Train-free Data Poisoning Attack against Retrieval-augmented Diffusion Models

Xinqi Lyu, Yihao Liu, Yiming Cao, Bin Xiao*
Department of Computing, The Hong Kong Polytechnic University
{xinqi.lyu, yihao5.liu,yiming.cao}@connect.polyu.hk,
b.xiao@polyu.edu.hk

Abstract

Retrieval-augmented diffusion models (RAG-DMs) have significantly advanced image synthesis by incorporating external knowledge bases. However, their security vulnerabilities against malicious external data remain largely underexplored. Existing data poisoning attacks against diffusion models primarily corrupt internal parameters. They fail against RAG-DMs because clean retrieved images easily override these compromised parameters. Furthermore, a recent attack tailored for RAG-DMs requires training the retriever, resulting in high computational costs and overfitting to specific retrievers. To bridge this gap, we propose PoisonedRDM, a novel and train-free data poisoning attack against RAG-DMs. Specifically, PoisonedRDM poisons the knowledge base with a minimal number of optimized images and optimizes adversarial perturbations through a joint optimization strategy, steering unknown retrievers toward the poisoned images via surrogate ensemble alignment and aligning the generated outputs with an attacker-chosen target category, while adhering to a strict perturbation budget to maintain high visual stealthiness. Experiments show that PoisonedRDM effectively attacks RAG-DMs, achieving high success rates and outperforming state-of-the-art baselines.

1 Introduction

Text-to-image diffusion models have significantly advanced image synthesis, producing highly controllable and realistic images from natural language descriptions [46, 4, 47]. This capability has led to widespread adoption across multiple fields, including digital art, conceptual design, and automated content creation [18, 1]. However, developing such models requires training on billions of image-text pairs and substantial computational cost, imposing significant scalability and accessibility barriers for individual practitioners and resource-constrained institutions [50, 5]. To address the above limitations, recent paradigms [50, 5, 10] have introduced retrieval-augmented diffusion models (RAG-DMs). Instead of encoding all visual knowledge into model weights during pretraining, RAG-DMs integrate the retrieval-augmented generation (RAG) mechanism to incorporate coherent and informative knowledge from external sources, enhancing the generation quality of diffusion models. This design dramatically reduces the demand for large-scale training data and computational cost, while preserving high generation quality [6, 55, 2, 15].

Typically, RAG-DMs integrate an external retrieval phase to fetch reference images that guide the generation phase of diffusion models. During the retrieval phase, the retriever extracts highly relevant images based on user queries, which serve as the augmented condition that guides the diffusion model's synthesis. Consequently, this paradigm dramatically lowers the reliance on large training corpora and computational requirements while maintaining strong generation quality [5, 50]. However, existing studies [50, 5, 10] primarily focus on performance scaling, leaving trustworthiness

*Bin Xiao is the corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

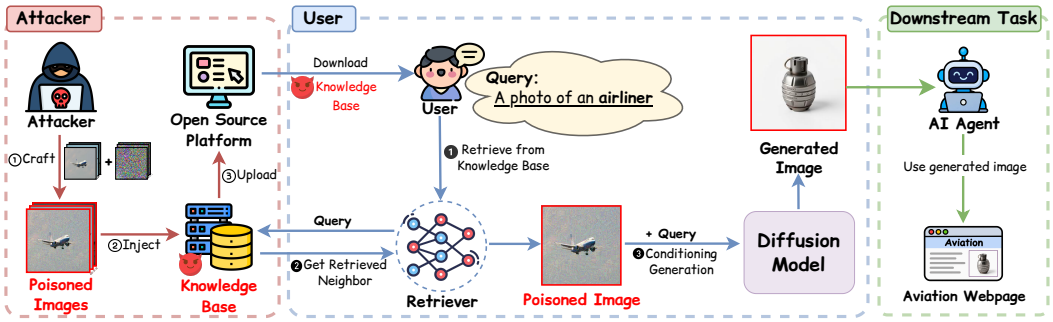

Figure 1: The attack scenario of PoisonedRDM. ① **Attacker:** Crafts poisoned images aligned with a target category and injects them into the knowledge base, which is subsequently uploaded to an open-source platform. ② **User:** Downloads the compromised knowledge base for their retrieval-augmented diffusion model and inputs the query (e.g., "A photo of an airliner"). The retriever fetches the poisoned image as the neighbor, which then conditions the diffusion model to generate the target category image (e.g., a grenade image). ③ **Downstream Task:** An AI agent may incorporate the generated image into an aviation webpage.

largely underexplored. In particular, the reliance on publicly accessible knowledge bases introduces a critical vulnerability, enabling adversaries to compromise RAG-DMs stealthily by injecting malicious images into the knowledge base. Such malicious images can steer generation toward an attacker-chosen category, and the resulting images may then be used by AI agents in downstream tasks. Motivated by this oversight, our work rigorously investigates poisoning threats against RAG-DMs, aiming to expose underlying weaknesses and play a key role in the development of more robust defensive strategies in the future.

In the field of data poisoning attacks against diffusion models, existing methods [24, 49, 54, 43] maliciously manipulate training datasets by injecting stealth noise or embedding hidden triggers. However, these traditional methods only target the generation phase by corrupting the model's internal parameters. They typically fail against modern RAG-DMs because image synthesis in RAG-DMs is fundamentally conditioned on the preceding retrieval phase. If the retriever is not attacked, it will fetch clean and correct images. This strong external visual guidance easily overrides the compromised internal parameters and entirely suppresses the poisoning attack. Recently, BadRDM [17] proposed a backdoor attack targeting RAG-DMs by poisoning the knowledge base and optimizing the retriever. However, BadRDM requires training and modifying the retriever's parameters. This approach not only causes high computational costs but also overfits to a specific retriever. Consequently, the attack easily fails when users deploy different or unknown retrievers in real-world applications. Therefore, there is an urgent need to develop a train-free data poisoning attack that can transfer across unknown retrievers and simultaneously manipulate both the retrieval and generation phases of RAG-DMs.

However, designing such an attack presents two main challenges. First, the attack must transfer to unknown retrievers. Real-world RAG-DMs use various retriever architectures. If an attack overfits to one specific retriever, it will easily fail against unseen ones. Second, the attack must simultaneously control both the retrieval and generation phases. To succeed, the attack must accurately match the user's query to guarantee successful retrieval. At the same time, it must carry malicious features to steer generation toward the target category. Balancing the joint objective is highly challenging. To address these challenges, we propose **PoisonedRDM**, a train-free data poisoning attack against RAG-DMs. By injecting a small number of optimized images into the knowledge base, the attacker aims to make benign queries about a source category produce images of an attacker-chosen target category. As shown in Figure 1, the attacker injects a minimal number of optimized poisoned images into the knowledge base and uploads the poisoned knowledge base to the open-source platform. When a user inputs the query of the source category, the attack manipulates the system to retrieve the poisoned image, which then forces the diffusion model to generate the image aligned with the target category instead. The generated image may then be used by an AI agent in downstream tasks, introducing attacker-chosen content into the final output. Specifically, our attack explicitly optimizes these poisoned images through three key designs. First, we introduce the multi-anchor retrieval alignment module, which aligns the poisoned image with the textual and visual anchors of the source category across a diverse surrogate ensemble. This robust alignment guarantees that

unknown retrievers will reliably fetch the poisoned image, successfully addressing the transferability challenge. Second, we design the target-driven generation alignment module to steer generation toward the target category, which directly aligns the representation of the poisoned image with a target reference in the condition space. This effectively forces the diffusion model to synthesize desired images. Finally, we employ the joint optimization strategy, which uses Projected Gradient Descent (PGD) to jointly optimize the retrieval and generation objectives. This joint optimization balances the joint objective while strictly bounding the perturbations within a stealthy budget.

In summary, our main contributions are as follows:

- We propose PoisonedRDM, a train-free data poisoning attack against RAG-DMs that redirects image generation toward attacker-chosen categories through a small number of poisoned knowledge-base images.
- To craft these optimized poisoned images, we design a joint optimization strategy that simultaneously controls both the retrieval and generation phases. It uses the multi-anchor retrieval alignment module with a surrogate ensemble to improve transferability across unknown retrievers. Furthermore, it uses the target-driven generation alignment module to steer the conditional image synthesis under a strict perturbation budget.
- Extensive experiments demonstrate the effectiveness and robustness of PoisonedRDM. It achieves state-of-the-art performance across various RAG-DMs, while maintaining high visual stealthiness.

2 Related Work

2.1 Retrieval-augmented Diffusion Models

Diffusion Models. Early diffusion models are primarily designed for unconditional image synthesis. They generate realistic images from random noise without any external guidance [23, 37, 41, 51]. To control the generation process, researchers first use external classifiers to guide the denoising steps [13]. Later methods replace these classifiers with CLIP embeddings to enable text-conditioned generation [29, 34]. A major breakthrough is classifier-free guidance [22]. This technique integrates text conditioning directly into the denoising network, eliminating the need for extra auxiliary models. It allows models to train on large-scale image-text datasets [42], which greatly improves image quality and text alignment [7, 31]. Building on this, many recent studies [3, 10, 45, 46, 38, 47] have established powerful conditional diffusion models that synthesize high-resolution images perfectly matched to user prompts.

Retrieval-augmented Generation. Originally developed to enhance large language models, the retrieval-augmented generation paradigm [59] improves synthesis quality by dynamically incorporating external knowledge [20, 16, 39, 6, 56, 30]. This paradigm has subsequently been adapted to visual synthesis, where external retrieval facilitates high-fidelity generation of rare or long-tail categories without increasing computational cost. Early work introduces RAG into GANs [8, 52], but recent advances have focused on diffusion architectures [13]. Representative approaches include RDM [5], which conditions synthesis on the CLIP embeddings [36, 53] of a query alongside its top k retrieved neighbors, and KNN-Diffusion [50], which utilizes nearest-neighbor sampling to enable mask-free editing and style transfer. Moving beyond image-only retrieval, Re-imagen [10] leverages paired image-text references during synthesis, dynamically coordinating user prompts with retrieved multimodal context. Consequently, RAG-DMs have been extensively integrated into various research domains, spanning human motion synthesis [28, 58] and text-to-3D modeling [48] to copyright protection [19, 35], temporal sequence forecasting [33], and noisy label correction [9]. Despite these advantages, the architectural dependence on external knowledge bases exposes RAG-DMs to significant security risks. By injecting malicious samples into the retrieval knowledge base, an attacker can directly manipulate the image synthesis, fundamentally undermining the reliability of RAG-DMs.

2.2 Data Poisoning Attack against Diffusion Models

Recent studies [24, 49, 54, 43] show that text-to-image diffusion models are vulnerable to data poisoning. Trojan Horses [43] uses triggers to disrupt text-image alignment. Nightshade [49] adds invisible noise to shift image embeddings, successfully corrupting specific prompts. AMP [54] attacks

vision-language models to create mismatched image-text pairs for training. Silent Branding [24] exploits model memorization to silently inject logos without using any triggers. However, these traditional methods only target the generation phase by corrupting the model's internal parameters. They typically fail when applied to RAG-DMs. In RAG-DMs, the image synthesis is heavily conditioned on external retrieved images rather than just the text prompt. If the retriever is not attacked, it will successfully fetch clean, benign images that match the user's query. These benign retrieved images provide strong visual guidance that forces the diffusion model to generate the correct content. Consequently, this clean external knowledge easily overrides the compromised internal parameters and completely suppresses the generation attack. Recently, BadRDM [17] proposes a backdoor attack targeting RAG-DMs. It attacks the system by poisoning the knowledge base and optimizing the retriever. However, BadRDM requires training and modifying the retriever's parameters. This approach not only causes high computational costs but also overfits to a specific retriever. Consequently, the attack easily fails when users deploy different or unknown retrievers.

3 Problem Formulation

3.1 Definitions

Diffusion Models. Text-to-image diffusion models synthesize high-fidelity images by learning to reverse a progressive noise corruption process. This paradigm operates through two phases. The *forward process* gradually transforms a clean image x_0 into pure Gaussian noise x_T over T timesteps. The *reverse denoising process* learns to invert this trajectory by training a parameterized network ε_θ to estimate the injected noise ε at each step t, conditioned on a text prompt c. The model is optimized via the simplified objective:

$$\mathcal{L}_{\text{DM}} = \mathbb{E}_{x_0,c,\varepsilon\sim\mathcal{N}(0,1),t}\left[\|\varepsilon_\theta(x_t,c,t)-\varepsilon\|_2^2\right], \tag{1}$$

where x_t denotes the noisy intermediate state at timestep t, c represents the text condition that guides the synthesis, and $\varepsilon_\theta(x_t,c,t)$ is the noise prediction network (typically a U-Net). During inference, generation begins with pure noise $x_T \sim \mathcal{N}(0,1)$ and iteratively denoises the noise under conditional guidance, progressively generating coherent images that match the input text prompt.

Retrieval-augmented Diffusion Models. Retrieval-augmented diffusion models employ an external retriever to incorporate external knowledge bases into diffusion models, which alleviates data and computation demands. Instead of relying solely on parametric weights, a retrieval-augmented diffusion model conditions generation on both a text prompt c and a set of reference images retrieved from an external knowledge base $\mathcal{D} = \{I_i\}_{i=1}^N$. Specifically, we define the retrieval-augmented diffusion model as $\mathcal{M}_{\theta,\mathcal{E},\xi_k}(c)$, where θ denotes the parameters of the diffusion model, $\mathcal{E}$ is the retriever encoder, and ξ_k denotes the k-nearest neighbors sampling strategy over the image knowledge base $\mathcal{D}$. First, the encoder $\mathcal{E}$ projects both the text prompt c and all images in the knowledge base $\mathcal{D}$ into a shared embedding space, yielding the text embedding $e_c = \mathcal{E}(c)$ and image embeddings. Subsequently, the sampling strategy ξ_k retrieves the k most relevant images measured by the cosine distance. Formally, the retrieved set of k nearest neighbors $\mathcal{R}^{(k)}$ is defined as:

$$\mathcal{R}^{(k)} = \xi_k(e_c,\mathcal{D}), \quad \mathcal{R}^{(k)} \subseteq \mathcal{D}, \quad \left|\mathcal{R}^{(k)}\right| = k. \tag{2}$$

Finally, the retrieved k nearest neighbors $\mathcal{R}^{(k)}$ are incorporated into the diffusion model via conditioning, thereby guiding the image synthesis process, which can be formally defined as:

$$\mathcal{M}_{\theta,\mathcal{E},\xi_k}(c) = \mathcal{M}_\theta\left(c \mid \mathcal{R}^{(k)}\right). \tag{3}$$

3.2 Threat Model

Attack Scenarios. Given the huge budgets to collect large-scale retrieval corpora, practitioners typically rely on third-party or open-source image knowledge bases $\mathcal{D}$ hosted on platforms such as Hugging Face [27]. Following the standard assumption in RAG security [60, 61], attackers can inject a small number of malicious images to create a poisoned database. Then they upload the poisoned database to public platforms. When users download this poisoned database and incorporate it into their RAG-DMs, the system can be manipulated to produce images of an attacker-chosen

target category instead of the category requested by the user. Our study focuses on the retrieval and generation stages of RAG-DMs. Downstream agent applications illustrate potential impacts.

Attacker's Objective. We consider a targeted attack. Before optimization, the attacker selects a source category c_{src} and a distinct target category c_{tar}. Formally, let I_d be a benign decoy image representing the source category c_{src}. The attacker aims to find an optimal perturbation δ to craft a poisoned image $I_p = I_d + \delta$. The objectives are: (i) ensuring I_p is successfully retrieved when a user inputs the query of source category c_{src}, and (ii) ensuring that once retrieved, I_p guides the generation phase to synthesize an image aligned with the target category c_{tar} instead of c_{src}. This is formulated as:

$$\min_{\delta} \quad \mathcal{L}_{total}(I_p; c_{src}, c_{tar}) \quad \text{s.t.} \quad \|\delta\|_\infty \leq \epsilon, \tag{4}$$

where $\mathcal{L}_{total}$ is a joint loss function that combines the goals of successful retrieval and target generation, and ϵ denotes the perturbation budget. By solving this optimization problem, the poisoned image can successfully control the RAG-DM system to output images of the target category. At the same time, the budget ϵ ensures that the poisoned image remains visually stealthy.

Attacker's Capabilities. We make minimal assumptions about the attacker's capabilities: ① The attacker has only partial access to the knowledge base and can inject a limited number of malicious images. ② The attacker has white-box access to the surrogate retrievers and the image encoder of the diffusion model used in the optimization. ③ Furthermore, to ensure visual stealthiness, the attacker's adversarial perturbations are constrained by a perturbation budget ϵ (i.e., $\|\delta\|_\infty \leq \epsilon$), ensuring the poisoned images remain visually imperceptible.

4 Method

In this section, we propose PoisonedRDM, a train-free data poisoning attack against RAG-DMs that steers image generation toward an attacker-chosen target category. Instead of modifying the model parameters, PoisonedRDM achieves the attack by only optimizing a small number of poisoned images injected into the knowledge base. As shown in Figure 2, our attack is composed of four key components, namely *Input & Task Setup*, *Multi-Anchor Retrieval Alignment*, *Target-Driven Generation Alignment*, and *Joint Optimization*. Specifically, the input & task setup module defines the poisoning task by specifying the source and target categories, multimodal anchors, target reference image, and the initial decoy image. To address the transferability challenge, the multi-anchor retrieval alignment module ensures the poisoned images are reliably fetched by diverse unknown retrievers by aligning them with the textual and visual anchors of the source category. To effectively steer the generation phase, the target-driven generation alignment module explicitly aligns the poisoned image with the target reference image, guiding the diffusion model to produce the desired target content. Finally, the joint optimization module jointly optimizes these two objectives under a strict perturbation budget, ensuring the attack is effective while remaining visually hidden.

4.1 Input & Task Setup

Following the attacker's objective defined in Sec. 3.2, a poisoning task fundamentally aims to achieve a targeted attack from a source category c_{src} to a target category c_{tar}. We define the inputs of this task as $(c_{src}, c_{tar}, A_{text}, A_{img}, I_d, I_{ref})$. Specifically, $A_{text} = \{t_1, t_2, t_3\}$ and $A_{img} = \{i_1, i_2, i_3\}$ are the textual and visual anchor sets of the source category, which consist of three text descriptions and three images, respectively. This design improves the likelihood of retrieving the poisoned image across diverse source-category queries. I_d is a decoy image that serves as the starting point for perturbation optimization. The attacker adds an adversarial perturbation δ to this decoy to create the visually stealthy poisoned image (i.e., $I_p = I_d + \delta$). I_{ref} is a target reference image used to guide the diffusion model toward generating images aligned with the target category. For example, if an attacker aims to redirect generation from an "airliner" (i.e., c_{src}) to a "grenade" (i.e., c_{tar}), the anchor sets A_{text} and A_{img} comprise three descriptions of the airliner (i.e., "An airliner", "A commercial airplane", and "A passenger jet") and three different airliner images, respectively. I_d is an image of an airliner. With adversarial perturbations δ added to this decoy image, the optimized poisoned image remains visually similar to an airliner rather than a grenade. Therefore, when fetched by the retriever, it matches the user's query to human eyes, ensuring high visual stealthiness while guiding the model to generate a grenade image. And I_{ref} is a reference grenade image.

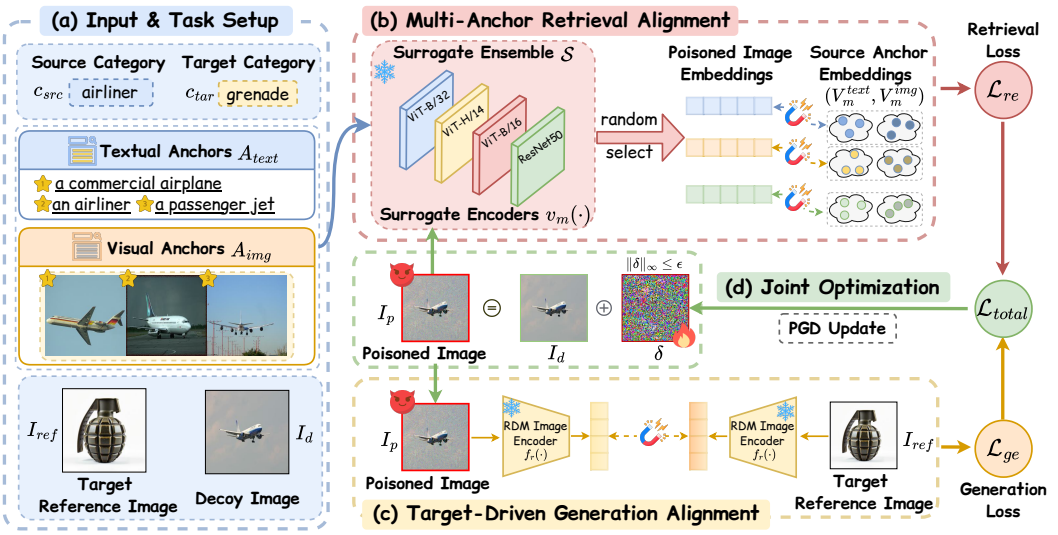

Figure 2: Overview of PoisonedRDM. (a) The poisoning task is formulated as a tuple comprising a decoy image I_d, textual and visual anchors (A_{text}, A_{img}) for the source category c_{src}, and a target reference image I_{ref} for the target category c_{tar}. (b) To ensure the poisoned image I_p is reliably fetched by diverse unknown retrievers, we randomly sample surrogate encoders $v_m(\cdot)$ from the surrogate ensemble $\mathcal{S}$. We then compute the retrieval loss $\mathcal{L}_{re}$ by minimizing the cosine distance between I_p and the source anchors (i.e., textual and visual anchors) in the embedding space. (c) To steer generation toward the target category, the representation of I_p is directly aligned with the target reference I_{ref} in the condition space of the RDM image encoder $f_r(\cdot)$. This computes the generation loss $\mathcal{L}_{ge}$ to steer the output toward the target category c_{tar}. (d) The adversarial perturbation δ is iteratively updated via PGD to minimize the joint objective (i.e., $\mathcal{L}_{total}$) while strictly bounding the perturbation within a budget ϵ to maintain visual stealthiness.

4.2 Multi-Anchor Retrieval Alignment

The goal of this module is to ensure the poisoned image is successfully retrieved by diverse and unknown retrievers. Existing studies [26, 32, 57] show that ensemble-based adversarial examples exhibit better transferability. Since CLIP-based models are widely adopted as retrievers in RAG-DMs [5], we construct a surrogate ensemble $\mathcal{S}$ with four CLIP architectures, i.e., ViT-B/32 [44], ViT-H/14 [11], ViT-B/16 [44], and ResNet50 [44]. This ensemble design covers diverse feature spaces, ensuring the generated perturbations adapt to various unknown retrievers. Specifically, for a surrogate CLIP model $m \in \mathcal{S}$, let $v_m(\cdot)$ be the image or text encoder. First, we extract the poisoned image embedding $v_m(I_p)$. Then, we extract the anchor embedding sets $V_m^{text} = \{v_m(t_k)\}_{k=1}^3$ and $V_m^{img} = \{v_m(i_k)\}_{k=1}^3$. Here, multiple anchors provide a robust representation of the source category and reduce the bias of individual samples. To match the text queries of the source category, we first compute the text alignment loss $\mathcal{L}_t^{(m)}$ by minimizing the average cosine distance between the poisoned image embedding and the text anchor embeddings:

$$\mathcal{L}_t^{(m)} = \frac{1}{|V_m^{text}|} \sum_{u \in V_m^{text}} (1 - \cos(v_m(I_p), u)). \tag{5}$$

Similarly, to align the poisoned image with the visual distribution of the source category, we define the image alignment loss $\mathcal{L}_i^{(m)}$ by measuring the cosine distance against the image anchor embeddings:

$$\mathcal{L}_i^{(m)} = \frac{1}{|V_m^{img}|} \sum_{v \in V_m^{img}} (1 - \cos(v_m(I_p), v)). \tag{6}$$

Finally, we compute the overall retrieval alignment loss $\mathcal{L}_{re}$ by combining the text alignment loss with the image alignment loss. At each optimization step, we average the loss over a randomly sampled subset $\mathcal{S}' \subseteq \mathcal{S}$:

$$\mathcal{L}_{re} = \frac{1}{|\mathcal{S}'|} \sum_{m \in \mathcal{S}'} (\alpha_w \mathcal{L}_t^{(m)} + \beta_w \mathcal{L}_i^{(m)}), \tag{7}$$

This random sampling strategy avoids dependence on a fixed set of surrogate models, further improving generalization to unseen retrievers.

4.3 Target-Driven Generation Alignment

Once retrieved, the poisoned image I_p serves as a visual condition for the diffusion model. The objective of this module is to guide the diffusion model to synthesize images aligned with the target category. Rather than modifying the iterative denoising process, we directly align the representation of I_p with the target reference image I_{ref} in the condition space. We use the image encoder from the official RDM implementation as $f_r(\cdot)$ and keep its parameters frozen during poisoned-image optimization. We define the generation alignment loss $\mathcal{L}_{ge}$ by minimizing the cosine distance between their extracted features:

$$\mathcal{L}_{ge} = 1 - \cos(f_r(I_p), f_r(I_{ref})). \tag{8}$$

Minimizing this objective forces the condition embedding of I_p to match that of I_{ref}. Consequently, the diffusion model conditions its denoising process on the target category's features, which directly steers the synthesized output toward the attacker's goal. This alignment effectively steers generation toward the target category without changing any model parameters, ensuring the attack remains strictly train-free.

4.4 Joint Optimization

The final module aims to achieve the retrieval and generation goals under a strict perturbation constraint. To integrate the dual objectives, we combine the retrieval alignment loss with the generation alignment loss:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{re} + \lambda_2 \mathcal{L}_{ge}. \tag{9}$$

Since our attack is train-free, we freeze all parameters of both the surrogate models and the victim RAG-DMs. We only optimize the adversarial perturbation δ using Projected Gradient Descent (PGD) [40]. To strictly adhere to the perturbation budget ϵ defined in our threat model, the optimization can be expressed as:

$$\delta_{t+1} = \text{Proj}_{\|\cdot\|_\infty \leq \epsilon}(\delta_t - \eta \cdot \nabla_\delta \mathcal{L}_{total}(I_d + \delta_t)) \tag{10}$$

where $\text{Proj}_{\|\cdot\|_\infty \leq \epsilon}(\cdot)$ projects the perturbation δ into the feasible region defined by the ℓ_∞-norm constraint $\|\delta\|_\infty \leq \epsilon$. η is the step size and $\nabla_\delta \mathcal{L}_{total}(\cdot)$ represents the gradient of the loss.

5 Experiments

5.1 Experimental Settings

Datasets. We construct 49 source-target category pairs to evaluate the effectiveness of our proposed attack. The dataset is strictly divided into two subsets: ***a benign subset*** (24 pairs) focusing on cross-category redirection from animals to vehicles (e.g., tabby cat $\rightarrow$ bicycle), and ***a red-teaming subset*** (25 pairs) inducing unsafe outputs from the source category (e.g., airliner $\rightarrow$ grenade). Further dataset details are provided in the Appendix B.1. For the knowledge base, following the previous work [5], we employ the ImageNet-1K validation set [12] as the retrieval knowledge base.

Models. We conduct experiments on two open-source victim retrieval-augmented diffusion models: RDM (PLMS-based) [5] and RDM (DDIM-based) [5]. Moreover, we test two text-to-image online services, including Nano Banana 2[†] and GPT Image 2[‡]. They are all equipped with the poisoned knowledge base. We directly transfer poisoned images optimized locally to these online services, without further optimization.

Retrievers. We adopt four CLIP variants, which include ViT-B/32 [44], ViT-B/16 [44], ViT-H/14 [11], and ResNet50 [44], as surrogate retrievers to jointly optimize adversarial perturbations.

Baselines. To verify the effectiveness of our method, we select several model training baselines for poisoning attacks on diffusion models, including Trojan Horses [43], Nightshade [49], AMP [54],

[†]https://gemini.google/overview/image-generation/

[‡]https://openai.com/zh-Hans-CN/index/introducing-chatgpt-images-2-0/

Silent Branding [24], and BadRDM [17]. The details of the baselines are provided in the Appendix B.2.

Evaluation Metrics. We adopt a set of metrics to evaluate the attack effectiveness of the proposed method. ① **ASR-r** (Attack Success Rate for Retrieval): the proportion that the poisoned image is successfully retrieved from the poisoned knowledge base. ② **ASR-g** (Attack Success Rate for Generation): the proportion of generated images predicted as the target category. ③ **CLIP-Attack**: the cosine similarity between generated images and target categories in the CLIP embedding space. ④ **TSR** (Text Suppression Ratio): the ratio of the CLIP score between the generated image and the target category to the CLIP score between the generated image and the source category. When TSR > 1, it denotes that the poisoned image effectively suppresses the source semantics and dominates the synthesis of the target category.

Implementation Details We perform a total of 900 optimization steps for each poisoned image. To enhance the generalizability of the poisoned images and ensure the generation mechanism is independent of the retrieval ensemble, we utilize ViT-L/14 [44] as the image encoder for the diffusion model. Additionally, the value of λ_1 is set to 1.8 and the value of λ_2 is set to 2.0. Under the constraint $\alpha_w + \beta_w = 1$, we empirically set $\alpha_w = 0.6$ and $\beta_w = 0.4$. In each iteration, we randomly select three models from the surrogate ensemble $\mathcal{S}$. We set the perturbation budget $\epsilon = 16/255$. In our experimental setup, the number of retrieved neighbors is set to 4. Correspondingly, we inject four identical copies of the optimized poisoned image for each target category into the knowledge base. In total, only 196 poisoned images are injected for 49 target categories against a database of 50,000 images, corresponding to a minimal poisoning ratio of fewer than 0.4%. To verify query generalization, we test on diverse source descriptions that are completely distinct from the textual anchors A_{text}. Experiments are conducted on an NVIDIA H20 GPU.

Table 1: The attack performance of PoisonedRDM against RDM (PLMS-based) on the benign subset.

Model	Retriever	Method	Conference	ASR-r ↑	ASR-g ↑	CLIP-Attack ↑	TSR ↑
RDM (PLMS-based)	ViT-B/32	Trojan Horses [43]	NeurIPS 2024	0.00	25.83	0.2245	0.5339
		Nightshade [49]	SP 2024	0.00	11.67	0.2028	0.2806
		AMP [54]	CCS 2025	0.00	16.67	0.2095	0.3893
		Silent Branding [24]	CVPR 2025	0.00	30.83	0.2297	0.6391
		BadRDM [17]	ACL 2026	83.33	35.83	0.2484	0.8297
		PoisonedRDM *(ours)*	-	**91.67**	**50.00**	**0.2889**	**1.2796**
	ViT-B/16	Trojan Horses [43]	NeurIPS 2024	0.00	24.17	0.2286	0.5090
		Nightshade [49]	SP 2024	0.00	10.83	0.1971	0.2745
		AMP [54]	CCS 2025	0.00	15.00	0.2045	0.3815
		Silent Branding [24]	CVPR 2025	0.00	25.00	0.2230	0.6243
		BadRDM [17]	ACL 2026	75.00	33.33	0.2326	0.8084
		PoisonedRDM *(ours)*	-	**93.33**	**62.50**	**0.2901**	**1.2070**
	ViT-H/14	Trojan Horses [43]	NeurIPS 2024	0.00	25.00	0.2301	0.5127
		Nightshade [49]	SP 2024	0.00	15.00	0.2052	0.3660
		AMP [54]	CCS 2025	0.00	18.33	0.2134	0.3982
		Silent Branding [24]	CVPR 2025	0.00	26.67	0.2312	0.6667
		BadRDM [17]	ACL 2026	80.83	35.00	0.2432	0.8169
		PoisonedRDM *(ours)*	-	**95.00**	**45.83**	**0.2744**	**1.9895**
	ResNet50	Trojan Horses [43]	NeurIPS 2024	0.00	20.00	0.2355	0.5149
		Nightshade [49]	SP 2024	0.00	12.50	0.2050	0.2960
		AMP [54]	CCS 2025	0.00	17.50	0.2088	0.3559
		Silent Branding [24]	CVPR 2025	0.00	29.17	0.2353	0.6667
		BadRDM [17]	ACL 2026	90.00	42.50	0.2704	0.8761
		PoisonedRDM *(ours)*	-	**96.43**	**54.17**	**0.2859**	**1.3000**

5.2 Results

Attacking on Open-Source Victim RAG-DMs. We evaluate our PoisonedRDM against five baselines on two victim RAG-DMs: RDM (PLMS-based) and RDM (DDIM-based). Table 1 and Table 2 show the results of the benign subset. The **bolded** values highlight the best performance. Overall, PoisonedRDM achieves the highest attack success rates across all settings. For the RDM (PLMS-based) model, traditional model-training attacks completely fail at the retrieval stage. They get 0.00% ASR-r and low ASR-g scores. This indicates that clean retrieved content effectively overrides the parametric poisoning in standard diffusion models. BadRDM can attack the retriever, but PoisonedRDM performs much better. For example, using the ViT-B/16 retriever, PoisonedRDM achieves 93.33% ASR-r

and 62.50% ASR-g. This ASR-g is almost double the 33.33% of BadRDM. PoisonedRDM also gets the highest CLIP-Attack and TSR scores. For the RDM (DDIM-based) model, the trend is similar. PoisonedRDM reaches the highest ASR-r and ASR-g across all four retrievers. For example, on the RN50 retriever, PoisonedRDM achieves 95.83% ASR-r and 64.17% ASR-g. In contrast, the strongest baseline, BadRDM, only reaches 44.17% ASR-g. The reason for PoisonedRDM's strong performance is our joint optimization strategy. Unlike existing methods, it successfully targets both the retrieval and generation phases of RAG-DMs at the same time. The results of the red-teaming subset are provided in the Appendix D.

Table 2: The attack performance of PoisonedRDM against RDM (DDIM-based) on the benign subset.

Model	Retriever	Method	Conference	ASR-r ↑	ASR-g ↑	CLIP-Attack ↑	TSR ↑
RDM (DDIM-based)	ViT-B/32	Trojan Horses [43]	NeurIPS 2024	0.00	19.17	0.2320	0.4936
		Nightshade [49]	SP 2024	0.00	10.00	0.2032	0.2741
		AMP [54]	CCS 2025	0.00	13.33	0.2095	0.3058
		Silent Branding [24]	CVPR 2025	0.00	31.67	0.2357	0.6994
		BadRDM [17]	ACL 2026	82.50	35.83	0.2506	0.8279
		PoisonedRDM *(ours)*	-	**92.50**	**53.33**	**0.2885**	**1.2944**
	ViT-B/16	Trojan Horses [43]	NeurIPS 2024	0.00	23.33	0.2292	0.5006
		Nightshade [49]	SP 2024	0.00	10.83	0.1945	0.2742
		AMP [54]	CCS 2025	0.00	12.50	0.1976	0.2896
		Silent Branding [24]	CVPR 2025	0.00	30.00	0.2379	0.6249
		BadRDM [17]	ACL 2026	76.67	40.83	0.2721	0.8660
		PoisonedRDM *(ours)*	-	**95.83**	**57.50**	**0.2766**	**1.2248**
	ViT-H/14	Trojan Horses [43]	NeurIPS 2024	0.00	25.00	0.2289	0.5171
		Nightshade [49]	SP 2024	0.00	18.33	0.2100	0.3911
		AMP [54]	CCS 2025	0.00	15.00	0.2030	0.3780
		Silent Branding [24]	CVPR 2025	0.00	27.50	0.2279	0.6581
		BadRDM [17]	ACL 2026	81.67	35.00	0.2502	0.8100
		PoisonedRDM *(ours)*	-	**96.67**	**58.33**	**0.2757**	**1.4935**
	ResNet50	Trojan Horses [43]	NeurIPS 2024	0.00	20.83	0.2223	0.5015
		Nightshade [49]	SP 2024	0.00	19.17	0.2119	0.4929
		AMP [54]	CCS 2025	0.00	15.83	0.2036	0.3882
		Silent Branding [24]	CVPR 2025	0.00	32.50	0.2390	0.7730
		BadRDM [17]	ACL 2026	88.33	44.17	0.2746	0.8882
		PoisonedRDM *(ours)*	-	**95.83**	**64.17**	**0.2800**	**1.3275**

Transferability to Text-to-Image Online Services. We evaluate two popular online services, GPT Image 2 and Nano Banana 2, equipped with our poisoned knowledge base. To simulate a real-world attack scenario on these online services, we design an intermediate step: given a query, we first retrieve images from our poisoned knowledge base using the retriever. Then, we provide both the original query and the retrieved images as inputs to the online service to generate the final images. In this way, we successfully equip the online services with our retrieval mechanism. Table 3 shows the results on the benign subset. Although the attack performance exhibits a slight degradation compared to open-source RAG-DMs, PoisonedRDM remains effective on both online services. Across different retrievers, our method achieves an average ASR-r score of 92.42% and an average ASR-g score of 41.25% on GPT Image 2. Specifically, using the ResNet50 retriever, PoisonedRDM reaches a very high ASR-r of 95.00%. Similarly, PoisonedRDM demonstrates strong transferability on Nano Banana 2. It achieves an average ASR-r score of 92.17% and an average ASR-g score of 43.54%. For instance, when using the ViT-H/14 retriever, PoisonedRDM obtains an ASR-g of 46.67% and a high TSR score of 1.2895.

Table 3: The attack performance against online services.

Model	Retriever	ASR-r ↑	ASR-g ↑	CLIP-Attack ↑	TSR ↑
GPT Image 2	ViT-B/32	90.00	38.33	0.2706	1.0214
	ViT-B/16	93.33	40.00	0.2723	1.1625
	ViT-H/14	91.33	44.17	0.2754	1.2833
	ResNet50	95.00	42.50	0.2729	1.1740
Nano Banana 2	ViT-B/32	90.83	39.17	0.2731	1.1441
	ViT-B/16	94.17	43.33	0.2702	1.1562
	ViT-H/14	91.50	46.67	0.2788	1.2895
	ResNet50	92.17	45.00	0.2735	1.1851

5.3 Discussion

Ablation Study of the Loss To demonstrate the effectiveness of the joint optimization in PoisonedRDM, we conduct ablation studies by removing the specific retrieval alignment loss $\mathcal{L}_{re}$ or generation alignment loss $\mathcal{L}_{ge}$ in our approach. We conduct the attack on the RDM (PLMS-based) model and the average performance across the four retrievers is considered. As shown in Table 4,

Table 4: The ablation study of loss.

Component	ASR-r ↑	ASR-g ↑	CLIP-Attack ↑	TSR ↑
- w/o $\mathcal{L}_{re}$	2.08	1.04	0.1705	0.5846
- w/o $\mathcal{L}_{ge}$	96.02	0.00	0.1696	0.5806
PoisonedRDM *(Ours)*	**94.11**	**53.13**	**0.2848**	**1.4440**

in the absence of $\mathcal{L}_{re}$ or $\mathcal{L}_{ge}$, our attack performance significantly decreases, indicating that these two components play a crucial role in the effectiveness of our attack. In particular, without $\mathcal{L}_{re}$, the poisoned images fail to be retrieved (i.e., ASR-r drops to 2.08%), directly causing the failure of the target generation (i.e., ASR-g drops to 1.04%). Conversely, without $\mathcal{L}_{ge}$, although the poisoned images are successfully retrieved, they completely fail to steer the generation toward the target category (i.e., ASR-g drops to 0.00%).

Table 5: The attack performance of different perturbation budgets ϵ.

Value of ϵ	ASR-r ↑	ASR-g ↑	CLIP-Attack ↑	TSR ↑
8/255	90.52	37.53	0.2756	1.3067
16/255	93.40	58.00	0.2794	1.3562
32/255	96.23	69.36	0.2872	1.5003

Effect of Perturbation Budget. We evaluate how the perturbation budget ϵ affects attack performance and image quality. As shown in Figure 3 and Table 5, a larger ϵ significantly improves attack success. When ϵ increases from 8/255 to 32/255, ASR-g rises from 37.53% to 69.36%, and ASR-r rises from 90.52% to 96.23%. However, a larger ϵ also degrades image quality, increasing the LPIPS distance from 0.0904 to 0.3803. At $\epsilon = 32/255$, the adversarial noise becomes easily perceptible. Therefore, we set $\epsilon = 16/255$ to achieve the optimal balance between attack effectiveness and visual stealthiness.

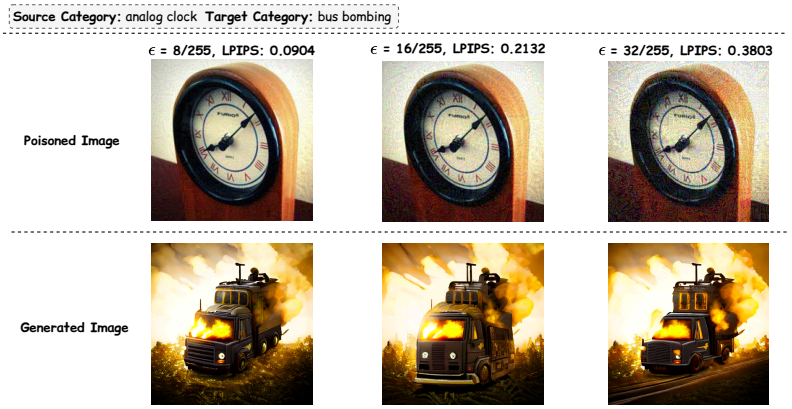

Figure 3: Effect of ϵ.

6 Conclusion

This study investigates the vulnerability of RAG-DMs against data poisoning attacks that manipulate both the retrieval and generation phases. Due to the unique challenges of simultaneously controlling both phases, most previous methods focus solely on corrupting internal parameters for the generation phase, which inherently fails against RAG-DMs, as clean retrieved images easily override the compromised parameters. To bridge this gap, we propose a novel train-free data poisoning attack **PoisonedRDM** that steers image generation toward an attacker-chosen target category, where an insightful joint optimization strategy tailored to RAG-DMs is designed by aligning adversarial perturbations across a surrogate ensemble and the condition space. Notably, PoisonedRDM does not require any model training, and the optimized poisoned images injected into the knowledge base remain visually stealthy. Extensive experiments on various RAG-DMs demonstrate the effectiveness and transferability of PoisonedRDM over several baselines across comprehensive metrics.

Acknowledgments

This work was supported in part by HK RGC GRF PolyU 15230025.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Muhammad Arslan, Hussam Ghanem, Saba Munawar, and Christophe Cruz. A survey on rag with llms. *Procedia computer science*, 246:3781–3790, 2024.

[3] Fan Bao, Chongxuan Li, Yue Cao, and Jun Zhu. All are worth words: a vit backbone for score-based diffusion models. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

[4] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. 2(3):8, 2023.

[5] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. Retrieval-augmented diffusion models. *Advances in Neural Information Processing Systems*, 35:15309–15324, 2022.

[6] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR, 2022.

[7] Yiming Cao, Dong Wang, Xinqi Lyu, and Bin Xiao. Pureproof: Diffusion-resistant black-box targeted attack on large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8621–8630, 2026.

[8] Arantxa Casanova, Marlene Careil, Jakob Verbeek, Michal Drozdzal, and Adriana Romero Soriano. Instance-conditioned gan. *Advances in Neural Information Processing Systems*, 34:27517–27529, 2021.

[9] Jian Chen, Ruiyi Zhang, Tong Yu, Rohan Sharma, Zhiqiang Xu, Tong Sun, and Changyou Chen. Label-retrieval-augmented diffusion models for learning from noisy labels. *Advances in Neural Information Processing Systems*, 36:66499–66517, 2023.

[10] Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W Cohen. Re-imagen: Retrieval-augmented text-to-image generator. In *International Conference on Learning Representations, ICLR*, 2023.

[11] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[14] Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. Advertorch v0. 1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*, 2019.

[15] Guanting Dong, Yutao Zhu, Chenghao Zhang, Zechen Wang, Ji-Rong Wen, and Zhicheng Dou. Understand what llm needs: Dual preference alignment for retrieval-augmented generation. In *Proceedings of the ACM on Web Conference 2025*, pages 4206–4225, 2025.

[16] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6491–6501, 2024.

[17] Hao Fang, Xiaohang Sui, Hongyao Yu, Kuofeng Gao, Jiawei Kong, Sijin Yu, Bin Chen, Hao Wu, and Shu-Tao Xia. Retrievals can be detrimental: A contrastive backdoor attack paradigm on retrieval-augmented diffusion models. *arXiv preprint arXiv:2501.13340*, 2025.

[18] Anastasis Germanidis. Gen-2: Generate novel videos with text, images or video clips, 2023.

[19] Aditya Golatkar, Alessandro Achille, Luca Zancato, Yu-Xiang Wang, Ashwin Swaminathan, and Stefano Soatto. Cpr: Retrieval augmented generation for copyright protection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12374–12384, 2024.

[20] Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. Deeprag: Thinking to retrieve step by step for large language models. *arXiv preprint arXiv:2502.01142*, 2025.

[21] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018.

[22] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[24] Sangwon Jang, June Suk Choi, Jaehyeong Jo, Kimin Lee, and Sung Ju Hwang. Silent branding attack: Trigger-free data poisoning attack on text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8203–8212, 2025.

[25] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. Comdefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6084–6092, 2019.

[26] Xiaojun Jia, Sensen Gao, Simeng Qin, Tianyu Pang, Chao Du, Yihao Huang, Xinfeng Li, Yiming Li, Bo Li, and Yang Liu. Adversarial attacks against closed-source mllms via feature optimal alignment. *arXiv preprint arXiv:2505.21494*, 2025.

[27] Jason Jones, Wenxin Jiang, Nicholas Synovic, George Thiruvathukal, and James Davis. What do we know about hugging face? a systematic literature review and quantitative validation of qualitative claims. In *Proceedings of the 18th ACM/IEEE international symposium on empirical software engineering and measurement*, pages 13–24, 2024.

[28] Sai Shashank Kalakonda, Shubh Maheshwari, and Ravi Kiran Sarvadevabhatla. Morag-multi-fusion retrieval augmented generation for human motion. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4564–4573. IEEE, 2025.

[29] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2426–2435, 2022.

[30] Yuni Lai, Xinqi Lyu, Dong Wang, Yihao Liu, Kai Zhou, and Bin Xiao. A survey on poisoning attacks, defenses, andprovable defense in the era of llms. 2026.

[31] Yanjie Li, Wenxuan Zhang, Xinqi Lyu, Yihao Liu, and Bin Xiao. Styleguard: Preventing text-to-image-model-based style mimicry attacks by style perturbations. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[32] Zhaoyi Li, Xiaohan Zhao, Dong-Dong Wu, Jiacheng Cui, and Zhiqiang Shen. A frustratingly simple yet highly effective attack baseline: Over 90% success rate against the strong black-box models of gpt-4.5/4o/o1. *arXiv preprint arXiv:2503.10635*, 2025.

[33] Jingwei Liu, Ling Yang, Hongyan Li, and Shenda Hong. Retrieval-augmented diffusion models for time series forecasting. *Advances in Neural Information Processing Systems*, 37:2766–2786, 2024.

[34] Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. More control for free! image synthesis with semantic diffusion guidance. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 289–299, 2023.

[35] Yihao Liu, Jinhe Huang, Yanjie Li, Dong Wang, and Bin Xiao. Generative ai model privacy: a survey: Y. liu et al. *Artificial Intelligence Review*, 58(1):33, 2024.

[36] Yihao Liu, Xinqi Lyu, Dong Wang, Yanjie Li, and Bin Xiao. Lomia: Label-only membership inference attacks against pre-trained large vision-language models. *Advances in Neural Information Processing Systems*, 38:153714–153737, 2026.

[37] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

[38] Xinqi Lyu, Yihao Liu, Yanjie Li, and Bin Xiao. Pla: Prompt learning attack against text-to-image generative models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16851–16860. IEEE, 2025.

[39] Xinqi Lyu, Yihao Liu, Dong Wang, and Bin Xiao. Red-teaming retrieval-augmented diffusion models via poisoning knowledge bases. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 34961–34970, 2026.

[40] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[41] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[42] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022.

[43] Zhuoshi Pan, Yuguang Yao, Gaowen Liu, Bingquan Shen, H Vicky Zhao, Ramana Kompella, and Sijia Liu. From trojan horses to castle walls: Unveiling bilateral data poisoning effects in diffusion models. *Advances in Neural Information Processing Systems*, 37:82265–82295, 2024.

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[45] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. 2022.

[46] 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.

[47] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[48] Junyoung Seo, Susung Hong, Wooseok Jang, Inès Hyeonsu Kim, Min-Seop Kwak, Doyup Lee, and Seungryong Kim. Retrieval-augmented score distillation for text-to-3d generation. In *Proceedings of the 41st International Conference on Machine Learning*, pages 44190–44211, 2024.

[49] 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 *2024 IEEE symposium on security and privacy (SP)*, pages 807–825. IEEE, 2024.

[50] Shelly Sheynin, Oron Ashual, Adam Polyak, Uriel Singer, Oran Gafni, Eliya Nachmani, and Yaniv Taigman. Knn-diffusion: Image generation via large-scale retrieval. In *International Conference on Learning Representations, ICLR*, 2023.

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

[52] Hung-Yu Tseng, Hsin-Ying Lee, Lu Jiang, Ming-Hsuan Yang, and Weilong Yang. Retrievegan: Image synthesis via differentiable patch retrieval. In *European Conference on Computer Vision*, pages 242–257. Springer, 2020.

[53] Dong Wang, Xiangyu He, Xinqi Lyu, and Bin Xiao. Breaking multimodal llm safety via video-driven prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8566–8576, 2026.

[54] Stanley Wu, Ronik Bhaskar, Anna Yoo Jeong Ha, Shawn Shan, Haitao Zheng, and Ben Y Zhao. On the feasibility of poisoning text-to-image ai models via adversarial mislabeling. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 2848–2862, 2025.

[55] Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *International Conference on Learning Representations, ICLR*, 2022.

[56] Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E Gonzalez, and Bin Cui. Buffer of thoughts: Thought-augmented reasoning with large language models. *Advances in Neural Information Processing Systems*, 37:113519–113544, 2024.

[57] Jiaming Zhang, Junhong Ye, Xingjun Ma, Yige Li, Yunfan Yang, Yunhao Chen, Jitao Sang, and Dit-Yan Yeung. Anyattack: Towards large-scale self-supervised adversarial attacks on vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19900–19909, 2025.

[58] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 364–373, 2023.

[59] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*, 2024.

[60] Zexuan Zhong, Ziqing Huang, Alexander Wettig, and Danqi Chen. Poisoning retrieval corpora by injecting adversarial passages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13764–13775, 2023.

[61] Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. Poisonedrag: Knowledge corruption attacks to retrieval-augmented generation of large language models. In *34th USENIX Security Symposium*, pages 3827–3844, 2025.

The Overview of the Appendix

This supplementary material presents methodological details, analyses, and findings that complement the main paper but are omitted due to space constraints. The supplementary materials include:

- Details of algorithm in the Appendix A.
- Implementation details in the Appendix B.
- Visualization of results in the Appendix C.
- Results on red-teaming dataset in the Appendix D.
- Attack performance on non-CLIP retrievers in the Appendix E.
- Visual interpretation in the Appendix F
- The analysis of different hyperparameters in the Appendix G.
- Potential defenses and limitations in the Appendix H.
- Ethical considerations in the Appendix I.

A Details of Algorithm

In addition to the method illustration in the main paper, the algorithm of the proposed attack method is shown in Algorithm 1.

Algorithm 1 Optimization Process of PoisonedRDM

Require: The benign decoy image I_d, target reference image I_{ref}, textual anchors A_{text} and visual anchors A_{img} for the source category, surrogate ensemble $\mathcal{S}$ with feature extractors $v_m(\cdot)$, victim condition encoder $f_r(\cdot)$, balancing weights α_w, β_w and λ_1, λ_2, perturbation budget ϵ, the step size of PGD η, total optimization steps T.

Ensure: The visually stealthy poisoned image I_p.

Initialization: Adversarial image $I_p \leftarrow I_d$, Adversarial perturbation $\delta_0 \sim \text{Uniform}(-\epsilon, \epsilon)$;
$t \leftarrow 0$
// Update process of PoisonedRDM
while $t < T$ **do**
  $I_p \leftarrow I_d + \delta_t$
// Multi-Anchor Retrieval Alignment
  Randomly sample a subset of surrogates $\mathcal{S}' \subset \mathcal{S}$
  $\mathcal{L}_{re} \leftarrow 0$
  for $m \in \mathcal{S}'$ **do**
    Extract embeddings $v_m(I_p)$, and anchor embeddings V_m^{text}, V_m^{img}
    $\mathcal{L}_t^{(m)} = \frac{1}{|V_m^{text}|} \sum_{u \in V_m^{text}} (1 - \cos(v_m(I_p), u))$
    $\mathcal{L}_i^{(m)} = \frac{1}{|V_m^{img}|} \sum_{v \in V_m^{img}} (1 - \cos(v_m(I_p), v))$
  end for
  $\mathcal{L}_{re} = \frac{1}{|\mathcal{S}'|} \sum_{m \in \mathcal{S}'} (\alpha_w \mathcal{L}_t^{(m)} + \beta_w \mathcal{L}_i^{(m)})$
// Target-Driven Generation Alignment
  $\mathcal{L}_{ge} \leftarrow 1 - \cos(f_r(I_p), f_r(I_{ref}))$
// Objective of Joint Optimization
  $\mathcal{L}_{total} \leftarrow \lambda_1 \mathcal{L}_{re} + \lambda_2 \mathcal{L}_{ge}$
// Update the perturbation via PGD
  $\delta_{t+1} \leftarrow \text{Proj}_{||\cdot||_\infty \leq \epsilon}(\delta_t - \eta \cdot \nabla_\delta \mathcal{L}_{total}(I_d + \delta_t))$
  $t \leftarrow t + 1$
end while

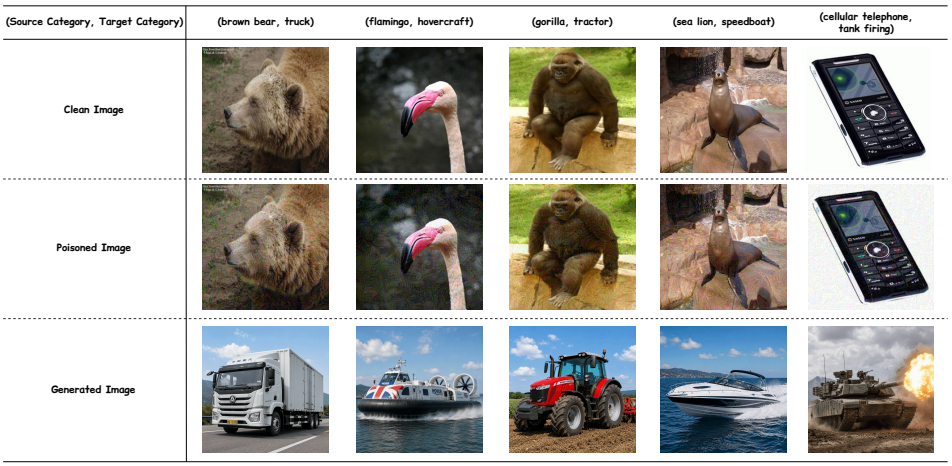

Figure 4: Visualization results of our PoisonedRDM.

B Implementation Details

B.1 Details of Datasets

Source Categories. We select 49 source categories from the ImageNet-1K validation set [12]. To extract decoy images and anchor images, we rank all validation images per category using CLIP [44] similarity with the prompt "a photo of a [category]". The top-4 highest-scoring images are retained: the top-1 serves as the decoy image, while the remaining three act as anchor images.

Target Categories. For the target categories of the benign subset, we select the categories from the ImageNet-1K validation set [12]. And we utilize Gemini 3.1 Pro[§] to generate the target categories of the red-teaming subset. We employ FLUX[¶] to synthesize target reference images, ensuring high semantic fidelity and visual clarity.

B.2 Details of Baselines

We select four state-of-the-art poisoning attack methods against the traditional diffusion models as our baselines, including Trojan Horses [43], Nightshade [49], AMP [54], Silent Branding [24]. Moreover, we select BadRDM [17] as the baseline, which executes a backdoor attack against RAG-DMs. We provide a detailed introduction to these baseline methods. ① *Trojan Horses* injects mislabeled pairs into the training corpus to induce generation misalignment by using BadNets-like data poisoning attack. ② *Nightshade* employs a stealthy poisoning strategy by injecting invisible perturbations into benign images, ultimately disabling the generation of semantically aligned images for specific categories. ③ *AMP* manipulates VLM-based captioning pipelines to automatically synthesize mismatched image–text pairs for diffusion model training. ④ *Silent Branding* is a trigger-free data poisoning attack that uses diffusion models' inherent memorization of recurrent visual patterns to stealthily embed target logos into clean training images. ⑤ *BadRDM* attacks against RAG-DMs by optimizing the retriever and poisoning the knowledge bases.

C Visualization of Results

Figure 4 presents images generated by GPT Image 2. As shown in the first two rows, the poisoned images are visually indistinguishable from the clean images, demonstrating that our adversarial perturbations are highly stealthy. However, when these poisoned images are retrieved, they successfully manipulate the generation process. As seen in the third row, the generated images completely ignore the original source categories (e.g., brown bear or flamingo) and perfectly align with the attacker's target categories (e.g., truck or hovercraft).

[§]https://deepmind.google/models/gemini/pro/
[¶]https://huggingface.co/black-forest-labs/FLUX.1-schnell

D Results on Red-teaming Dataset

We also evaluate PoisonedRDM and the baselines on the red-teaming subset to test their ability to induce unsafe generations, as shown in Table 6. Overall, PoisonedRDM maintains the highest attack success rates across all settings. For the RDM (PLMS-based) model, traditional poisoning attacks still fail to retrieve any poisoned images, resulting in 0.00% ASR-r. BadRDM performs better but still struggles to dominate the generation. In contrast, PoisonedRDM achieves the best results. For example, on the ResNet50 retriever, PoisonedRDM achieves 96.00% ASR-r and 60.80% ASR-g. This is 20 percentage points higher than BadRDM (i.e., 40.80%). PoisonedRDM also obtains the highest TSR score of 1.5498 on the ViT-H/14 retriever, showing a strong target category injection. For the RDM (DDIM-based) model, PoisonedRDM again consistently outperforms all baselines. PoisonedRDM secures the highest ASR-r and ASR-g on all four retrievers. For instance, using the ViT-B/32 retriever, PoisonedRDM reaches a very high ASR-g of 68.00%. Meanwhile, the strongest baseline, BadRDM, only reaches 36.80% ASR-g in the same setting. These results prove that PoisonedRDM is highly effective in red-teaming scenarios. Even for malicious categories, our optimization strategy ensures that the poisoned images are successfully retrieved and accurately guide the final generation toward the target.

E Attack Performance on non-CLIP Retrievers

We have further evaluated two **unseen non-CLIP retrievers**: GME-Qwen2-VL and Llama-Nemotron-Embed-VL, with two RDM variants and two online services. The results are shown in the tables below (from Table 7 to Table 11). First, compared to surrogate retrievers used during optimization, the unseen non-CLIP retrievers cause slight performance drops, but PoisonedRDM still achieves satisfactory 80.00–84.17% ASR-r and 43.33–52.50% ASR-g. Second, compared to BadRDM with unseen retrievers, our PoisonedRDM yields higher ASR-r (up to 2.50%) and ASR-g (up to 12.50%), with consistently higher CLIP-Attack and TSR. Third, we evaluate these unseen retrievers in online services. PoisonedRDM achieves 81.67%/39.17% ASR-r/ASR-g on GPT Image 2 and 79.17%/37.50% on Nano Banana 2, demonstrating strong transfer to unseen non-CLIP retrievers.

F Visual Interpretation

Figure 5 visualizes the attention maps (AM) based on image-category similarities. For the clean image and the target reference image, the AM correctly highlights their respective main objects. When computing the similarity between the poisoned image and the source category (i.e., Arabian camel), the AM specifically highlights the hump. Meanwhile, the AM for the poisoned image and the target category (i.e., dune buggy) shows highly concentrated attention in the central area. This spatial distribution perfectly matches the target reference image. Therefore, the poisoned image successfully tricks the model into treating it as the target category.

G The Analysis of Different Hyperparameters

Value of λ_1. We use RDM (PLMS-based) as the victim model and evaluate five numbers: $\{1, 1.2, 1.5, 1.8, 2\}$. As shown in Figure 6, 1.8 is optimal for all metrics. λ_1 controls the weight of the retrieval alignment loss. An excessively high or low λ_1 results in reduced performance, because it disrupts the crucial balance between the retrieval and generation objectives, leading to a drop in the generation attack success rate.

Value of λ_2. We use RDM (PLMS-based) as the victim model and evaluate five numbers: $\{1, 1.2, 1.5, 1.8, 2\}$. As shown in Figure 6, 2 is optimal for all metrics. λ_2 determines the weight of the target-driven generation alignment. Setting this value too low fails to provide enough guidance for the generation phase, whereas a higher value effectively forces the diffusion model to synthesize the malicious target category, maximizing the attack success rates.

Value of α_w. We use RDM (PLMS-based) as the victim model and evaluate seven numbers: $\{0, 0.2, 0.4, 0.5, 0.6, 0.7, 0.8\}$. As shown in Figure 6, 0.6 is optimal for all metrics. α_w balances the textual and visual anchors during the retrieval optimization. When α_w is set too low (e.g., 0), the

Table 6: The attack performance of PoisonedRDM against RAG-DMs on the red-teaming subset.

Model	Retriever	Method	Conference	ASR-r ↑	ASR-g ↑	CLIP-Attack ↑	TSR ↑
RDM (PLMS-based)	ViT-B/32	Trojan Horses [43]	NeurIPS 2024	0.00	21.60	0.2278	0.5335
		Nightshade [49]	SP 2024	0.00	12.00	0.1960	0.2939
		AMP [54]	CCS 2025	0.00	13.60	0.1966	0.2988
		Silent Branding [24]	CVPR 2025	0.00	28.80	0.2313	0.6420
		BadRDM [17]	ACL 2026	80.80	34.40	0.2498	0.8069
		PoisonedRDM *(ours)*	-	**92.80**	**56.80**	**0.2701**	**1.3306**
	ViT-B/16	Trojan Horses [43]	NeurIPS 2024	0.00	19.20	0.2162	0.4924
		Nightshade [49]	SP 2024	0.00	10.40	0.2035	0.2730
		AMP [54]	CCS 2025	0.00	15.20	0.2074	0.3882
		Silent Branding [24]	CVPR 2025	0.00	30.40	0.2346	0.6553
		BadRDM [17]	ACL 2026	77.60	35.20	0.2370	0.8175
		PoisonedRDM *(ours)*	-	**93.60**	**58.40**	**0.2886**	**1.3140**
	ViT-H/14	Trojan Horses [43]	NeurIPS 2024	0.00	20.80	0.2219	0.5141
		Nightshade [49]	SP 2024	0.00	12.80	0.2005	0.3012
		AMP [54]	CCS 2025	0.00	13.60	0.2036	0.3027
		Silent Branding [24]	CVPR 2025	0.00	26.40	0.2311	0.5843
		BadRDM [17]	ACL 2026	80.00	34.40	0.2474	0.8024
		PoisonedRDM *(ours)*	-	**91.20**	**56.00**	**0.2722**	**1.5498**
	ResNet50	Trojan Horses [43]	NeurIPS 2024	0.00	23.20	0.2299	0.5060
		Nightshade [49]	SP 2024	0.00	15.20	0.2055	0.3926
		AMP [54]	CCS 2025	0.00	18.40	0.2142	0.3944
		Silent Branding [24]	CVPR 2025	0.00	29.60	0.2334	0.6459
		BadRDM [17]	ACL 2026	87.20	40.80	0.2792	0.8627
		PoisonedRDM *(ours)*	-	**96.00**	**60.80**	**0.2865**	**1.2305**
RDM (DDIM-based)	ViT-B/32	Trojan Horses [43]	NeurIPS 2024	0.00	21.60	0.2263	0.5368
		Nightshade [49]	SP 2024	0.00	12.80	0.1994	0.3171
		AMP [54]	CCS 2025	0.00	15.20	0.2061	0.4016
		Silent Branding [24]	CVPR 2025	0.00	30.40	0.2376	0.6498
		BadRDM [17]	ACL 2026	83.20	36.80	0.2603	0.8446
		PoisonedRDM *(ours)*	-	**92.00**	**68.00**	**0.2895**	**1.2710**
	ViT-B/16	Trojan Horses [43]	NeurIPS 2024	0.00	20.00	0.2208	0.5058
		Nightshade [49]	SP 2024	0.00	20.80	0.2187	0.5185
		AMP [54]	CCS 2025	0.00	15.20	0.2062	0.4036
		Silent Branding [24]	CVPR 2025	0.00	36.00	0.2444	0.8475
		BadRDM [17]	ACL 2026	77.60	42.40	0.2755	0.8733
		PoisonedRDM *(ours)*	-	**94.40**	**55.20**	**0.2760**	**1.3007**
	ViT-H/14	Trojan Horses [43]	NeurIPS 2024	0.00	20.80	0.2221	0.5168
		Nightshade [49]	SP 2024	0.00	18.40	0.2135	0.3839
		AMP [54]	CCS 2025	0.00	20.80	0.2225	0.5279
		Silent Branding [24]	CVPR 2025	0.00	25.60	0.2290	0.6396
		BadRDM [17]	ACL 2026	82.40	37.60	0.2631	0.8517
		PoisonedRDM *(ours)*	-	**96.00**	**47.20**	**0.2790**	**1.7866**
	ResNet50	Trojan Horses [43]	NeurIPS 2024	0.00	24.00	0.2279	0.6653
		Nightshade [49]	SP 2024	0.00	20.00	0.2203	0.5172
		AMP [54]	CCS 2025	0.00	19.60	0.2178	0.5067
		Silent Branding [24]	CVPR 2025	0.00	27.20	0.2306	0.6646
		BadRDM [17]	ACL 2026	89.60	46.40	0.2774	0.8911
		PoisonedRDM *(ours)*	-	**97.60**	**51.20**	**0.2811**	**1.2342**

Table 7: Results of non-CLIP retriever (GME-Qwen2-VL) with RDM (PLMS-based).

Method	Conference	ASR-r	ASR-g	CLIP-Attack	TSR
Trojan Horses [43]	NeurIPS 2024	0.00	26.67	0.2261	0.5372
Nightshade [49]	SP 2024	0.00	12.50	0.2041	0.2783
AMP [54]	CCS 2025	0.00	15.83	0.2108	0.3921
Silent Branding [24]	CVPR 2025	0.00	30.83	0.2283	0.6415
BadRDM [17]	ACL 2026	82.50	31.67	0.2471	0.8302
PoisonedRDM *(ours)*	-	**83.33**	**44.17**	**0.2740**	**1.1639**

Table 8: Results of non-CLIP retriever (GME-Qwen2-VL) with RDM (DDIM-based).

Method	Conference	ASR-r	ASR-g	CLIP-Attack	TSR
Trojan Horses [43]	NeurIPS 2024	0.00	25.00	0.2238	0.5361
Nightshade [49]	SP 2024	0.00	10.83	0.2015	0.2795
AMP [54]	CCS 2025	0.00	17.50	0.2082	0.3893
Silent Branding [24]	CVPR 2025	0.00	30.00	0.2212	0.6375
BadRDM [17]	ACL 2026	81.67	34.17	0.2497	0.8305
PoisonedRDM *(ours)*	-	**84.17**	**45.00**	**0.2793**	**1.1742**

Table 9: Results of non-CLIP retriever (Llama-Nemotron-Embed-VL) with RDM (PLMS-based).

Method	Conference	ASR-r	ASR-g	CLIP-Attack	TSR
Trojan Horses [43]	NeurIPS 2024	0.00	24.17	0.2262	0.5358
Nightshade [49]	SP 2024	0.00	13.33	0.2045	0.2789
AMP [54]	CCS 2025	0.00	18.33	0.2078	0.3916
Silent Branding [24]	CVPR 2025	0.00	32.50	0.2283	0.6412
BadRDM [17]	ACL 2026	79.17	37.50	0.2492	0.8261
PoisonedRDM *(ours)*	-	**80.00**	**43.33**	**0.2705**	**1.1642**

Table 10: Results of non-CLIP retriever (Llama-Nemotron-Embed-VL) with RDM (DDIM-based).

Method	Conference	ASR-r	ASR-g	CLIP-Attack	TSR
Trojan Horses [43]	NeurIPS 2024	0.00	25.00	0.2238	0.5352
Nightshade [49]	SP 2024	0.00	12.50	0.2041	0.2818
AMP [54]	CCS 2025	0.00	15.83	0.2112	0.3876
Silent Branding [24]	CVPR 2025	0.00	31.67	0.2281	0.6415
BadRDM [17]	ACL 2026	80.83	45.00	0.2472	0.8256
PoisonedRDM *(ours)*	-	**82.50**	**52.50**	**0.2711**	**1.1768**

Table 11: Results of non-CLIP retriever (GME-Qwen2-VL) with online services.

Model	ASR-r	ASR-g	CLIP-Attack	TSR
GPT Image 2	81.67	39.17	0.2693	1.1058
Nano Banana 2	79.17	37.50	0.2641	1.0042

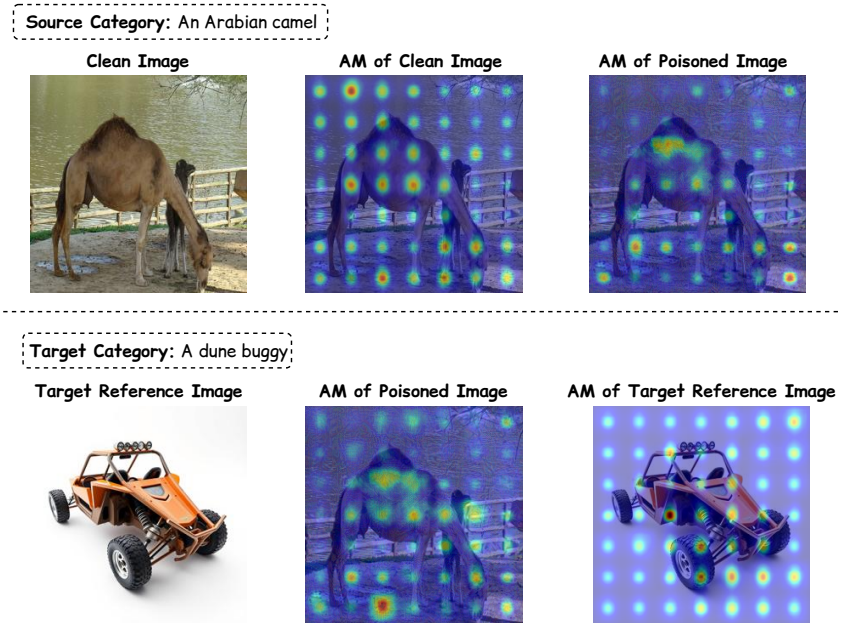

Figure 5: Visual interpretation.

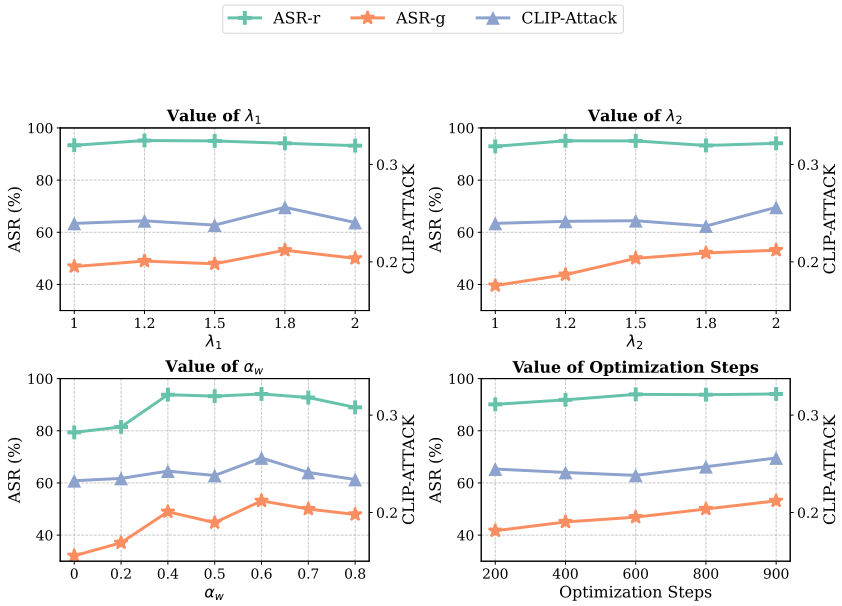

Figure 6: The analysis of different hyperparameters.

method relies entirely on visual anchors, resulting in a significant performance drop. An optimal balance at 0.6 ensures the poisoned image aligns robustly with the source category.

Optimization Steps. We use RDM (PLMS-based) as the victim model and evaluate five numbers: $\{200, 400, 600, 800, 900\}$. As shown in Figure 6, 900 is optimal for all metrics. Insufficient optimization steps fail to fully align the poisoned image's features within the condition space, whereas increasing the steps to 900 guarantees that the adversarial perturbation is sufficiently optimized to successfully manipulate the generation process. Furthermore, 900 optimization steps take an average of 141.67 seconds per image, proving that our method is highly efficient and does not introduce excessive time overhead.

H Potential Defenses and Limitations

We evaluate PoisonedRDM against defenses including smoothing filters [14], JPEG compression [21], and Comdefend [25] in Table 12. Across all defenses, the ASR-r remains remarkably high. This is because these defense methods process the retrieved images immediately before they enter the generation phase, thus leaving the retrieval process entirely unaffected. Furthermore, PoisonedRDM consistently outperforms BadRDM. For example, under Comdefend, PoisonedRDM achieves 92.23% ASR-r and 31.88% ASR-g, which are higher than the 89.35% and 27.93% of BadRDM. However, our method exhibits a limitation. The ASR-g drops under these defenses. Despite this limitation, PoisonedRDM still demonstrates stronger resilience than the baseline. Crucially, the ASR-g of BadRDM also drops significantly. Since BadRDM relies on clean images without adversarial perturbations, this drop demonstrates that such defenses inherently destroy normal image semantics, thereby degrading the overall generation quality.

I Ethical Considerations

Our research explores the security vulnerabilities of retrieval-augmented diffusion models. The positive impact is that exposing these flaws helps developers build more secure generative systems. However, we acknowledge potential negative impacts. Attackers could use PoisonedRDM to inject stealthy images into open-source platforms. This might cause public models to generate harmful or violent content from benign queries. To mitigate this, we evaluated defense methods like image compression to provide immediate protective strategies. Furthermore, we have implemented strict safeguards. We will not publicly release the optimized poisoned images related to the red-teaming subset, which includes unsafe target categories. We will only provide the visual examples shown in

Table 12: Attack performance against RAG-DMs after defense methods.

Defense Strategy	Method	ASR-r ↑	ASR-g ↑
Gaussian [14]	BadRDM	80.79	18.03
	PoisonedRDM *(Ours)*	**90.24**	**27.96**
Median [14]	BadRDM	84.95	20.34
	PoisonedRDM *(Ours)*	**92.36**	**35.78**
Average [14]	BadRDM	89.42	21.91
	PoisonedRDM *(Ours)*	**91.67**	**26.14**
JPEG [21]	BadRDM	88.24	27.02
	PoisonedRDM *(Ours)*	**91.82**	**36.06**
Comdefend [25]	BadRDM	89.35	27.93
	PoisonedRDM *(Ours)*	**92.23**	**31.88**

the paper. Upon publication, we will release our attack code and benign datasets under a restrictive license. This license prohibits commercial or malicious use, restricting the tools to academic and defensive research. We strongly condemn using our method to attack real-world applications.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have thoroughly detailed the background, motivation, scope, main experimental results, and contributions of our work in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide discussions on limitations and potential defenses of our method in the Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: There is no assumption or theoretical result in this work.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our paper provides a detailed account of implementation details and hyperparameter settings of our method for reproducing in Section 5.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The used datasets are publicly available. We will consider releasing the code upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide implementation details in the Section 5.1 and analyze the choice of hyperparameters in the Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We fix the random seed in experiments for fair comparison.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide implementation details in Section 5.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The conducted research conforms in every respect with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We provide ethical considerations, which discuss social impacts in the Appendix I.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: The paper has described safeguards for responsible release of data or models that have a high risk for misuse in the Appendix I.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the datasets and models used and give a detail introduction.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The code will be made public upon acceptance.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [No]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.