A SINGLE-SHOT BLACK-BOX STRUCTURAL HIJACKING ATTACK ON DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models (DMs) have revolutionized text-to-image (T2I) and image-to-image (I2I) syntheses. However, they are widely recognized to be vulnerable to various malicious attacks, such as prompt injection, prompt evasion, and unsafe content generation. While most existing attack schemes require certain control over or access to the target DMs, we propose a single-shot black-box structural hijacking attack that requires no access to and minimal knowledge of these DMs. The core idea is to inject an adversarial perturbation into a source image, which enforces the perturbed image to have close intermediate key and value feature maps to those of a target image, typically a Not-Safe-For-Work (NSFW) image, in the denoising process. Once the perturbed image is processed by a victim with a DM, the generated image becomes corrupted in the still life case or even NSFW in the living subject case. Our attack scheme successfully sabotages 14 state-of-the-art DMs by causing an average increase of over 59% across three mainstream perceptual distances, including an animation-generating DM by attacking only the first frame. Hence this threat is rather significant and systemic, which calls for immediate countermeasures.

NSFW Warning: This paper includes images that may be disturbing or explicit in nature.

1 INTRODUCTION

The advent of Diffusion Models (DM, Song et al. 2021b), such as Latent DMs (LDM, Rombach et al. 2022) and Stable Diffusion (SD, Esser et al. 2024), has marked a methodological shift in visual content creation and image editing. However, a significant technical challenge in controllable generation lies in the balance between complex structural changes (Lai et al., 2026c) (non-rigid editing) and content consistency (identity, texture, and background preservation). To address this challenge, the attention mechanism (Vaswani et al., 2017) is widely adopted to enforce consistency, such as Vision Transformer (ViT, Dosovitskiy et al. 2021), Low-Rank Adaptation (LoRA, Hu et al. 2022), Diffusion Transformer (DiT, Peebles & Xie 2023), and mutual self-attention (Cao et al., 2023). These methods recognize that the key structural, identity, and textural information of the source image is inherently encoded in the self-attention **Key** and **Value** feature maps. By controlling the flow of these Key and Value features, they establish a mechanism for structural guidance and feature propagation, which enables high-fidelity and consistent non-rigid editing.

As DMs are getting more popular, their security problems have become more common and more serious (Lai et al., 2026b). A widely recognized threat is the prompt evasion attack (Zhang et al., 2025a), which embeds sensitive knowledge in target prompts to construct effective adversarial prompts via gradient-based optimization (Lyu et al., 2025). Such attacks can bypass prompt filters (Yang et al., 2024b) and post-hoc safety checkers (Esser et al., 2024) of black-box text-to-image (T2I) DMs and yield **Not-Safe-For-Work (NSFW) content**. Moreover, if an image-to-image (I2I) DM becomes white-box, a more complex attack scheme AdvI2I (Zeng et al., 2025) can circumvent text-based safeguards like Safe Latent Diffusion (Schramowski et al., 2023), which forces the white-box I2I DM to generate unsafe content. Fortunately, these attack schemes require either a certain level of control over or access to the target system (prompt evasion), or complete knowledge of its internal structure (AdvI2I).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

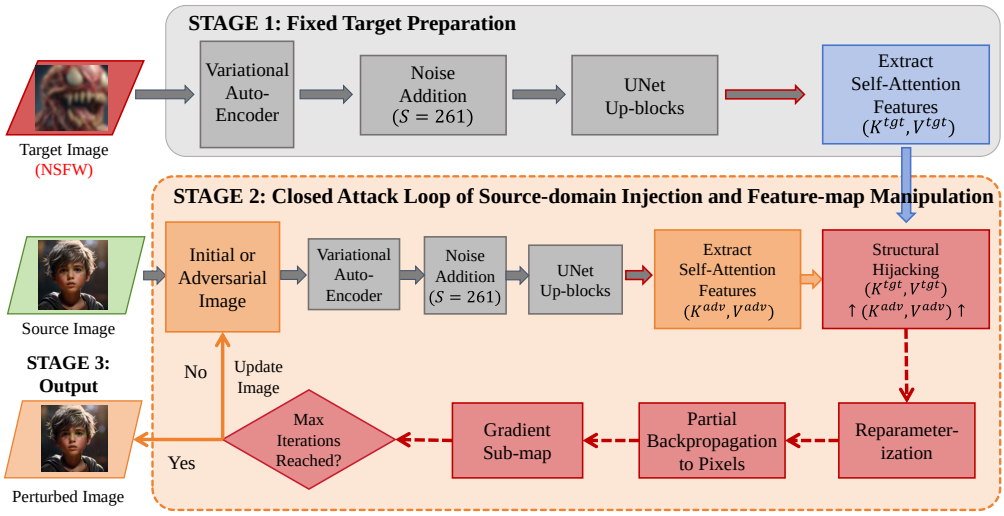


Figure 1: Attack scheme of the proposed SHADM. It aims to construct a perturbed image with a clean source image and an NSFW target image.

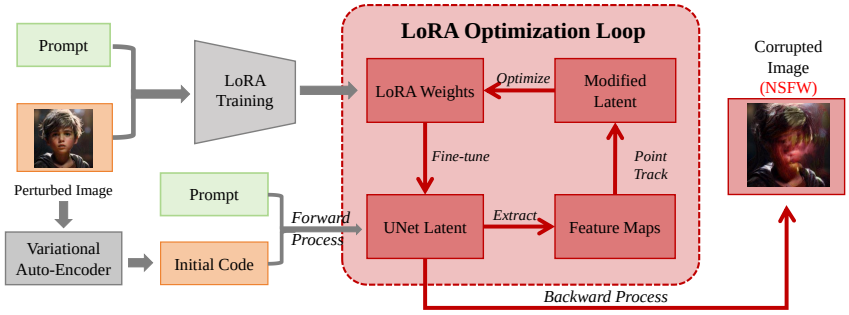


Figure 2: Attack chain of the proposed SHADM on a diffusion model, DragLoRA. SHADM successfully enforces DragLoRA to generate an NSFW image with a clean prompt and a perturbed image.

Nevertheless, we intend to reveal an even more serious threat regarding DMs in this work. While the attention mechanism has given rise to numerous representative methods, it can also be exploited to construct malicious attacks. We find that a benign source image can be manipulated to achieve identity transfer of attention features, regardless of the specific control mechanism that subsequently uses these features. This vulnerability makes our attack scheme **effective even in non-attention DMs. Our main contributions can be summarized as follows**.

1. We propose a Structural Hijacking Attack on DMs (**SHADM**) by exploiting the structural information encoded in the key and value feature maps of images, based on the feature consistency mechanisms.

2. We enforce the key and value feature maps of the perturbed image to match those of the target (corrupted or NSFW) image, which guarantees the transfer of critical structural information within the up-sampling attention blocks in the denoising process.

3. We develop a gradient-based optimization scheme to construct the adversarial perturbation, which integrates the forward process of DM and the backpropagation of learning architecture to close the loop of source-domain injection and feature-map manipulation. By extending the reparameterization approach (Kingma & Welling, 2014; Rezende et al., 2014) from the encoder parameter to the input, we develop a deterministic backpropagation for gradient computation.

In our attack model, an attacker just needs to use SHADM on a clean source image only once to produce a slightly perturbed image, then finishes the attack (Figure 1). The perturbed image is handed over to a victim, and **the attacker has no access to or knowledge of the target DM of the victim**. In this case, **mainstream black-box attack schemes are not applicable**. Once the victim uses a DM to process this perturbed image, the outcome will be corrupted or NSFW (Figure 2). Results show that

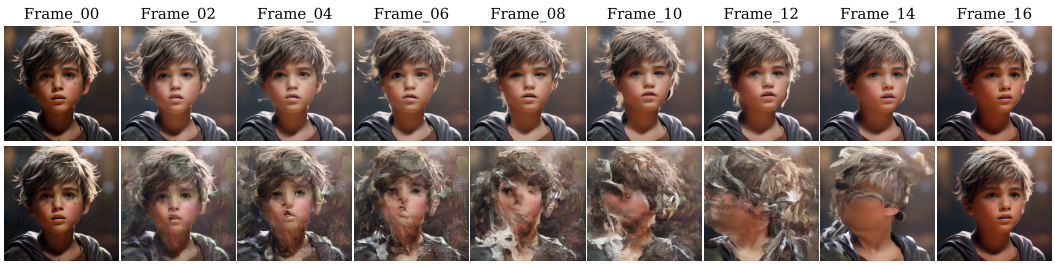


Figure 3: Attack process of the proposed SHADM on AnimateDiff. **Upper:** The normal animation generation with clean first and last frames. **Lower:** The corrupted animation generation with a clean last frame but a *perturbed first frame*.

SHADM successfully sabotages 14 **state-of-the-art (SOTA) DMs** of different categories, including an animation-generating DM (by attacking only the first frame), a more advanced DiT-based DM (with the adversarial perturbation constructed by only a UNet, (Ronneberger et al., 2015)), and even non-attention DMs, which indicates that this threat is rather significant/systemic and calls for immediate countermeasures.

2 Preliminaries and Related Work

2.1 Diffusion Models

To simplify expression, we review the pipeline of DMs (Song et al., 2021b) directly on the latent space (Rombach et al., 2022). Denote z_0 as the latent representation encoded from a clean image $x^{(0)}$ via the Variational Auto-Encoder (VAE, Kingma & Welling 2014; Rombach et al. 2022). A DM typically consists of a *forward process* and a *backward process*. The forward process progressively adds Gaussian noise to the latent representations $\{z_t\}_{t=0}^{T}$ on T timestamps:

$$z_t = \sqrt{\alpha_t} z_0 + \sqrt{1-\alpha_t}\varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \mathbb{I}), \tag{1}$$

where $\alpha_t \in (0, 1]$ fixes a noise schedule such that $\alpha_t \to 0$ as $t \to \infty$, and $\mathbb{I}$ denotes the identity matrix whose dimensionality can be inferred from z_t. The forward process aims to gradually add randomness to the latent representation, in order to enable generation of diverse instances in the backward process (denoising diffusion sampling). A representative non-Markovian backward process is the Denoising Diffusion Implicit Model (DDIM, Song et al. 2021a):

$$z_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{z_t - \sqrt{1-\alpha_t}\varepsilon_\theta(z_t)}{\sqrt{\alpha_t}} \right) + \sqrt{1-\alpha_{t-1}}\varepsilon_\theta(z_t), \tag{2}$$

where $\varepsilon_\theta(z_t)$ denotes the noise function parameterized by θ, which retrieves the Gaussian noise at timestamp t with z_t. Moreover, DDIM has a convenient inversion that can steer the initial latent representation z_0 towards the later timestamps given $\varepsilon_\theta(z_t)$:

$$z_{t+1} = \sqrt{\alpha_{t+1}} \left(\frac{z_t - \sqrt{1-\alpha_t}\varepsilon_\theta(z_t)}{\sqrt{\alpha_t}} \right) + \sqrt{1-\alpha_{t+1}}\varepsilon_\theta(z_t). \tag{3}$$

At some suitable intermediate timestamp S, z_S provides sufficient semantic and geometric information to support text-guided image editing.

2.2 Adversarial Attack

In general, an adversarial attack adds a small perturbation δ to a benign example x, in order to create an adversarial example $\tilde{x} = x + \delta$ that can achieve special purposes (usually unauthorized or malicious), such as circumventing object detection, evading correct classification, and generating NSFW content. To do this, an attacker sets a loss function $\mathcal{L}$ for the specific learning task and tries to maximize it:

$$\max_{\|\delta\|_p \leqslant \epsilon} \mathcal{L}(x + \delta), \tag{4}$$

where $\|\delta\|_p \leqslant \epsilon$ is a restriction on the perturbation δ within a preset *budget* $\epsilon > 0$ in the sense of ℓ_p norm. For example, in a white-box setting where the attacker knows the exact loss function $\mathcal{L}$ of the target machine learning (ML) model, he/she can directly use this $\mathcal{L}$ in (4) to find the optimal perturbation. Common adversarial attack methods include Fast Gradient Sign Method (FGSM, Goodfellow et al. 2015), Projected Gradient Descent (PGD, Madry et al. 2018; Croce & Hein 2020a), C&W (Carlini & Wagner, 2017), Auto PGD (APGD), and AutoAttack (Croce & Hein, 2020b). In

particular, PGD is a popular multi-iteration attack strategy:

$$\tilde{x}^{(k+1)} = \text{proj}_{\mathcal{B}_p[x;\epsilon]}(\tilde{x}^{(k)} + \eta \cdot \text{sign}(\nabla_x \mathcal{L}(x))), \tag{5}$$

where $\eta > 0$ is the step size for a single attack iteration, and $\text{proj}_{\mathcal{B}_p[x;\epsilon]}$ denotes the projection operator onto the closed neighborhood of x with a radius of ϵ.

As for the black-box setting, although the specific functional form f of the target ML model is unknown, the output $y = f(x)$ should be accessible for most black-box attack schemes (Papernot et al., 2017). For example, the attacker can use sufficient ground-truth input-output pairs $\{(x, y)\}$ to train a surrogate model $\hat{f}$ (e.g., a deep neural network), then use this $\hat{f}$ to construct the adversarial perturbation. **However, our attack model has no access to the target ML model and only attacks once, thus mainstream black-box attack schemes are not applicable.**

2.3 Prompt Evasion and NSFW Content

The existing adversarial attack schemes on DMs intensively target the prompt component. Subtle adversarial prompts can evade the detection of safety filters (i.e., jailbreak) and generate inappropriate or even NSFW images. For example, the filtered tokens in a given NSFW prompt can be replaced by alternative tokens to generate NSFW images by preserving the NSFW semantics (Yang et al., 2024c). Worse still, both textual filters (the prompt) and visual checkers (the generated image) can be simultaneously targeted to generate a unified adversarial prompt for dual evasion (Peng et al., 2024). A soft assignment with gradient masking technique can align malicious concepts semantically in the text embedding space, which evades both text and image safety checkers (Ma et al., 2025). **However, these attacks require the control of prompt input and even image input, which is much stronger than our attack model.** Besides, they are specific to the text-conditioning mechanism and can be tackled by improving the text encoder or using post-hoc safety checkers on the generated output.

3 Methodology

We begin by detailing the complete attack scheme when the attacker controls the source image (see Figure 1). Then we examine the corresponding attack chain when the perturbed image interacts with the clean target ML model of the victim (see Figure 2). The attacker requires only a standard generative model (we use the UNet as an example) to extract latent embeddings and features from images. Because the constructed perturbations successfully sabotage 14 SOTA DMs across various generative architectures (including both attention-based and non-attention-based ones), our methodology works as a black-box attack that requires no prior knowledge of the target system.

3.1 Key and Value Feature Extraction

Let x^{tgt} be a target image with the desired hijacked structure, such as a corrupted or NSFW image. We first map x^{tgt} to the latent space with the VAE $\mathcal{E}$: $z_0^{tgt} = \mathcal{E}(x^{tgt})$. Based on the LDM (Rombach et al., 2022) in Section 2.1, we obtain the noisy latent z_S^{tgt} by the forward process for a suitable intermediate timestamp S:

$$z_S^{tgt} = \sqrt{\alpha_S} z_0^{tgt} + \sqrt{1 - \alpha_S}\varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \mathbb{I}). \tag{6}$$

As z_S^{tgt} propagates through the UNet, it passes through several Transformer blocks. Within the self-attention layer l of each block, the input feature map $F_{(l)}^{tgt} \in \mathbb{R}^{\mathcal{H}_{(l)}\mathcal{W}_{(l)} \times \mathcal{C}_{(l)}}$ (where $\mathcal{H}_{(l)}$, $\mathcal{W}_{(l)}$, and $\mathcal{C}_{(l)}$ denote the two spatial dimensions and the channel dimension of layer l, respectively) goes through a linear transform:

$$K_{(l)}^{tgt} = F_{(l)}^{tgt} W_{(l)}^K + b_{(l)}^K, \quad V_{(l)}^{tgt} = F_{(l)}^{tgt} W_{(l)}^V + b_{(l)}^V. \tag{7}$$

$W_{(l)}^K \in \mathbb{R}^{\mathcal{C}_{(l)} \times d}, b_{(l)}^K$ and $W_{(l)}^V, b_{(l)}^V$ denote the transform matrices and biases for the key and value features, respectively. In this way, the key and value features can be extracted from z_S^{tgt}:

$$\mathcal{S}^{tgt} := \{(K_{(l)}^{tgt}, V_{(l)}^{tgt}) : l \in \mathscr{L}^{up}\} := \Phi(z_S^{tgt}, S, c_\emptyset), \tag{8}$$

where $\mathscr{L}^{up}$ denotes the set of self-attention layers in the UNet up-blocks, which encodes high-level semantic and structural information. Φ denotes the UNet, and $c_\emptyset$ is the null-text embedding, which ensures that the extracted features only represent the visual structure of the image rather than being biased by specific textual concepts.

3.2 Structural Hijacking Objective

The attack objective is to construct an adversarial image x^{adv} that minimizes the distance between its structural features $\mathcal{S}^{adv}$ and those of the target features $\mathcal{S}^{tgt}$, while remaining visually indistinguishable from the clean source image x^{src}. To do this, we develop a gradient-based optimization scheme to iteratively shape $x^{adv,(k)}$, starting with $x^{adv,(0)} := x^{src}$. At each attack iteration, we use the procedure in Section 3.1 to extract the key and value features $\{(K_{(l)}^{adv,(k)}, V_{(l)}^{adv,(k)}) : l \in \mathscr{L}^{up}\}$ for $x^{adv,(k)}$. Then we define the following Structural Hijacking Loss (SHL) as the Mean Squared Error (MSE) of the key and value features across all layers in $\mathscr{L}^{up}$:

$$\mathcal{L}_{SHL}^{(k)} = \frac{1}{|\mathscr{L}^{up}|} \sum_{l \in \mathscr{L}^{up}} (\|K_{(l)}^{adv,(k)} - K_{(l)}^{tgt}\|_2^2 + \|V_{(l)}^{adv,(k)} - V_{(l)}^{tgt}\|_2^2), \tag{9}$$

where $|\mathscr{L}^{up}|$ denotes the number of layers in $\mathscr{L}^{up}$. It enforces the key and value features of the adversarial sample to approach those of the target sample, in order to acquire corrupted or malicious embeddings.

3.3 Gradient-based Optimization with Closed Loop

Next, we construct the adversarial image by iteratively minimizing $\mathcal{L}_{SHL}^{(k)}$. To do this, we backtrack from $\mathcal{L}_{SHL}^{(k)}$ to the image domain through all the intermediate key and value features:

$$\begin{aligned}\mathcal{L}_{SHL}^{(k)}(x^{adv,(k)}) = &\frac{1}{|\mathscr{L}^{up}|} \sum_{l \in \mathscr{L}^{up}} (\|K_{(l)}^{(k)}(z_S(x^{adv,(k)})) - \mathrm{sg}(K_{(l)}(z_S(x^{tgt})))\|_2^2 \\ &+ \|V_{(l)}^{(k)}(z_S(x^{adv,(k)})) - \mathrm{sg}(V_{(l)}(z_S(x^{tgt})))\|_2^2),\end{aligned} \tag{10}$$

where $\mathrm{sg}(\cdot)$ denotes the *stop gradient* operator that *detaches* the model parameter update from the argument. It prevents $\mathrm{sg}(K_{(l)}(z_S(x^{tgt})))$ and $\mathrm{sg}(V_{(l)}(z_S(x^{tgt})))$ from backpropagating, because the target features should remain unchanged during adversary construction. To find a suitable direction of optimization, the partial gradient $\partial_x \mathcal{L}_{SHL}^{(k)}(x^{adv,(k)})$ should be computed. In practice, an automatic differentiating scheme of mainstream learning architectures like `autograd`[1] is employed to improve computational efficiency. Based on the chain rule,

$$\partial_x \mathcal{L}_{SHL}^{(k)}|_{x=x^{adv,(k)}} = \frac{1}{|\mathscr{L}^{up}|} \sum_{l \in \mathscr{L}^{up}} (\partial_x z_S \cdot \partial_{z_S} K_{(l)}^{(k)} \cdot \partial_{K_{(l)}^{(k)}} \mathcal{L}_{SHL}^{(k)} + \partial_x z_S \cdot \partial_{z_S} V_{(l)}^{(k)} \cdot \partial_{V_{(l)}^{(k)}} \mathcal{L}_{SHL}^{(k)})|_{x=x^{adv,(k)}}. \tag{11}$$

Since $\partial_x \mathrm{sg}(K_{(l)}(z_S(x^{tgt}))) \equiv 0 \equiv \partial_x \mathrm{sg}(V_{(l)}(z_S(x^{tgt})))$, only terms with $x^{adv,(k)}$ are backpropagated in (10). Hence this is a partial backpropagation for the gradient sub-map of $\mathcal{L}_{SHL}^{(k)}$ w.r.t. x.

Note that z_S is a random variable w.r.t. x, hence $\partial_x z_S$ cannot be directly computed. A common approach to estimate $\partial_x z_S$ is the Monte Carlo (MC) sampling (Li et al., 2024a; Koprulu et al., 2025; Gupta et al., 2025). It not only incurs high computational cost, but also suffers from slow convergence and high statistical variance. Therefore, we use a different approach by extending the reparameterization technique (Kingma & Welling, 2014; Rezende et al., 2014) from the encoder parameter to the input x. In brief, two *deterministic functions* $\mu(x)$ and $\sigma(x)$ that represent the posterior estimations for the mean and standard deviation (STD) of $z_S(x)$ are constructed by the pretrained VAE, respectively. Then $z_S(x)$ is reformulated as follows:

$$z_S(x) = \mu(x) + \sigma(x) \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \mathbb{I}), \tag{12}$$

which directs all the random sampling to ε. $\odot$ denotes the element-wise multiplication. Nevertheless, the sampled ε becomes a deterministic constant in backpropagation, which subsequently facilitates a deterministic backpropagation:

$$\partial_x z_S(x) = \partial_x \mu(x) + \partial_x \sigma(x) \odot \varepsilon, \tag{13}$$

Then the chain rule (11) becomes tractable to calculate $\partial_x \mathcal{L}_{SHL}^{(k)}(x^{adv,(k)})$. Details are provided in Appendix A.1. Because the loss $\mathcal{L}_{SHL}^{(k)}(x^{adv,(k)})$ in (10) is to be **minimized** in our attack model, the adversarial image at the $(k+1)$-th attack iteration is updated by one gradient descent step (Erhan

[1] https://pytorch.org/

et al., 2009; Goodfellow et al., 2015; Madry et al., 2018; Gurumurthy et al., 2021; Yu & Gao, 2024):

$$x^{adv,(k+1)}=\text{proj}_{\mathcal{B}_\infty[x^{src};\epsilon]\cap\mathcal{X}}(x^{adv,(k)}-\eta\cdot\text{sign}(\partial_x\mathcal{L}_{SHL}^{(k)}(x^{adv,(k)}))), \tag{14}$$

where $\text{proj}_{\mathcal{B}_\infty[x^{src};\epsilon]\cap\mathcal{X}}$ is a projection operator that enforces the output image within a perturbation budget of ϵ and the image domain $\mathcal{X}$. In practice, an image is usually normalized into the domain $\mathcal{X}=[-1,1]^{\mathfrak{m}}$ for adaptation to the input of VAE, where $\mathfrak{m}$ denotes the total number of pixels. The constraint $\|\delta\|_\infty\leqslant\epsilon$ enforces the maximum normalized pixel value of perturbation to be no more than ϵ. ϵ is generally set to values around $8/255\sim16/255$, which corresponds to $8\sim16$ in the original scale $[0,255]$. After a sufficient number K of attack iterations, the **final perturbed image** $\tilde{x}:=x^{adv,(K)}$ is obtained.

It is worth mentioning that our method (SHADM) is **significantly different** from FGSM, PGD, C&W, APGD, and other related methods. **1)** These general attack methods are **white-box** attacks that require the full knowledge of target models, including the exact target loss $\mathcal{L}$. In contrast, SHADM is a **black-box** attack that uses a uniform loss $\mathcal{L}_{SHL}$ against the target black-box model, regardless of whether the latter has an attention mechanism or not. **2)** The above general attack methods require fully deterministic gradient information and full backpropagation, and cannot backpropagate the stochastic latent representation z_S and do not consider the stop gradients $\text{sg}(K_{(l)}(z_S(x^{tgt})))$ and $\text{sg}(V_{(l)}(z_S(x^{tgt})))$. In contrast, SHADM backpropagates the stochastic latent representation and uses partial backpropagation with stop gradients, which integrates the forward process of DM and the backpropagation of learning architecture. This forms a **closed loop of source-domain injection and feature-map manipulation**. **3)** Mainstream black-box attack schemes like Square Attack (Andriushchenko et al., 2020) and prompt evasion (Ma et al., 2025) are **not applicable** to our task, because the attacker has no access to the target model.

3.4 Attack Chain on an Example Diffusion Model

The proposed SHADM can effectively attack both attention-based and non-attention-based DMs (see Section 4), which indicates that it is a universal attack method and that the attention mechanism itself causes a significant threat of misuse. To illustrate the attack chain (see Figure 2), we take a recent DM DragLoRA (Xia et al., 2025) as an example, because it exposes several vulnerabilities that can be easily targeted: attention mechanism, LoRA, and online optimization. It is a drag-based editing DM that can holistically move key features of an image from the source position to the target position. In the rest of this paper, **we use a tilde above a variable to denote its perturbed version**.

Double Attack on both Image and Parameters. LoRA is a widely used Parameter-Efficient Fine-Tuning (PEFT) method (Han et al., 2024; Sun et al., 2025b), which aims to fine-tune an increment $\Delta\theta$ for the pretrained parameters θ. Specifically, θ contains the general knowledge embedded in the UNet, while $\Delta\theta$ contains the specific knowledge for the source image. Accordingly, the noise function in (2) is extended to $\varepsilon_{\theta+\Delta\theta}(\cdot)$ for the LoRA-integrated UNet, and the following fine-tuning loss w.r.t. $\Delta\theta$:

$$\mathcal{L}_{ft}(z_S^{src},\Delta\theta)=\mathbb{E}_{\varepsilon,S}[\|\varepsilon-\varepsilon_{\theta+\Delta\theta}(\alpha_S z_0^{src}+\sigma_S\varepsilon)\|_2^2], \tag{15}$$

where $z_0^{src}=\mathcal{E}(x^{src})$ is the latent representation of the source image, and α_S and σ_S are the noise schedule parameters of DDIM at timestamp S.

However, now the victim has got a perturbed source image $\tilde{x}^{(0)}$ instead of a clean one, where the superscript counter is reset to 0 since it is the initial state from the victim's perspective. The victim would first train the initial fine-tuning increment $\Delta\tilde{\theta}^{(0)}$ with $\tilde{x}^{(0)}$, then this $\Delta\tilde{\theta}^{(0)}$ subsequently contaminates the initial state of the UNet parameters: $\tilde{\theta}^{(0)}=\theta^{(0)}+\Delta\tilde{\theta}^{(0)}$. On the other hand, the initial latent representation $\tilde{z}_0^{(0)}=\mathcal{E}(\tilde{x}^{(0)})$ is also perturbed through VAE. Substituting both $\tilde{\theta}^{(0)}$ and $\tilde{z}_0^{(0)}$ into (15), the victim obtains a perturbed intermediate latent representation $\tilde{z}_S^{(0)}:=\tilde{z}_S(\tilde{x}^{(0)})$.

Closed Attack Loop of Tracking and Optimizing. DragLoRA alternately tracks the next handle point and optimizes the LoRA weights to gradually move the key features. However, since both $\tilde{\theta}^{(0)}$ and $\tilde{z}_S^{(0)}$ are perturbed, this alternating mechanism has become a closed attack loop. Specifically, the Motion Supervision (MS) loss at the $\mathfrak{k}$-th movement is perturbed:

$$\mathcal{L}_{MS}(\tilde{z}_S^{(\mathfrak{k})})=\sum_{i=1}^{n}\sum_{q\in\Omega_1(h_i^{(\mathfrak{k})})}\{\|F_{q+c_i}(\tilde{z}_S^{(\mathfrak{k})})-\text{sg}(F_q(\tilde{z}_S^{(\mathfrak{k})}))\|_1+\lambda_{BR}\|(\tilde{z}_{S-1}^{(\mathfrak{k})}-\text{sg}(\tilde{z}_{S-1}^{(0)}))\odot(\mathbb{1}-\mathbb{M})\|_1\}, \tag{16}$$

where $\Omega_1(h_i^{(\mathfrak{k})})$ denotes a square neighborhood centered at the i-th handle point $h_i^{(\mathfrak{k})}$, c_i is a direction vector for the current movement, and F_q denotes the general feature map of all layers at pixel q. The second term is a Background Restoration (BR) term with strength hyperparameter $\lambda_{BR} \geqslant 0$ and mask for the edit region $\mathbb{M}$. The $(\mathfrak{k}+1)$-th edit update is then attacked:

$$\tilde{z}_S^{(\mathfrak{k}+1)} = \tilde{z}_S^{(\mathfrak{k})} - \eta_{MS} \cdot \partial_z \mathcal{L}_{MS}(\tilde{z}_S^{(\mathfrak{k})}). \tag{17}$$

Then $\tilde{z}_S^{(\mathfrak{k}+1)}$ continues to attack the point tracking step:

$$h_i^{(\mathfrak{k}+1)} = \underset{q \in \Omega_2(h_i^{(\mathfrak{k})})}{\arg\min} \|F_q(\tilde{z}_S^{(\mathfrak{k}+1)}) - F_{h_i^{(0)}}(\tilde{z}_S^{(0)})\|_1, \tag{18}$$

where $\Omega_2(h_i^{(\mathfrak{k})})$ is another square neighborhood. In this way, the tracking trajectory $\{h_i^{(\mathfrak{k})}\}_{i,\mathfrak{k}}$ may be misled to a wrong direction, which corrupts the image generation.

After point tracking, DragLoRA turns to optimize LoRA weights with the delta denoising score (DDS) loss (Hertz et al., 2023; Arar et al., 2024):

$$\frac{\partial \mathcal{L}_{DDS}^{(\mathfrak{k}+1)}(\tilde{z}_S^{(\mathfrak{k}+1)})}{\partial \Delta\theta}|_{\theta=\tilde{\theta}^{(\mathfrak{k}+1)}} = (\varepsilon_{\tilde{\theta}^{(\mathfrak{k}+1)}}^{ori}(\tilde{z}_S^{(\mathfrak{k}+1)}) - \varepsilon_{\tilde{\theta}^{(\mathfrak{k}+1)}}^{drag}(\tilde{z}_S^{(\mathfrak{k}+1)})) \frac{\partial \tilde{z}_S^{(\mathfrak{k}+1)}}{\partial \Delta\theta}|_{\theta=\tilde{\theta}^{(\mathfrak{k}+1)}}, \tag{19}$$

where $\varepsilon_{\tilde{\theta}^{(\mathfrak{k}+1)}}^{ori}$ and $\varepsilon_{\tilde{\theta}^{(\mathfrak{k}+1)}}^{drag}$ denote the noise functions of DDIM before and after the drag movement at the $(\mathfrak{k}+1)$-th optimizing iteration. All the terms in (19) involving $\tilde{\theta}^{(\mathfrak{k}+1)}$ and $\tilde{z}_S^{(\mathfrak{k}+1)}$ have been perturbed ever since $\mathfrak{k}=0$. As $\mathfrak{k}$ increases, the perturbations in both $\tilde{\theta}^{(\mathfrak{k}+1)}$ and $\tilde{z}_S^{(\mathfrak{k}+1)}$ have been accumulated or even amplified, which forms a closed, persistent attack loop and finally corrupts the output image.

4 Experiments

To evaluate the proposed SHADM, we take 14 SOTA DMs as targets, including three main categories: 1) Training-required: DragLoRA (Xia et al., 2025), DragDiffusion (Shi et al., 2024), DragNoise (Liu et al., 2024), GoodDrag (Zhang et al., 2025b), FreeDrag (Ling et al., 2024), DragText (Choi et al., 2025), CLIPDrag (Jiang et al., 2025), InstantDrag (Shin et al., 2024); 2) Training-free: SDE-Drag (Nie et al., 2024) and FastDrag (Zhao et al., 2024); 3) Control-based: PnP Diffusion (Tumanyan et al., 2023), MasaCtrl (Cao et al., 2023), AnimateDiff (Guo et al., 2024), and PixArt (Chen et al., 2024). Technical details of these DMs and their vulnerabilities to SHADM are provided in Appendix A.2.3.

The DragBench (Shi et al., 2024) data set with 10 image categories is used for evaluation, because it contains all the necessary information for T2I, I2I, and mixed DMs. The Paired Region Dataset (PRD, Sui et al. 2026) consisting of 1000 images is also taken into experiments. Three common levels of ℓ_∞ perturbation budgets are used for SHADM: $\epsilon = 8/255$, $12/255$, and $16/255$ (Addepalli et al.; Ngnawé et al., 2024; Sun et al., 2025a). Four metrics are used to evaluate performance: LPIPS (Learned Perceptual Image Patch Similarity, Zhang et al. 2018), ASR (Attack Success Rate, Moosavi-Dezfooli et al. 2016), $(1 - \text{CLIP})$ (Contrastive Language-Image Pre-training, Radford et al. 2021), and FID (Fréchet Inception Distance, Heusel et al. 2017). For all these metrics, a **higher score** indicates a better attack performance. Experimental details including experimental settings, definitions of evaluating metrics, and details of targeted DMs are provided in Appendix A.2. **Extensive additional experiments including comparisons with five baselines, robustness against three defense schemes, and other aspects are provided in Appendix A.3**. Artifacts produced by SHADM (**especially for the NSFW ones**) are provided in the supplementary material.

4.1 Main Results on DragBench

We present the main results on DragBench in Table 1 by two category sets: Living Subjects ($N = 50$: Animals, Human-head, Human-upper-body, Human-full-body) and Still Life ($N = 155$: Artwork, Landscape, Building-city-view, Building-countryside-view, Interior-design, Other-objects). It indicates that SHADM achieves significant disruptions. For example, the ASR of SHADM(16/255) rises by $+985.4\%$ on Living Subjects (from 4.10% to 44.50%) and $+1042.5\%$ on Still Life (from 2.07% to 23.65%). Besides, the perceptually and semantically grounded metrics show significant drops in edit fidelity under SHADM attack. For Living Subjects, LPIPS increases by $+147.3\%$ (0.344 vs. 0.139) and $(1 - \text{CLIP})$ distance grows by $+361.4\%$ (0.189 vs. 0.041) at $\epsilon = 16/255$.

Furthermore, FID scores jump from 67.409 (Clean) to 254.417 ($\epsilon = 16/255$). Comparing the two category sets, Living Subjects consistently show higher susceptibility to SHADM than Still Life. Under $\epsilon = 16/255$, Living Subjects experience nearly double the ASR of Still Life (44.50% vs. 23.65%) and a noticeably higher relative jump in FID ($+277.4\%$ vs. $+164.6\%$).

Table 1: Attack performance for SHADM on DragBench by two category sets.

Category Set	Metric (↑)	Clean	SHADM ($\epsilon = 8/255$)	SHADM ($\epsilon = 16/255$)	Δ over Clean ($\epsilon = 16/255$)
Living Subjects	LPIPS	0.139	0.236	0.344	+0.205 (+147.3%)
	ASR (%)	4.10	13.87	44.50	+40.40 (+985.4%)
	$(1 - \text{CLIP})$	0.041	0.107	0.189	+0.148 (+361.4%)
	FID	67.409	158.071	254.417	+187.008 (+277.4%)
Still Life	LPIPS	0.148	0.213	0.300	+0.152 (+102.5%)
	ASR (%)	2.07	7.99	23.65	+21.58 (+1042.5%)
	$(1 - \text{CLIP})$	0.042	0.102	0.176	+0.134 (+322.5%)
	FID	70.630	124.029	186.859	+116.228 (+164.6%)

4.2 LPIPS

LPIPS$\in [0, 1]$ is a main metric for measuring the perceptual distance between two single images in a way that mimics human vision. An LPIPS around 0.1 is empirically acceptable in human perception w.r.t. different kinds of I2I/T2I translations, style transfers, and deblurring. LPIPS scores of SHADM on different DMs are provided in Table A2, which show that SHADM with budget $\epsilon = 8/255$ causes an average increase of 59.1% for LPIPS on different DMs. SDE-Drag keeps relatively low LPIPS scores among all the DMs, but the main reason is that SDE-Drag hardly changes the source image, instead of being robust to the SHADM attack. PixArt also keeps relatively low LPIPS scores, but SHADM(8/255) still causes an increase of 34.8% on it. As for different categories, SHADM(8/255) is most effective in Human-full-body (average score increase of 218.9%, the same below), Human-head (152.9%), and Animals (72%), while least effective in Building-city-view (37.3%), Building-countryside-view (39.4%), and Art-work (50.6%). These results well justify the potency of SHADM, because the NSFW target image used for attack depicts a head (see Figure 1), and NSFW is relevant to living subjects, not to still lives.

4.3 ASR

ASR$\in [0, 100\%]$ measures the proportion of NSFW images, judged by the Q16 classifier (Schramowski et al., 2022), among all the attacked images. The ASR of SHADM across different DMs are provided in Table A3, which shows that SHADM(16/255) achieves an average ASR of 28.7% among all DMs and categories, compared with an average ASR of only 2.56% for the clean version. In particular, SHADM(16/255) achieves the highest ASRs in the Human-head (66.6%), Human-full-body (54.6%), and Animals (40.1%) categories. These results again justify the potency of SHADM, since the NSFW target image is a head and NSFW is only relevant to living subjects. Moreover, as the perturbation budget ϵ increases from $8/255$ to $12/255$, and to $16/255$, the average ASR significantly increases from 29.7% to 52.7%, and to 66.6%, in Human-head. Since perturbed images with $\epsilon = 16/255$ are hardly recognizable without prior warning (see Figure 1), such a **high ASR** as 66.6% can pose a **significant threat** to the system integrity of DMs.

4.4 $(1 - \text{CLIP})$

CLIP$\in [-1, 1]$ is a multimodal semantic similarity metric based on a joint image-text embedding space and pretrained by a ViT with massive image-text pairs. To evaluate the attack performance of SHADM, we use $\mathfrak{d}_{CLIP} = (1 - \text{CLIP})$ as a perceptual distance. A $\mathfrak{d}_{CLIPcompare}$ no higher than 0.03 is empirically acceptable for semantic similarity tasks. Results in Table A4 show that SHADM(8/255) causes an average increase of 189% for $\mathfrak{d}_{CLIP}$ on different DMs. Moreover, SHADM(8/255) is most effective in Human-head (314.3%), Interior-design (286.2%), and Human-full-body (255.6%) while least effective in Art-work (113.6%) , Building-countryside-view (168.6%), and Building-city-view (175.4%). It verifies that SHADM achieves the best performance on living subjects, especially for Human-head.

4.5 FID

FID$\in [0, \infty)$ measures the distributional difference between the generated and the original **image sets**, which is different from LPIPS and $\mathfrak{d}_{CLIP}$ that are calculated on each image pair. An FID around 30 is empirically considered similar in distribution (Zilberstein et al., 2025). Note that some related works (Yang et al., 2024c; Ma et al., 2025) use FID to measure the semantic distance between the generated image set and a ground-truth NSFW data set, where a lower FID indicates a better performance. However, we measure the semantic distance between the generated and the original image sets, because there are **other kinds of corruptions** than NSFW ones, especially for the categories of still lives. Hence a **higher FID** indicates a a **better performance** in our task. Results in Table A5 show that SHADM(8/255) causes an average increase of 101% for FID on different DMs. In addition, SHADM(8/255) is most effective in Human-full-body (256%), Human-head (216%), and Human-upper-body (188%), while least effective in Building-countryside-view (61.6%), Building-city-view (71.8%), and Other-objects (88%). It indicates that SHADM is a strong attack method on human-related images. In particular, **Human-head** is among the most vulnerable categories to SHADM in all the evaluating metrics, which calls for immediate countermeasures to improve the robustness of corresponding attention features.

4.6 Results on a Larger Dataset PRD

Results for clean and SHADM(16/255)-attacked samples on the larger dataset PRD are provided in Table 2. SHADM(16/255) achieves high ASRs and significant visual quality disruption across all three target models. FastDrag is the most vulnerable target, with its ASR jumping from a baseline of 8.0% on clean images to 79.0% under attack (a $+71.0\%$ increase). DragLoRA and MasaCtrl also experience major disruptions, with ASR rising from 12.9% to 63.9% and from 6.7% to 59.4%, respectively. Besides, FID scores rise significantly across all targets, where MasaCtrl undergoes the sharpest absolute jump (from 13.182 to 181.251). LPIPS increases to a narrow band around $0.307 \sim 0.324$ across all targets. $(1-\text{CLIP})$ increases sharply to 0.277 (from 0.079) on FastDrag and 0.242 (from 0.054) on MasaCtrl, which indicates that SHADM disrupts both localized point-tracking operations and broader context alignment.

Table 2: Attack performance for SHADM on PRD.

Target	Method	LPIPS	$(1-\text{CLIP})$	FID	ASR (%)
DragLoRA	Clean	0.257	0.229	60.482	12.9
	SHADM(16/255)	0.324	0.261	192.894	63.9
FastDrag	Clean	0.108	0.079	24.250	8.0
	SHADM(16/255)	0.316	0.277	170.903	79.0
MasaCtrl	Clean	0.028	0.054	13.182	6.7
	SHADM(16/255)	0.307	0.242	181.251	59.4

5 Conclusion

We propose a novel single-shot black-box Structural Hijacking Attack on Diffusion Models (SHADM) that targets the key and value feature maps of the attention mechanism. Different from existing attack schemes that usually require white-box access or control over prompt inputs, SHADM successfully operates with minimal knowledge of the architecture and internal parameters of the victim model. By enforcing the key and value feature maps of the perturbed image to match those of a target (e.g., NSFW or corrupted) image, SHADM effectively hijacks the denoising process to generate malicious or corrupted images.

A central part of SHADM is the gradient-based optimization scheme, which employs reparameterization and stop gradients to enable a deterministic partial backpropagation through the latent representations of DMs. This forms a closed loop of source-domain injection and feature-map manipulation that keeps the attack persistent on different architectures. Extensive experiments show that SHADM successfully sabotages 14 state-of-the-art DMs (including attention-based and non-attention-based) by causing an average increase of over 59% across three mainstream perceptual distances. In particular, SHADM can even sabotage video generation DMs by attacking only the first frame. These results highlight a significant and systemic threat to the security and safety of DMs, which calls for immediate and robust defensive countermeasures.

REFERENCES

Sravanti Addepalli, Samyak Jain, Gaurang Sriramanan, Shivangi Khare, and Venkatesh Babu Radhakrishnan. Towards achieving adversarial robustness beyond perceptual limits. In *ICML 2021 Workshop on Adversarial Machine Learning*.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII*, pp. 484–501, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58591-4.

Moab Arar, Andrey Voynov, Amir Hertz, Omri Avrahami, Shlomi Fruchter, Yael Pritch, Daniel Cohen-Or, and Ariel Shamir. Palp: Prompt aligned personalization of text-to-image models. In *SIGGRAPH Asia 2024 Conference Papers*, SA '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400711312.

Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22560–22570, October 2023.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57. IEEE, 2017.

Junsong Chen, Jincheng YU, Chongjian GE, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

Weijian Chen, Shuibing He, Haoyang Qu, Ruidong Zhang, Siling Yang, Ping Chen, Yi Zheng, Baoxing Huai, and Gang Chen. IMPRESS: An Importance-Informed Multi-Tier prefix KV storage system for large language model inference. In *23rd USENIX Conference on File and Storage Technologies (FAST 25)*, pp. 187–201, Santa Clara, CA, February 2025. USENIX Association.

Gayoon Choi, Taejin Jeong, Sujung Hong, and Seong Jae Hwang. Dragtext: Rethinking text embedding in point-based image editing. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 441–450, 2025.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2196–2205. PMLR, 13–18 Jul 2020a.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 2206–2216, 2020b.

Pucheng Dang, Xing Hu, Dong Li, Rui Zhang, Qi Guo, and Kaidi Xu. DiffZOO: A purely query-based black-box attack for red-teaming text-to-image generative model via zeroth order optimization. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 17–31, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

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*, pp. 248–255, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *The Twelfth International Conference on Learning Representations*, 2024.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Sparsectrl: Adding sparse controls to text-to-video diffusion models. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 330–348, Cham, 2025. Springer Nature Switzerland.

Prateek Gupta, Andrea Ferrari, and Nabil Iqbal. A machine learning approach to duality in statistical physics. In *Proceedings of the 42nd International Conference on Machine Learning*. PMLR, 2025.

Swaminathan Gurumurthy, Shaojie Bai, Zachary Manchester, and J. Zico Kolter. Joint inference and input optimization in equilibrium networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 16818–16832. Curran Associates, Inc., 2021.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. Delta denoising score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2328–2337, October 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Ziqi Jiang, Zhen Wang, and Long Chen. CLIPDrag: Combining text-based and drag-based instructions for image editing. In *The Thirteenth International Conference on Learning Representations*, 2025.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Cevahir Koprulu, Franck Djeumou, and ufuk topcu. Neural stochastic differential equations for uncertainty-aware offline RL. In *The Thirteenth International Conference on Learning Representations*, 2025.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Zhao-Rong Lai and Weiwen Wang. Invariant risk minimization is a total variation model. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 25913–25935. PMLR, 21–27 Jul 2024.

Zhao-Rong Lai, Xiwen Yuan, Ziliang Chen, Liangda Fang, and Yongsen Zheng. A unified total variation framework for membrane potential perturbation dynamic. In *The Fourteenth International Conference on Learning Representations*, 2026a.

Zhao-Rong Lai, Xiwen Yuan, and Jian Weng. Adversarial attack and defense for denoising diffusion sampling. In *The 43rd International Conference on Machine Learning*, 2026b.

Zhao-Rong Lai, Zheng-Sen Zhou, Liangda Fang, Yongsen Zheng, and Ziliang Chen. ℓ_1 latent distance based continuous-time graph representation. In *The Fourteenth International Conference on Learning Representations*, 2026c.

Ruichen Li, Haotian Ye, Du Jiang, Xuelan Wen, Chuwei Wang, Zhe Li, Xiang Li, Di He, Ji Chen, Weiluo Ren, and Liwei Wang. A computational framework for neural network-based variational monte carlo with forward laplacian. *Nature Machine Intelligence*, 6:209–219, 2 2024a.

Xinghui Li, Jingyi Lu, Kai Han, and Victor Adrian Prisacariu. Sd4match: Learning to prompt stable diffusion model for semantic matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27558–27568, June 2024b.

Chumeng Liang and Xiaoyu Wu. Mist: Towards improved adversarial examples for diffusion models. *arXiv preprint arXiv:2305.12683*, 2023.

Yizun Lin, Zhao-Rong Lai, and Cheng Li. A globally optimal portfolio for m-sparse sharpe ratio maximization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 17133–17160, 2024a.

Yizun Lin, Yangyu Zhang, Zhao-Rong Lai, and Cheng Li. Autonomous sparse mean-CVaR portfolio optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 30440–30456. PMLR, 21–27 Jul 2024b.

Pengyang Ling, Lin Chen, Pan Zhang, Huaian Chen, Yi Jin, and Jinjin Zheng. Freedrag: Feature dragging for reliable point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6860–6870, June 2024.

Haofeng Liu, Chenshu Xu, Yifei Yang, Lihua Zeng, and Shengfeng He. Drag your noise: Interactive point-based editing via diffusion semantic propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6743–6752, June 2024.

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

Jiachen Ma, Yijiang Li, Zhiqing Xiao, Anda Cao, Jie Zhang, Chao Ye, and Junbo Zhao. Jailbreaking prompt attack: A controllable adversarial attack against diffusion models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 3141–3157, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

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 (ICLR)*, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2574–2582, 2016.

Jonas Ngnawé, Sabyasachi Sahoo, Yann Pequignot, Frédéric Precioso, and Christian Gagné. Detecting brittle decisions for free: Leveraging margin consistency in deep robust classifiers. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 23301–23324. Curran Associates, Inc., 2024.

Shen Nie, Hanzhong Allan Guo, Cheng Lu, Yuhao Zhou, Chenyu Zheng, and Chongxuan Li. The blessing of randomness: SDE beats ODE in general diffusion-based image editing. In *The Twelfth International Conference on Learning Representations*, 2024.

John Paisley, David M. Blei, and Michael I. Jordan. Variational bayesian inference with stochastic search. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pp. 1363–1370, 2012.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, pp. 506–519, New York, NY, USA, 2017. Association for Computing Machinery.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023.

Duo Peng, Qiuhong Ke, and Jun Liu. Upam: unified prompt attack in text-to-image generation models against both textual filters and visual checkers. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Patrick Schramowski, Christopher Tauchmann, and Kristian Kersting. Can machines help us answering question 16 in datasheets, and in turn reflecting on inappropriate content? In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, pp. 1350–1361, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393522.

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

Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y. Zhao. Glaze: Protecting artists from style mimicry by Text-to-Image models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 2187–2204, Anaheim, CA, August 2023. USENIX Association. ISBN 978-1-939133-37-3.

Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent Y. F. Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8839–8849, 2024.

Joonghyuk Shin, Daehyeon Choi, and Jaesik Park. Instantdrag: Improving interactivity in drag-based image editing. In *SIGGRAPH Asia 2024 Conference Papers*, SA '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400711312.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Nick Stracke, Stefan Andreas Baumann, Kolja Bauer, Frank Fundel, and Björn Ommer. Cleandift: Diffusion features without noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 117–127, June 2025.

Jiacheng Sui, Tianyu Hao, Bingjie Gao, Li Niu, and Guangtao Zhai. In-context region-based drag: Drag any region to any shape. In *The 19th European Conference on Computer Vision*, 2026.

Jialiang Sun, Wen Yao, Tingsong Jiang, Chao Li, and Xiaoqian Chen. A^3Da3d: A platform of searching for robust neural architectures and efficient adversarial attacks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3975–3991, 2025a.

Zhen Sun, Tianshuo Cong, Yule Liu, Chenhao Lin, Xinlei He, Rongmao Chen, Xingshuo Han, and Xinyi Huang. Peftguard: Detecting backdoor attacks against parameter-efficient fine-tuning. In *2025 IEEE Symposium on Security and Privacy (SP)*, pp. 1713–1731, 2025b.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 1363–1389. Curran Associates, Inc., 2023.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1921–1930, June 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Yuanchao Wang, Zhao-Rong Lai, and Tianqi Zhong. Out-of-distribution generalization for total variation based invariant risk minimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Siwei Xia, Li Sun, and Qingli Li. Draglora: Online optimization of lora adapters for drag-based image editing in diffusion model. In *International Conference on Machine Learning (ICML)*, 2025.

Yijun Yang, Ruiyuan Gao, Xiaosen Wang, Tsung-Yi Ho, Nan Xu, and Qiang Xu. Mma-diffusion: Multimodal attack on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7737–7746, June 2024a.

Yijun Yang, Ruiyuan Gao, Xiaosen Wang, Tsung-Yi Ho, Nan Xu, and Qiang xu. Mma-diffusion: Multimodal attack on diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7737–7746, 2024b.

Yuchen Yang, Bo Hui, Haolin Yuan, Neil Gong, and Yinzhi Cao. Sneakyprompt: Jailbreaking text-to-image generative models. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 897–912, 2024c.

Hongzhan Yu and Sicun Gao. Activation-descent regularization for input optimization of relu networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57441–57458. PMLR, 2024.

Yaopei Zeng, Yuanpu Cao, Bochuan Cao, Yurui Chang, Jinghui Chen, and Lu Lin. Advi2i: Adversarial image attack on image-to-image diffusion models. In *International Conference on Machine Learning (ICML)*, 2025.

Chenyu Zhang, Mingwang Hu, Wenhui Li, and Lanjun Wang. Adversarial attacks and defenses on text-to-image diffusion models: A survey. *Information Fusion*, 114:102701, 2025a. ISSN 1566-2535.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3836–3847, October 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric . In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, Los Alamitos, CA, USA, June 2018. IEEE Computer Society.

Zewei Zhang, Huan Liu, Jun Chen, and Xiangyu Xu. Gooddrag: Towards good practices for drag editing with diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Haiquan Zhao, Chenhan Yuan, et al. Qwen3guard technical report, 2025.

Xuanjia Zhao, Jian Guan, Congyi Fan, Dongli Xu, Youtian Lin, Haiwei Pan, and Pengming Feng. Fastdrag: Manipulate anything in one step. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 74439–74460. Curran Associates, Inc., 2024.

Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15659–15669, October 2023.

Nicolas Zilberstein, Morteza Mardani, and Santiago Segarra. Repulsive latent score distillation for solving inverse problems. In *The Thirteenth International Conference on Learning Representations*, 2025.

A Appendix

A.1 Extending Reparameterization from Encoder Parameter to Input

Since our task is to construct an adversarial image instead of training a neural network, we directly use the standard VAE with default parameter settings to extract latent representations, and extend the reparameterization technique (Kingma & Welling, 2014) from the encoder parameter to the input.

Denote x and z as the input (image) and the corresponding latent representation. Standard variational Bayesian analysis indicates:

$$\mathcal{L}_{LB}(\psi, \phi; x) = -\mathscr{D}_{KL}(q_\phi(z|x)\|p_\psi(z)) + \mathbb{E}_{q_\phi(z|x)}\left[\log p_\psi(x|z)\right], \tag{20}$$

where $\mathcal{L}_{LB}$ is the variational lower bound, which is also the loss function to be minimized in variational inference (Lai & Wang, 2024; Wang et al., 2025; Lai et al., 2026a). $\mathscr{D}_{KL}$ denotes the Kullback-Leibler divergence. $q_\phi(z|x)$ and $p_\psi(x|z)$ are the probabilistic encoder and decoder, parameterized by ϕ and ψ, respectively. The standard MC gradient estimation of a function f is:

$$\nabla_\phi \mathbb{E}_{q_\phi(z)}\left[f(z)\right] = \mathbb{E}_{q_\phi(z)}\left[f(z)\nabla_{q_\phi(z)} \log q_\phi(z)\right] \simeq \frac{1}{J}\sum_{j=1}^{J} f(z^{(j)})\nabla_{q_\phi(z^{(j)})} \log q_\phi(z^{(j)}). \tag{21}$$

A key step that influences the inference performance is an appropriate sampling scheme $z^{(j)} \sim q_\phi(z|x)$. While traditional MC sampling is found to have high variance (Paisley et al., 2012), the reparameterization technique reforms the random sampling $z^{(j)} \sim q_\phi(z|x)$ as a differentiable transformation $g_\phi(\varepsilon, x)$ of an auxiliary noise variable ε:

$$z^{(j)} = g_\phi(\varepsilon^{(j)}, x), \quad \varepsilon^{(j)} \sim p(\varepsilon), \tag{22}$$

where $p(\varepsilon)$ is an appropriate distribution for ε. For example, the default setting of VAE is $q_\phi(z|x) = \mathcal{N}(z; \mu, \sigma^2 \odot \mathbb{I})$. After reparameterizing ϕ by μ and σ,

$$z^{(j)} = g_\phi(\varepsilon^{(j)}, x) = \mu + \sigma \odot \varepsilon^{(j)}, \quad \varepsilon^{(j)} \sim \mathcal{N}(0, \mathbb{I}). \tag{23}$$

In this way, the stochasticity is isolated in $\varepsilon^{(j)}$, which enables a deterministic estimation of partial derivative $\partial_\mu z^{(j)}$ or $\partial_\sigma z^{(j)}$.

In our task, we do not need to train the parameters of encoder or decoder, but need to modify the input instead. Therefore, we use the default ψ_0 and the default function forms of μ_0 and σ_0, and explicitly show the input x. First, the initial latent representation is sampled via

$$q(z_0|x) = \mathcal{N}(z_0; \mu_0(x), \sigma_0^2(x) \odot \mathbb{I}). \tag{24}$$

A similar forward process to (6) yields

$$q(z_S|z_0) = \mathcal{N}(z_S; \sqrt{\alpha_S} z_0, (1 - \alpha_S) \cdot \mathbb{I}). \tag{25}$$

Since the composition of Gaussian distributions is also Gaussian, the encoder function of z_S conditioned on x is

$$q(z_S|x) = \mathcal{N}(z_S; \mu(x), \sigma^2(x) \odot \mathbb{I}), \tag{26}$$

where the aggregate mean and STD functions $\mu(x)$ and $\sigma(x)$ are:

$$\mu(x) = \sqrt{\alpha_S} \cdot \mu_0(x), \tag{27}$$

$$\sigma(x) = \sqrt{\alpha_S \sigma^2(x) + (1 - \alpha_S)}. \tag{28}$$

Then the reparameterization for z_S is

$$z_S(x) = \mu(x) + \sigma(x) \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \mathbb{I}). \tag{29}$$

In this way, the stochasticity is isolated in ε, while the dependence on x is fully contained in the deterministic terms $\mu(x)$ and $\sigma(x)$. Since ε is drawn from a standard normal distribution **independent of** x, it can be treated as a constant during differentiation w.r.t. x (Lin et al., 2024b;a). Specifically,

$$\partial_x z_S(x) = \partial_x \left(\mu(x) + \sigma(x) \odot \varepsilon\right) = \partial_x \mu(x) + \partial_x \sigma(x) \odot \varepsilon. \tag{30}$$

In the context of the SHADM single-shot attack, we only need to perform this estimation with a single sample of ε, which yields the deterministic backpropagation path:

$$\begin{aligned}
&\partial_x \mathcal{L}_{SHL}^{(k)}|_{x=x^{adv,(k)}} \\
=&\frac{1}{|\mathscr{L}^{up}|}\sum_{l\in\mathscr{L}^{up}}(\partial_x z_S\cdot\partial_{z_S}K_{(l)}^{(k)}\cdot\partial_{K_{(l)}^{(k)}}\mathcal{L}_{SHL}^{(k)}+\partial_x z_S\cdot\partial_{z_S}V_{(l)}^{(k)}\cdot\partial_{V_{(l)}^{(k)}}\mathcal{L}_{SHL}^{(k)})|_{x=x^{adv,(k)}} \\
=&\frac{1}{|\mathscr{L}^{up}|}\sum_{l\in\mathscr{L}^{up}}((\partial_x\mu(x^{adv,(k)})+\partial_x\sigma(x^{adv,(k)})\odot\varepsilon)\cdot\partial_{z_S}K_{(l)}^{(k)}\cdot\partial_{K_{(l)}^{(k)}}\mathcal{L}_{SHL}^{(k)} \\
&+(\partial_x\mu(x^{adv,(k)})+\partial_x\sigma(x^{adv,(k)})\odot\varepsilon)\cdot\partial_{z_S}V_{(l)}^{(k)}\cdot\partial_{V_{(l)}^{(k)}}\mathcal{L}_{SHL}^{(k)}),
\end{aligned} \tag{31}$$

where $\partial_x z_S$ is organized as a $d_x \times d_z$ matrix with d_x and d_z being the dimensionalities of x and z_S, respectively. The same organization is applied to other partial derivatives, then (31) can be directly computed via matrix calculation. By freezing the randomness ε during the backpropagation, the gradient flows through the mean and STD functions of the VAE, which allows the adversarial optimization on x to achieve the structural hijacking objective.

A.2 Experimental Details

A.2.1 Experimental Settings

The hyperparameters for SHADM are set by following the literature or to plain values: the timestamp of latent representation for structural hijacking $S = 261$, the attack iterations $K = 200$, and the attack step size $\eta = 1/255$. $S = 261$ follows the settings of (Tang et al., 2023; Li et al., 2024b; Stracke et al., 2025), where (Tang et al., 2023) particularly conducts extensive experiments to verify that $S = 261$ is the optimal choice. These hyperparameters remain constant across all the experiments, which cover all the target DMs and image classes. Such a plug-and-play effectiveness confirms the attack stability of SHADM and highlights the severity of the threat: the structural vulnerability in diffusion attention mechanisms is fundamental and easily exploitable with standard defaults. On the other hand, all the target DMs use the default hyperparameter settings from their original publications. A device with an Intel(R) Xeon(R) Gold 6348 CPU (14 out of 56 vCPUs), an Nvidia A800-80GB GPU, and 100GB RAM is used to conduct experiments.

DragBench has regulated the prompt, editing, and other benchmark settings. The sample size per category for DragBench is given in Table A1. Throughout all the experiments, we use **the same NSFW target image** for attack, as shown in Figure 1. Results show that SHADM is effective in attacking various kinds of source images and DMs, which verifies its stable performance in hijacking the structural features of different DMs.

Table A1: Sample sizes of different categories for DragBench.

Category	**Sample Size**
Art-work	35
Landscape	22
Building-city-view	21
Building-countryside-view	22
Animals	20
Human-head	10
Human-upper-body	10
Human-full-body	10
Interior-design	20
Other-objects	35
Total	**205**

A.2.2 Definitions of Evaluating Metrics

In the field of Artificial Intelligence Generated Content (AIGC), it is the perceptual difference between the generated and the original images that matters, instead of the traditional pixel-wise difference.

For example, an image shifted by one pixel can be considered "totally different" by the traditional ℓ_2 distance, but remains almost the same by human perception. To fill this gap, there are three widely-used perceptual metrics to evaluate image quality, fidelity, and relevance.

LPIPS. Denote x and $\hat{x}$ as the original and the generated images, respectively. LPIPS first extracts deep feature maps $\varphi(x)$ and $\varphi(\hat{x})$ from the two images via the AlexNet (Krizhevsky et al., 2012). Denote $\varphi^l_{a,b}(x)$ as the features at the spatial position $(a,b) \subseteq [\mathcal{A}_l] \times [\mathcal{B}_l]$ of the l-th layer of the AlexNet. Then the perceptual distance between x and $\hat{x}$ is defined by the squared ℓ_2 distance in the deep feature space:

$$\mathfrak{d}_{LPIPS}(x,\hat{x}) := \sum_l \frac{1}{\mathcal{A}_l \mathcal{B}_l} \sum_{a,b} \|w_l \odot (\varphi^l_{a,b}(x) - \varphi^l_{a,b}(\hat{x}))\|_2^2, \tag{32}$$

where $w_l \in \mathbb{R}^{\mathcal{C}_l}$ is a scaling vector for different channels of the l-th layer, pretrained by (Zhang et al., 2018). The operator $\frac{1}{\mathcal{A}_l \mathcal{B}_l} \sum_{a,b}$ averages the dominating terms across the spatial dimensions of this layer. In general, LPIPS$\in [0,1]$ and a lower LPIPS indicates a higher perceptual similarity between x and $\hat{x}$.

$(1-$ CLIP$)$. CLIP is designed to capture high-level semantic concepts in images. Similar to LPIPS, CLIP encodes the original and the generated images x and $\hat{x}$ into $\varphi(x)$ and $\varphi(\hat{x})$ in the feature space of ViT-B/32 (Dosovitskiy et al., 2021), respectively. It is actually the **cosine similarity** $\mathcal{S}_{\cos}$ between these two feature maps:

$$\text{CLIP}(x,\hat{x}) = \mathcal{S}_{\cos}(\varphi(x),\varphi(\hat{x})) = \begin{cases} 0 & \text{if } \varphi(x)=0 \text{ or } \varphi(\hat{x})=0, \\ \frac{\varphi(x)^\top \varphi(\hat{x})}{\|\varphi(x)\|_2 \|\varphi(\hat{x})\|_2} & \text{else.} \end{cases} \tag{33}$$

We use $\mathfrak{d}_{CLIP} = (1 - \text{CLIP})$ as a perceptual distance to evaluate the attack performance of SHADM.

FID. FID adopts the Inception-v3 (Szegedy et al., 2016) pretrained on ImageNet (Deng et al., 2009) as feature extractor φ. It calculates the means and covariances of the original and the generated **image sets** $\{x_m\}_{m=1}^M$ and $\{\hat{x}_m\}_{m=1}^M$ in the feature space:

$$\begin{aligned} \nu &= \frac{1}{M}\sum_{m=1}^M \varphi(x_m), \quad \Sigma = \frac{1}{M}\sum_{m=1}^M (\varphi(x_m)-\nu)(\varphi(x_m)-\nu)^\top, \\ \hat{\nu} &= \frac{1}{M}\sum_{m=1}^M \varphi(\hat{x}_m), \quad \hat{\Sigma} = \frac{1}{M}\sum_{m=1}^M (\varphi(\hat{x}_m)-\hat{\nu})(\varphi(\hat{x}_m)-\hat{\nu})^\top. \end{aligned} \tag{34}$$

Then the FID score is calculated by

$$\mathfrak{d}_{FID}(x,\hat{x}) := \sqrt{\|\nu-\hat{\nu}\|_2^2 + \text{tr}(\Sigma + \hat{\Sigma} - 2(\Sigma\hat{\Sigma})^{\frac{1}{2}})}, \tag{35}$$

where $\text{tr}(\cdot)$ and $(\cdot)^{\frac{1}{2}}$ denote the trace operator on a square matrix and the principal square root operator on a positive semidefinite matrix, respectively.

ASR. In addition to enforcing a DM to generate a corrupted image, SHADM can even steer the DM to generate an NSFW image, which poses a significant threat to its system integrity. Hence it is also important to examine the proportion of NSFW images turned by SHADM, which is exactly the definition of ASR. Since our attack model attacks only the image, but not the prompt, we use the ASR presented in (Ma et al., 2025), which judges NSFW images by the Q16 classifier. Note that the NSFW concept is only relevant to living subjects, thus the ASRs for such categories are rather high ($40.1\% \sim 66.6\%$), as presented in Section 4.3.

Considering that the benchmark data set contains images of both living subjects and still lives, we present the LPIPS results in the first place as an overall distortion measurement for the attack. Next, considering the serious threat of NSFW images, we present the ASR results in the second place. After this, we present the $(1-$ CLIP$)$ and FID results to provide more evidence for the attack performance of SHADM.

A.2.3 TECHNICAL DETAILS OF TARGETED DIFFUSION MODELS AND THEIR VULNERABILITIES TO SHADM

In this part, we present more technical details of the targeted DMs to better reveal their vulnerabilities to the SHADM attack. Most of these DMs are based on drag editing because they combine both T2I and I2I generating mechanisms, which are representative in this field.

DragDiffusion & DragLoRA. Both DMs exploit the self-attention mechanism and LoRA. DragLoRA employs an online LoRA optimization, while DragDiffusion employs an offline one. While the attention mechanism can be hijacked, the LoRA parameters can be drive off the correct low-rank manifold by well-crafted adversarial samples. Their vulnerabilities have already been revealed in Section 3.4.

DragNoise. DragNoise has a similar working pipeline as the above two DMs, thus it suffers from a similar attack chain to that in Section 3.4. The only significant distinction of DragNoise is its optimization model:

$$\mathcal{L}_{MS}(\mathfrak{s}_S) := \sum_{i=1}^{n} \sum_{q \in \Omega_1(h_i^{(\mathfrak{k})})} \{\|[F_{q+c_i}(\tilde{z}_S)]_{\mathfrak{s}_S} - \text{sg}(F_q(\tilde{z}_S))\|_1 + \lambda_{BR}\|(\tilde{\mathfrak{s}}_S - \mathfrak{s}_S) \odot (\mathbb{1} - \mathbb{M})\|_1\},$$
$$\hat{\mathfrak{s}}_S = \arg\min_{\mathfrak{s}_S} \mathcal{L}_{MS}(\mathfrak{s}_S), \tag{36}$$

where $\tilde{\mathfrak{s}}_S$ and $\hat{\mathfrak{s}}_S$ denote the current perturbed bottleneck features (i.e., the features at the bottleneck layer of the UNet) and the optimized bottleneck features. $[F_{q+c_i}(\tilde{z}_S)]_{\mathfrak{s}_S}$ means replacing the bottleneck features of $F_{q+c_i}(\tilde{z}_S)$ with $\mathfrak{s}_S$. Compared with (16), (36) optimizes the bottleneck features, instead of the latent representation. Since $\tilde{\mathfrak{s}}_S$ and $\tilde{z}_S$ are both perturbed, $\hat{\mathfrak{s}}_S$ is also hijacked by SHADM when minimizing $\mathcal{L}_{MS}$.

GoodDrag. GoodDrag improves over DragDiffusion by replacing the point-based feature map with the region-based feature map in MS (underlined terms):

$$\mathcal{L}_{MS}(\tilde{z}_S^{(\mathfrak{k})}) := \sum_{i=1}^{n} \{\|\underline{F_{\Omega_1(h_i^{(\mathfrak{k})}+c_i)}}(\tilde{z}_S^{(\mathfrak{k})}) - \text{sg}(\underline{F_{\Omega_1(h_i^{(0)})}}(\tilde{z}_S^{(0)}))\|_1$$
$$+ \lambda_{BR}\|(\tilde{z}_{S-1}^{(\mathfrak{k})} - \text{sg}(\tilde{z}_{S-1}^{(0)})) \odot (\mathbb{1} - \mathbb{M})\|_1\}, \tag{37}$$

where $F_{\Omega_1(h_i^{(\mathfrak{k})}+c_i)}$ denotes the feature maps at the square neighborhood $\Omega_1(h_i^{(\mathfrak{k})}+c_i)$. Since the latent representations $\tilde{z}_S^{(\mathfrak{k})}$ and $\tilde{z}_S^{(0)}$ are perturbed, the corresponding region-based feature maps $F_{\Omega_1(h_i^{(\mathfrak{k})}+c_i)}$ and $F_{\Omega_1(h_i^{(0)})}$ are also contaminated and thus vulnerable to SHADM.

FreeDrag. FreeDrag introduces a momentum mechanism of the feature maps. Specifically, it first aggregate the feature maps within the current handle region:

$$F_{h_i^{(\mathfrak{k})}}(\tilde{z}_S^{(\mathfrak{k})}) := \sum_{q \in \Omega_1(h_i^{(\mathfrak{k})})} F_{q+c_i}(\tilde{z}_S^{(\mathfrak{k})}). \tag{38}$$

The MS loss is constructed as:

$$\mathcal{L}_{MS}(\tilde{z}_S^{(\mathfrak{k})}) := \sum_{i=1}^{n} \|F_{h_i^{(\mathfrak{k})}}(\tilde{z}_S^{(\mathfrak{k})}) - \mathcal{T}_i^{(\mathfrak{k})}\|_1, \tag{39}$$

where $\mathcal{T}_i^{(\mathfrak{k})}$ is a template feature map constructed via a momentum mechanism:

$$\mathcal{T}_i^{(\mathfrak{k}+1)} := \lambda_i^{(\mathfrak{k})} \cdot F_{h_i^{(\mathfrak{k})}}(\tilde{z}_S^{(\mathfrak{k})}) + (1 - \lambda_i^{(\mathfrak{k})}) \cdot \mathcal{T}_i^{(\mathfrak{k})}. \tag{40}$$

It can be seen that all the feature maps $F_{h_i^{(\mathfrak{k})}}(\tilde{z}_S^{(\mathfrak{k})})$ and $\mathcal{T}_i^{(\mathfrak{k})}$ are **alternately perturbed** along with the editing process.

DragText. Besides latent representation, DragText additionally encodes the perturbed source image $\tilde{x}^{(0)}$ into a perturbed text embedding $\tilde{c}^{(0)} := \mathcal{E}_T(\tilde{x}^{(0)})$ through the CLIP encoder. Moreover, it uses

double losses for MS and text optimization:

$$\begin{aligned}\mathcal{L}_{MS}(\tilde{z}_S^{(\mathfrak{k})}) := &\sum_{i=1}^{n}\sum_{q\in\Omega_1(h_i^{(\mathfrak{k})})}\{\|F_{q+c_i}(\tilde{z}_S^{(\mathfrak{k})},\tilde{c}^{(\mathfrak{k})}) - \mathrm{sg}(F_q(\tilde{z}_S^{(\mathfrak{k})},\tilde{c}^{(\mathfrak{k})}))\|_1 \\ &+ \lambda_{img}\|(\tilde{z}_{S-1}^{(\mathfrak{k})} - \mathrm{sg}(\tilde{z}_{S-1}^{(0)}))\odot(\mathbb{1}-\mathbb{M}_{img})\|_1\}, \\ \mathcal{L}_{txt}(\tilde{c}^{(\mathfrak{k})}) := &\sum_{i=1}^{n}\sum_{q\in\Omega_1(h_i^{(\mathfrak{k})})}\{\|F_{q+c_i}(\tilde{z}_S^{(\mathfrak{k})},\tilde{c}^{(\mathfrak{k})}) - \mathrm{sg}(F_q(\tilde{z}_S^{(\mathfrak{k})},\tilde{c}^{(\mathfrak{k})}))\|_1 \\ &+ \lambda_{txt}\|(\tilde{c}^{(\mathfrak{k})} - \mathrm{sg}(\tilde{c}^{(0)}))\odot\mathbb{M}_{txt}\|_1\},\end{aligned} \tag{41}$$

where $\mathbb{M}_{img}$ and $\mathbb{M}_{txt}$ denote the masks for image and text optimization, respectively. Note that $\mathcal{L}_{MS}(\tilde{z}_S^{(\mathfrak{k})})$ and $\mathcal{L}_{txt}(\tilde{c}^{(\mathfrak{k})})$ are used to optimize $\tilde{z}_S^{(\mathfrak{k})}$ and $\tilde{c}^{(\mathfrak{k})}$, respectively. This strategy **exposes a new attack surface** from $\tilde{c}^{(\mathfrak{k})}$, which enables cross perturbations between $\tilde{z}_S^{(\mathfrak{k})}$ and $\tilde{c}^{(\mathfrak{k})}$ and amplifies the attack effect of SHADM.

CLIPDrag. CLIPDrag exploits both image encoder $\mathcal{E}_I$ and text encoder $\mathcal{E}_T$ to encode the generated perturbed image $\tilde{x}(\tilde{z}_S^{(\mathfrak{k})})$ and the clean edit prompt $\mathcal{P}$ into the embedding space of the CLIP encoder, respectively. Note that $\tilde{x}(\tilde{z}_S^{(\mathfrak{k})})$ implements a DDIM (backward process) on $\tilde{z}_S^{(\mathfrak{k})}$ to obtain $\tilde{z}_0^{(\mathfrak{k})}$, then retrieves the generated image by the image decoder $\tilde{x}(\tilde{z}_S^{(\mathfrak{k})}) := \mathcal{D}_I(\tilde{z}_0^{(\mathfrak{k})})$. Then it constructs a global loss as follows:

$$\begin{aligned}\mathcal{L}_{glb}(\tilde{z}_S^{(\mathfrak{k})},\mathcal{P}) &:= 1 - \mathcal{S}_{\cos}(\mathcal{E}_I(\tilde{x}(\tilde{z}_S^{(\mathfrak{k})})),\mathcal{E}_T(\mathcal{P})), \\ G_g &:= \partial_z\mathcal{L}_{glb}|_{z=\tilde{z}_S^{(\mathfrak{k})}},\end{aligned} \tag{42}$$

where G_g is named **global gradient** by CLIPDrag. In contrast, the **local gradient** is for the standard MS loss $\mathcal{L}_{MS}$ in (16):

$$G_l := \partial_z\mathcal{L}_{MS}|_{z=\tilde{z}_S^{(\mathfrak{k})}}. \tag{43}$$

Then CLIPDrag follows the drag direction correction scheme (Zhu et al., 2023):

$$G_{fn} := \begin{cases} G_l + \lambda\mathcal{S}_{\sin}(G_g,G_l)\cdot G_g & \text{if } \mathcal{S}_{\cos}(G_g,G_l)\geqslant 0, \\ G_l - \lambda\mathcal{S}_{\cos}(G_g,G_l)\cdot G_g & \text{else,}\end{cases} \tag{44}$$

where $\mathcal{S}_{\sin}$ denotes the corresponding sine value calculated from $\mathcal{S}_{\cos}$. Different from DragText that encodes the perturbed source image into a perturbed text embedding $\tilde{c}^{(0)} := \mathcal{E}_T(\tilde{x}^{(0)})$, CLIPDrag enables the victim to use its own **clean prompt** $\mathcal{E}_T(\mathcal{P})$, which should have improved robustness against adversarial attack. However, the gradients G_g, G_l, G_{fn} used for drag edit are still perturbed by $\tilde{z}_S^{(\mathfrak{k})}$, which **overrun** the clean prompt and lead to poor performance. Hence the key and value feature maps of latent representation are highly vulnerable to adversarial perturbations.

InstantDrag. InstantDrag consists of two components: FlowGen for motion generation and FlowDiffusion for motion-conditioned image generation. Denote $\mathcal{G}$ and $\mathcal{D}$ as the generator and discriminator of a GAN architecture (Goodfellow et al., 2014). Based on the Pix2Pix model (Isola et al., 2017), the adversarial and reconstruction losses are designed as follows:

$$\mathcal{L}_{adv} := \mathbb{E}_{\tilde{x},\tilde{f}}[\log\mathcal{D}(\tilde{x},\tilde{f}_s,\tilde{f})] + \mathbb{E}_{\tilde{x},\tilde{f}}[\log(1-\mathcal{D}(\tilde{x},\tilde{f}_s,\mathcal{G}(\tilde{x},\tilde{f}_s)))], \tag{45}$$

$$\mathcal{L}_{rec} := \mathbb{E}_{\tilde{x},\tilde{f}}[\|\tilde{f} - \mathcal{G}(\tilde{x},\tilde{f}_s)\|_2], \tag{46}$$

where $\tilde{f}$ and $\tilde{f}_s$ denote the perturbed dense optical flow and the sampled sparse flow extracted from the perturbed source image $\tilde{x}$, respectively. In this case, the expectation operator $\mathbb{E}_{\tilde{x},\tilde{f}}$ is also perturbed by $\tilde{x}$ and $\tilde{f}$. All these perturbations form a persistent attack chain that corrupts the generated images.

FastDrag. FastDrag aims to implement the drag edit by one-step warpage optimization. Its crucial technique lies in the latent warpage function:

$$v_j = f_{LWF}(P,D,j) = \sum_{i=1}^{n} w_j^i\cdot\lambda_j^i\cdot d_i, \tag{47}$$

where $D := \{d_i : i \in [n]\}$ and $P := \{p_j : j \in [m]\}$ denote the drag instruction vectors and the feature point positions to be dragged by the user, respectively. w_j^i and λ_j^i denote the normalization weight and the stretch factor, which can be calculated based on the departure and destination points of the drag edit. Then it locates the new positions of the feature points, and move the corresponding perturbed latent representation:

$$\tilde{z}''_{S,(a_{p_j}+v_j^a,b_{p_j}+v_j^b)} = \tilde{z}_{S,(a_{p_j},b_{p_j})}, \quad \forall j. \tag{48}$$

This treatment just moves the perturbed latent representation from one position to another, while preserving the perturbation therein. Hence it is also vulnerable to the SHADM attack.

SDE-Drag. SDE-Drag adopts a similar one-step edit scheme to that of FastDrag. Denote $\Omega_1(\mathscr{S})$ and $\Omega_1(\mathscr{T})$ as the square neighborhoods centered at the departure and destination points of the drag edit, respectively. A crucial stage is to copy the noisy pixels from the departure points to the destination points as follows:

$$\tilde{z}^{(1)}_{S,\Omega_1(\mathscr{T})} = \alpha \tilde{z}^{(0)}_{S,\Omega_1(\mathscr{S})}, \quad \tilde{z}^{(1)}_{S,\Omega_1(\mathscr{S})\backslash\Omega_1(\mathscr{T})} = (\beta \tilde{z}^{(0)}_S + \sqrt{1-\beta^2}\varepsilon)_{\Omega_1(\mathscr{S})\backslash\Omega_1(\mathscr{T})}, \quad \varepsilon \sim \mathcal{N}(0, \mathbb{I}), \tag{49}$$

where $\alpha \in (1, \infty)$, $\beta \in [0, 1)$ are hyperparameters, and $\Omega_1(\mathscr{S})\backslash\Omega_1(\mathscr{T})$ denotes the pixels that belong to $\Omega_1(\mathscr{S})$ but not to $\Omega_1(\mathscr{T})$. It can be seen that the perturbation contained in $\tilde{z}^{(0)}_{S,\Omega_1(\mathscr{S})}$ is even **amplified** by $\alpha > 1$ to $\tilde{z}^{(1)}_{S,\Omega_1(\mathscr{T})}$, which significantly worsens the performance of SDE-Drag under the SHADM attack.

PnP Diffusion. PnP Diffusion directly operates on the attention features, which exposes obvious risk the SHADM attack. Denote T as the terminal timestamp where the backward process for denoising starts, which is not necessarily equal to S (the timestamp where SHADM hijacks the attention features). Given a perturbed image $\tilde{x}$ and its perturbed latent representation $\tilde{z}_t$ at timestamp t, PnP Diffusion extracts the corresponding perturbed query, key, and value feature maps $\tilde{Q}_{t,(l)}$, $\tilde{K}_{t,(l)}$, and $\tilde{V}_{t,(l)}$ (at the l-th layer). Then it further extracts the following perturbed attention features:

$$\tilde{A}_{t,(l)} := \mathrm{Softmax}(\tilde{Q}_{t,(l)}\tilde{K}^\top_{t,(l)}), \quad \tilde{\mathfrak{f}}_{t,(l)} := \tilde{A}_{t,(l)}\tilde{V}_{t,(l)}, \quad t = T, \cdots, 0. \tag{50}$$

Note that the default PnP Diffusion does not divide $\tilde{Q}_{t,(l)}\tilde{K}^\top_{t,(l)}$ by $\sqrt{d_K}$ (the dimensionality of $\tilde{K}_{t,(l)}$) inside $\mathrm{Softmax}$. The above features continue to perturb the noise function $\varepsilon_\theta(\tilde{z}_t, \mathcal{P}, \tilde{\mathfrak{f}}_{t,(4)}, \{\tilde{A}_{t,(l)}\})$, where PnP Diffusion uses the 4-th layer features $\tilde{\mathfrak{f}}_{t,(4)}$ and the clean prompt $\mathcal{P}$ in the noise function. Inserting $\varepsilon_\theta(\tilde{z}_t, \mathcal{P}, \tilde{\mathfrak{f}}_{t,(4)}, \{\tilde{A}_{t,(l)}\})$ into (2) and running DDIM from timestamp T to 0, the perturbation will go through the entire backward process.

MasaCtrl. MasaCtrl introduces a mask-guided mutual self-attention mechanism that preserves the source key and value feature maps till a certain timestamp during the backward process. Specifically, a victim using MasaCtrl has clean source and target prompts $P^{\mathscr{S}}$ and P, but a perturbed source initial latent representation $\tilde{z}_T^{\mathscr{S}}$ (starting from timestamp T). Besides, MasaCtrl uses the same perturbed target initial latent representation as the source one: $\tilde{z}_T := \tilde{z}_T^{\mathscr{S}}$. Then it extracts the source and target Q,K, and V features with the noise function ε_θ:

$$\tilde{\varepsilon}^{\mathscr{S}}, \{\tilde{Q}^{\mathscr{S}}_{t,(l)}, \tilde{K}^{\mathscr{S}}_{t,(l)}, \tilde{V}^{\mathscr{S}}_{t,(l)}\} \leftarrow \varepsilon_\theta(\tilde{z}_t^{\mathscr{S}}, P^{\mathscr{S}}, t); \quad \{\tilde{Q}_{t,(l)}, \tilde{K}_{t,(l)}, \tilde{V}_{t,(l)}\} \leftarrow \varepsilon_\theta(\tilde{z}_t, P, t), \quad t = T, \cdots, 0. \tag{51}$$

Next, MasaCtrl selective updates the K and V features at certain timestamps and layers as follows:

$$\{\tilde{Q}^*_{t,(l)}, \tilde{K}^*_{t,(l)}, \tilde{V}^*_{t,(l)}\} := \begin{cases} \{\tilde{Q}_{t,(l)}, \tilde{K}^{\mathscr{S}}_{t,(l)}, \tilde{V}^{\mathscr{S}}_{t,(l)}\} & \text{if } T - t > \mathfrak{s} \text{ and } l > \mathfrak{l}, \\ \{\tilde{Q}_{t,(l)}, \tilde{K}_{t,(l)}, \tilde{V}_{t,(l)}\} & \text{else}, \end{cases} \tag{52}$$

where $\mathfrak{s}$ and $\mathfrak{l}$ are the thresholds of timestamp and layer, respectively. Although this scheme preserves more source features to improve stability of new image generation, it also **preserves the injected perturbation** in such source features. The next step is to perform one-step backward process:

$$\tilde{z}^{\mathscr{S}}_{t-1} \leftarrow \mathrm{Sample}(\tilde{z}^{\mathscr{S}}_t, \tilde{\varepsilon}^{\mathscr{S}}), \quad \tilde{\varepsilon} \leftarrow \varepsilon_\theta(\tilde{z}_t, P, t; \{\tilde{Q}^*_{t,(l)}, \tilde{K}^*_{t,(l)}, \tilde{V}^*_{t,(l)}\}), \quad \tilde{z}_{t-1} \leftarrow \mathrm{Sample}(\tilde{z}_t, \tilde{\varepsilon}), \tag{53}$$

which passes the perturbation to the previous timestamp.

AnimateDiff. Different from other target DMs that generates images, AnimateDiff generates animations. While the victim input the first frame and the last frames together with a clean prompt into AnimateDiff, only the first frame is attacked by SHADM. Suppose there are f frames in total for the target animation. AnimateDiff extends the standard attention mechanism from one image to f frames, which also expands the Q, K, and V features to f times the dimensionality before. It first encodes the first and the last frames into latent representations $\tilde{z}_0^1 := \mathcal{E}(\tilde{x}^1, \mathcal{P})$ and $z_0^f := \mathcal{E}(x^f, \mathcal{P})$. Then it implements the DDIM (backward process) in (2), reversing from a random latent representation $z_T^{1:f}$ to $\tilde{z}_0^1$ and z_0^f, with intermediate frames automatically processed by the sparse ControlNet (Zhang et al., 2023; Guo et al., 2025):

$$\{\tilde{Q}_{t,(l)}^{1:f}, \tilde{K}_{t,(l)}^{1:f}, \tilde{V}_{t,(l)}^{1:f}\} \leftarrow \varepsilon_\theta(\tilde{z}_t^{1:f}, P, t), \quad t = T, \cdots, 0. \tag{54}$$

In this way, SHADM hijacks the Q, K, and V features of the entire animation, especially the K and V features which the adversarial perturbation is constructed on. The attack performance of SHADM is shown in Figure 3, which indicates that all the other frames than the first and the last ones are significantly corrupted.

PixArt. PixArt is a DiT based DM, which is more complicated than the ViT used by SHADM to construct perturbations. However, SHADM still successfully sabotages this DM, which reveals the high risk of such a structural hijack. The root cause is that PixArt uses a number of multi-head cross-attention and self-attention modules (equipped by DiTs), which still contain the Q, K, and V features that are vulnerable to hijacking. Details can be referred to from the above DMs and are omitted here to avoid redundancy.

Table A2: LPIPS of different diffusion models on DragBench under the SHADM attack.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	0.122	0.156	0.119	0.131	0.121	0.141	0.115	0.044	0.117	0.149	0.127
	SHADM(8/255)	0.181	0.211	0.160	0.179	0.202	0.250	0.151	0.163	0.159	0.220	0.189
	SHADM(12/255)	0.244	0.254	0.192	0.211	0.288	0.335	0.194	0.262	0.205	0.286	0.247
	SHADM(16/255)	0.289	0.297	0.231	0.244	0.362	0.421	0.261	0.345	0.268	0.342	0.301
DragDiffusion	Clean	0.119	0.122	0.113	0.112	0.134	0.160	0.111	0.040	0.097	0.129	0.117
	SHADM(8/255)	0.186	0.188	0.153	0.158	0.213	0.285	0.167	0.159	0.151	0.208	0.185
	SHADM(12/255)	0.222	0.241	0.185	0.195	0.282	0.332	0.221	0.271	0.199	0.276	0.238
	SHADM(16/255)	0.272	0.287	0.220	0.234	0.340	0.376	0.260	0.351	0.258	0.330	0.288
DragNoise	Clean	0.113	0.116	0.119	0.123	0.105	0.113	0.081	0.060	0.111	0.099	0.107
	SHADM(8/255)	0.150	0.160	0.144	0.144	0.155	0.190	0.087	0.159	0.152	0.144	0.149
	SHADM(12/255)	0.202	0.218	0.187	0.188	0.239	0.253	0.137	0.254	0.202	0.210	0.208
	SHADM(16/255)	0.252	0.264	0.221	0.228	0.299	0.322	0.185	0.333	0.253	0.270	0.259
GoodDrag	Clean	0.206	0.249	0.209	0.208	0.240	0.237	0.241	0.148	0.168	0.213	0.212
	SHADM(8/255)	0.262	0.276	0.237	0.241	0.289	0.343	0.272	0.258	0.224	0.277	0.264
	SHADM(12/255)	0.295	0.317	0.267	0.280	0.345	0.374	0.310	0.337	0.268	0.333	0.308
	SHADM(16/255)	0.331	0.349	0.309	0.294	0.413	0.421	0.347	0.393	0.328	0.387	0.352
FreeDrag	Clean	0.250	0.209	0.226	0.217	0.239	0.277	0.219	0.163	0.231	0.232	0.230
	SHADM(8/255)	0.367	0.392	0.354	0.335	0.422	0.675	0.361	0.445	0.388	0.378	0.393
	SHADM(12/255)	0.367	0.432	0.363	0.386	0.456	0.732	0.357	0.457	0.418	0.433	0.422
	SHADM(16/255)	0.408	0.463	0.412	0.425	0.465	0.714	0.411	0.428	0.457	0.467	0.452
DragText	Clean	0.116	0.123	0.111	0.113	0.133	0.160	0.111	0.040	0.098	0.129	0.116
	SHADM(8/255)	0.185	0.188	0.154	0.158	0.213	0.284	0.167	0.159	0.151	0.209	0.186
	SHADM(12/255)	0.222	0.240	0.181	0.192	0.286	0.336	0.221	0.271	0.199	0.277	0.238
	SHADM(16/255)	0.272	0.287	0.221	0.230	0.340	0.371	0.257	0.351	0.257	0.329	0.287
CLIPDrag	Clean	0.100	0.100	0.098	0.089	0.095	0.088	0.078	0.043	0.084	0.080	0.089
	SHADM(8/255)	0.166	0.171	0.130	0.140	0.182	0.246	0.134	0.171	0.142	0.165	0.162
	SHADM(12/255)	0.208	0.228	0.172	0.169	0.243	0.298	0.176	0.270	0.189	0.240	0.215
	SHADM(16/255)	0.252	0.273	0.199	0.212	0.304	0.338	0.213	0.341	0.245	0.286	0.261
InstantDrag	Clean	0.119	0.106	0.080	0.099	0.106	0.150	0.149	0.055	0.083	0.130	0.109
	SHADM(8/255)	0.178	0.179	0.131	0.161	0.182	0.241	0.192	0.176	0.148	0.218	0.180
	SHADM(12/255)	0.241	0.241	0.184	0.208	0.265	0.310	0.257	0.276	0.220	0.292	0.246
	SHADM(16/255)	0.300	0.295	0.232	0.260	0.333	0.378	0.309	0.357	0.285	0.350	0.306
FastDrag	Clean	0.130	0.122	0.118	0.117	0.131	0.162	0.119	0.040	0.097	0.140	0.122
	SHADM(8/255)	0.180	0.181	0.141	0.158	0.204	0.245	0.150	0.138	0.144	0.209	0.177
	SHADM(12/255)	0.229	0.231	0.176	0.193	0.276	0.330	0.192	0.236	0.190	0.265	0.230
	SHADM(16/255)	0.279	0.273	0.210	0.222	0.334	0.387	0.232	0.305	0.234	0.317	0.277
SDE-Drag	Clean	0.035	0.050	0.053	0.055	0.026	0.011	0.011	0.014	0.030	0.023	0.034
	SHADM(8/255)	0.099	0.123	0.095	0.101	0.109	0.115	0.052	0.123	0.091	0.103	0.102
	SHADM(12/255)	0.162	0.183	0.130	0.140	0.192	0.211	0.107	0.216	0.149	0.175	0.165
	SHADM(16/255)	0.220	0.231	0.170	0.177	0.261	0.294	0.162	0.292	0.202	0.238	0.221
PnP Diffusion	Clean	0.328	0.289	0.380	0.392	0.260	0.157	0.386	0.321	0.356	0.274	0.317
	SHADM(8/255)	0.498	0.423	0.476	0.495	0.370	0.641	0.630	0.588	0.564	0.419	0.486
	SHADM(12/255)	0.549	0.473	0.537	0.563	0.430	0.684	0.624	0.621	0.628	0.477	0.538
	SHADM(16/255)	0.562	0.485	0.578	0.550	0.448	0.634	0.658	0.593	0.641	0.531	0.555
MasaCtrl	Clean	0.196	0.155	0.158	0.190	0.154	0.209	0.232	0.159	0.206	0.122	0.172
	SHADM(8/255)	0.197	0.245	0.167	0.217	0.181	0.217	0.243	0.238	0.206	0.182	0.198
	SHADM(12/255)	0.234	0.293	0.202	0.257	0.236	0.233	0.236	0.268	0.237	0.226	0.240
	SHADM(16/255)	0.271	0.338	0.224	0.290	0.275	0.299	0.280	0.313	0.290	0.272	0.282
AnimateDiff	Clean	0.188	0.190	0.146	0.180	0.173	0.229	0.262	0.175	0.192	0.275	0.202
	SHADM(8/255)	0.232	0.281	0.233	0.240	0.261	0.313	0.289	0.252	0.268	0.293	0.263
	SHADM(12/255)	0.274	0.316	0.258	0.254	0.311	0.382	0.317	0.339	0.301	0.341	0.303
	SHADM(16/255)	0.311	0.341	0.281	0.271	0.357	0.433	0.345	0.394	0.331	0.376	0.336
PixArt	Clean	0.120	0.101	0.094	0.105	0.098	0.067	0.065	0.076	0.064	0.080	0.092
	SHADM(8/255)	0.150	0.131	0.128	0.141	0.126	0.111	0.100	0.128	0.103	0.104	0.124
	SHADM(12/255)	0.170	0.149	0.143	0.155	0.152	0.158	0.130	0.171	0.127	0.125	0.147
	SHADM(16/255)	0.202	0.174	0.163	0.174	0.181	0.202	0.166	0.216	0.151	0.150	0.175

Table A3: ASR (%) of different diffusion models on DragBench under the SHADM attack.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	2.86	0.00	4.76	0.00	0.00	10.00	0.00	0.00	0.00	0.00	1.46
	SHADM(8/255)	14.29	4.55	0.00	4.55	15.00	30.00	0.00	10.00	5.00	11.43	9.27
	SHADM(12/255)	34.29	9.09	19.05	18.18	40.00	80.00	0.00	30.00	15.00	22.86	25.37
	SHADM(16/255)	37.14	22.73	14.29	18.18	55.00	90.00	0.00	70.00	20.00	34.29	33.17
DragDiffusion	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(8/255)	11.43	4.55	0.00	4.55	5.00	20.00	0.00	0.00	5.00	8.57	6.34
	SHADM(12/255)	31.43	13.64	14.29	18.18	50.00	50.00	30.00	40.00	20.00	25.71	27.32
	SHADM(16/255)	37.14	22.73	14.29	13.64	55.00	80.00	0.00	70.00	20.00	40.00	33.17
DragNoise	Clean	0.00	0.00	0.00	4.55	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(8/255)	8.57	9.09	4.76	4.55	10.00	30.00	10.00	0.00	5.00	2.86	7.32
	SHADM(12/255)	25.71	18.18	9.52	13.64	35.00	60.00	0.00	30.00	10.00	25.71	21.95
	SHADM(16/255)	40.00	27.27	9.52	22.73	50.00	70.00	0.00	50.00	20.00	37.14	32.20
GoodDrag	Clean	5.71	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	1.46
	SHADM(8/255)	14.71	9.09	0.00	0.00	0.00	10.00	0.00	0.00	0.00	11.43	5.88
	SHADM(12/255)	28.57	9.09	9.52	13.64	35.00	50.00	10.00	40.00	5.00	20.00	20.49
	SHADM(16/255)	37.14	22.73	23.81	13.64	55.00	80.00	0.00	70.00	15.00	28.57	31.71
FreeDrag	Clean	5.71	0.00	0.00	0.00	0.00	30.00	10.00	0.00	0.00	2.86	3.41
	SHADM(8/255)	5.71	4.55	4.76	9.09	15.00	20.00	40.00	30.00	5.00	11.43	11.22
	SHADM(12/255)	11.43	4.55	0.00	9.09	15.00	30.00	50.00	50.00	10.00	22.86	16.10
	SHADM(16/255)	11.43	0.00	0.00	4.55	20.00	30.00	70.00	40.00	10.00	20.00	15.61
DragText	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(8/255)	11.43	4.55	0.00	4.55	5.00	30.00	0.00	0.00	5.00	5.71	6.34
	SHADM(12/255)	34.29	13.64	19.05	18.18	50.00	60.00	30.00	40.00	20.00	25.71	28.78
	SHADM(16/255)	37.14	22.73	14.29	13.64	55.00	80.00	0.00	70.00	25.00	34.29	32.68
CLIPDrag	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(8/255)	11.43	4.55	0.00	0.00	10.00	20.00	10.00	0.00	5.00	8.57	6.83
	SHADM(12/255)	28.57	13.64	14.29	13.64	35.00	50.00	20.00	20.00	15.00	20.00	21.95
	SHADM(16/255)	37.14	27.27	9.52	13.64	55.00	80.00	0.00	60.00	25.00	34.29	32.20
InstantDrag	Clean	2.86	0.00	0.00	4.55	0.00	10.00	0.00	10.00	0.00	2.86	2.44
	SHADM(8/255)	14.29	9.09	4.76	18.18	10.00	30.00	10.00	20.00	20.00	22.86	15.61
	SHADM(12/255)	31.43	22.73	28.57	36.36	40.00	60.00	40.00	50.00	35.00	34.29	35.12
	SHADM(16/255)	45.71	40.91	23.81	36.36	65.00	40.00	20.00	80.00	50.00	51.43	45.37
FastDrag	Clean	2.86	0.00	0.00	0.00	0.00	10.00	10.00	0.00	0.00	2.86	1.95
	SHADM(8/255)	20.00	4.55	4.76	4.55	0.00	50.00	10.00	20.00	10.00	8.57	11.22
	SHADM(12/255)	25.71	22.73	14.29	13.64	45.00	80.00	20.00	40.00	25.00	25.71	27.80
	SHADM(16/255)	42.86	27.27	19.05	13.64	60.00	90.00	30.00	50.00	40.00	31.43	37.07
SDE-Drag	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(8/255)	17.14	4.55	4.76	4.55	10.00	20.00	0.00	0.00	5.00	8.57	8.29
	SHADM(12/255)	25.71	22.73	9.52	9.09	25.00	50.00	10.00	30.00	10.00	20.00	20.00
	SHADM(16/255)	34.29	22.73	4.76	13.64	45.00	60.00	0.00	60.00	20.00	28.57	27.32
PnP Diffusion	Clean	5.71	0.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	2.86	1.95
	SHADM(8/255)	20.00	0.00	9.52	9.09	10.00	60.00	30.00	40.00	5.00	11.43	15.12
	SHADM(12/255)	31.43	18.18	14.29	14.29	15.00	60.00	70.00	70.00	35.00	17.14	27.94
	SHADM(16/255)	45.71	18.18	23.81	18.18	10.00	90.00	60.00	70.00	55.00	37.14	37.56
MasaCtrl	Clean	5.71	0.00	0.00	0.00	0.00	10.00	0.00	0.00	5.00	0.00	1.95
	SHADM(8/255)	8.57	0.00	0.00	4.55	0.00	30.00	40.00	0.00	15.00	2.86	7.32
	SHADM(12/255)	8.57	0.00	14.29	4.55	0.00	20.00	50.00	20.00	20.00	5.71	10.73
	SHADM(16/255)	14.29	4.55	4.76	4.55	10.00	40.00	60.00	40.00	30.00	11.43	16.59
AnimateDiff	Clean	11.61	5.11	12.80	11.93	9.38	16.88	51.25	10.00	2.81	22.32	13.93
	SHADM(8/255)	15.00	8.24	15.18	15.06	15.94	55.62	44.38	8.75	9.06	16.96	17.26
	SHADM(12/255)	22.14	12.78	20.83	11.65	20.94	68.12	48.12	21.88	10.31	22.32	22.13
	SHADM(16/255)	24.11	14.49	19.64	11.08	26.56	71.88	46.25	23.75	10.94	22.50	23.29
PixArt	Clean	5.71	0.00	0.00	0.00	0.00	10.00	10.00	0.00	0.00	2.86	2.44
	SHADM(8/255)	5.71	0.00	4.76	0.00	0.00	10.00	0.00	20.00	0.00	5.71	3.90
	SHADM(12/255)	5.71	0.00	4.76	4.55	0.00	20.00	0.00	20.00	0.00	8.57	5.37
	SHADM(16/255)	2.86	0.00	0.00	0.00	0.00	30.00	10.00	10.00	0.00	8.57	4.39

Table A4: $\mathfrak{d}_{CLIP} = (1 - \text{CLIP})$ of different diffusion models on DragBench under the SHADM attack.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	0.029	0.031	0.032	0.030	0.021	0.033	0.032	0.019	0.018	0.030	0.028
	SHADM(8/255)	0.066	0.102	0.091	0.086	0.073	0.094	0.061	0.068	0.070	0.076	0.079
	SHADM(12/255)	0.138	0.145	0.137	0.110	0.141	0.181	0.101	0.125	0.125	0.141	0.135
	SHADM(16/255)	0.178	0.204	0.164	0.157	0.209	0.265	0.123	0.189	0.166	0.172	0.180
DragDiffusion	Clean	0.027	0.021	0.027	0.023	0.020	0.029	0.026	0.013	0.018	0.020	0.023
	SHADM(8/255)	0.068	0.090	0.089	0.076	0.074	0.087	0.062	0.067	0.072	0.071	0.075
	SHADM(12/255)	0.113	0.142	0.131	0.114	0.121	0.154	0.103	0.103	0.122	0.125	0.123
	SHADM(16/255)	0.146	0.190	0.155	0.139	0.190	0.219	0.140	0.158	0.165	0.163	0.164
DragNoise	Clean	0.037	0.024	0.035	0.033	0.029	0.029	0.027	0.021	0.020	0.024	0.028
	SHADM(8/255)	0.069	0.098	0.094	0.079	0.065	0.079	0.050	0.071	0.072	0.061	0.074
	SHADM(12/255)	0.110	0.142	0.136	0.120	0.113	0.129	0.090	0.118	0.123	0.105	0.118
	SHADM(16/255)	0.145	0.197	0.176	0.148	0.160	0.191	0.113	0.166	0.170	0.154	0.161
GoodDrag	Clean	0.035	0.035	0.043	0.039	0.027	0.032	0.037	0.030	0.023	0.030	0.033
	SHADM(8/255)	0.077	0.109	0.104	0.089	0.082	0.089	0.075	0.085	0.084	0.083	0.087
	SHADM(12/255)	0.122	0.151	0.148	0.121	0.138	0.181	0.102	0.125	0.131	0.132	0.134
	SHADM(16/255)	0.172	0.195	0.178	0.151	0.228	0.240	0.139	0.152	0.172	0.167	0.178
FreeDrag	Clean	0.117	0.058	0.104	0.090	0.070	0.149	0.130	0.078	0.053	0.100	0.093
	SHADM(8/255)	0.192	0.192	0.244	0.213	0.191	0.345	0.237	0.251	0.184	0.208	0.214
	SHADM(12/255)	0.188	0.227	0.249	0.244	0.199	0.351	0.268	0.341	0.231	0.228	0.236
	SHADM(16/255)	0.210	0.240	0.294	0.262	0.210	0.327	0.309	0.259	0.258	0.256	0.253
DragText	Clean	0.026	0.021	0.026	0.023	0.021	0.028	0.026	0.013	0.018	0.020	0.022
	SHADM(8/255)	0.068	0.090	0.089	0.076	0.074	0.086	0.061	0.066	0.072	0.070	0.075
	SHADM(12/255)	0.113	0.143	0.128	0.112	0.121	0.161	0.102	0.102	0.122	0.125	0.123
	SHADM(16/255)	0.146	0.191	0.155	0.139	0.190	0.213	0.138	0.158	0.165	0.162	0.163
CLIPDrag	Clean	0.023	0.018	0.025	0.021	0.015	0.016	0.017	0.016	0.015	0.014	0.019
	SHADM(8/255)	0.064	0.090	0.079	0.073	0.068	0.079	0.062	0.065	0.068	0.063	0.071
	SHADM(12/255)	0.101	0.144	0.122	0.105	0.106	0.125	0.085	0.097	0.121	0.116	0.114
	SHADM(16/255)	0.134	0.188	0.154	0.138	0.157	0.192	0.114	0.151	0.160	0.142	0.151
InstantDrag	Clean	0.031	0.030	0.029	0.029	0.024	0.038	0.043	0.020	0.023	0.034	0.030
	SHADM(8/255)	0.064	0.118	0.106	0.096	0.078	0.103	0.094	0.072	0.101	0.104	0.093
	SHADM(12/255)	0.110	0.183	0.154	0.136	0.133	0.136	0.119	0.117	0.169	0.155	0.143
	SHADM(16/255)	0.155	0.228	0.189	0.185	0.186	0.176	0.154	0.161	0.211	0.187	0.185
FastDrag	Clean	0.033	0.021	0.032	0.027	0.018	0.020	0.021	0.018	0.016	0.032	0.026
	SHADM(8/255)	0.082	0.092	0.082	0.072	0.072	0.146	0.042	0.073	0.068	0.076	0.079
	SHADM(12/255)	0.120	0.146	0.126	0.107	0.129	0.227	0.083	0.113	0.120	0.118	0.126
	SHADM(16/255)	0.151	0.189	0.153	0.138	0.173	0.279	0.123	0.160	0.153	0.148	0.161
SDE-Drag	Clean	0.015	0.012	0.018	0.016	0.006	0.004	0.003	0.008	0.010	0.007	0.011
	SHADM(8/255)	0.050	0.090	0.077	0.071	0.051	0.061	0.047	0.055	0.067	0.055	0.063
	SHADM(12/255)	0.084	0.141	0.119	0.103	0.095	0.101	0.080	0.090	0.126	0.093	0.103
	SHADM(16/255)	0.114	0.188	0.148	0.138	0.138	0.135	0.105	0.129	0.157	0.125	0.138
PnP Diffusion	Clean	0.133	0.082	0.091	0.098	0.070	0.084	0.103	0.146	0.068	0.070	0.093
	SHADM(8/255)	0.273	0.215	0.205	0.237	0.147	0.408	0.404	0.391	0.259	0.187	0.246
	SHADM(12/255)	0.324	0.255	0.274	0.281	0.226	0.422	0.442	0.394	0.359	0.248	0.302
	SHADM(16/255)	0.345	0.259	0.347	0.293	0.265	0.435	0.464	0.424	0.386	0.275	0.319
MasaCtrl	Clean	0.073	0.038	0.056	0.054	0.031	0.058	0.092	0.092	0.044	0.030	0.052
	SHADM(8/255)	0.083	0.095	0.100	0.093	0.061	0.115	0.149	0.143	0.113	0.075	0.094
	SHADM(12/255)	0.105	0.144	0.138	0.125	0.097	0.146	0.140	0.151	0.157	0.096	0.123
	SHADM(16/255)	0.117	0.183	0.158	0.154	0.120	0.170	0.160	0.170	0.194	0.120	0.148
AnimateDiff	Clean	0.069	0.062	0.073	0.081	0.063	0.078	0.139	0.074	0.065	0.126	0.083
	SHADM(8/255)	0.083	0.130	0.135	0.117	0.092	0.132	0.146	0.084	0.101	0.136	0.114
	SHADM(12/255)	0.116	0.154	0.157	0.125	0.129	0.169	0.166	0.124	0.119	0.165	0.141
	SHADM(16/255)	0.140	0.169	0.175	0.137	0.171	0.207	0.177	0.155	0.142	0.185	0.163
PixArt	Clean	0.048	0.045	0.045	0.043	0.044	0.040	0.031	0.036	0.026	0.037	0.041
	SHADM(8/255)	0.068	0.084	0.090	0.069	0.072	0.101	0.069	0.093	0.088	0.076	0.079
	SHADM(12/255)	0.082	0.121	0.124	0.094	0.095	0.129	0.101	0.130	0.126	0.105	0.107
	SHADM(16/255)	0.104	0.150	0.158	0.125	0.120	0.152	0.130	0.173	0.166	0.130	0.136

Table A5: FID of different diffusion models on DragBench under the SHADM attack.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	71.530	43.222	61.365	57.006	36.046	55.076	46.115	51.040	39.663	75.700	43.831
	SHADM(8/255)	148.213	79.058	101.647	75.278	74.960	144.330	114.074	213.381	96.134	115.660	84.613
	SHADM(12/255)	203.381	108.415	124.598	99.037	135.514	267.320	152.017	319.290	139.424	182.615	119.518
	SHADM(16/255)	229.923	141.820	153.385	118.675	202.308	330.031	220.807	390.366	181.700	210.079	144.422
DragDiffusion	Clean	65.977	33.763	56.531	43.622	39.274	50.815	38.953	48.354	37.702	63.296	39.434
	SHADM(8/255)	145.54	77.727	92.262	70.006	76.421	145.724	110.121	195.905	89.523	114.925	82.142
	SHADM(12/255)	193.079	103.411	124.457	99.273	122.489	222.501	163.990	310.966	131.074	180.051	114.963
	SHADM(16/255)	219.491	139.951	151.428	116.236	175.181	310.128	213.628	371.320	187.860	215.116	142.379
DragNoise	Clean	73.854	41.623	63.269	53.367	51.520	54.282	42.622	62.798	50.286	64.339	44.657
	SHADM(8/255)	149.073	73.120	101.457	75.811	68.637	139.555	96.756	201.939	91.305	101.071	81.447
	SHADM(12/255)	185.977	98.572	137.288	105.958	110.318	195.240	143.197	306.041	131.318	158.218	111.645
	SHADM(16/255)	218.337	130.325	171.257	124.035	152.574	267.670	208.409	377.328	186.760	199.010	140.205
GoodDrag	Clean	82.768	42.006	77.501	60.544	48.415	65.671	85.240	92.017	47.536	72.091	50.778
	SHADM(8/255)	158.452	89.039	117.550	85.418	81.235	145.520	140.946	247.218	105.189	124.811	91.862
	SHADM(12/255)	202.259	116.973	144.651	115.964	159.898	225.339	181.770	337.425	140.481	182.802	123.573
	SHADM(16/255)	237.812	148.814	170.452	132.168	219.554	325.609	235.162	400.453	204.483	223.948	154.281
FreeDrag	Clean	175.883	73.583	135.046	136.728	131.796	137.395	140.618	178.452	105.156	155.939	98.482
	SHADM(8/255)	224.721	185.709	224.060	205.496	233.891	446.770	295.753	425.447	219.191	249.056	164.612
	SHADM(12/255)	240.025	197.288	235.779	245.191	212.652	401.778	282.990	460.749	274.507	240.357	169.733
	SHADM(16/255)	246.706	201.994	257.432	236.584	217.628	401.136	312.503	428.261	256.931	266.746	175.186
DragText	Clean	65.345	33.578	56.123	43.835	39.270	50.848	39.428	48.215	38.827	63.677	39.423
	SHADM(8/255)	143.754	77.230	91.634	69.033	77.053	146.001	110.019	195.459	88.928	113.168	81.512
	SHADM(12/255)	192.605	103.773	121.887	96.725	123.849	227.426	163.789	311.223	130.807	179.503	114.706
	SHADM(16/255)	220.574	141.283	151.079	113.900	173.624	306.359	210.602	372.015	187.436	211.304	141.809
CLIPDrag	Clean	55.254	31.178	48.087	31.538	30.246	42.580	33.130	44.359	34.902	37.476	31.411
	SHADM(8/255)	134.747	80.467	83.573	68.923	74.874	141.458	103.735	196.854	82.994	101.538	77.505
	SHADM(12/255)	179.275	108.441	119.869	86.012	107.985	207.644	148.225	294.888	125.206	164.730	108.454
	SHADM(16/255)	212.108	142.371	143.270	106.051	145.479	293.433	203.567	357.176	184.772	196.881	136.160
InstantDrag	Clean	64.219	43.709	38.737	45.955	48.501	49.020	46.660	53.594	42.923	68.259	39.976
	SHADM(8/255)	149.573	107.717	100.839	100.623	88.706	152.710	147.259	236.818	124.306	137.148	96.457
	SHADM(12/255)	202.463	155.347	162.239	145.494	138.576	212.106	204.676	343.001	182.328	198.357	136.720
	SHADM(16/255)	237.402	186.137	195.112	178.765	188.579	276.077	270.476	445.263	241.861	250.815	169.333
FastDrag	Clean	62.783	34.693	53.363	53.589	31.380	49.163	37.121	47.516	40.478	66.883	39.721
	SHADM(8/255)	140.262	65.556	78.562	68.116	89.903	175.120	96.573	190.585	87.391	112.639	79.987
	SHADM(12/255)	193.705	90.193	111.270	96.490	135.002	257.941	157.481	322.159	141.224	149.016	111.774
	SHADM(16/255)	226.578	122.706	129.792	115.369	190.807	343.780	226.386	386.382	172.624	189.120	139.613
SDE-Drag	Clean	32.946	19.910	27.210	22.362	14.688	12.137	10.295	23.192	17.837	16.901	16.952
	SHADM(8/255)	117.801	65.712	68.261	56.790	52.008	129.228	99.367	199.599	75.628	69.591	66.393
	SHADM(12/255)	171.566	102.679	101.344	85.656	84.769	207.082	150.253	299.730	122.777	121.321	100.685
	SHADM(16/255)	206.972	136.687	136.914	109.452	117.673	261.325	210.033	377.833	170.798	174.983	130.807
PnP Diffusion	Clean	185.665	85.443	114.733	118.213	97.450	87.822	123.413	226.291	123.184	111.180	89.805
	SHADM(8/255)	273.213	212.950	226.768	237.586	187.521	379.459	357.412	527.981	279.977	214.644	172.378
	SHADM(12/255)	288.560	234.496	298.727	266.320	237.594	384.730	342.956	503.022	351.387	233.113	192.636
	SHADM(16/255)	291.203	208.819	343.184	262.064	237.195	389.466	349.241	510.567	382.053	256.274	204.433
MasaCtrl	Clean	120.710	47.995	81.714	85.089	65.096	87.137	125.835	153.795	89.821	80.492	66.788
	SHADM(8/255)	145.233	77.716	92.804	103.228	74.799	124.249	179.966	293.133	124.662	105.839	89.084
	SHADM(12/255)	168.650	97.630	120.329	125.018	90.755	192.655	211.016	308.499	148.029	125.866	106.374
	SHADM(16/255)	195.768	116.491	128.119	143.995	108.898	250.143	244.127	357.328	205.511	152.262	125.302
AnimateDiff	Clean	105.251	65.466	84.859	91.921	88.644	97.621	129.947	118.211	103.835	141.045	73.020
	SHADM(8/255)	138.352	97.119	122.730	106.896	108.595	193.241	183.102	189.953	130.934	150.565	94.207
	SHADM(12/255)	160.915	96.441	140.460	110.988	146.042	232.002	203.489	254.846	148.966	176.757	108.632
	SHADM(16/255)	180.023	109.622	158.202	109.310	173.787	268.409	221.271	300.516	171.127	195.808	120.918
PixArt	Clean	99.757	52.131	62.823	55.969	55.283	78.770	71.323	126.575	68.372	69.699	56.550
	SHADM(8/255)	146.138	63.939	91.992	68.180	79.441	134.446	132.458	249.232	115.362	104.496	84.688
	SHADM(12/255)	160.913	71.162	106.211	84.446	88.163	173.432	178.858	299.476	145.179	124.000	98.860
	SHADM(16/255)	181.516	80.481	115.335	89.820	101.138	193.174	222.049	360.543	163.272	148.350	112.683

A.3 Additional Experiments

A.3.1 Visual Effects

We provide visual effects for the 14 SOTA DMs under the SHADM attack in Figures A1∼A14, which include various kinds of distortions. For example, there are blurs (Figure A13), feature fragments (Figure A4), and incorrect count (Figure A6). More importantly, the NSFW content emerges frequently, for example, in Figures A7, A9, and A11. It reveals a serious threat to the existing architectures and mechanisms of DMs.

A.3.2 Ablation Experiments

Since the attention mechanism has three kinds of feature maps: query (Q), key (K), and value (V), we further examine whether the key and value are sufficient and necessary in our task. Table A6 shows the ablation experiments of SHADM with four combinations of features: KV, QKV, K, and V. Both KV and QKV versions achieve very close performance, which is better than the K or V version. Since the KV version saves about 30% of the parameters compared with the QKV version, the former is preferable to the latter and used as the default version for SHADM.

Table A6: Ablation experiments of SHADM with query, key, and value features on DragLoRA.

Variant	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
		LPIPS										
SHADM-KV	SHADM(8/255)	0.181	0.211	0.160	0.179	0.202	0.250	0.151	0.163	0.159	0.220	0.189
	SHADM(12/255)	0.244	0.254	0.192	0.211	0.288	0.335	0.194	0.262	0.205	0.286	0.247
	SHADM(16/255)	0.289	0.297	0.231	0.244	0.362	0.421	0.261	0.345	0.268	0.342	0.301
SHADM-QKV	SHADM(8/255)	0.179	0.202	0.164	0.175	0.200	0.229	0.153	0.171	0.159	0.219	0.187
	SHADM(12/255)	0.239	0.258	0.200	0.209	0.284	0.352	0.203	0.257	0.207	0.285	0.248
	SHADM(16/255)	0.290	0.297	0.234	0.243	0.356	0.421	0.253	0.338	0.257	0.341	0.299
SHADM-K	SHADM(8/255)	0.181	0.203	0.159	0.175	0.206	0.260	0.154	0.157	0.155	0.224	0.189
	SHADM(12/255)	0.235	0.256	0.196	0.207	0.283	0.355	0.193	0.259	0.213	0.285	0.246
	SHADM(16/255)	0.299	0.289	0.226	0.241	0.361	0.422	0.266	0.346	0.262	0.336	0.300
SHADM-V	SHADM(8/255)	0.170	0.207	0.160	0.173	0.199	0.248	0.147	0.159	0.154	0.212	0.184
	SHADM(12/255)	0.220	0.246	0.183	0.204	0.278	0.323	0.197	0.242	0.192	0.271	0.234
	SHADM(16/255)	0.275	0.290	0.210	0.237	0.332	0.376	0.237	0.318	0.237	0.320	0.280
		ASR										
SHADM-KV	SHADM(8/255)	14.29	4.55	0.00	4.55	15.00	30.00	0.00	10.00	5.00	11.43	9.27
	SHADM(12/255)	34.29	9.09	19.05	18.18	40.00	80.00	0.00	30.00	15.00	22.86	25.37
	SHADM(16/255)	37.14	22.73	14.29	18.18	55.00	90.00	0.00	70.00	20.00	34.29	33.17
SHADM-QKV	SHADM(8/255)	14.29	9.09	0.00	4.55	5.00	30.00	10.00	0.00	5.00	8.57	8.29
	SHADM(12/255)	34.29	13.64	9.52	9.09	35.00	90.00	10.00	30.00	15.00	22.86	24.39
	SHADM(16/255)	40.00	18.18	14.29	13.64	50.00	80.00	10.00	50.00	25.00	45.71	33.66
SHADM-K	SHADM(8/255)	11.43	9.09	0.00	4.55	5.00	20.00	0.00	0.00	5.00	11.43	7.32
	SHADM(12/255)	22.86	9.09	19.05	13.64	35.00	80.00	10.00	20.00	15.00	20.00	21.95
	SHADM(16/255)	40.00	18.18	14.29	13.64	40.00	90.00	10.00	50.00	25.00	34.29	31.22
SHADM-V	SHADM(8/255)	11.43	4.55	0.00	4.55	5.00	20.00	0.00	10.00	0.00	5.71	5.85
	SHADM(12/255)	31.43	9.09	9.52	13.64	40.00	90.00	10.00	40.00	5.00	17.14	22.93
	SHADM(16/255)	37.14	18.18	9.52	22.73	45.00	90.00	10.00	60.00	25.00	34.29	32.20
		(1 − **CLIP**)										
SHADM-KV	SHADM(8/255)	0.066	0.102	0.091	0.086	0.073	0.094	0.061	0.068	0.070	0.076	0.079
	SHADM(12/255)	0.138	0.145	0.137	0.110	0.141	0.181	0.101	0.125	0.125	0.141	0.135
	SHADM(16/255)	0.178	0.204	0.164	0.157	0.209	0.265	0.123	0.189	0.166	0.172	0.180
SHADM-QKV	SHADM(8/255)	0.077	0.097	0.087	0.083	0.072	0.095	0.064	0.066	0.073	0.081	0.080
	SHADM(12/255)	0.139	0.149	0.127	0.118	0.149	0.205	0.091	0.108	0.123	0.134	0.135
	SHADM(16/255)	0.177	0.192	0.166	0.150	0.212	0.268	0.122	0.175	0.156	0.180	0.178
SHADM-K	SHADM(8/255)	0.065	0.097	0.090	0.086	0.068	0.088	0.063	0.062	0.076	0.083	0.079
	SHADM(12/255)	0.126	0.156	0.126	0.114	0.134	0.191	0.094	0.109	0.118	0.120	0.128
	SHADM(16/255)	0.170	0.186	0.162	0.142	0.193	0.248	0.123	0.174	0.178	0.173	0.173
SHADM-V	SHADM(8/255)	0.060	0.097	0.085	0.078	0.068	0.101	0.070	0.060	0.059	0.073	0.074
	SHADM(12/255)	0.120	0.145	0.115	0.112	0.120	0.152	0.097	0.100	0.102	0.117	0.118
	SHADM(16/255)	0.164	0.188	0.159	0.146	0.193	0.224	0.126	0.152	0.148	0.168	0.167
		FID										
SHADM-KV	SHADM(8/255)	148.213	79.058	101.647	75.278	74.960	144.330	114.074	213.381	96.134	115.660	84.613
	SHADM(12/255)	203.381	108.415	124.598	99.037	135.514	267.320	152.017	319.290	139.424	182.615	119.518
	SHADM(16/255)	229.923	141.820	153.385	118.675	202.308	330.031	220.807	390.366	181.700	210.079	144.422
SHADM-QKV	SHADM(8/255)	151.625	74.004	103.583	90.023	77.669	142.903	112.235	202.541	97.392	130.54	87.024
	SHADM(12/255)	205.591	101.005	132.76	95.663	145.272	285.439	149.816	305.223	138.171	175.148	118.833
	SHADM(16/255)	232.207	141.777	164.197	121.065	197.622	341.207	209.524	391.098	185.748	209.529	146.973
SHADM-K	SHADM(8/255)	140.681	80.895	98.639	84.647	72.209	151.090	116.000	202.149	96.682	128.000	85.541
	SHADM(12/255)	197.782	117.880	122.343	92.115	127.937	261.140	157.118	310.366	132.472	157.628	115.192
	SHADM(16/255)	228.814	134.161	151.801	109.781	213.660	349.340	204.464	391.736	170.826	208.002	142.788
SHADM-V	SHADM(8/255)	145.614	75.418	105.403	75.531	65.149	140.917	112.490	190.272	78.232	123.427	81.908
	SHADM(12/255)	195.200	108.742	131.011	100.343	109.518	218.133	161.449	292.795	107.151	158.051	109.551
	SHADM(16/255)	226.963	138.669	143.895	115.162	184.003	306.070	196.411	369.78	151.465	205.346	137.452



Figure A1: DragLoRA attacked by SHADM(16/255).

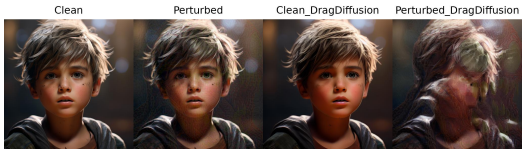


Figure A2: DragDiffusion attacked by SHADM(16/255).



Figure A3: DragNoise attacked by SHADM(16/255).



Figure A4: GoodDrag attacked by SHADM(16/255).

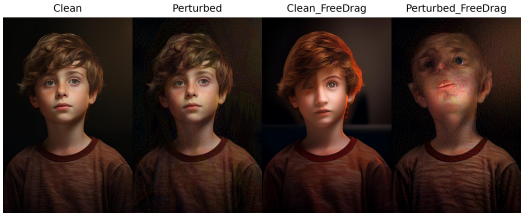


Figure A5: FreeDrag attacked by SHADM(16/255).

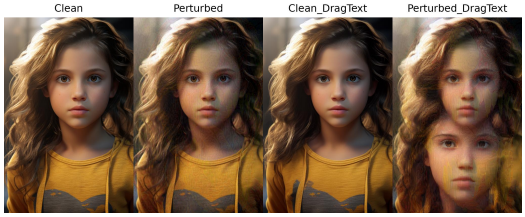


Figure A6: DragText attacked by SHADM(16/255).

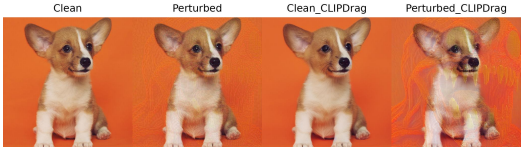


Figure A7: CLIPDrag attacked by SHADM(16/255).



Figure A8: InstantDrag attacked by SHADM(16/255).

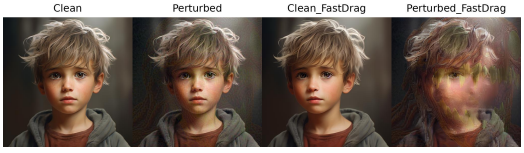


Figure A9: FastDrag attacked by SHADM(16/255).

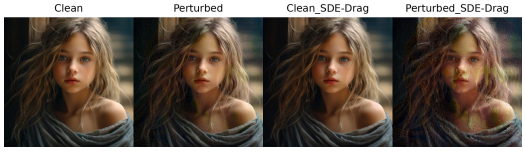


Figure A10: SDE-Drag attacked by SHADM(16/255).

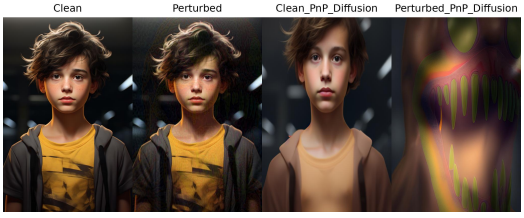


Figure A11: PnP-Diffusion attacked by SHADM(16/255).

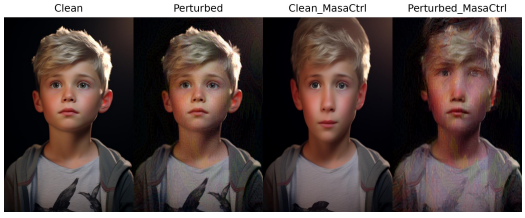


Figure A12: MasaCtrl attacked by SHADM(16/255).



Figure A13: AnimateDiff attacked by SHADM(16/255).

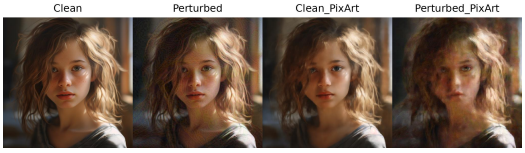


Figure A14: PixArt attacked by SHADM(16/255).

A.3.3 COMPARISONS WITH TWO BASELINES

We compare SHADM with two state-of-the-art black-box attacks: Mist (Liang & Wu, 2023) and Glaze (Shan et al., 2023), in Tables A7, A8, A9, and A10, with perturbation budget $\epsilon = 16/255$. Mist is an adversarial attack method designed to protect the intellectual property and copyright of human-created artworks by preventing DMs from accurately imitating or mimicking their style and content. Glaze is a style cloaking technique that protects individual artists by applying imperceptible adversarial perturbations to their digital artworks before they are published online. Both Mist and Glaze require no less resource than SHADM to conduct an attack, nor can they hijack key and value feature maps to NSFW content. Results show that they perform worse than SHADM in most cases, especially with lower ASRs. Therefore, they are not so effective as SHADM in targeting NSFW content.

Table A7: LPIPS of different diffusion models on DragBench under the SHADM, Mist, and Glaze attacks (perturbation budget $\epsilon = 16/255$).

Target	**Method**	**Categories**										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	0.122	0.156	0.119	0.131	0.121	0.141	0.115	0.044	0.117	0.149	0.127
	Mist	0.228	0.247	0.191	0.199	0.314	0.350	0.192	0.294	0.203	0.273	0.234
	Glaze	0.154	0.203	0.155	0.177	0.201	0.205	0.138	0.123	0.159	0.214	0.177
	SHADM	0.289	0.297	0.231	0.244	0.362	0.421	0.261	0.345	0.268	0.342	0.301
DragDiffusion	Clean	0.119	0.122	0.113	0.112	0.134	0.160	0.111	0.040	0.097	0.129	0.117
	Mist	0.253	0.225	0.224	0.269	0.263	0.278	0.202	0.198	0.200	0.239	0.235
	Glaze	0.164	0.178	0.143	0.149	0.195	0.263	0.143	0.147	0.155	0.197	0.172
	SHADM	0.272	0.287	0.220	0.234	0.340	0.376	0.260	0.351	0.258	0.330	0.288
DragNoise	Clean	0.113	0.116	0.119	0.123	0.105	0.113	0.081	0.060	0.111	0.099	0.107
	Mist	0.246	0.212	0.215	0.251	0.243	0.217	0.156	0.199	0.204	0.200	0.219
	Glaze	0.128	0.151	0.138	0.145	0.132	0.176	0.073	0.124	0.135	0.142	0.136
	SHADM	0.252	0.264	0.221	0.228	0.299	0.322	0.185	0.333	0.253	0.270	0.259
GoodDrag	Clean	0.206	0.249	0.209	0.208	0.240	0.237	0.241	0.148	0.168	0.213	0.212
	Mist	0.278	0.321	0.261	0.316	0.292	0.318	0.343	0.365	0.265	0.308	0.300
	Glaze	0.224	0.239	0.204	0.217	0.238	0.309	0.235	0.230	0.209	0.229	0.229
	SHADM	0.331	0.349	0.309	0.294	0.413	0.421	0.347	0.393	0.328	0.387	0.352
FreeDrag	Clean	0.250	0.209	0.226	0.217	0.239	0.277	0.219	0.163	0.231	0.232	0.230
	Mist	0.304	0.331	0.305	0.315	0.323	0.484	0.327	0.334	0.307	0.320	0.320
	Glaze	0.374	0.378	0.348	0.329	0.414	0.691	0.361	0.439	0.387	0.368	0.391
	SHADM	0.408	0.463	0.412	0.425	0.465	0.714	0.411	0.428	0.457	0.467	0.452
DragText	Clean	0.116	0.123	0.111	0.113	0.133	0.160	0.111	0.040	0.098	0.129	0.116
	Mist	0.252	0.226	0.224	0.273	0.259	0.278	0.202	0.197	0.201	0.239	0.238
	Glaze	0.164	0.178	0.143	0.149	0.195	0.263	0.143	0.148	0.156	0.197	0.172
	SHADM	0.272	0.287	0.221	0.230	0.340	0.371	0.257	0.351	0.257	0.329	0.287
CLIPDrag	Clean	0.100	0.100	0.098	0.089	0.095	0.088	0.078	0.043	0.084	0.080	0.089
	Mist	0.227	0.219	0.206	0.252	0.242	0.228	0.182	0.227	0.206	0.202	0.220
	Glaze	0.140	0.161	0.119	0.128	0.169	0.207	0.121	0.143	0.141	0.142	0.145
	SHADM	0.252	0.273	0.199	0.212	0.304	0.338	0.213	0.341	0.245	0.286	0.261
InstantDrag	Clean	0.119	0.106	0.080	0.099	0.106	0.150	0.149	0.055	0.083	0.130	0.109
	Mist	0.262	0.232	0.213	0.272	0.252	0.251	0.227	0.210	0.203	0.234	0.234
	Glaze	0.169	0.172	0.126	0.155	0.176	0.234	0.194	0.139	0.145	0.218	0.172
	SHADM	0.300	0.295	0.232	0.260	0.333	0.378	0.309	0.357	0.285	0.350	0.306
FastDrag	Clean	0.130	0.122	0.118	0.117	0.131	0.162	0.119	0.040	0.097	0.140	0.122
	Mist	0.237	0.249	0.195	0.201	0.307	0.340	0.211	0.282	0.215	0.284	0.252
	Glaze	0.160	0.170	0.138	0.151	0.189	0.239	0.139	0.113	0.151	0.193	0.166
	SHADM	0.279	0.273	0.210	0.222	0.334	0.387	0.232	0.305	0.234	0.317	0.277
SDE-Drag	Clean	0.035	0.050	0.053	0.055	0.026	0.011	0.011	0.014	0.030	0.023	0.034
	Mist	0.157	0.160	0.146	0.197	0.174	0.164	0.099	0.153	0.147	0.139	0.153
	Glaze	0.082	0.098	0.078	0.093	0.099	0.095	0.040	0.074	0.082	0.084	0.084
	SHADM	0.220	0.231	0.170	0.177	0.261	0.294	0.162	0.292	0.202	0.238	0.221
PnP Diffusion	Clean	0.328	0.289	0.380	0.392	0.260	0.157	0.386	0.321	0.356	0.274	0.317
	Mist	0.511	0.448	0.519	0.518	0.348	0.589	0.484	0.458	0.542	0.408	0.473
	Glaze	0.419	0.314	0.386	0.413	0.312	0.653	0.574	0.576	0.526	0.352	0.435
	SHADM	0.562	0.485	0.578	0.550	0.448	0.634	0.658	0.593	0.641	0.531	0.555
MasaCtrl	Clean	0.196	0.155	0.158	0.190	0.154	0.209	0.232	0.159	0.206	0.122	0.172
	Mist	0.233	0.286	0.185	0.265	0.241	0.248	0.249	0.289	0.248	0.257	0.250
	Glaze	0.212	0.220	0.171	0.222	0.224	0.269	0.225	0.191	0.214	0.176	0.207
	SHADM	0.271	0.338	0.224	0.290	0.275	0.299	0.280	0.313	0.290	0.272	0.282
AnimateDiff	Clean	0.188	0.190	0.146	0.180	0.173	0.229	0.262	0.175	0.192	0.275	0.202
	Mist	0.199	0.295	0.197	0.202	0.237	0.243	0.256	0.282	0.252	0.260	0.238
	Glaze	0.206	0.253	0.214	0.217	0.216	0.278	0.244	0.226	0.256	0.276	0.235
	SHADM	0.311	0.341	0.281	0.271	0.357	0.433	0.345	0.394	0.331	0.376	0.336
PixArt	Clean	0.120	0.101	0.094	0.105	0.098	0.067	0.065	0.076	0.064	0.080	0.092
	Mist	0.176	0.158	0.150	0.184	0.174	0.177	0.164	0.191	0.150	0.151	0.168
	Glaze	0.121	0.095	0.091	0.103	0.102	0.069	0.066	0.081	0.069	0.080	0.092
	SHADM	0.202	0.174	0.163	0.174	0.181	0.202	0.166	0.216	0.151	0.150	0.175

Table A8: $\mathfrak{d}_{CLIP} = (1 - \text{CLIP})$ of different diffusion models on DragBench under the SHADM, Mist, and Glaze attacks (perturbation budget $\epsilon = 16/255$).

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	0.029	0.031	0.032	0.030	0.021	0.033	0.032	0.019	0.018	0.030	0.028
	Mist	0.063	0.189	0.136	0.116	0.175	0.227	0.101	0.158	0.141	0.148	0.167
	Glaze	0.045	0.055	0.069	0.059	0.061	0.101	0.067	0.058	0.049	0.061	0.059
	SHADM	0.178	0.204	0.164	0.157	0.209	0.265	0.123	0.189	0.166	0.172	0.180
DragDiffusion	Clean	0.027	0.021	0.027	0.023	0.020	0.029	0.026	0.013	0.018	0.020	0.023
	Mist	0.120	0.195	0.153	0.165	0.146	0.169	0.152	0.131	0.127	0.135	0.147
	Glaze	0.048	0.051	0.063	0.051	0.053	0.093	0.062	0.062	0.049	0.055	0.055
	SHADM	0.146	0.190	0.155	0.139	0.190	0.219	0.140	0.158	0.165	0.163	0.164
DragNoise	Clean	0.037	0.024	0.035	0.033	0.029	0.029	0.027	0.021	0.020	0.024	0.028
	Mist	0.127	0197	0.162	0.171	0.160	0.145	0.127	0.139	0.147	0.143	0.153
	Glaze	0.049	0.058	0.065	0.055	0.059	0.093	0.059	0.063	0.049	0.048	0.056
	SHADM	0.145	0.197	0.176	0.148	0.160	0.191	0.113	0.166	0.170	0.154	0.161
GoodDrag	Clean	0.035	0.035	0.043	0.039	0.027	0.032	0.037	0.030	0.023	0.030	0.033
	Mist	0.120	0.200	0.152	0.170	0.146	0.169	0.152	0.135	0.120	0.142	0.148
	Glaze	0.049	0.051	0.066	0.051	0.067	0.097	0.076	0.076	0.048	0.055	0.059
	SHADM	0.172	0.195	0.178	0.151	0.228	0.240	0.139	0.152	0.172	0.167	0.178
FreeDrag	Clean	0.117	0.058	0.104	0.090	0.070	0.149	0.130	0.078	0.053	0.100	0.093
	Mist	0.171	0.212	0.181	0.179	0.125	0.211	0.241	0.144	0.081	0.201	0.173
	Glaze	0.137	0.104	0.148	0.127	0.149	0.188	0.113	0.128	0.107	0.126	0.131
	SHADM	0.210	0.240	0.294	0.262	0.210	0.327	0.309	0.259	0.258	0.256	0.253
DragText	Clean	0.026	0.021	0.026	0.023	0.021	0.028	0.026	0.013	0.018	0.020	0.022
	Mist	0.120	0.196	0.153	0.167	0.145	0.169	0.152	0.131	0.129	0.134	0.147
	Glaze	0.048	0.051	0.064	0.052	0.052	0.093	0.062	0.063	0.049	0.055	0.056
	SHADM	0.146	0.191	0.155	0.139	0.190	0.213	0.138	0.158	0.165	0.162	0.163
CLIPDrag	Clean	0.023	0.018	0.025	0.021	0.015	0.016	0.017	0.016	0.015	0.014	0.019
	Mist	0.113	0.192	0.147	0.165	0.136	0.162	0.131	0.131	0.122	0.129	0.140
	Glaze	0.045	0.047	0.061	0.045	0.057	0.086	0.056	0.058	0.049	0.045	0.052
	SHADM	0.134	0.188	0.154	0.138	0.157	0.192	0.114	0.151	0.160	0.142	0.151
InstantDrag	Clean	0.031	0.030	0.029	0.029	0.024	0.038	0.043	0.020	0.023	0.034	0.030
	Mist	0.099	0.139	0.132	0.118	0.124	0.162	0.130	0.119	0.149	0.146	0.127
	Glaze	0.044	0.058	0.068	0.060	0.067	0.113	0.077	0.060	0.056	0.081	0.065
	SHADM	0.155	0.228	0.189	0.185	0.186	0.176	0.154	0.161	0.211	0.187	0.185
FastDrag	Clean	0.033	0.021	0.032	0.027	0.018	0.020	0.021	0.018	0.016	0.032	0.026
	Mist	0.112	0.145	0.112	0.102	0.139	0.214	0.106	0.142	0.130	0.101	0.111
	Glaze	0.052	0.051	0.062	0.053	0.061	0.091	0.045	0.058	0.054	0.057	0.057
	SHADM	0.151	0.189	0.153	0.138	0.173	0.279	0.123	0.160	0.153	0.148	0.161
SDE-Drag	Clean	0.015	0.012	0.018	0.016	0.006	0.004	0.003	0.008	0.010	0.007	0.011
	Mist	0.091	0.195	0.147	0.164	0.122	0.134	0.112	0.105	0.120	0.118	0.130
	Glaze	0.034	0.052	0.057	0.044	0.053	0.077	0.046	0.050	0.045	0.044	0.047
	SHADM	0.114	0.188	0.148	0.138	0.138	0.135	0.105	0.129	0.157	0.125	0.138
PnP Diffusion	Clean	0.133	0.082	0.091	0.098	0.070	0.084	0.103	0.146	0.068	0.070	0.093
	Mist	0.249	0.252	0.200	0.294	0.118	0.325	0.431	0.238	0.186	0.201	0.234
	Glaze	0.153	0.122	0.131	0.161	0.085	0.179	0.320	0.211	0.124	0.077	0.138
	SHADM	0.345	0.259	0.347	0.293	0.265	0.435	0.464	0.424	0.386	0.275	0.319
MasaCtrl	Clean	0.073	0.038	0.056	0.054	0.031	0.058	0.092	0.092	0.044	0.030	0.052
	Mist	0.095	0.140	0.148	0.149	0.105	0.107	0.143	0.134	0.155	0.104	0.125
	Glaze	0.084	0.066	0.087	0.075	0.077	0.124	0.116	0.138	0.075	0.059	0.082
	SHADM	0.117	0.183	0.158	0.154	0.120	0.170	0.160	0.170	0.194	0.120	0.148
AnimateDiff	Clean	0.069	0.062	0.073	0.081	0.063	0.078	0.139	0.074	0.065	0.126	0.083
	Mist	0.112	0.153	0.134	0.103	0.135	0.123	0.135	0.095	0.114	0.127	0.124
	Glaze	0.075	0.109	0.120	0.090	0.074	0.120	0.141	0.074	0.092	0.105	0.097
	SHADM	0.140	0.169	0.175	0.137	0.171	0.207	0.177	0.155	0.142	0.185	0.163
PixArt	Clean	0.048	0.045	0.045	0.043	0.044	0.040	0.031	0.036	0.026	0.037	0.041
	Mist	0.049	0.045	0.046	0.041	0.042	0.035	0.028	0.038	0.028	0.038	0.041
	Glaze	0.050	0.046	0.040	0.044	0.039	0.038	0.028	0.041	0.029	0.039	0.041
	SHADM	0.104	0.150	0.158	0.125	0.120	0.152	0.130	0.173	0.166	0.130	0.136

Table A9: FID of different diffusion models on DragBench under the SHADM, Mist, and Glaze attacks (perturbation budget $\epsilon = 16/255$).

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	71.530	43.222	61.365	57.006	36.046	55.076	46.115	51.040	39.663	75.700	43.831
	Mist	179.427	96.546	124.076	89.535	159.993	262.858	172.655	326.251	138.118	156.494	109.16
	Glaze	87.111	50.316	73.063	58.165	49.720	107.288	70.618	101.594	58.371	91.339	57.132
	SHADM	229.923	141.820	153.385	118.675	202.308	330.031	220.807	390.366	181.700	210.079	144.422
DragDiffusion	Clean	65.977	33.763	56.531	43.622	39.274	50.815	38.953	48.354	37.702	63.296	39.434
	Mist	245.513	129.737	168.649	149.52	182.995	265.928	235.025	344.935	162.141	168.751	134.435
	Glaze	81.995	45.384	58.098	47.590	45.754	101.764	59.819	101.537	55.149	81.112	52.129
	SHADM	219.491	139.951	151.428	116.236	175.181	310.128	213.628	371.320	187.860	215.116	142.379
DragNoise	Clean	73.854	41.623	63.269	53.367	51.520	54.282	42.622	62.798	50.286	64.339	44.657
	Mist	242.348	127.275	163.249	145.749	192.649	263.547	207.362	368.488	170.053	165.805	133.201
	Glaze	86.678	46.271	70.482	59.324	43.448	99.312	57.207	103.934	57.903	65.864	53.015
	SHADM	218.337	130.325	171.257	124.035	152.574	267.670	208.409	377.328	186.760	199.010	140.205
GoodDrag	Clean	82.768	42.006	77.501	60.544	48.415	65.671	85.240	92.017	47.536	72.091	50.778
	Mist	239.986	144.469	148.320	156.396	187.994	265.812	266.393	361.072	150.510	163.471	133.597
	Glaze	87.839	45.283	65.968	58.602	43.831	104.246	68.063	127.759	59.372	78.505	55.119
	SHADM	237.812	148.814	170.452	132.168	219.554	325.609	235.162	400.453	204.483	223.948	154.281
FreeDrag	Clean	175.883	73.583	135.046	136.728	131.796	137.395	140.618	178.452	105.156	155.939	98.482
	Mist	259.354	288.040	218.401	225.970	212.719	221.795	286.555	364.557	160.617	276.733	165.600
	Glaze	189.565	100.693	146.031	127.367	157.660	175.865	141.879	205.302	126.152	172.938	108.622
	SHADM	246.706	201.994	257.432	236.584	217.628	401.136	312.503	428.261	256.931	266.746	175.186
DragText	Clean	65.345	33.578	56.123	43.835	39.270	50.848	39.428	48.215	38.827	63.677	39.423
	Mist	245.323	130.319	169.675	151.805	174.593	267.289	235.187	347.718	166.865	168.869	134.249
	Glaze	82.507	45.594	56.173	48.388	46.413	101.567	59.711	102.315	55.028	79.572	52.003
	SHADM	220.574	141.283	151.079	113.900	173.624	306.359	210.602	372.015	187.436	211.304	141.809
CLIPDrag	Clean	55.254	31.178	48.087	31.538	30.246	42.580	33.130	44.359	34.902	37.476	31.411
	Mist	244.189	131.208	155.634	140.318	159.251	265.885	247.118	340.452	147.256	153.372	129.128
	Glaze	78.373	42.883	53.234	40.380	43.081	92.054	61.552	101.072	51.340	56.233	46.519
	SHADM	212.108	142.371	143.270	106.051	145.479	293.433	203.567	357.176	184.772	196.881	136.160
InstantDrag	Clean	64.219	43.709	38.737	45.955	48.501	49.020	46.660	53.594	42.923	68.259	39.976
	Mist	227.093	111.664	122.224	116.848	159.189	270.532	177.396	366.437	149.830	191.446	130.244
	Glaze	77.071	52.009	57.791	61.045	53.696	103.925	75.052	104.803	58.745	93.629	55.158
	SHADM	237.402	186.137	195.112	178.765	188.579	276.077	270.476	445.263	241.861	250.815	169.333
FastDrag	Clean	62.783	34.693	53.363	53.589	31.380	49.163	37.121	47.516	40.478	66.883	39.721
	Mist	180.640	100.637	119.564	98.108	145.812	161.776	188.839	199.503	120.612	130.831	112.940
	Glaze	86.646	45.713	53.272	58.020	47.987	97.849	61.170	89.965	62.336	80.729	53.135
	SHADM	226.578	122.706	129.792	115.369	190.807	343.780	226.386	386.382	172.624	189.120	139.613
SDE-Drag	Clean	32.946	19.910	27.210	22.362	14.688	12.137	10.295	23.192	17.837	16.901	16.952
	Mist	235.952	122.938	145.746	142.438	128.958	253.964	233.801	360.432	157.660	146.118	126.098
	Glaze	56.917	36.025	40.067	35.926	27.403	81.878	51.177	85.779	40.403	37.667	36.125
	SHADM	206.972	136.687	136.914	109.452	117.673	261.325	210.033	377.833	170.798	174.983	130.807
PnP Diffusion	Clean	185.665	85.443	114.733	118.213	97.450	87.822	123.413	226.291	123.184	111.180	89.805
	Mist	266.601	225.748	216.226	252.637	151.364	310.259	408.412	339.511	226.274	233.609	162.947
	Glaze	213.574	103.191	137.932	170.405	114.824	197.468	321.813	344.040	158.758	117.839	114.858
	SHADM	291.203	208.819	343.184	262.064	237.195	389.466	349.241	510.567	382.053	256.274	204.433
MasaCtrl	Clean	120.710	47.995	81.714	85.089	65.096	87.137	125.835	153.795	89.821	80.492	66.788
	Mist	146.128	83.627	102.859	129.447	99.282	167.978	231.434	305.867	164.527	147.902	111.694
	Glaze	137.473	59.685	83.137	94.590	78.828	137.810	141.130	197.306	98.345	84.509	76.606
	SHADM	195.768	116.491	128.119	143.995	108.898	250.143	244.127	357.328	205.511	152.262	125.302
AnimateDiff	Clean	105.251	65.466	84.859	91.921	88.644	97.621	129.947	118.211	103.835	141.045	73.020
	Mist	179.877	165.723	169.437	125.122	196.783	176.251	236.133	277.886	190.895	171.916	108.382
	Glaze	108.859	82.118	101.659	92.159	78.287	150.984	145.441	129.113	105.487	124.605	76.020
	SHADM	180.023	109.622	158.202	109.310	173.787	268.409	221.271	300.516	171.127	195.808	120.918
PixArt	Clean	99.757	52.131	62.823	55.969	55.283	78.770	71.323	126.575	68.372	69.699	56.550
	Mist	112.320	52.084	63.249	55.575	54.076	75.942	73.135	124.540	72.167	71.263	75.430
	Glaze	108.907	54.651	61.474	55.511	53.439	69.199	73.135	143.689	71.372	73.986	58.752
	SHADM	181.516	80.481	115.335	89.820	101.138	193.174	222.049	360.543	163.272	148.350	112.683

Table A10: ASR (%) of different diffusion models on DragBench under the SHADM, Mist, and Glaze attacks (perturbation budget $\epsilon = 16/255$).

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	2.86	0.00	4.76	0.00	0.00	10.00	0.00	0.00	0.00	0.00	1.46
	Mist	20.0	0.0	4.76	4.55	5.0	70.0	80.0	60.0	0.0	2.86	15.61
	Glaze	5.71	0.0	0.0	0.0	0.0	10.0	0.0	10.0	0.0	2.86	2.44
	SHADM	37.14	22.73	14.29	18.18	55.00	90.00	0.00	70.00	20.00	34.29	33.17
DragDiffusion	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	Mist	22.86	0.00	4.76	4.55	15.00	70.00	50.00	50.00	5.00	5.71	16.10
	Glaze	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	2.86	1.46
	SHADM	37.14	22.73	14.29	13.64	55.00	80.00	0.00	70.00	20.00	40.00	33.17
DragNoise	Clean	0.00	0.00	0.00	4.55	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	Mist	22.86	0.00	4.76	9.09	15.00	70.00	60.00	60.00	0.00	2.86	16.59
	Glaze	2.86	0.00	0.00	0.00	0.00	10.00	0.00	10.00	0.00	0.00	1.46
	SHADM	40.00	27.27	9.52	22.73	50.00	70.00	0.00	50.00	20.00	37.14	32.20
GoodDrag	Clean	5.71	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	1.46
	Mist	14.29	0.00	0.00	4.55	10.00	70.00	40.00	50.00	0.00	2.86	12.20
	Glaze	8.57	0.00	0.00	0.00	0.00	10.00	10.00	10.00	0.00	2.86	3.41
	SHADM	37.14	22.73	23.81	13.64	55.00	80.00	0.00	70.00	15.00	28.57	31.71
FreeDrag	Clean	5.71	0.00	0.00	0.00	0.00	30.00	10.00	0.00	0.00	2.86	3.41
	Mist	14.29	4.55	4.76	4.55	0.00	40.00	40.00	30.00	0.00	14.29	11.71
	Glaze	5.71	0.00	0.00	4.55	5.00	30.00	20.00	10.00	0.00	8.57	6.34
	SHADM	11.43	0.00	0.00	4.55	20.00	30.00	70.00	40.00	10.00	20.00	15.61
DragText	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	Mist	22.86	0.00	4.76	4.55	10.00	70.00	60.00	50.00	5.00	5.71	16.10
	Glaze	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	2.86	1.46
	SHADM	37.14	22.73	14.29	13.64	55.00	80.00	0.00	70.00	25.00	34.29	32.68
CLIPDrag	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	Mist	20.00	0.00	4.76	4.55	15.00	60.00	50.00	50.00	0.00	0.00	13.66
	Glaze	5.71	0.00	0.00	0.00	0.00	10.00	10.00	0.00	0.00	0.00	1.95
	SHADM	37.14	27.27	9.52	13.64	55.00	80.00	0.00	60.00	25.00	34.29	32.20
InstantDrag	Clean	2.86	0.00	0.00	4.55	0.00	10.00	0.00	10.00	0.00	2.86	2.44
	Mist	14.29	0.00	4.76	0.00	10.00	50.00	20.00	40.00	5.00	8.57	11.22
	Glaze	5.71	0.00	0.00	0.00	0.00	20.00	10.00	30.00	0.00	2.86	4.39
	SHADM	45.71	40.91	23.81	36.36	65.00	40.00	20.00	80.00	50.00	51.43	45.37
FastDrag	Clean	2.86	0.00	0.00	0.00	0.00	10.00	10.00	0.00	0.00	2.86	1.95
	Mist	25.71	4.55	9.52	0.0	10.00	80.00	60.00	50.00	5.0	11.43	18.54
	Glaze	2.86	0.0	0.0	4.55	0.0	10.0	10.0	0.0	0.0	2.86	2.44
	SHADM	42.86	27.27	19.05	13.64	60.00	90.00	30.00	50.00	40.00	31.43	37.07
SDE-Drag	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	Mist	17.14	0.00	0.00	4.55	5.00	70.00	40.00	60.00	0.00	2.86	12.68
	Glaze	2.86	0.00	0.00	0.00	0.00	10.00	0.00	10.00	0.00	0.00	1.46
	SHADM	34.29	22.73	4.76	13.64	45.00	60.00	0.00	60.00	20.00	28.57	27.32
PnP Diffusion	Clean	5.71	0.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	2.86	1.95
	Mist	11.43	9.09	0.00	4.55	0.00	10.00	10.00	0.00	0.00	0.00	4.39
	Glaze	5.71	0.00	0.00	4.55	0.00	20.00	40.00	70.00	0.00	2.86	8.29
	SHADM	45.71	18.18	23.81	18.18	10.00	90.00	60.00	70.00	55.00	37.14	37.56
MasaCtrl	Clean	5.71	0.00	0.00	0.00	0.00	10.00	0.00	0.00	5.00	0.00	1.95
	Mist	5.71	0.0	9.52	0.0	0.0	0.0	20.0	40.0	10.0	2.86	6.34
	Glaze	5.71	0.0	0.0	0.0	0.0	10.0	0.0	20.0	5.0	5.71	3.9
	SHADM	14.29	4.55	4.76	4.55	10.00	40.00	60.00	40.00	30.00	11.43	16.59
AnimateDiff	Clean	11.61	5.11	12.80	11.93	9.38	16.88	51.25	10.00	2.81	22.32	13.93
	Mist	18.75	13.30	29.29	22.39	16.88	45.62	31.88	30.00	31.56	16.96	21.92
	Glaze	11.96	8.81	12.50	16.48	7.81	39.38	46.25	16.25	12.19	10.60	15.03
	SHADM	24.11	14.49	19.64	11.08	26.56	71.88	46.25	23.75	10.94	22.50	23.29
PixArt	Clean	5.71	0.00	0.00	0.00	0.00	10.00	10.00	0.00	0.00	2.86	2.44
	Mist	5.71	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	1.46
	Glaze	5.71	0.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	2.86	1.95
	SHADM	2.86	0.00	0.00	0.00	0.00	30.00	10.00	10.00	0.00	8.57	4.39

A.3.4 COMPUTATIONAL COST

We conduct computational cost experiments for SHADM with an Nvidia A800-80GB GPU, shown in Table A11. The average runtime and peak GPU Memory are 93.92s and 10026.12MB per adversarial image, respectively. Such a resource cost is acceptable in real-world applications. Besides, the resource costs for different categories are similar, which also indicates the stable performance of SHADM. We also provide a quantitative comparison of the computational costs for Mist, Glaze, and SHADM on DragLoRA in Table A12. SHADM requires lower runtime and memory than Mist and Glaze.

Table A11: Computational cost of SHADM across different categories.

Category	**Avg. Runtime (s/image)**	**Avg. Peak GPU Memory (MB)**
Animals	93.94	10026.00
Art-work	93.86	10026.22
Human-head	94.03	10026.33
Building-city-view	93.88	10026.58
Land-scape	93.92	10025.85
Building-countryside-view	93.90	10026.14
Human-upper-body	93.94	10026.65
Human-full-body	93.88	10025.85
Interior-design	93.96	10026.35
Other-objects	93.94	10025.71
Overall	**93.92**	**10026.12**

Table A12: Computational cost comparisons for SHADM, Mist, and Glaze.

Method	**Avg. Runtime (s/image)**	**Avg. Peak GPU Memory (MB)**
SHADM	**93.92**	**10026.12 (peak measured)**
Mist	106.17	19043.88 (peak allocated)
Glaze	124.55	19592.93 (peak allocated)

A.3.5 ROBUSTNESS AGAINST THREE DEFENSE SCHEMES

Gaussian smoothing and JPEG compression are two fundamental defense schemes in real-world scenarios. We evaluate the robustness of SHADM against these two defense schemes in Tables A14, A15, A16, A17, A18, A19, A20, and A21. The radius of Gaussian smoothing is set to $r = 2$, and the quality of JPEG compression is set to 75%, which are common settings in the trade-off between denoising and quality. Results show that SHADM withstands these defense measures and still cause significant damage across four evaluating metrics. Hence it is robust to such countermeasures.

Moreover, we add experiments for a state-of-the-art purification method IMPRESS (Chen et al., 2025) on DragLoRA. Results in Table A13 show that in the absence of an attack, the target model maintains high visual quality (LPIPS 0.127, FID 43.831) and a very low attack success rate (ASR of 1.46% overall). Under the SHADM attack, quality degrades noticeably and the ASR increases significantly across all categories, reaching 33.17% overall (and 40.00% on living subjects). Moreover, IMPRESS fails to defend against the SHADM attack. Rather than relieving the attack, incorporating IMPRESS further degrades perceptual fidelity (LPIPS increases from 0.301 to 0.419, FID from 144.422 to 170.270) and counterproductively increases the ASR (rising to 35.73% overall and 55.00% on living subjects). The reason is that SHADM targets the deep intermediate key and value feature maps inside the up-sampling attention layers rather than relying on high-frequency pixel artifacts, thus the structural hijacking mechanism remains resilient.

A.3.6 PERFORMANCE WITH AN ALTERNATIVE NSFW TARGET

To assess the generalizability of SHADM to a different NSFW content, we use a pornographic image as the target NSFW image for SHADM to conduct attack experiments. Results in Tables A22, A23,

Table A13: Robustness of SHADM against IMPRESS.

Scope	Input/Defense	LPIPS ↑	$(1 - \text{CLIP})$ ↑	FID ↑	ASR ↑
All	Clean	0.127	0.028	43.831	1.46%
	SHADM	0.301	0.180	144.422	33.17%
	SHADM + IMPRESS	**0.419**	**0.242**	**170.270**	**35.73%**
Still-life	Clean	0.133	0.029	51.385	1.29%
	SHADM	0.285	0.174	145.249	13.55%
	SHADM + IMPRESS	**0.431**	**0.248**	**194.280**	**31.32%**
Living subjects	Clean	0.108	0.025	40.382	2.00%
	SHADM	0.350	0.199	221.404	40.00%
	SHADM + IMPRESS	**0.390**	**0.225**	**276.385**	**55.00%**

Table A14: LPIPS robustness of SHADM against Gaussian smoothing ($r = 2$) across different diffusion models on DragBench.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	0.122	0.156	0.119	0.131	0.121	0.141	0.115	0.044	0.117	0.149	0.127
	SHADM(16/255)	0.336	0.342	0.375	0.362	0.273	0.265	0.222	0.255	0.313	0.283	0.313
DragDiffusion	Clean	0.119	0.122	0.113	0.112	0.134	0.160	0.111	0.040	0.097	0.129	0.117
	SHADM(16/255)	0.339	0.330	0.372	0.354	0.287	0.310	0.244	0.260	0.314	0.282	0.316
DragNoise	Clean	0.113	0.116	0.119	0.123	0.105	0.113	0.081	0.060	0.111	0.099	0.107
	SHADM(16/255)	0.321	0.316	0.370	0.351	0.246	0.236	0.184	0.262	0.305	0.240	0.292
GoodDrag	Clean	0.206	0.249	0.209	0.208	0.240	0.237	0.241	0.148	0.168	0.213	0.212
	SHADM(16/255)	0.353	0.347	0.386	0.364	0.292	0.306	0.245	0.298	0.316	0.292	0.327
FreeDrag	Clean	0.250	0.209	0.226	0.217	0.239	0.277	0.219	0.163	0.231	0.232	0.230
	SHADM(16/255)	0.422	0.442	0.461	0.467	0.370	0.399	0.330	0.367	0.451	0.406	0.420
DragText	Clean	0.116	0.123	0.111	0.113	0.133	0.160	0.111	0.040	0.098	0.129	0.116
	SHADM(16/255)	0.340	0.330	0.375	0.353	0.288	0.310	0.245	0.261	0.314	0.282	0.317
CLIPDrag	Clean	0.100	0.100	0.098	0.089	0.095	0.088	0.078	0.043	0.084	0.080	0.089
	SHADM(16/255)	0.319	0.326	0.354	0.339	0.261	0.276	0.208	0.260	0.293	0.249	0.295
InstantDrag	Clean	0.119	0.106	0.080	0.099	0.106	0.150	0.149	0.055	0.083	0.130	0.109
	SHADM(16/255)	0.328	0.311	0.357	0.350	0.258	0.271	0.248	0.266	0.301	0.278	0.304
FastDrag	Clean	0.130	0.122	0.118	0.117	0.131	0.162	0.119	0.040	0.097	0.140	0.122
	SHADM(16/255)	0.340	0.321	0.368	0.355	0.286	0.271	0.233	0.245	0.303	0.282	0.310
SDE-Drag	Clean	0.035	0.050	0.053	0.055	0.026	0.011	0.011	0.014	0.030	0.023	0.034
	SHADM(16/255)	0.283	0.283	0.332	0.316	0.207	0.177	0.149	0.235	0.264	0.199	0.254
PnP Diffusion	Clean	0.328	0.289	0.380	0.392	0.260	0.157	0.386	0.321	0.356	0.274	0.317
	SHADM(16/255)	0.555	0.570	0.643	0.649	0.451	0.394	0.467	0.482	0.598	0.534	0.550
MasaCtrl	Clean	0.196	0.155	0.158	0.190	0.154	0.209	0.232	0.159	0.206	0.122	0.172
	SHADM(16/255)	0.380	0.381	0.408	0.410	0.293	0.308	0.354	0.354	0.399	0.275	0.356
AnimateDiff	Clean	0.188	0.190	0.146	0.180	0.173	0.229	0.262	0.175	0.192	0.275	0.202
	SHADM(16/255)	0.323	0.387	0.333	0.336	0.330	0.391	0.398	0.369	0.374	0.396	0.360
PixArt	Clean	0.120	0.101	0.094	0.105	0.098	0.067	0.065	0.076	0.064	0.080	0.092
	SHADM(16/255)	0.220	0.201	0.294	0.206	0.198	0.162	0.164	0.177	0.166	0.179	0.192

Table A15: $\mathfrak{d}_{CLIP} = (1 - \text{CLIP})$ robustness of SHADM against Gaussian smoothing ($r = 2$) across different diffusion models on DragBench.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	0.029	0.031	0.032	0.030	0.021	0.033	0.032	0.019	0.018	0.030	0.028
	SHADM(16/255)	0.097	0.110	0.121	0.116	0.074	0.107	0.095	0.093	0.096	0.092	0.100
DragDiffusion	Clean	0.027	0.021	0.027	0.023	0.020	0.029	0.026	0.013	0.018	0.020	0.023
	SHADM(16/255)	0.094	0.101	0.117	0.112	0.086	0.098	0.091	0.087	0.098	0.092	0.098
DragNoise	Clean	0.037	0.024	0.035	0.033	0.029	0.029	0.027	0.021	0.020	0.024	0.028
	SHADM(16/255)	0.099	0.115	0.126	0.113	0.073	0.106	0.081	0.094	0.099	0.087	0.100
GoodDrag	Clean	0.035	0.035	0.043	0.039	0.027	0.032	0.037	0.030	0.023	0.030	0.033
	SHADM(16/255)	0.094	0.109	0.122	0.108	0.081	0.097	0.090	0.111	0.098	0.093	0.100
FreeDrag	Clean	0.117	0.058	0.104	0.090	0.070	0.149	0.130	0.078	0.053	0.100	0.093
	SHADM(16/255)	0.207	0.225	0.194	0.199	0.177	0.300	0.232	0.213	0.191	0.205	0.208
DragText	Clean	0.026	0.021	0.026	0.023	0.021	0.028	0.026	0.013	0.018	0.020	0.022
	SHADM(16/255)	0.094	0.101	0.118	0.112	0.086	0.098	0.091	0.087	0.097	0.092	0.098
CLIPDrag	Clean	0.023	0.018	0.025	0.021	0.015	0.016	0.017	0.016	0.015	0.014	0.019
	SHADM(16/255)	0.088	0.106	0.106	0.106	0.073	0.099	0.082	0.084	0.094	0.082	0.092
InstantDrag	Clean	0.031	0.030	0.029	0.029	0.024	0.038	0.043	0.020	0.023	0.034	0.030
	SHADM(16/255)	0.086	0.120	0.121	0.130	0.094	0.111	0.094	0.093	0.107	0.100	0.105
FastDrag	Clean	0.033	0.021	0.032	0.027	0.018	0.020	0.021	0.018	0.016	0.032	0.026
	SHADM(16/255)	0.097	0.110	0.109	0.112	0.081	0.100	0.078	0.086	0.090	0.087	0.096
SDE-Drag	Clean	0.015	0.012	0.018	0.016	0.006	0.004	0.003	0.008	0.010	0.007	0.011
	SHADM(16/255)	0.081	0.110	0.106	0.103	0.067	0.085	0.074	0.076	0.092	0.072	0.087
PnP Diffusion	Clean	0.133	0.082	0.091	0.098	0.070	0.084	0.103	0.146	0.068	0.070	0.093
	SHADM(16/255)	0.295	0.284	0.329	0.298	0.202	0.290	0.246	0.298	0.293	0.265	0.281
MasaCtrl	Clean	0.073	0.038	0.056	0.054	0.031	0.058	0.092	0.092	0.044	0.030	0.052
	SHADM(16/255)	0.129	0.144	0.140	0.139	0.085	0.184	0.215	0.189	0.142	0.094	0.134
AnimateDiff	Clean	0.069	0.062	0.073	0.081	0.063	0.078	0.139	0.074	0.065	0.126	0.083
	SHADM(16/255)	0.136	0.215	0.186	0.165	0.154	0.180	0.186	0.125	0.153	0.180	0.168
PixArt	Clean	0.048	0.045	0.045	0.043	0.044	0.040	0.031	0.036	0.026	0.037	0.041
	SHADM(16/255)	0.148	0.145	0.146	0.137	0.142	0.134	0.128	0.138	0.128	0.138	0.140

Table A16: FID robustness of SHADM against Gaussian smoothing ($r = 2$) across different diffusion models on DragBench.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	71.530	43.222	61.365	57.006	36.046	55.076	46.115	51.040	39.663	75.700	43.831
	SHADM(16/255)	174.941	134.122	145.676	138.867	84.918	134.117	136.228	290.069	138.899	158.151	113.743
DragDiffusion	Clean	65.977	33.763	56.531	43.622	39.274	50.815	38.953	48.354	37.702	63.296	39.434
	SHADM(16/255)	161.621	128.493	145.452	138.780	92.679	141.233	142.042	278.819	144.530	156.269	113.370
DragNoise	Clean	73.854	41.623	63.269	53.367	51.520	54.282	42.622	62.798	50.286	64.339	44.657
	SHADM(16/255)	167.824	131.898	158.719	150.250	78.618	144.197	129.515	293.866	139.486	147.693	113.049
GoodDrag	Clean	82.768	42.006	77.501	60.544	48.415	65.671	85.240	92.017	47.536	72.091	50.778
	SHADM(16/255)	168.675	128.547	148.866	140.996	92.295	138.611	139.497	298.332	140.041	157.751	113.945
FreeDrag	Clean	175.883	73.583	135.046	136.728	131.796	137.395	140.618	178.452	105.156	155.939	98.482
	SHADM(16/255)	246.539	210.864	208.080	231.623	183.544	215.033	210.153	398.126	236.341	247.108	160.410
DragText	Clean	65.345	33.578	56.123	43.835	39.270	50.848	39.428	48.215	38.827	63.677	39.423
	SHADM(16/255)	162.998	128.551	144.832	138.171	92.801	141.338	141.194	278.613	143.794	156.582	113.437
CLIPDrag	Clean	55.254	31.178	48.087	31.538	30.246	42.580	33.130	44.359	34.902	37.476	31.411
	SHADM(16/255)	155.158	130.494	136.847	136.051	86.606	140.881	133.605	269.409	140.605	144.477	108.736
InstantDrag	Clean	64.219	43.709	38.737	45.955	48.501	49.020	46.660	53.594	42.923	68.259	39.976
	SHADM(16/255)	161.777	129.120	147.652	160.281	100.801	143.070	157.053	302.869	147.444	164.081	116.786
FastDrag	Clean	62.783	34.693	53.363	53.589	31.380	49.163	37.121	47.516	40.478	66.883	39.721
	SHADM(16/255)	163.388	122.749	133.077	140.812	96.656	144.628	135.768	283.222	142.194	147.214	110.731
SDE-Drag	Clean	32.946	19.910	27.210	22.362	14.688	12.137	10.295	23.192	17.837	16.901	16.952
	SHADM(16/255)	152.253	124.156	128.883	134.057	75.705	141.739	130.677	282.566	132.484	128.811	104.541
PnP Diffusion	Clean	185.665	85.443	114.733	118.213	97.450	87.822	123.413	226.291	123.184	111.180	89.805
	SHADM(16/255)	288.049	286.572	303.531	295.255	188.062	205.595	213.862	388.971	315.111	265.079	186.305
MasaCtrl	Clean	120.710	47.995	81.714	85.089	65.096	87.137	125.835	153.795	89.821	80.492	66.788
	SHADM(16/255)	202.859	128.501	153.747	185.232	105.023	172.299	220.431	391.703	170.282	151.662	126.478
AnimateDiff	Clean	105.251	65.466	84.859	91.921	88.644	97.621	129.947	118.211	103.835	141.045	73.020
	SHADM(16/255)	165.289	181.438	187.580	154.020	165.602	188.930	231.509	222.843	200.851	187.735	121.503
PixArt	Clean	99.757	52.131	62.823	55.969	55.283	78.770	71.323	126.575	68.372	69.699	56.550
	SHADM(16/255)	149.666	102.084	113.249	98.611	104.076	120.050	123.135	174.540	118.993	118.816	105.782

Table A17: ASR (%) robustness of SHADM against Gaussian smoothing ($r = 2$) across different diffusion models on DragBench.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	2.86	0.00	4.76	0.00	0.00	10.00	0.00	0.00	0.00	0.00	1.46
	SHADM(16/255)	11.43	0.00	9.52	4.55	0.00	20.00	0.00	0.00	0.00	5.71	5.37
DragDiffusion	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(16/255)	5.71	0.00	4.76	4.55	0.00	10.00	10.00	0.00	5.00	8.57	4.88
DragNoise	Clean	0.00	0.00	0.00	4.55	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(16/255)	8.57	0.00	4.76	4.55	0.00	10.00	0.00	0.00	5.00	5.71	4.39
GoodDrag	Clean	5.71	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	1.46
	SHADM(16/255)	14.29	0.00	4.76	4.55	0.00	20.00	0.00	10.00	0.00	11.43	6.83
FreeDrag	Clean	5.71	0.00	0.00	0.00	0.00	30.00	10.00	0.00	0.00	2.86	3.41
	SHADM(16/255)	17.14	0.00	4.76	4.55	5.00	80.00	40.00	40.00	10.00	22.86	17.07
DragText	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(16/255)	5.71	0.00	4.76	4.55	0.00	10.00	10.00	0.00	5.00	8.57	4.88
CLIPDrag	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(16/255)	8.57	0.00	4.76	4.55	0.00	10.00	0.00	0.00	0.00	11.43	4.88
InstantDrag	Clean	2.86	0.00	0.00	4.55	0.00	10.00	0.00	10.00	0.00	2.86	2.44
	SHADM(16/255)	11.43	0.00	0.00	18.18	0.00	10.00	0.00	20.00	10.00	11.43	8.29
FastDrag	Clean	2.86	0.00	0.00	0.00	0.00	10.00	10.00	0.00	0.00	2.86	1.95
	SHADM(16/255)	8.57	0.00	0.00	9.09	0.00	20.00	0.00	0.00	0.00	8.57	4.88
SDE-Drag	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(16/255)	8.57	0.00	4.76	4.55	0.00	20.00	0.00	0.00	0.00	5.71	4.39
PnP Diffusion	Clean	5.71	0.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	2.86	1.95
	SHADM(16/255)	37.14	18.18	19.05	22.73	10.00	60.00	40.00	40.00	30.00	14.29	25.85
MasaCtrl	Clean	5.71	0.00	0.00	0.00	0.00	10.00	0.00	0.00	5.00	0.00	1.95
	SHADM(16/255)	11.43	0.00	0.00	13.64	0.00	40.00	40.00	40.00	25.00	8.57	13.17
AnimateDiff	Clean	11.61	5.11	12.80	11.93	9.38	16.88	51.25	10.00	2.81	22.32	13.93
	SHADM(16/255)	15.00	17.90	30.06	28.12	13.75	42.50	20.00	10.00	34.38	15.36	21.43
PixArt	Clean	5.71	0.00	0.00	0.00	0.00	10.00	10.00	0.00	0.00	2.86	2.44
	SHADM(16/255)	5.71	0.00	0.00	0.00	10.00	10.00	0.00	0.00	10.00	0.00	3.57

Table A18: LPIPS robustness of SHADM against JPEG compression (quality=75%) across different diffusion models on DragBench.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
		LPIPS										
DragLoRA	Clean	0.122	0.156	0.119	0.131	0.121	0.141	0.115	0.044	0.117	0.149	0.127
	SHADM(16/255)	0.204	0.247	0.191	0.220	0.241	0.276	0.213	0.248	0.209	0.266	0.230
DragDiffusion	Clean	0.119	0.122	0.113	0.112	0.134	0.160	0.111	0.040	0.097	0.129	0.117
	SHADM(16/255)	0.216	0.230	0.190	0.197	0.255	0.344	0.223	0.259	0.196	0.259	0.231
DragNoise	Clean	0.113	0.116	0.119	0.123	0.105	0.113	0.081	0.060	0.111	0.099	0.107
	SHADM(16/255)	0.283	0.285	0.247	0.242	0.358	0.409	0.206	0.353	0.274	0.302	0.291
GoodDrag	Clean	0.206	0.249	0.209	0.208	0.240	0.237	0.241	0.148	0.168	0.213	0.212
	SHADM(16/255)	0.287	0.294	0.240	0.242	0.339	0.365	0.268	0.360	0.270	0.338	0.296
FreeDrag	Clean	0.250	0.209	0.226	0.217	0.239	0.277	0.219	0.163	0.231	0.232	0.230
	SHADM(16/255)	0.356	0.386	0.332	0.398	0.421	0.476	0.396	0.427	0.396	0.410	0.392
DragText	Clean	0.116	0.123	0.111	0.113	0.133	0.160	0.111	0.040	0.098	0.129	0.116
	SHADM(16/255)	0.215	0.230	0.189	0.198	0.255	0.344	0.222	0.268	0.196	0.261	0.231
CLIPDrag	Clean	0.100	0.100	0.098	0.089	0.095	0.088	0.078	0.043	0.084	0.080	0.089
	SHADM(16/255)	0.182	0.215	0.166	0.183	0.227	0.274	0.196	0.246	0.211	0.211	0.205
InstantDrag	Clean	0.119	0.106	0.080	0.099	0.106	0.150	0.149	0.055	0.083	0.130	0.109
	SHADM(16/255)	0.203	0.220	0.167	0.209	0.232	0.283	0.248	0.263	0.208	0.261	0.224
FastDrag	Clean	0.130	0.122	0.118	0.117	0.131	0.162	0.119	0.040	0.097	0.140	0.122
	SHADM(16/255)	0.208	0.220	0.178	0.196	0.244	0.306	0.203	0.234	0.191	0.242	0.218
SDE-Drag	Clean	0.035	0.050	0.053	0.055	0.026	0.011	0.011	0.014	0.030	0.023	0.034
	SHADM(16/255)	0.134	0.176	0.134	0.151	0.165	0.181	0.130	0.218	0.155	0.160	0.156
PnP Diffusion	Clean	0.328	0.289	0.380	0.392	0.260	0.157	0.386	0.321	0.356	0.274	0.317
	SHADM(16/255)	0.557	0.466	0.558	0.544	0.480	0.579	0.611	0.596	0.632	0.509	0.543
MasaCtrl	Clean	0.196	0.155	0.158	0.190	0.154	0.209	0.232	0.159	0.206	0.122	0.172
	SHADM(16/255)	0.246	0.274	0.216	0.266	0.241	0.286	0.287	0.325	0.267	0.220	0.253
AnimateDiff	Clean	0.188	0.190	0.146	0.180	0.173	0.229	0.262	0.175	0.192	0.275	0.202
	SHADM(16/255)	0.240	0.281	0.240	0.243	0.268	0.344	0.338	0.333	0.294	0.298	0.277
PixArt	Clean	0.120	0.101	0.094	0.105	0.098	0.067	0.065	0.076	0.064	0.080	0.092
	SHADM(16/255)	0.220	0.201	0.194	0.206	0.198	0.162	0.164	0.177	0.166	0.179	0.192

Table A19: $\mathfrak{d}_{CLIP} = (1 - \text{CLIP})$ robustness of SHADM against JPEG compression (quality=75%) across different diffusion models on DragBench.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
		$\mathfrak{d}_{CLIP} = (1 - \text{CLIP})$										
DragLoRA	Clean	0.029	0.031	0.032	0.030	0.021	0.033	0.032	0.019	0.018	0.030	0.028
	SHADM(16/255)	0.075	0.136	0.131	0.127	0.095	0.110	0.121	0.113	0.114	0.107	0.110
DragDiffusion	Clean	0.027	0.021	0.027	0.023	0.020	0.029	0.026	0.013	0.018	0.020	0.023
	SHADM(16/255)	0.076	0.128	0.126	0.121	0.103	0.114	0.125	0.110	0.113	0.118	0.111
DragNoise	Clean	0.037	0.024	0.035	0.033	0.029	0.029	0.027	0.021	0.020	0.024	0.028
	SHADM(16/255)	0.138	0.195	0.185	0.149	0.191	0.186	0.119	0.167	0.176	0.152	0.164
GoodDrag	Clean	0.035	0.035	0.043	0.039	0.027	0.032	0.037	0.030	0.023	0.030	0.033
	SHADM(16/255)	0.153	0.195	0.160	0.145	0.185	0.206	0.132	0.150	0.161	0.165	0.165
FreeDrag	Clean	0.117	0.058	0.104	0.090	0.070	0.149	0.130	0.078	0.053	0.100	0.093
	SHADM(16/255)	0.217	0.204	0.236	0.252	0.257	0.363	0.294	0.286	0.234	0.256	0.248
DragText	Clean	0.026	0.021	0.026	0.023	0.021	0.028	0.026	0.013	0.018	0.020	0.022
	SHADM(16/255)	0.076	0.128	0.125	0.120	0.103	0.116	0.125	0.113	0.114	0.119	0.111
CLIPDrag	Clean	0.023	0.018	0.025	0.021	0.015	0.016	0.017	0.016	0.015	0.014	0.019
	SHADM(16/255)	0.071	0.124	0.120	0.119	0.092	0.104	0.115	0.104	0.120	0.100	0.104
InstantDrag	Clean	0.031	0.030	0.029	0.029	0.024	0.038	0.043	0.020	0.023	0.034	0.030
	SHADM(16/255)	0.078	0.158	0.147	0.148	0.114	0.119	0.139	0.117	0.142	0.140	0.128
FastDrag	Clean	0.033	0.021	0.032	0.027	0.018	0.020	0.021	0.018	0.016	0.032	0.026
	SHADM(16/255)	0.089	0.129	0.129	0.117	0.100	0.178	0.104	0.117	0.102	0.101	0.111
SDE-Drag	Clean	0.015	0.012	0.018	0.016	0.006	0.004	0.003	0.008	0.010	0.007	0.011
	SHADM(16/255)	0.063	0.129	0.119	0.119	0.088	0.095	0.105	0.096	0.113	0.092	0.099
PnP Diffusion	Clean	0.133	0.082	0.091	0.098	0.070	0.084	0.103	0.146	0.068	0.070	0.093
	SHADM(16/255)	0.362	0.260	0.356	0.279	0.273	0.470	0.445	0.427	0.362	0.296	0.334
MasaCtrl	Clean	0.073	0.038	0.056	0.054	0.031	0.058	0.092	0.092	0.044	0.030	0.052
	SHADM(16/255)	0.116	0.148	0.146	0.148	0.105	0.181	0.188	0.218	0.163	0.106	0.139
AnimateDiff	Clean	0.069	0.062	0.073	0.081	0.063	0.078	0.139	0.074	0.065	0.126	0.083
	SHADM(16/255)	0.087	0.130	0.142	0.111	0.104	0.151	0.175	0.105	0.110	0.132	0.120
PixArt	Clean	0.048	0.045	0.045	0.043	0.044	0.040	0.031	0.036	0.026	0.037	0.041
	SHADM(16/255)	0.068	0.065	0.066	0.057	0.062	0.054	0.048	0.058	0.048	0.058	0.060

Table A20: FID robustness of SHADM against JPEG compression (quality=75%) across different diffusion models on DragBench.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
		FID										
DragLoRA	Clean	71.530	43.222	61.365	57.006	36.046	55.076	46.115	51.040	39.663	75.700	43.831
	SHADM(16/255)	148.523	78.968	117.745	92.557	74.371	168.208	179.366	273.468	121.996	138.577	98.244
DragDiffusion	Clean	65.977	33.763	56.531	43.622	39.274	50.815	38.953	48.354	37.702	63.296	39.434
	SHADM(16/255)	145.079	88.232	114.740	86.911	75.244	172.437	169.439	270.328	118.541	148.576	99.148
DragNoise	Clean	73.854	41.623	63.269	53.367	51.520	54.282	42.622	62.798	50.286	64.339	44.657
	SHADM(16/255)	227.437	142.062	171.709	137.558	185.593	304.789	204.166	389.091	180.424	205.982	145.948
GoodDrag	Clean	82.768	42.006	77.501	60.544	48.415	65.671	85.240	92.017	47.536	72.091	50.778
	SHADM(16/255)	221.452	135.615	143.744	124.182	193.876	295.381	211.741	383.541	180.341	221.330	144.578
FreeDrag	Clean	175.883	73.583	135.046	136.728	131.796	137.395	140.618	178.452	105.156	155.939	98.482
	SHADM(16/255)	244.330	177.816	204.897	237.503	253.198	292.343	310.063	469.343	248.811	261.121	169.582
DragText	Clean	65.345	33.578	56.123	43.835	39.270	50.848	39.428	48.215	38.827	63.677	39.423
	SHADM(16/255)	146.853	87.029	111.632	89.431	75.007	172.731	169.115	274.015	117.799	145.455	98.797
CLIPDrag	Clean	55.254	31.178	48.087	31.538	30.246	42.580	33.130	44.359	34.902	37.476	31.411
	SHADM(16/255)	135.805	88.312	97.178	84.816	74.633	175.866	157.617	265.301	124.184	120.465	93.457
InstantDrag	Clean	64.219	43.709	38.737	45.955	48.501	49.020	46.660	53.594	42.923	68.259	39.976
	SHADM(16/255)	151.436	93.330	119.805	121.961	100.071	171.858	205.700	327.198	132.984	160.443	109.812
FastDrag	Clean	62.783	34.693	53.363	53.589	31.380	49.163	37.121	47.516	40.478	66.883	39.721
	SHADM(16/255)	146.569	76.272	103.139	93.301	100.266	193.734	180.605	260.562	114.605	130.162	98.055
SDE-Drag	Clean	32.946	19.910	27.210	22.362	14.688	12.137	10.295	23.192	17.837	16.901	16.952
	SHADM(16/255)	130.667	82.520	93.982	81.293	63.058	163.546	172.750	280.973	111.168	103.310	89.099
PnP Diffusion	Clean	185.665	85.443	114.733	118.213	97.450	87.822	123.413	226.291	123.184	111.180	89.805
	SHADM(16/255)	308.015	256.229	332.194	273.399	258.318	377.530	366.280	524.689	362.094	291.820	215.977
MasaCtrl	Clean	120.710	47.995	81.714	85.089	65.096	87.137	125.835	153.795	89.821	80.492	66.788
	SHADM(16/255)	179.488	81.694	114.677	126.311	103.474	189.615	222.451	416.682	163.675	128.112	111.508
AnimateDiff	Clean	105.251	65.466	84.859	91.921	88.644	97.621	129.947	118.211	103.835	141.045	73.020
	SHADM(16/255)	133.601	80.208	123.685	94.576	112.266	178.065	215.795	211.793	132.507	151.043	93.134
PixArt	Clean	99.757	52.131	62.823	55.969	55.283	78.770	71.323	126.575	68.372	69.699	56.550
	SHADM(16/255)	119.680	72.128	83.278	68.616	74.091	90.052	93.100	144.551	89.010	88.838	75.785

Table A21: ASR robustness of SHADM against JPEG compression (quality=75%) across different diffusion models on DragBench.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
		ASR										
DragLoRA	Clean	2.86	0.00	4.76	0.00	0.00	10.00	0.00	0.00	0.00	0.00	1.46
	SHADM(16/255)	11.43	9.09	0.00	9.09	5.00	60.00	10.00	10.00	5.00	11.43	10.73
DragDiffusion	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(16/255)	17.14	9.09	0.00	13.64	0.00	30.00	0.00	20.00	5.00	11.43	10.24
DragNoise	Clean	0.00	0.00	0.00	4.55	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(16/255)	34.29	18.18	9.52	22.73	40.00	60.00	0.00	50.00	20.00	34.29	28.29
GoodDrag	Clean	5.71	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	1.46
	SHADM(16/255)	34.29	22.73	14.29	13.64	50.00	80.00	0.00	40.00	15.00	34.29	29.27
FreeDrag	Clean	5.71	0.00	0.00	0.00	0.00	30.00	10.00	0.00	0.00	2.86	3.41
	SHADM(16/255)	17.14	4.55	4.76	9.09	35.00	60.00	70.00	40.00	10.00	17.14	20.49
DragText	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(16/255)	17.14	9.09	0.00	13.64	0.00	20.00	0.00	30.00	5.00	14.29	10.73
CLIPDrag	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(16/255)	17.14	9.09	4.76	9.09	5.00	60.00	10.00	30.00	5.00	11.43	13.17
InstantDrag	Clean	2.86	0.00	0.00	4.55	0.00	10.00	0.00	10.00	0.00	2.86	2.44
	SHADM(16/255)	17.14	0.00	4.76	13.64	10.00	40.00	10.00	20.00	15.00	20.00	14.15
FastDrag	Clean	2.86	0.00	0.00	0.00	0.00	10.00	10.00	0.00	0.00	2.86	1.95
	SHADM(16/255)	17.14	9.09	0.00	9.09	5.00	80.00	20.00	30.00	10.00	11.43	14.63
SDE-Drag	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(16/255)	8.57	13.64	4.76	4.55	5.00	30.00	0.00	10.00	5.00	11.43	8.78
PnP Diffusion	Clean	5.71	0.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	2.86	1.95
	SHADM(16/255)	42.86	18.18	19.05	31.82	50.00	100.00	60.00	90.00	25.00	31.43	39.51
MasaCtrl	Clean	5.71	0.00	0.00	0.00	0.00	10.00	0.00	0.00	5.00	0.00	1.95
	SHADM(16/255)	14.29	0.00	0.00	9.09	0.00	70.00	70.00	70.00	25.00	8.57	17.56
AnimateDiff	Clean	11.61	5.11	12.80	11.93	9.38	16.88	51.25	10.00	2.81	22.32	13.93
	SHADM(16/255)	10.36	7.67	18.45	10.23	9.38	52.50	33.12	13.75	12.19	15.71	15.21
PixArt	Clean	5.71	0.00	0.00	0.00	0.00	10.00	10.00	0.00	0.00	2.86	2.44
	SHADM(16/255)	8.71	0.00	0.00	0.00	10.00	10.00	0.00	0.00	0.00	0.00	2.87

A24, and A25 show that SHADM still achieves effective performance across four evaluating metrics. Hence it is generalizable to other NSFW content.

Table A22: LPIPS generalizability evaluation of SHADM using an alternative NSFW target (pornographic) with distinct semantics.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	0.122	0.156	0.119	0.131	0.121	0.141	0.115	0.044	0.117	0.149	0.127
	SHADM(16/255)	0.279	0.285	0.221	0.222	0.334	0.447	0.247	0.305	0.249	0.312	0.284
DragDiffusion	Clean	0.119	0.122	0.113	0.112	0.134	0.160	0.111	0.040	0.097	0.129	0.117
	SHADM(16/255)	0.259	0.270	0.217	0.219	0.317	0.410	0.262	0.316	0.227	0.312	0.273
DragNoise	Clean	0.113	0.116	0.119	0.123	0.105	0.113	0.081	0.060	0.111	0.099	0.107
	SHADM(16/255)	0.240	0.252	0.217	0.206	0.278	0.357	0.192	0.290	0.253	0.247	0.247
GoodDrag	Clean	0.206	0.249	0.209	0.208	0.240	0.237	0.241	0.148	0.168	0.213	0.212
	SHADM(16/255)	0.346	0.332	0.317	0.284	0.396	0.417	0.343	0.372	0.293	0.359	0.341
FreeDrag	Clean	0.250	0.209	0.226	0.217	0.239	0.277	0.219	0.163	0.231	0.232	0.230
	SHADM(16/255)	0.404	0.449	0.378	0.378	0.448	0.631	0.377	0.407	0.395	0.470	0.428
DragText	Clean	0.116	0.123	0.111	0.113	0.133	0.160	0.111	0.040	0.098	0.129	0.116
	SHADM(16/255)	0.259	0.271	0.215	0.219	0.317	0.410	0.262	0.316	0.228	0.313	0.274
CLIPDrag	Clean	0.100	0.100	0.098	0.089	0.095	0.088	0.078	0.043	0.084	0.080	0.089
	SHADM(16/255)	0.247	0.262	0.198	0.203	0.296	0.387	0.237	0.314	0.228	0.261	0.254
InstantDrag	Clean	0.119	0.106	0.080	0.099	0.106	0.150	0.149	0.055	0.083	0.130	0.109
	SHADM(16/255)	0.269	0.285	0.213	0.245	0.311	0.380	0.297	0.328	0.260	0.326	0.285
FastDrag	Clean	0.130	0.122	0.118	0.117	0.131	0.162	0.119	0.040	0.097	0.140	0.122
	SHADM(16/255)	0.261	0.260	0.198	0.209	0.315	0.396	0.216	0.281	0.220	0.297	0.262
SDE-Drag	Clean	0.035	0.050	0.053	0.055	0.026	0.011	0.011	0.014	0.030	0.023	0.034
	SHADM(16/255)	0.197	0.219	0.160	0.159	0.240	0.296	0.156	0.260	0.186	0.210	0.203
PnP Diffusion	Clean	0.328	0.289	0.380	0.392	0.260	0.157	0.386	0.321	0.356	0.274	0.317
	SHADM(16/255)	0.507	0.446	0.512	0.518	0.456	0.608	0.597	0.578	0.596	0.489	0.516
MasaCtrl	Clean	0.196	0.155	0.158	0.190	0.154	0.209	0.232	0.159	0.206	0.122	0.172
	SHADM(16/255)	0.258	0.312	0.231	0.272	0.266	0.292	0.276	0.303	0.299	0.262	0.273
AnimateDiff	Clean	0.188	0.190	0.146	0.180	0.173	0.229	0.262	0.175	0.192	0.275	0.202
	SHADM(16/255)	0.296	0.313	0.252	0.252	0.351	0.454	0.356	0.411	0.313	0.352	0.321
PixArt	Clean	0.120	0.101	0.094	0.105	0.098	0.067	0.065	0.076	0.064	0.080	0.092
	SHADM(16/255)	0.120	0.111	0.114	0.106	0.108	0.082	0.084	0.077	0.086	0.109	0.100

Table A23: $\mathfrak{d}_{CLIP} = (1 - \text{CLIP})$ generalizability evaluation of SHADM using an alternative NSFW target (pornographic) with distinct semantics.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	0.029	0.031	0.032	0.030	0.021	0.033	0.032	0.019	0.018	0.030	0.028
	SHADM(16/255)	0.166	0.197	0.155	0.123	0.191	0.237	0.116	0.157	0.153	0.165	0.165
DragDiffusion	Clean	0.027	0.021	0.027	0.023	0.020	0.029	0.026	0.013	0.018	0.020	0.023
	SHADM(16/255)	0.143	0.182	0.150	0.124	0.161	0.192	0.125	0.139	0.148	0.154	0.151
DragNoise	Clean	0.037	0.024	0.035	0.033	0.029	0.029	0.027	0.021	0.020	0.024	0.028
	SHADM(16/255)	0.137	0.188	0.161	0.130	0.150	0.196	0.118	0.133	0.168	0.143	0.151
GoodDrag	Clean	0.035	0.035	0.043	0.039	0.027	0.032	0.037	0.030	0.023	0.030	0.033
	SHADM(16/255)	0.158	0.192	0.176	0.137	0.193	0.229	0.128	0.147	0.163	0.176	0.170
FreeDrag	Clean	0.117	0.058	0.104	0.090	0.070	0.149	0.130	0.078	0.053	0.100	0.093
	SHADM(16/255)	0.196	0.238	0.225	0.228	0.205	0.291	0.272	0.240	0.202	0.238	0.226
DragText	Clean	0.026	0.021	0.026	0.023	0.021	0.028	0.026	0.013	0.018	0.020	0.022
	SHADM(16/255)	0.142	0.181	0.152	0.126	0.162	0.192	0.128	0.139	0.148	0.155	0.152
CLIPDrag	Clean	0.023	0.018	0.025	0.021	0.015	0.016	0.017	0.016	0.015	0.014	0.019
	SHADM(16/255)	0.137	0.183	0.141	0.124	0.150	0.190	0.116	0.124	0.150	0.139	0.145
InstantDrag	Clean	0.031	0.030	0.029	0.029	0.024	0.038	0.043	0.020	0.023	0.034	0.030
	SHADM(16/255)	0.133	0.218	0.168	0.172	0.163	0.164	0.145	0.141	0.203	0.180	0.170
FastDrag	Clean	0.033	0.021	0.032	0.027	0.018	0.020	0.021	0.018	0.016	0.032	0.026
	SHADM(16/255)	0.141	0.182	0.138	0.120	0.154	0.226	0.113	0.147	0.146	0.165	0.152
SDE-Drag	Clean	0.015	0.012	0.018	0.016	0.006	0.004	0.003	0.008	0.010	0.007	0.011
	SHADM(16/255)	0.105	0.179	0.139	0.115	0.127	0.136	0.103	0.110	0.153	0.123	0.129
PnP Diffusion	Clean	0.133	0.082	0.091	0.098	0.070	0.084	0.103	0.146	0.068	0.070	0.093
	SHADM(16/255)	0.269	0.200	0.289	0.262	0.194	0.401	0.400	0.387	0.317	0.235	0.273
MasaCtrl	Clean	0.073	0.038	0.056	0.054	0.031	0.058	0.092	0.092	0.044	0.030	0.052
	SHADM(16/255)	0.116	0.175	0.150	0.145	0.113	0.163	0.181	0.182	0.182	0.117	0.144
AnimateDiff	Clean	0.069	0.062	0.073	0.081	0.063	0.078	0.139	0.074	0.065	0.126	0.083
	SHADM(16/255)	0.131	0.158	0.155	0.113	0.166	0.229	0.182	0.152	0.120	0.167	0.151
PixArt	Clean	0.048	0.045	0.045	0.043	0.044	0.040	0.031	0.036	0.026	0.037	0.041
	SHADM(16/255)	0.098	0.095	0.096	0.087	0.092	0.084	0.078	0.088	0.078	0.088	0.080

Table A24: FID generalizability evaluation of SHADM using an alternative NSFW target (pornographic) with distinct semantics.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	71.530	43.222	61.365	57.006	36.046	55.076	46.115	51.040	39.663	75.700	43.831
	SHADM(16/255)	226.901	133.224	144.942	100.902	186.235	336.098	193.016	406.555	141.978	195.111	137.822
DragDiffusion	Clean	65.977	33.763	56.531	43.622	39.274	50.815	38.953	48.354	37.702	63.296	39.434
	SHADM(16/255)	214.686	144.197	136.788	103.544	151.949	305.148	188.092	351.391	139.399	176.385	128.836
DragNoise	Clean	73.854	41.623	63.269	53.367	51.520	54.282	42.622	62.798	50.286	64.339	44.657
	SHADM(16/255)	224.065	134.823	145.256	103.477	144.931	294.012	196.369	360.305	156.925	178.305	132.000
GoodDrag	Clean	82.768	42.006	77.501	60.544	48.415	65.671	85.240	92.017	47.536	72.091	50.778
	SHADM(16/255)	242.138	150.539	163.413	111.265	199.768	317.290	207.244	393.590	148.939	195.080	141.908
FreeDrag	Clean	175.883	73.583	135.046	136.728	131.796	137.395	140.618	178.452	105.156	155.939	98.482
	SHADM(16/255)	238.299	191.027	220.241	212.935	203.086	310.546	287.653	336.189	211.470	257.410	157.499
DragText	Clean	65.345	33.578	56.123	43.835	39.270	50.848	39.428	48.215	38.827	63.677	39.423
	SHADM(16/255)	215.229	144.437	136.409	103.396	151.616	304.969	187.790	350.915	141.056	176.624	129.073
CLIPDrag	Clean	55.254	31.178	48.087	31.538	30.246	42.580	33.130	44.359	34.902	37.476	31.411
	SHADM(16/255)	206.012	133.168	129.439	96.509	133.477	283.277	192.411	366.702	143.480	165.440	124.527
InstantDrag	Clean	64.219	43.709	38.737	45.955	48.501	49.020	46.660	53.594	42.923	68.259	39.976
	SHADM(16/255)	225.441	190.417	154.441	167.541	159.284	295.437	283.231	426.403	214.528	226.242	158.410
FastDrag	Clean	62.783	34.693	53.363	53.589	31.380	49.163	37.121	47.516	40.478	66.883	39.721
	SHADM(16/255)	216.953	119.753	119.847	103.753	163.001	334.106	190.616	395.466	145.822	180.846	130.251
SDE-Drag	Clean	32.946	19.910	27.210	22.362	14.688	12.137	10.295	23.192	17.837	16.901	16.952
	SHADM(16/255)	192.188	130.478	116.400	87.380	103.213	288.380	200.204	378.153	141.617	137.640	118.118
PnP Diffusion	Clean	185.665	85.443	114.733	118.213	97.450	87.822	123.413	226.291	123.184	111.180	89.805
	SHADM(16/255)	266.938	186.902	295.324	234.007	197.550	347.663	341.738	464.358	318.629	236.646	175.437
MasaCtrl	Clean	120.710	47.995	81.714	85.089	65.096	87.137	125.835	153.795	89.821	80.492	66.788
	SHADM(16/255)	184.637	108.582	124.204	128.178	104.909	208.974	236.273	386.489	176.702	151.780	119.323
AnimateDiff	Clean	105.251	65.466	84.859	91.921	88.644	97.621	129.947	118.211	103.835	141.045	73.020
	SHADM(16/255)	171.873	98.779	131.695	103.529	162.096	298.069	222.373	319.402	141.188	172.404	113.940
PixArt	Clean	99.757	52.131	62.823	55.969	55.283	78.770	71.323	126.575	68.372	69.699	56.550
	SHADM(16/255)	119.666	72.084	83.249	68.611	74.076	90.050	93.135	144.540	88.993	88.816	75.782

Table A25: ASR (%) generalizability evaluation of SHADM using an alternative NSFW target (pornographic) with distinct semantics.

Target	Method	Categories										
		Art-work	Landscape	Building-city-view	Building-countryside -view	Animals	Human-head	Human-upper-body	Human-full-body	Interior-design	Other-objects	All
DragLoRA	Clean	2.86	0.00	4.76	0.00	0.00	10.00	0.00	0.00	0.00	0.00	1.46
	SHADM(16/255)	34.29	9.09	4.76	4.55	25.00	90.00	20.00	40.00	10.00	8.57	20.00
DragDiffusion	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(16/255)	28.57	4.55	4.76	4.55	25.00	90.00	10.00	20.00	0.00	11.43	16.59
DragNoise	Clean	0.00	0.00	0.00	4.55	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(16/255)	28.57	0.00	4.76	0.00	25.00	100.00	20.00	40.00	5.00	14.29	18.54
GoodDrag	Clean	5.71	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	1.46
	SHADM(16/255)	25.71	4.55	0.00	0.00	15.00	90.00	20.00	30.00	5.00	17.14	16.59
FreeDrag	Clean	5.71	0.00	0.00	0.00	0.00	30.00	10.00	0.00	0.00	2.86	3.41
	SHADM(16/255)	14.29	0.00	4.76	0.00	5.00	60.00	50.00	30.00	15.00	11.43	13.66
DragText	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(16/255)	28.57	4.55	4.76	0.00	25.00	90.00	20.00	20.00	0.00	11.43	16.59
CLIPDrag	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(16/255)	28.57	4.55	0.00	4.55	20.00	90.00	0.00	40.00	10.00	5.71	16.10
InstantDrag	Clean	2.86	0.00	0.00	4.55	0.00	10.00	0.00	10.00	0.00	2.86	2.44
	SHADM(16/255)	31.43	0.00	4.76	18.18	40.00	90.00	50.00	60.00	35.00	22.86	28.78
FastDrag	Clean	2.86	0.00	0.00	0.00	0.00	10.00	10.00	0.00	0.00	2.86	1.95
	SHADM(16/255)	28.57	4.55	4.76	4.55	35.00	100.00	30.00	50.00	15.00	20.00	23.41
SDE-Drag	Clean	2.86	0.00	0.00	0.00	0.00	10.00	0.00	0.00	0.00	0.00	0.98
	SHADM(16/255)	25.71	0.00	0.00	0.00	25.00	90.00	0.00	30.00	5.00	8.57	14.63
PnP Diffusion	Clean	5.71	0.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	2.86	1.95
	SHADM(16/255)	14.29	4.55	4.76	9.09	15.00	60.00	50.00	30.00	20.00	8.57	16.10
MasaCtrl	Clean	5.71	0.00	0.00	0.00	0.00	10.00	0.00	0.00	5.00	0.00	1.95
	SHADM(16/255)	14.29	0.00	0.00	0.00	0.00	90.00	80.00	60.00	10.00	5.71	15.61
AnimateDiff	Clean	11.61	5.11	12.80	11.93	9.38	16.88	51.25	10.00	2.81	22.32	13.93
	SHADM(16/255)	18.04	6.82	17.86	13.92	24.38	80.00	37.50	32.50	11.56	18.21	21.07
PixArt	Clean	5.71	0.00	0.00	0.00	0.00	10.00	10.00	0.00	0.00	2.86	2.44
	SHADM(16/255)	5.71	0.00	0.00	0.00	10.00	10.00	0.00	0.00	10.00	0.00	3.571

A.3.7 Experiments with Two Additional NSFW Detectors

We also evaluate the generated images using two additional detectors: NudeNet[2] and QwenGuard (a state-of-the-art VLM-based safety checker, Zhao et al. 2025). The results are provided in Table A26. **Both NudeNet and QwenGuard show high agreement with the Q16 classifier** across all settings (overall agreement $> 98.0\%$, with Cohen's $\kappa > 0.92$). Particularly in the **Living subjects** subset, the agreement between Q16 and QwenGuard reaches up to 100.0% ($\kappa = 1.000$).

Table A26: Two additional NSFW detectors and their agreements with the Q16 classifier.

Target	Scope	Q16	NudeNet	QwenGuard	Q16–NudeNet Agreement	Cohen's κ	Q16–QwenGuard Agreement	Cohen's κ
DragLoRA	All	41/205 (20.0%)	39/205 (19.0%)	40/205 (19.5%)	98.0%	0.938	98.5%	0.954
DragLoRA	Still-life	21/155 (13.5%)	20/155 (12.9%)	20/155 (12.9%)	98.1%	0.916	98.1%	0.916
DragLoRA	Living subjects	20/50 (40.0%)	19/50 (38.0%)	20/50 (40.0%)	98.0%	0.958	100.0%	1.000
FastDrag	All	48/205 (23.4%)	46/205 (22.4%)	47/205 (22.9%)	98.0%	0.945	98.5%	0.959
FastDrag	Still-life	23/155 (14.8%)	22/155 (14.2%)	23/155 (14.8%)	98.1%	0.922	98.7%	0.949
FastDrag	Living subjects	25/50 (50.0%)	24/50 (48.0%)	24/50 (48.0%)	98.0%	0.960	98.0%	0.960
MasaCtrl	All	32/205 (15.6%)	30/205 (14.6%)	31/205 (15.1%)	98.0%	0.924	98.5%	0.944
MasaCtrl	Still-life	9/155 (5.8%)	9/155 (5.8%)	8/155 (5.2%)	98.7%	0.882	99.4%	0.938
MasaCtrl	Living subjects	23/50 (46.0%)	21/50 (42.0%)	23/50 (46.0%)	96.0%	0.919	96.0%	0.919

The false positive and false negative rates are provided in Table A27. Across all target models, both detectors maintain **very low** False Positive Rates (FPR ranging between 1.15% and 1.89%) and False Negative Rates (FNR ranging between 2.13% and 3.33%). This confirms that our evaluation pipeline is highly reliable and free from classifier-specific bias. For non-living images, FPR drops as low as 0.68%, which shows that visual degradation or non-explicit feature shifts **rarely trigger false alarms**. On living-subject samples, FNR reaches 0.00% across almost all settings, which ensures that **genuine explicit violations are captured** without escaping detection.

[2] https://github.com/platelminto/NudeNetClassifier

Table A27: False positive and false negative rates of NudeNet and QwenGuard.

Target	Scope	vs. NudeNet			vs. QwenGuard		
		FP/FN	FPR	FNR	FP/FN	FPR	FNR
DragLoRA	All	3/1	1.81%	2.56%	2/1	1.21%	2.50%
	Still-life	2/1	1.48%	5.00%	2/1	1.48%	5.00%
	Living subjects	1/0	3.23%	0.00%	0/0	0.00%	0.00%
FastDrag	All	3/1	1.89%	2.17%	2/1	1.27%	2.13%
	Still-life	2/1	1.50%	4.55%	1/1	0.76%	4.35%
	Living subjects	1/0	3.85%	0.00%	1/0	3.85%	0.00%
MasaCtrl	All	3/1	1.71%	3.33%	2/1	1.15%	3.23%
	Still-life	1/1	0.68%	11.11%	1/0	0.68%	0.00%
	Living subjects	2/0	6.90%	0.00%	1/1	3.70%	4.35%

A.3.8 Comparisons with NSFW-inducing Attacks and Threat Model

Existing NSFW-inducing attacks, such as AdvI2I (Zeng et al., 2025), MMA-Diffusion (Yang et al., 2024a), and DiffZOO (Dang et al., 2025), operate under substantially stronger threat assumptions than SHADM:

- **AdvI2I** is a *white-box attack* that requires direct access to gradients of the target model.
- **MMA-Diffusion** and **DiffZOO** are **query-heavy black-box attacks** requiring hundreds or thousands of interactive queries or text-prompt optimization.
- **SHADM** operates under a **single-shot, zero-query, transfer-based black-box threat model**. The attacker constructs the perturbation offline using a local surrogate UNet, with **zero query or structural access** to the victim model or prompt input.

The threat models for these methods are shown in Table A28.

Table A28: Threat models for different attack methods.

Method	Target Access	Query Requirement	Input Control Required	Primary Attack Surface
AdvI2I	White-Box	N/A (Direct Gradients)	Source Latent / Model	White-box latent space trajectory optimization
MMA-Diffusion	Black-Box	High ($> 10^3$ queries)	Text Prompts / Latents	Multi-query prompt & latent optimization
DiffZOO	Black-Box	High ($> 10^3$ queries)	Source Image / Prompts	Zeroth-order gradient estimation
Mist / Glaze	Black-Box	Zero-Query	Source Image	Perceptual disruption / Feature watermarking
SHADM (Ours)	**Black-Box**	**Zero-Query (Single-Shot)**	**Source Image Only**	**Key/Value Attention Feature Hijacking**

We also add experiments to compare these methods in Table A29. To ensure a consistent attack model across different methods, AdvI2I, MMA-Diffusion, and SHADM construct their adversarial perturbations with their default settings via the same surrogate model UNet, then they attack the target models with zero query. DiffZOO directly attacks the target models with at most 1968 queries for each prompt. Results show that SHADM significantly outperforms other baselines, especially in the Living Subjects category with both Q16 and NudeNet metrics. It indicates that SHADM is effective in structural hijacking to generate NSFW contents.

A.3.9 Image Quality Degradation and NSFW Signals

Based on the experimental results in Table A2 (LPIPS) and Table A3 (ASR), across Still-life categories, ASR does not simply correlate with image degradation or LPIPS distortion.

1. High distortion is not equivalent to High ASR: Under SHADM(16/255), FreeDrag shows severe perceptual distortion across Still-life categories: for example, an LPIPS of 0.463 on Landscape and 0.457 on Interior-design. However, its corresponding ASR on Landscape is 0.00% and on Interior-design is 10.00%. Under SHADM(16/255) on Landscape, DragDiffusion has a significantly lower LPIPS distortion (0.287) compared to FreeDrag (0.463), but DragDiffusion has a substantially higher ASR (22.73%) than FreeDrag (0.00%).

Table A29: Comparison of different NSFW-inducing attack methods across different target models.

Attack	Target	All			Still-life			Living		
		LPIPS	Q16 ↑	NudeNet ↑	LPIPS	Q16 ↑	NudeNet ↑	LPIPS	Q16 ↑	NudeNet ↑
SHADM	DragLoRA	0.301	20.00	19.00	0.285	13.50	12.90	0.350	40.00	38.00
	FastDrag	0.277	23.40	22.40	0.263	14.80	14.20	0.318	50.00	48.00
	MasaCtrl	0.282	15.60	14.60	0.280	5.80	5.80	0.288	46.00	42.00
AdvI2I	DragLoRA	0.191	6.34	0.00	0.203	5.16	0.00	0.296	10.00	0.00
	FastDrag	0.153	3.90	0.49	0.147	3.23	0.65	0.172	6.00	0.00
	MasaCtrl	0.215	4.39	0.00	0.214	2.58	0.00	0.219	10.00	0.00
MMA-Diffusion	DragLoRA	0.188	2.44	0.49	0.174	2.58	0.65	0.214	2.00	0.00
	FastDrag	0.118	2.44	0.49	0.115	1.94	0.65	0.129	4.00	0.00
	MasaCtrl	0.176	2.93	0.00	0.174	1.29	0.00	0.182	8.00	0.00
DiffZOO	DragLoRA	0.196	2.44	0.00	0.217	1.94	0.65	0.214	2.00	0.00
	FastDrag	0.133	2.84	0.49	0.115	1.42	0.00	0.101	2.00	0.00
	MasaCtrl	0.158	3.01	0.00	0.174	1.24	0.00	0.172	4.00	0.00

2. Similar Distortion but Vastly Different ASRs Across Models: Under SHADM(16/255) on Art-work, DragLoRA (LPIPS: 0.289, ASR: 37.14%), FastDrag (LPIPS: 0.279, ASR: 42.86%), and AnimateDiff (LPIPS: 0.311, ASR: 24.11%) suffer similar levels of LPIPS degradation. However, their ASRs range widely from 24.11% to 42.86%. Similarly, for Other-objects under SHADM(16/255), InstantDrag and GoodDrag have comparable LPIPS scores (0.350 vs. 0.387), but InstantDrag achieves an ASR of 51.43% compared to 28.57% of GoodDrag.

3. Disproportionate ASR Jumps Relative to LPIPS Scaling: On Landscape under DragDiffusion, moving from SHADM(8/255) to SHADM(12/255) increases LPIPS moderately from 0.188 to 0.241 ($+28.2\%$), but the ASR jumps over three times from 4.55% to 13.64%.

We also conduct Pearson and Spearman correlation analyses between LPIPS and ASR on Still-life categories under SHADM(16/255) in Table A30. Results show that none of the p-values get smaller than 0.05, which means that **LPIPS and ASR are not correlated with statistical significance**. These comparisons indicate that perceptual degradation (LPIPS) alone is insufficient to trigger high ASR / Q16 safety flags on Still-life categories. Instead, the attack success is driven by adversarial semantic features targeted by the SHADM attack rather than generic image degradation.

Table A30: Pearson and Spearman correlation analyses between LPIPS and ASR under SHADM(16/255).

Target	Pearson r	p-value	Spearman ρ	p-value
DragLoRA	0.260	0.619	0.429	0.397
FastDrag	0.617	0.192	0.609	0.200
MasaCtrl	-0.088	0.869	-0.246	0.638

A.3.10 Experiments on Non-NSFW Target Images

We conduct experiments for SHADM(16/255) on non-NSFW target images in Table A31. Results show that both Q16 ($\leqslant 4.81\%$) and NudeNet ASRs ($\leqslant 0.49\%$) drop to near-zero levels. It indicates that the safety violations observed in our main experiments are strictly driven by the explicit features of the target image rather than arbitrary feature corruption or random model instability.

A.3.11 Risk Asymmetry and Precautionary Safety Margins (Zero-Tolerance Alignment)

We call on the community to pay sufficient attention to the systemic flaws revealed in this paper, due to the following reasons:

Asymmetric Cost of Failure: In real-world AI deployment and content moderation, the harm associated with exposing users to NSFW material is **strictly asymmetric**. The penalty for a **false negative** (failing to detect explicit or boundary-crossing material) incurs severe ethical, legal, and

Table A31: Performance of SHADM(16/255) on non-NSFW target images.

Target	Scope	LPIPS	$(1 - \text{CLIP})$	FID	Q16 ASR	NudeNet ASR
FastDrag	All	0.265	0.144	139.640	2.68%	0.00%
	Still-life	0.250	0.142	136.835	4.81%	0.00%
	Living subjects	0.313	0.150	218.012	4.00%	0.00%
MasaCtrl	All	0.269	0.133	121.682	3.34%	0.00%
	Still-life	0.265	0.133	122.152	3.87%	0.00%
	Living subjects	0.281	0.135	180.522	4.00%	0.00%
DragLoRA	All	0.370	0.193	149.134	3.34%	0.49%
	Still-life	0.378	0.206	156.739	3.10%	0.00%
	Living subjects	0.343	0.154	207.349	4.00%	0.00%

regulatory consequences, while the penalty for a **false positive** (conservatively flagging borderline content) carries negligible cost.

Precautionary Evaluation Standard: Because safety guardrails operate under a **zero-tolerance** policy, safety evaluations should not be tied to a hyper-conservative and overly narrow definition of NSFW content. Demanding such strict precision risks underestimating severe vulnerabilities by ignoring partial, suggestive, or structurally distorted explicit features.

Alignment with Production Guardrails: Real-world generative AI platforms (e.g., OpenAI, Midjourney, Stability AI) employ sensitive and high-recall detection thresholds precisely because **any** emerging explicit feature constitutes a critical safety breach. Standard safety benchmarks (such as the Q16 classifier) intentionally reflect this zero-tolerance stance. Thus adopting a high-sensitivity standard ensures that our threat assessment aligns with real-world model deployment risks.

Broad Architectural Fragility: The vulnerability exposed by SHADM is not an isolated artifact of a single network implementation, but a fundamental security issue inherent to attention-driven latent generative pipelines. By successfully disrupting **14 diverse state-of-the-art (SOTA) diffusion models**, spanning classic UNet backbones, Transformer-based diffusion architectures (DiTs like PixArt), and interactive image-editing frameworks (e.g., DragDiffusion, DragLoRA, DragText), SHADM reveals that deep key/value structural hijacking represents a pervasive and industry-wide threat.

Societal Impact

This research strictly adheres to the ethical guidelines for security and privacy research, which prioritizes responsible methodology and disclosure. The primary goal of this work is fundamentally defensive: to uncover and understand the vulnerabilities within current adversarial machine learning systems and to inform the design of more robust countermeasures.

All experiments are conducted exclusively in controlled, academic environments. The data sets used in this study are publicly available, widely-used in the academic community, and conform to their respective licenses. No personally identifiable information or sensitive private data is collected, stored, or processed throughout the course of this research.

As this research focuses solely on technical system analysis and publicly available data, it does not involve human subjects.

AI Use Statement

In this work, we used generative AI tools only for grammar correction and minor language edits to improve readability. We have not used generative AI tools for other tasks. We have reviewed all AI-assisted work, and take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.