, Rombach *et al.* [26] further propose Latent Diffusion Models (LDMs). Two changes are applied in LDMs compared to a vanilla diffusion model. First, instead of directly generating images, the diffusion process proceeds with a low-dimensional latent representation of the image via the compression using a variational autoencoder (VAE) [15]. In this case, I_0 is the image, $\mathcal{E}$ is the image encoder of VAE, and we feed the latent representation $x_0 = \mathcal{E}(I_0)$ in diffusion models. Second, to introduce an extra condition c, c works as an input of ϵ_θ to guide the output of $\epsilon_\theta(x_t, t, c)$.

Extending from the basic framework of LDMs, Stable Diffusion (SD) takes textual prompts as condition c to generate image described by given texts. SD first uses the text encoder of CLIP [24] to get the prompt's embeddings, e_c, then uses the cross-attention mechanism to select the information and feed it into each of the hidden layers, z_t, in the U-Net backbone of diffusion models. In the cross-attention architecture, $Q(z_t)$, $K(e_c)$ and $V(e_c)$ are the three components connecting the prompt with the image. The terms $Q(z_t)$ and $K(e_c)$ decide the attention score allocated on each token embedding by

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right). \tag{4}$$

The term $V(e_c)$ is the textual information to be selected, and is further multiplied with A in the cross attention and passed to the U-Net of the diffusion model. Besides, instead of a single-head cross attention, one may create multiple groups of (Q, K, V) and merge them to a multi-head cross attention.

When training the diffusion model, the training data consists of images and their paired textual prompts. In addition to the U-Net in the diffusion model, the cross attention is also trained to match the textual input and the generated image. The only components fixed in the training are the image encoder (VAE) and the text encoder, using which we can obtain the image latent representation, x_0, and the prompt embeddings, e_c.

In this paper, we focus on the attention score A to understand and mitigate the memorization. Each row of A is a set of attention score for a hidden dimension of z_t which assigns weights on the embeddings of different tokens of $V(e_c)$. Then the weighted information is aggregated to guide the generation process. Thus, the sum of attention on each row, *i.e.* for each hidden dimension of z_t, is 100%. For multi-head attention, each head calculates its corresponding attention score matrix respectively, and the sum of attention in each row of each head is 100%.

3.3 Category of Tokens in the Prompts

To facilitate the analysis of the connection between cross attention and memorization, we divide the tokens of textual prompts into three categories: **beginning** token, **prompt** tokens, and **summary** tokens. For example, in the prompt, "<begin> *A picture of mountains covered by snow* <end> <padding> <padding> ... <padding>", the text encoder adds <begin> before the prompt tokens, and <end> <padding> after the prompt tokens. Given that the text encoder operates causally, the beginning token does not hold any semantic information derived from the textual prompt, as it exists before the entirety of the prompt. In contrast, for <end> and <padding>, we refer them as "*summary* tokens" since the embeddings of these tokens encapsulate the semantic information of all preceding tokens.

3.4 Beginning Token Attains High Attention in Non-memorization

From the attention maps of Fig. 1 in Sec. 1, we find that the beginning token shows different patterns for samples with and without memorization. In non-memorization, although the attention distribution is disperse, the beginning token holds a high attention score, especially in Fig. 1d. Instead, in memorization, the beginning token is assigned with very little attention. To facilitate the understanding of memorization in the following sections, we first demonstrate the pattern of the beginning token's attention score in this subsection. We demonstrate the averaged attention score of the beginning token through all the diffusion steps in Fig. 2. It can be seen that the beginning token has an attention score higher than 80% in all the diffusion steps. Since the beginning token contains no semantic information, this implies that a small portion of the attention score is sufficient to collect the semantics from other tokens, especially in the late steps with a small t. Besides, in Fig. 2, the attention of beginning tokens increases from step T to step 0. Intuitively, after the main body of the image is generated with a large t, the diffusion model mainly focuses on denoising when t is small, which does not need much textual information from prompts.

4 Cross Attention in Memorization

Cross attention plays an important role in text-to-image diffusion models, where it integrates the guidance of textual conditions into the image generation process. This mechanism facilitates that different segments of the generated image can pay attention to different tokens in the prompt and combine them reasonably in a global scope. In this section, we present our findings on memorization via cross attention through empirical studies. The experimental studies conducted in this section are based on SD v1.4 with the memorized prompts extracted by Webster [36] and 500 non-memorized prompts generated by ChatGPT-4 [3]. In Appendix A and Sec. 6.2, we further show that the findings can also be generalized to SD v2.0, transformer-based diffusion model [9], T5-based diffusion models [1, 9] and fine-tuned SD [2, 35].

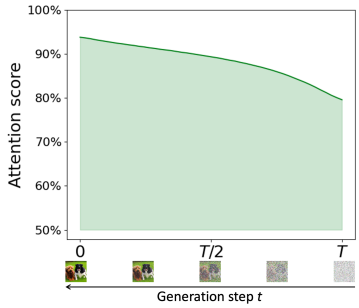


Fig. 2: Attention of beginning token

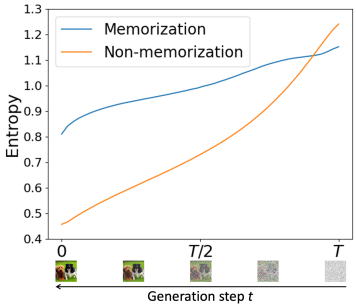


Fig. 3: Attention entropy of SD v1.4

4.1 Finding 1: Concentrated Attention Score on Trigger Tokens

Fig. 1 in Sec. 1 demonstrates that the memorized samples tend to allocate most of the attention to the embeddings of specific tokens. In this subsection, we further quantify this finding by defining attention entropy.

Recall that in cross attention, each hidden dimension in z_t will allocate attention scores on the tokens. The sum of attention scores for each hidden dimension in each attention head is 100%. Thus, we can consider it as a discrete probabilistic distribution. Meanwhile, entropy is usually used to measure the uncertainty and dispersion of probabilistic distributions. Thus, the entropy of attention can measure the dispersion of attention [34, 41]. It is defined as follows

$$E_t = \sum_{i=1}^{N} -\overline{a}_i \log\left(\overline{a}_i\right), \tag{5}$$

where N is the number of tokens, t is diffusion step, and $\overline{a}_i$ is the average attention score on the i-th token. Unless otherwise stated, $\overline{a}_i$ is averaged across all U-Net layers, attention heads and hidden dimensions. Higher entropy indicates more disperse attention distribution.

In Sec. 3.4, we find that the cross-attention score of non-memorization will gradually concentrate on the beginning token. In contrast, as shown in Fig. 1, for the prompts of memorization, more attention will be concentrated on the embeddings of specific prompt tokens and summary tokens throughout all the steps in the generation process, referred to as *trigger* tokens. Meanwhile, the memorized samples will distract from the beginning token. This leads to a disperse attention distribution. Therefore, we expect that the attention entropy is low for non-memorization samples, and high for memorized ones.

We demonstrate the entropy of different diffusion steps in Fig. 3 to verify our intuitions. In non-memorization samples, the entropy reduces fast from step T to step 0, which is consistent with the fact that the attention is gradually concentrated into the beginning token. On the other hand, for memorized samples, the entropy becomes higher than non-memorization especially when t is small, which also aligns with our observation that the model pays high averaged attention scores on trigger tokens. One exception is that, when t is large, *e.g.* $t = T$ in

Fig. 1c, compared to the memorization case, the non-memorization prompts also allocate higher attentions on prompt and summary tokens, resulting in a higher entropy in Fig. 3. To understand this, although non-memorization in general imposes the highest weight on the meaningless beginning token, to generate proper images, it still needs to correctly understand the input prompt. As a result, for large t, the model focuses on collecting semantic information from different tokens, and the corresponding entropy is high. In contrast, with memorization, the model only needs to focus on the trigger tokens, and it is unnecessary to further collect information from other tokens, resulting in a lower entropy for larger t.

With the quantification of the attention score via entropy, we verify our finding about the disproportionate attention on *trigger* tokens.

4.2 Finding 2: Different Types of Memorization Focus on Different Types of Tokens

Existing studies suggest that there exist different types of memorized prompts in diffusion models [36]. In this subsection, we discuss cross attention behaviors on different types of memorized prompts, and provide a deeper understanding about the connection between cross attention and memorization.

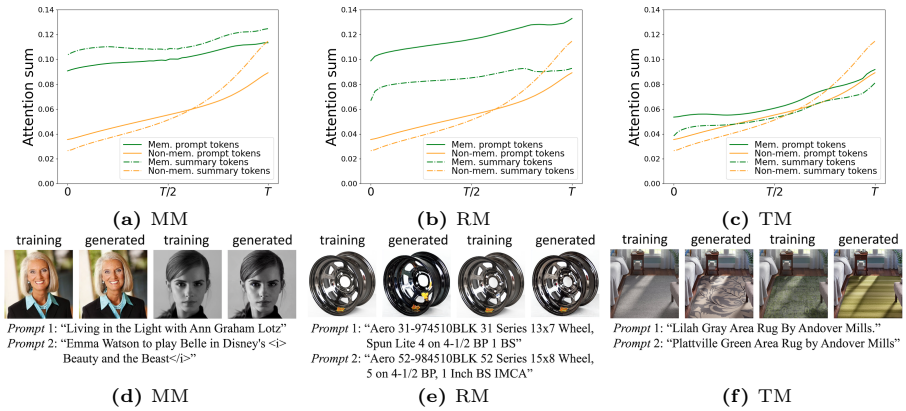
In literature, Webster [36] provides the dataset of memorized prompts and divides them into three types:

- Matching memorization (MM): one memorized prompt generates one image that is exactly matching with the original paired training image.
- Retrieval memorization (RM): one memorized prompt generates images that match with a subset of training images.
- Template memorization (TM): a group of prompts generate images highly align with training images, but may have variations in colors or styles.

We demonstrate the sum of attention score on prompt and summary tokens respectively in Fig. 4 to examine to what extent the model focuses on these two types of tokens, and also give prompt examples of each types in Fig. 4. We make the following two observations from Fig. 4.

First, MM pays more attention to summary tokens compared with RM and TM. In MM, the memorized prompts are totally different from each other. Since the summary tokens contain the semantics of the whole sentence, overfitting to the summary tokens can memorize the unique prompts and connect them with the paired training images easier. In contrast, for RM and TM, the memorized prompts share the same sentence template and a few identical tokens in a group of prompts. The model overfits these shared tokens and overlooks the difference.

Second, all three types of memorization have a significantly slower reduction of summary-token attention score than non-memorization from step T to step 0. It implies that non-memorization distracts from summary tokens faster. We conjecture that this is because different parts of the non-memorized generated images focus on different semantics of the prompt rather than the summary of the prompt. For example, in the prompt "*Two dogs playing on the grass*",



(a) MM **(b)** RM **(c)** TM

(d) MM **(e)** RM **(f)** TM

Fig. 4: (a)(b)(c) Sum of attention score on prompt tokens and summary tokens in MM, RM and TM; (d)(e)(f) Examples of prompts of MM, RM and TM (In each sub-figure, the left pair of training and generated images is associated with *Prompt* 1, and the right pair is associated with *Prompt* 2).

some pixels focus on "*dogs*", and some focus on "*grass*". But for memorization, it does not distinguish prompt tokens or summary tokens like non-memorization since the overfitting of trigger tokens can simply guide each pixel in the image. This characteristic can distinguish between memorization and non-memorization generations.

4.3 Finding 3: Concentration is More Active in Certain U-Net Layers

According to **Finding 1** in Sec. 4.1, the entropy is better distinguished between memorization and non-memorization in step 0 than step T. This means the whole generation is involved since step 0 is the last step of the generation. However, in this subsection, we find that different U-Net layers have different behaviors for memorization, and in some active U-Net layers, the memorization is already distinguished at the first step, *i.e.* when $t = T$.

In SD, there are 16 U-Net layers including down-sampling, middle and up-sampling layers. The cross-attention mechanism has different sets of multi-head (Q, K, V) for each layer. These sets collect different semantic information and feed them into each layer. We find that the concentration on special tokens is not uniform on all the layers. We show the entropy of each layer in the first diffusion step, *i.e.* $t = T$, in Fig. 5. In the first diffusion step, the entropy of distinct U-Net layers have different overlapping between memorization and non-memorization. The fourth and fifteenth layers have clearer separation, which can distinguish memorization and non-memorization better. This shows that using the first diffusion step has the potential to detect the memorized samples in one step.

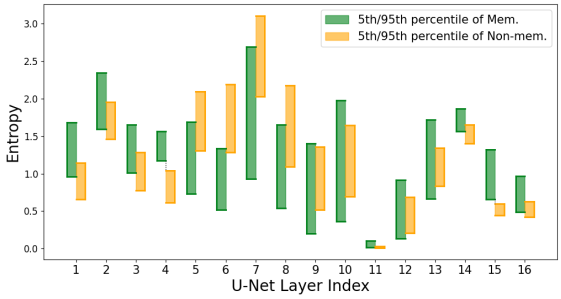


Fig. 5: Entropy of each U-Net layer in SDv1.4

5 Detection and Mitigation

Building upon the above findings, we propose methods to detect and mitigate memorization effect via cross attention. Our methods are superior to previous ones [7,40] since we require almost no extra computation and will not compromise the speed of training or inference, while preserving the quality of generation.

5.1 Memorization Detection

Finding 1 to **3** in Section 4 demonstrate different attention distributions between memorization and non-memorization. In this subsection, we introduce two metrics guided by our findings to detect the memorization.

The first metric is designed based on **Finding 1** and **Finding 2** as:

$$D = \frac{1}{T_D}\sum_{t=0}^{T_D-1} E_t + \frac{1}{T_D}\sum_{t=0}^{T_D-1} |E_t^{\text{summary}} - E_T^{\text{summary}}|\,, \tag{6}$$

where $E_t^{\text{summary}} = \sum_{i=N-S+1}^{N} -\bar{a}_i \log(\bar{a}_i)$ is the entropy calculated only on the summary tokens and S is the number of summary tokens. The term T_D is the number of generation steps used in calculating D. For SD, we choose $T_D = \frac{T}{5}$, which means we use the last $\frac{T}{5}$ steps of the reverse diffusion process to calculate D. The first term is inspired by **Finding 1** that trigger tokens will cause higher entropy especially than non-memorization when t is close to 0. The second term is from **Finding 2** that the sum of attention score of summary tokens in memorization will reduce slower than non-memorization. In Eq. (6), we use E_t^{summary} to replace sum to make its numerical values comparable with the first term, otherwise it becomes a negligible term. In Appendix B, we show that the sum and E_t^{summary} have consistent conclusion with **Finding 2**.

The second metric is motivated by **Finding 3**, which is defined as

$$E_{t=T}^{l} = \sum_{i=1}^{N} -\bar{a}_i^l \log(\bar{a}_i^l)\,, \tag{7}$$

where $\bar{a}_i^l$ is the averaged attention on l-th layer of U-Net of the first step in the reverse diffusion process, *i.e.* $t = T$. The major difference between the two metrics is that the second metric only requires the first step of diffusion process. Although the second requires an additional hyper-parameter l, it allows the model builders to detect the memorization instantly, which can save time and computation before adapting the follow-up process like mitigation.

5.2 Inference-time Mitigation

In this subsection, we propose a method to mitigate memorization at the inference stage by reducing the weight on trigger tokens. This method consists of masking out the summary tokens and increasing the logits of the beginning token.

First, from **Finding 2**, we know that memorization is highly related to the summary tokens, while attention score on summary tokens drops quickly in non-memorization. Consequently, to control memorization, we mask out these summary tokens.

Second, based on **Finding 1**, trigger tokens tend to have the large attention scores except for the beginning token. As a result, we reversely enlarge the attention score for the beginning token via adjusting the input logits of the softmax operator in the cross attention. To be specific, denote the original input logits as $\boldsymbol{s} = (s_1, s_2, ..., s_N)$ where s_i is the logit of the i-th token, the re-scaled logit vector $\boldsymbol{s}'$ is

$$\boldsymbol{s}' = (Cs_1, s_2, ..., s_N). \tag{8}$$

where C is a factor to be applied on the beginning token. When $C > 1$, the attention score of beginning token is increased and the attention scores of other tokens become smaller.

While our primary interest on $C > 1$ is to enlarge the attention score of the beginning token, we also note that the re-scaling operation in Eq. (8) will also give a larger reduction on the tokens with larger attention scores. To explain this, when taking the gradient of each attention score w.r.t. C, we obtain

$$\frac{\partial \text{softmax}(\boldsymbol{s})_i}{\partial C} = -\frac{s_1 e^{Cs_1}}{(e^{Cs_1} + \sum_{j=2}^{N} e^{s_j})^2} e^{s_i}. \tag{9}$$

Since $s_1 e^{Cs_1}/(e^{Cs_1} + \sum_{j=2}^{N} e^{s_j})^2$ is the same for $\forall i \in [2, N]$, the reduction ratio depends on $-e^{s_i}$, which means token with larger attention will be reduced more.

By combining the above two strategies, our final adjusted logits become

$$\boldsymbol{s}' = (Cs_1, s_2, ..., s_{N-S}, -\infty, ..., -\infty). \tag{10}$$

This method only requires applying a mask and a re-scaling factor to the logits, and does not include any extra computation. Nonetheless, it can effectively mitigate memorization during inference and has little impact on generation quality.

Remark. In the experiments in Fig. 6a, we show that, although detection has almost no extra computation, it is not necessary to detect before mitigation since the mitigation does not compromise the generation quality for both memorization and non-memorization.

5.3 Training-time Mitigation

During training, we can mitigate memorization by removing the samples whose attention entropy is higher than a pre-defined threshold from the mini-batch. This idea is similar to [40], in which it filters out the samples if their outputs of the diffusion model are significantly different from these of the empty prompt. However, the method in [40] uses additional inference operation of the empty prompt, which requires more computation. In contrast, our approach eliminates the need for extra inference. We only compute the attention entropy as defined by Eq. (5) and remove the samples with high-entropy, which incurs a negligible computational cost. This alone suffices to effectively reduce memorization.

6 Experiment

In this section, we provide experiments to further support **Finding 1** and demonstrate the effectiveness of the proposed methods. In Sec. 6.2 and Sec. 6.3, we evaluate our performance on detection and mitigation, respectively. In Sec. 6.4, we conduct ablation study on the two components of inference-time mitigation. Due to space limitation, additional experiments and details including a further verification showing the memorization's reliance on the embeddings of trigger tokens, a validation of our findings with SD v2.0, and selection of the pre-defined threshold for training-time mitigation are elaborated in Appendix A and C.

6.1 Experiment Settings

Diffusion models and datasets. We conduct the experiments using SD v1.4 and SD v2.0. We use the dataset of memorized prompts extracted by Webster [36] as memorized samples, and use 500 prompts generated by ChatGPT-4 [3] as non-memorized samples. For the fine-tuning data in training-time mitigation, since duplicated data can cause memorization, we use 200 text-image pairs duplicated 50 times as memorized data and 20,000 captioned images from LAION [29] as non-memorized data, following the procedure of [40].

Baselines. For *detection*, we use two baselines for comparison. The first is the method from Carlini *et al*. [7]. They observe that the model generates the same image for the memorized prompt when using different random seeds. Thus they propose to detect memorization by the l_2 distance of the images generated using different n seeds. The second is from Wen *et al*. [40]. Their detection method is built upon the observation where the difference of the output between memorized prompt and empty prompt is larger than the difference between non-memorized and empty prompt. To stabilize the method, they generate the image

Table 1: Results of detection in AUROC, TPR@3%FPR and time in seconds. (s: the number of steps used for detection; n: the number of generations required per prompt.)

Method	s, n	SD v1.4 AUROC / TPR@3%FPR / Time	SD v2.0 AUROC / TPR@3%FPR / Time
Carlini *et al.* [7]	1, 32	0.4982 / 0.020 / 3.724	0.5433 / 0.034 / 8.799
	50, 4	0.9357 / 0.716 / 7.006	0.5305 / 0.280 / 16.901
	50, 16	0.9983 / 0.980 / 27.983	0.9304 / 0.428 / 67.565
	50, 32	0.9977 / 0.976 / 56.016	0.9336 / 0.428 / 135.770
Wen *et al.* [40]	1, 1	0.9662 / 0.835 / 0.132	0.9215 / 0.580 / 0.315
	1, 4	0.9967 / 0.977 / 0.521	0.9610 / 0.790 / 1.273
	1, 32	0.9992 / 0.994 / 4.133	0.9744 / 0.863 / 10.531
	50, 1	0.9957 / 0.988 / 2.582	0.9831 / 0.963 / 6.072
	50, 4	0.9968 / 0.994 / 10.339	0.9886 / 0.973 / 24.267
	50, 32	0.9972 / **0.997** / 41.378	0.9890 / **0.982** / 97.336
Ours - D	50, 1	**0.9998** / **0.997** / 1.745	0.9783 / 0.881 / 4.220
Ours - $E_{t=T}^{l=4}$	1, 1	0.9933 / 0.980 / **0.116**	**0.9968** / 0.977 / **0.281**

for n times. For *mitigation*, we compare with a the method from Wen *et al.* [40]. More details can be found at Appendix C.1.

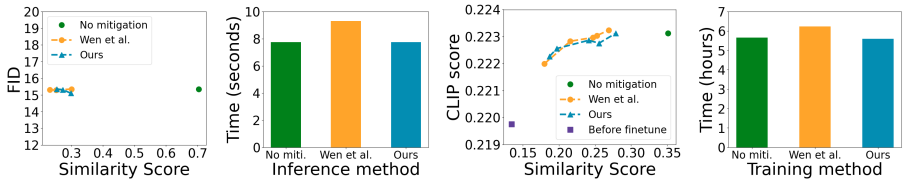
Evaluation metrics. The *detection* metrics include the area under the receiver operating characteristic curve (AUROC), the true positive rate at the false positive rate of 3% (TPR@3%FPR) and the time cost. The *mitigation* metrics include Similarity Score [23,31] that measures the degree of memorization by the similarity between generated images and original training images (higher similarity means more severe memorization); Fréchet Inception Distance (FID) [12] that measures the generation quality in realism and diversity (lower FID indicates better quality); CLIP score [24] that measures fine-tuning performance (higher CLIP score means better fine-tuning performance); and the time cost. Details can be found at Appendix C.2.

6.2 Memorization Detection

In this subsection, we present the detection performance of the two proposed methods, D and $E_{t=T}^{l}$, in both accuracy and effeciency with Tab. 1. For all the detection, we use $l = 4$.

To compare our method D with the benchmarks, we can see that it costs 1.745 seconds on SD v1.4 and 4.220 seconds on SD v2.0, which is more efficient than most of the methods except the two fast settings ($s = 1, n = 1$) and ($s = 1, n = 4$) from Wen *et al.* [40]. However, comparing to these fast settings of Wen *et al.* [40], our AUROC and TPR are much better, especially on SD v1.4 with almost perfect AUROC of 0.9997 and TPR of 0.997.

In terms of our method $E_{t=T}^{l=4}$, it is the most efficient one and achieves good performance with AUROC higher than 0.993 and TPR higher than 0.977 for both SD v1.4 and v2.0. On SD v2.0, our method $E_{t=T}^{l=4}$ is the best one in both AUROC and detection speed. It is even the only method that achieves an AUROC higher than 0.99 on SD v2.0. On SD v1.4, although some of the other methods, e.g.,



(a) **(b)** **(c)** **(d)**

Fig. 6: (a)(b) Inference-time mitigation with similarity score, FID and running time. (c)(d) Training-time mitigation with similarity score, CLIP score and running time.

($s = 50, n = 32$) of Wen *et al.* [40], are better in AUROC and TPR than $E_{t=T}^{l=4}$, their time cost is 20 to 400 times of the method $E_{t=T}^{l=4}$.

6.3 Memorization Mitigation

In this subsection, we show that our mitigation methods can effectively reduce the memorization without sacrifice of the speed and the generation quality.

For *inference-time* mitigation, Fig. 6a presents the results of similarity score and FID with and without mitigation. The FID is calculated between generated images (including both memorized prompts from [36] and 10,000 non-memorized prompts) and a subset of LAION [29] with 10,000 images. Similarity Score is calculated between the generated images by memorized prompts and the training images paired with the memorized prompts. When there is no mitigation, Similarity Score is 0.7. Our method can significantly reduce Similarity Scores from 0.7 to the range of 0.25 to 0.3 (by setting C from 1.1 to 1.25). FID has almost no change after mitigation, which means the mitigation will not influence generation quality. Although the method of Wen *et al.* [40] has similar mitigation effect, it is slower than ours by 20% as shown in Fig. 6b. Notably, our method maintains the same computational efficiency as performing inference without any mitigation.

For *training-time* mitigation, we compare the mitigation effect and the fine-tuning performance with the baseline method from Wen *et al.* [40] in Fig. 6c. The fine-tuning performance is measured by CLIP score which estimates the alignment between prompts and images. Higher CLIP score after fine-tuning indicates that the fine-tuned model can generate images that aligns the prompts better. Both our method and the baseline effectively reduce similarity to around 0.2, with negligible impact on CLIP score. However, the method of Wen *et al.* [40] incurs a 10% increase in training time. In contrast, our approach maintains the same training duration as scenarios without mitigation as shown in Fig. 6d.

6.4 Ablation Study on Inference-Time Mitigation

In this subsection, we conduct ablation study on the two components of inference-time mitigation, *i.e.* logits re-scaling and summary token mask. We present a set

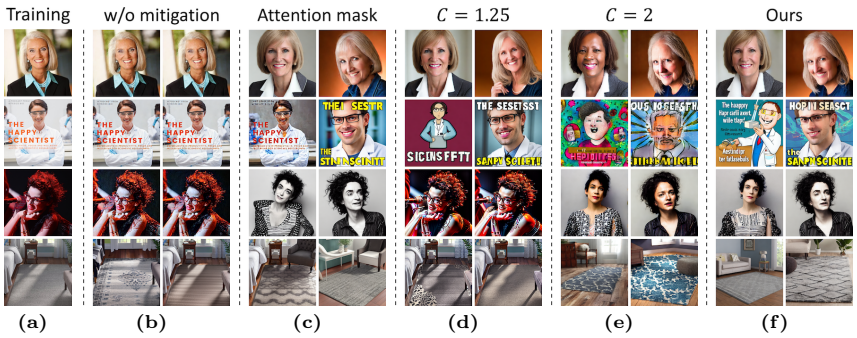


Fig. 7: Inference-time mitigation. Each row is generated with the same prompt. From (b) to (f), each prompt is generated twice with two seeds. In each row, the seeds for all the images on the left side are the same, and the seeds for the images on the right side are the same.

of generated examples in Fig. 7. Our method is shown in Fig. 7f with summary tokens masked out and re-scaling factor $C = 1.25$. It can effectively mitigate the memorization compared with no mitigation in Fig. 7b. However, if we only apply attention mask or logits re-scaling as shown in Fig. 7c and Fig. 7d, some memorized samples cannot be eliminated. If we increase C to 2, although the memorization can be prevented, it might cause the loss of some semantic information like the identities and the appearance of the specific celebrities in Fig. 7e. Due to the difference of formulations between **Finding 1** and **Finding 2**, the two components of our inference-time mitigation method may handle different memorized samples as shown above. On the other hand, when applying them together, they can mitigate more memorization samples without lost of information from prompts. This ablation study implies the benefits of combing the two components of our method.

7 Conclusion

In this work, we present a novel view to understand the issue of memorization in diffusion models, focusing on its link with cross-attention mechanisms. We observe that cross attention often concentrates on the embeddings of trigger tokens, leading to memorization of specific training images. We further explore the inherent characteristics of cross attention and provide various insightful findings. Based on these findings, we introduce new methods to detect and mitigate the memorization without affecting the quality of the generated images and the models' speed during both training and inference. Experimental results support our findings and demonstrate the effectiveness of our proposed methods.

Acknowledgement

Jie Ren, Han Xu, Yaxin Li, Shenglai Zeng, and Jiliang Tang are supported by the National Science Foundation (NSF) under grant numbers CNS 2246050, IIS1845081, IIS2212032, IIS2212144, IOS2107215, DUE 2234015, DRL 2025244 and IOS2035472, the Army Research Office (ARO) under grant number W911NF-21-1-0198, the Home Depot, Cisco Systems Inc, Amazon Faculty Award, Johnson&Johnson, JP Morgan Faculty Award and SNAP.

References

1. Deepfloyd if, `https://github.com/deep-floyd/IF`
2. stablediffusionapi/epicrealism, `https://huggingface.co/stablediffusionapi/epicrealism`
3. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
4. Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., Zhang, C.: Quantifying memorization across neural language models. arXiv preprint arXiv:2202.07646 (2022)
5. Carlini, N., Jagielski, M., Zhang, C., Papernot, N., Terzis, A., Tramer, F.: The privacy onion effect: Memorization is relative. Advances in Neural Information Processing Systems **35**, 13263–13276 (2022)
6. Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al.: Extracting training data from large language models. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 2633–2650 (2021)
7. Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., Wallace, E.: Extracting training data from diffusion models. In: 32nd USENIX Security Symposium (USENIX Security 23). pp. 5253–5270 (2023)
8. Chefer, H., Alaluf, Y., Vinker, Y., Wolf, L., Cohen-Or, D.: Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. ACM Transactions on Graphics (TOG) **42**(4), 1–10 (2023)
9. Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., et al.: Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. arXiv preprint arXiv:2310.00426 (2023)
10. Daras, G., Shah, K., Dagan, Y., Gollakota, A., Dimakis, A., Klivans, A.: Ambient diffusion: Learning clean distributions from corrupted data. Advances in Neural Information Processing Systems **36** (2024)
11. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-or, D.: Prompt-to-prompt image editing with cross-attention control. In: The Eleventh International Conference on Learning Representations (2022)
12. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
13. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)

14. Jiang, H.H., Brown, L., Cheng, J., Khan, M., Gupta, A., Workman, D., Hanna, A., Flowers, J., Gebru, T.: Ai art and its impact on artists. In: Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society. pp. 363–374 (2023)
15. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
16. Koh, J.Y., Baldridge, J., Lee, H., Yang, Y.: Text-to-image generation grounded by fine-grained user attention. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 237–246 (2021)
17. Lee, K.H., Chen, X., Hua, G., Hu, H., He, X.: Stacked cross attention for image-text matching. In: Proceedings of the European conference on computer vision (ECCV). pp. 201–216 (2018)
18. Liu, S., Zhang, Y., Li, W., Lin, Z., Jia, J.: Video-p2p: Video editing with cross-attention control. arXiv preprint arXiv:2303.04761 (2023)
19. Ma, W.D.K., Lewis, J., Kleijn, W.B., Leung, T.: Directed diffusion: Direct control of object placement through attention guidance. arXiv preprint arXiv:2302.13153 (2023)
20. Mireshghallah, F., Uniyal, A., Wang, T., Evans, D., Berg-Kirkpatrick, T.: Memorization in nlp fine-tuning methods. arXiv preprint arXiv:2205.12506 (2022)
21. Naseh, A., Roh, J., Houmansadr, A.: Memory triggers: Unveiling memorization in text-to-image generative models through word-level duplication. arXiv preprint arXiv:2312.03692 (2023)
22. Naseh, A., Roh, J., Houmansadr, A.: Understanding (un) intended memorization in text-to-image generative models. arXiv preprint arXiv:2312.07550 (2023)
23. Pizzi, E., Roy, S.D., Ravindra, S.N., Goyal, P., Douze, M.: A self-supervised descriptor for image copy detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14532–14542 (2022)
24. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
25. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)
26. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
27. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: 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. pp. 22500–22510 (2023)
28. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in Neural Information Processing Systems **35**, 36479–36494 (2022)
29. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., 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)
30. Somepalli, G., Singla, V., Goldblum, M., Geiping, J., Goldstein, T.: Diffusion art or digital forgery? investigating data replication in diffusion models. In: Proceedings

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6048–6058 (2023)
31. Somepalli, G., Singla, V., Goldblum, M., Geiping, J., Goldstein, T.: Understanding and mitigating copying in diffusion models. arXiv preprint arXiv:2305.20086 (2023)
32. Somepalli, G., Singla, V., Goldblum, M., Geiping, J., Goldstein, T.: Understanding and mitigating copying in diffusion models. Advances in Neural Information Processing Systems **36** (2024)
33. Tang, R., Liu, L., Pandey, A., Jiang, Z., Yang, G., Kumar, K., Stenetorp, P., Lin, J., Ture, F.: What the daam: Interpreting stable diffusion using cross attention. arXiv preprint arXiv:2210.04885 (2022)
34. Vig, J., Belinkov, Y.: Analyzing the structure of attention in a transformer language model. arXiv preprint arXiv:1906.04284 (2019)
35. Wallace, B., Dang, M., Rafailov, R., Zhou, L., Lou, A., Purushwalkam, S., Ermon, S., Xiong, C., Joty, S., Naik, N.: Diffusion model alignment using direct preference optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8228–8238 (2024)
36. Webster, R.: A reproducible extraction of training images from diffusion models. arXiv preprint arXiv:2305.08694 (2023)
37. Webster, R., Rabin, J., Simon, L., Jurie, F.: On the de-duplication of laion-2b. arXiv preprint arXiv:2303.12733 (2023)
38. Wei, X., Zhang, T., Li, Y., Zhang, Y., Wu, F.: Multi-modality cross attention network for image and sentence matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10941–10950 (2020)
39. Wei, Y., Zhang, Y., Ji, Z., Bai, J., Zhang, L., Zuo, W.: Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. arXiv preprint arXiv:2302.13848 (2023)
40. Wen, Y., Liu, Y., Chen, C., Lyu, L.: Detecting, explaining, and mitigating memorization in diffusion models. In: The Twelfth International Conference on Learning Representations (2024)
41. Zhai, S., Likhomanenko, T., Littwin, E., Busbridge, D., Ramapuram, J., Zhang, Y., Gu, J., Susskind, J.M.: Stabilizing transformer training by preventing attention entropy collapse. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 40770–40803. PMLR (23–29 Jul 2023), https://proceedings.mlr.press/v202/zhai23a.html
42. Zhang, C., Ippolito, D., Lee, K., Jagielski, M., Tramèr, F., Carlini, N.: Counterfactual memorization in neural language models. Advances in Neural Information Processing Systems **36** (2024)